Designing Data Warehouse : A group collaborative approach

Amir Sakka

THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par l'Université Toulouse 1 Capitole

Présentée et soutenue par

AMIR SAKKA

Le 17 septembre 2020

Designing Data Warehouse : A group collaborative approach

Ecole doctorale **EDMITT - Ecole Doctorale Mathématiques, Informatique et Télécommunications de Toulouse**

Spécialité **Informatique et Télécommunications**

Unité de recherche

IRIT : Institut de Recherche en Informatique de Toulouse

Thèse dirigée par

Pascale Zaraté et Sandro Bimonte

Jury

M. Bajwa Deepinder, Rapporteur
M. Robert Wrembel, Rapporteur
M. Vetschera Rudolf, Rapporteur
Mme Fadila Bentayeb, Examinatrice
Mme Lucile Sautot, Examinatrice
M. Guy Camilleri, Examinateur
Mme Pascale Zaraté, Directrice de thèse
M. Sandro Bimonte, Co-directeur de thèse

Aknowledgement

It is with immense gratitude that I address my most genuine acknowledgement to my PhD directors professor Pascale Zaraté and associate-professor Sandro Bimonte for their support and guidance throughout my research efforts, and to my supervisors Lucile Sautot and Guy Camilleri for their passionate attitude and encouragement that helped me to overcome many challenging difficulties of this research.

I am also sincerely grateful to both teams of Copain (INRAE) and ADRIA (UT1C) that have welcomed me during the last three years within their motivating and supportive research working environments.

I would like as well to express my extreme gratefulness to all organizers, researchers and volunteers of the VGI4BIO and RUC-APS projects that have consented to participate in our empirical experiments and data acquisition inquiries. I thank, especially, the teams of LPO Aquitaine, MNHN and FEDACOVA for their valuable collaboration and insightful contributions to different steps of our work.

Finally, I would like to thank my family that have been always supportive to my research ambitions and that have sacrificed at many stages to allow me to achieve my goals.

Abstract

Data warehouses (DWs) are widely known for their powerful analysis capabilities that serve either for historic data investigation or for predictions of potentially continuous phenomena. However, they are still in most cases limitedly used except by enterprises or governments while, with the huge amounts of data produced and collected by the Web2.0 technologies, many other unusual users might benefit from analysing their data if DWs are properly dedicated to their specific needs. They might be association adherents, online community members, observatory volunteers, etc. Unlike in classical contexts, requirements engineering RE with volunteers lacks group cohesion and straightforward strategic objectives. This is hence because they come with different backgrounds and they do not have an acknowledged representative leadership, which would very likely lead to multiple contradictory interpretations of the data and consequently of conflictual requirements. When stakeholders have divergent goals, it becomes problematic to maintain an agreement between them, especially when it comes to eliciting DW requirements whose future use is meant to serve as larger interested public as it possibly could. In this work, we propose a new generic and participative DW design methodology that relies on a Group Decision Support System (GDSS) to support the collaboration of the engaged volunteers. We suggest in this methodology two RE scenarios, (i) using GDSS for a collaborative elicitation when groups of users with common objectives are identifiable or (ii) with pivot tables and rapid prototyping formalisms when only individual volunteers are participating. Then, we reduce the number of the resulting models by fusing them based on their multidimensional (MD) similarities. The fused models require a further refinement that focuses on solving the remaining subject matter inconsistencies that are due to either erroneous definitions of unspecialized volunteers or to conceptually admissible, but irrelevant to the application domain, newly generated elements after the fusion. This is handled by the "collaborative resolution of requirement conflicts" step that we defined two methods for its execution. The first is a simplified collaborative method that we evaluate in which each model's MD elements against a reduced number of criteria that apply for each component's type using an existing GDSS that allows the collaborative process execution. The second is a profile-aware method that we suggest for which a more detailed set of evaluation criteria and adaptability of the collaborative process to allow its use by both crowdsourcing and enterprise DW design projects. As GDSS are designed to support a

group engaged in a collective decision process, which is the main tool that we rely on which in two stages of our methodology i.e. RE and collaborative refinement of the fused models, we also propose a new GDSS that we adopted in its architecture the concept of Thinklets i.e. a well-known design pattern for collaborative processes. In addition to the group activities reproducibility that offers the concept of Thinklets, we have as well implemented a recommender system prototype that is mainly based on a hierarchical division of decision categories and an automatization of certain assistive functionalities to allow a guided and appropriate use of the system devoted to the facilitator. This has been done after a set of experiments conducted with real volunteer users engaged in solving risk management and uncertainty group problems. The new GDSS that we suggest introduces a customized implementation of certain Thinklets in order to improve their suitability to our methodology as well as for novice and inexperienced users from a more general perspective. In addition to that, we propose a new Thinklet, namely CollaborativeDW, that allows a fluid configuration and dynamic execution of our second refinement method i.e. the profile-aware approach, and that we have tested with real users.

Keywords: Collaborative design, Requirement engineering, data warehouse, conceptual modelling, multidimensional modelling, OLAP systems, VGI, citizen science, crowdsourced data, recommender systems, GDSS, Thinklets.

Résumé

Les entrepôts de données (EDs) sont connus pour leurs puissantes capacités d'analyse qui servent soit à la fouille de données historiques soit à la prévision de phénomènes potentiellement continus. Cependant, dans la plupart des cas, ils sont encore utilisés de manière limitée, sauf par les entreprises ou les gouvernements alors que, avec les énormes quantités de données produites et collectées par les technologies Web 2.0, de nombreux autres utilisateurs inhabituels pourraient bénéficier de l'analyse de leurs données si les EDs sont correctement dédiés à leurs besoins spécifiques. Il peut s'agir d'adhérents à une association, de membres d'une communauté en ligne, de volontaire d'un observatoire, etc. Contrairement aux contextes classiques, l'ingénierie des exigences (IE) avec les volontaires manque de cohésion de groupe et d'objectifs stratégiques précis. En effet, ils viennent d'horizons différents et n'ont pas un leadership représentatif reconnu, ce qui conduirait très probablement à de multiples interprétations contradictoires des données et par conséquent à des exigences conflictuelles. Lorsque les parties prenantes ont des objectifs divergents, il devient problématique de maintenir un accord entre elles, en particulier lorsqu'il s'agit d'éliciter des exigences d'ED dont l'utilisation future est destinée à servir le plus grand public intéressé possible. Dans ce travail, nous proposons une nouvelle méthodologie de conception participative d'ED, qui s'appuie sur un système d'aide à la décision de groupe (GDSS). Nous proposons deux scénarios d'IE (i) utiliser le GDSS pour une élicitation collaborative lorsque des groupes d'utilisateurs ayant des objectifs communs sont identifiables ou (ii) avec des tableaux croisés dynamiques et des formalismes de prototypage rapide lorsque seuls des volontaires individuels y participent. Ensuite, nous réduisons le nombre de modèles résultants en les fusionnant en fonction de leurs similitudes multidimensionnelles (MD). Les modèles fusionnés nécessitent une amélioration supplémentaire qui se concentre sur la résolution des incohérences causées soit par des définitions erronées de volontaires non spécialisés, soit par des éléments, conceptuellement admissibles mais sans rapport avec le domaine d'application, qui sont nouvellement générés après la fusion. Ceci est géré par la résolution collaborative des conflits d'exigences que nous avons défini au travers des deux méthodes précédemment évoquées. (i) Une méthode collaborative simplifiée que nous évaluons dans laquelle les éléments MD par rapport à un nombre réduit de critères en utilisant un GDSS existant qui permet l'exécution du processus collaboratif (PC). (ii) Une méthode

sensible au profil que nous suggérons pour laquelle un ensemble plus détaillé de critères d'évaluation et une adaptabilité du PC pour permettre son utilisation à la fois dans des projets de crowdsourcing et d'entreprise. Nous proposons également un nouveau GDSS dans lequel nous nous sommes inspirés dans son architecture du concept de Thinklet, qui est un modèle de conception bien connu dans la littérature pour les PCs. En plus de la reproductibilité des activités de groupe qu'offrent les Thinklets, nous avons implémenté un prototype d'un système de recommandation qui est basé sur une hiérarchisation des catégories de décision pour permettre une utilisation guidée et appropriée du système pour le facilitateur. Cela a été testé par des expériences menées avec de vrais utilisateurs volontaires engagés dans la résolution de problèmes de gestion de risques et d'incertitude. Nous introduisons dans ce GDSS une implémentation personnalisée de certains Thinklets afin d'améliorer leur adéquation à notre méthodologie ainsi qu'aux utilisateurs novices et inexpérimentés dans une perspective plus générale. En plus de cela, nous proposons un nouveau Thinklet, à savoir CollaborativeDW, qui permet une configuration dynamique de notre deuxième méthode de résolution de conflits, et que nous avons testée avec de vrais utilisateurs.

Mots-clés: Conception collaborative, Ingénierie des exigences, entrepôt de données, modélisation conceptuelle, modélisation multidimensionnelle, systèmes OLAP, VGI, science citoyenne, données participatives, systèmes de recommandation, GDSS, Thinklets.

Contents

List of figures

List of Tables

Chapter I

GENERAL INTRODUCTION

I.1. General context

The monitoring and conservation of biodiversity in agricultural landscapes represent currently a major challenges, both because intensive agricultural practices are rapidly eroding biodiversity (Bommarco et al. 2013; Levrel et al. 2010; Sui et al. 2013), and because many promising alternatives to improve the sustainability of agriculture rely on the ecosystem services provided by biodiversity (Princé et al. 2012; Giraud et al. 2013; Bedard et al. 2007). However, financial and human resources limitation is facing nature observatories, scientific institutions, and environmental government services to collect the amounts of data required for an accurate assessment of biodiversity trends. To achieve meaningful results, we need to encompass a wide range of situations by collecting standardized data and relying on predefined sampling protocols of observations in large spatial and temporal scales. To collect such data, the only actual financially reasonable possibility is by the involvement of a large number of volunteer observers (citizen science programs) (Régnier et al. 2015), who are also producing a huge amount of opportunistic biodiversity data, i.e. non-standardized data. Therefore, we suggest that the use of Volunteered Geographic Information (VGI) technology in participative monitoring of biodiversity would have important social, economic, and environmental benefits. However, VGI systems do not support advanced analysis tools of Geo-Business Intelligence (GeoBI) systems (Bimonte et al. 2014). GeoBI systems allow stakeholders to analyse geo-referenced indicators using cartographic displays (Golfarelli et al., 2013). We argue that GeoBI technologies and, particularly, Spatial Data Warehouse (SDW) and Spatial OLAP (SOLAP) can be successfully used to analyse VGI data and should be developed for farmland biodiversity monitoring. Spatial On-Line Analytical Processing (SOLAP) platforms allow non-experienced IT users to easily make geo-decision analysis and data

exploration (Stefanovic et al. 2000), which is not the case with the VGI systems that allow only creating, assembling and geographically disseminating the collected data (Golfarelli et al. 2013). Thereby, the various overlapping VGI user categories involved in this data collection process, each with different interests and perspectives of biodiversity data, make the requirements definition and the design of a data warehouse model problematic.

I.2. Problem statement

In this work, we focus on issues related to requirements engineering and design of multidimensional models from crowdsourcing projects. Several design methodologies for DW have been proposed, however, when decision-makers are volunteers and are different from those who decide the relevance of the requirements, they only represent few potential users of the OLAP system. This makes their specific analysis needs very likely not those useful for most final users. Volunteer users can have different backgrounds (e.g. scientist, amateur citizen, etc.), which can lead, amongst others, to multiple contradictory interpretations of the same requirement. When stakeholders have divergent goals, it becomes problematic to maintain an agreement between them. In addition to that, most of the data collectors are not skilled in DW, and sometimes, even in Information Technologies (IT), which raises the possibility that they do not correctly or clearly formalize most of their needs. By this nature of these numerous participants, the conflicts' management becomes a complicated task, especially when they are not employed by the project, which leads to a limited involvement time, and so, they cannot exhaustively, accurately nor correctly define their requirements. This is because that necessitates, in the successful classical design cases as we highlight their key success factors in chapter 3, a full engagement of application domain qualified employees. Handling volunteers' involvement using the existing DW methodologies is not possible since the existing DW methodologies:

- Require advanced knowledge of OLAP main concepts.
- Assume that users are effectively involved in the overall project, which makes all their needed requirements well and completely defined.
- Can generate too many DW models when considering all the definitions of many users, therefore, a very expensive implementation.
- Can consider inconsistent combinations of meaningless or erroneously defined analysis indicators, which is probable with the lack of subject matter knowledge that qualifies the amateur volunteers.

I.3. Main contributions

To tackle these issues, we analyse the data warehouse literature to better approach our specific context accordingly. Then, in the state-of-the-art chapter, we illustrate the overview with the features that we have defined for papers' selection as well as the inspirations that guided therefore our work. Afterward, we define our new generic and participative DW design methodology, which relies on a Group Decision Support System (GDSS) to support the collaboration of volunteer users engaged in the design of the data warehouse models independently of their physical or temporal situations. GDSS are designed to support a group engaged in a collective decision process. Intended to provide computational support to participative decision-making processes, GDSS represent a widely used collaborative technology, which increases users' cohesiveness and decision-making quality. Having that as a group assistance and guidance asset, the proposed methodology is, therefore, participative since it allows, firstly, data warehouse designers to easier elicit requirements and, secondly, the groups of participants to design their DW models collectively. It is a generic methodology, to which we integrate the spatial dimension for the SOLAP part that encompasses the geographic data of VGI. The use of GDSS to help participants solving conflictual definitions has also led us to another interesting improvement of its design. It consists of considering the same nature of users i.e. technical unskillfulness, that encouraged our architectural choices when testing the first GDSS that we have worked with as well as the new one that we implemented to improve the system's suitability to novice and inexperienced users and to overcome the user-experience drawbacks that have been detected during the tests. For that matter, we chose to adopt an acknowledged design pattern from the literature i.e. Thinklets (Briggs and Vreede 2009), that we also suggest in this work improved versions of their scripts, configurations, and a customized implementation of their tools. Thus, we conducted a set of experiments with real volunteer users engaged in solving risk management and uncertainty group problems using the GDSS that we relied upon during our collaborative data warehouse methodology definition. In these experiments, we defined an experimentation protocol that we followed with all the different participating teams. Then, we analysed based on the results i.e. meeting reports and users' feedback, the technical pitfalls, and the user-experience limitations which are a fundamental success factor of such interactive user-interface based systems. With that at hand, we developed a new GDSS system inspired by some classical Thinklets that we introduce to which some user-friendliness related improvements to offer a better user experience as well as a new Thinklet for DW collective evaluation activities. Hence, with their functioning that was personalized to allow more interaction and intuitive representation of the used data, we conducted some experiments in order to validate the newly introduced functionalities. In addition to that, we also implement a recommender system prototype that is based on the user-experience experiments that we've done. It is mainly suggesting,

for inexperienced facilitators, a hierarchical division of decision categories followed by an automatization of certain assistive functionalities that we use for which a question-based recommendation approach.

I.4. Outline

This work is organized in nine chapters as follows:

— In chapter I, “General introduction”, we introduce the general context of our work with a statement of the problem that we are addressing which is mainly handling requirements engineering and design of multidimensional models for crowdsourcing projects. Then, we summarize our main contributions that will be further detailed each in its corresponding chapter.

— In chapter II, “General concepts”, we introduce the technological and theoretical concepts that we consider mandatory to recall their encapsulations, advantages, and newly raised research potentials. We do so by first defining the crowdsourcing context and second highlighting its technical support and various fields of application. After that, we give a definition and examples of the main concepts of the Online Analytical Processing (OLAP) which is a necessarily required knowledge for our work to be followed up. Then, since it is a main methodological and technical contribution of our work, we define the Group Decision Support Systems (GDSS) with their advantages and fundamental use prerequisites. Finally, we introduce the recommender systems that we suggest, in an attempt to simplify the group processes facilitation in GDSS systems, to employ one of the available techniques in the system that we developed, detailed afterwards in chapter VII.

— In chapter III, “Case study and motivational aspects”, we start by describing the VGI4BIO project that represents the main case study of our work and that allowed us to perform empirical tests and validations with real users. Afterwards, we answer three questions that we consider required in order to cover appropriately the state-of-the-art related to our proposal. These questions address (i) the interest of implementing DW systems for crowdsourced data, (ii) the differences between citizen science volunteers and enterprise employees, and (iii) the critical success factors of classical data warehouse systems, that will require a special design methodology to manage them properly.

— In chapter IV, “State of the art”, we first cover the two main data warehouse building phases i.e. requirements engineering and design, from a general perspective. Then, we define a set of features that we rely on to cover in detail from the literature the methodologies of these

two phases. Finally, we conclude with highlights of the interesting findings and a preliminary composition of our approach from a purely theoretical perspective.

— In chapter V, "Collaborative data warehouse design", We detail our methodology that starts with the 'requirements elicitation and modelling' step that we suggest in which two scenarios of collaboration. Next, in the 'solving subject matter issues of requirements' step, we propose to handle the potentially large number of resulting models by a fusion algorithm followed by a collaborative resolution of requirement conflicts. For this conflicts' resolution step, we propose two methods that consist of using a GDSS for the collaborative refinements. Further, we introduce the details of our implementation environment and used technologies for both semi-structured interviews of elicitation and validation and the fusion algorithm. Finally, we validate the methodology with the VGI4BIO case study by designing, with real users, a data warehouse for one of the project partners using the first method of the step 'Collaborative resolution of requirements conflicts' while the second will be tested in chapter VII since it requires a prior solution that we develop later within the new GDSS system that we propose afterwards.

— In chapter VI, "Experiments of the GDSS user-experience", we perform a set of experiments using the GDSS that we used with the first 'Collaborative resolution of requirements conflicts' method of our data warehouse design methodology in order to identify the critical user-experience limitations when used by inexperienced users, so we could tackle them correctly in the new GDSS that we propose in Chapter VII. The experiments have been done in the context of an international project, namely RUC-APS, with teams from four countries that are working on various topics related mainly to risks and uncertainty management when collaborating within the agriculture production systems community. To test the GDSS at hand with these users that have similar profiles to our VGIO4BIO project volunteers, we have put in place a protocol that normalizes the experimentation sessions in order to guarantee a minimum of comparability and reproducibility by other facilitators. We, then, conclude by detailing two of the six run experiments succeeded by the user-experience related feedback.

— In chapter VII, "A new GDSS system", we introduce GROUDA i.e. a new GDSS system called 'GROUp decision & DAta-warehouse', that we tend in which to solve the user-experience general issues that we concluded with in the chapter VI. We start by introducing the concept of Thinklets which is a concept of collaboration that is well-known in the literature and that serves as building blocks in collaborative processes. After that, we illustrate the general architecture of GROUDA followed by the conceptual design of a recommender system that assists the construction of group processes. Then, we detail the implemented Thinklets while focusing on the new interactivity aspects. Amongst these

Thinklets, we implement the CollaborativeDW which allows the collaborative evaluation of DW models and thus the execution of the second method of our methodology of data warehouse design, ‘Collaborative resolution of requirements conflicts’ namely the ‘profile-aware’ method. At the end of this implementation, we provide a set of tests of this Thinklet with the VGI4BIO case study that we validate with both the effectiveness of the collaborative method and the usefulness of the implemented tool.

— In chapter VIII, “General conclusion”, we conclude with a recapitulation of the contributions and the research perspectives of the GDSS systems user-experience and collaborative data warehouse requirements engineering, design, and conflicts’ management.

Chapter II

GENERAL CONCEPTS

Summary

In this chapter we draw the attention to some generalities to ease the access to notions and paradigms that we use later in our work. First, we present here the main concepts of crowdsourcing as it is the source of data that we exclusively use to feed the data models in our data warehouse design approach. Second, OLAP fundamentals which is the core exploitation technology of warehoused data analysis. Third, an overview about GSS that makes, in a part, the elicitation tool facilitating our requirements acquisition, and in the other, a subject of our contribution in this work. Fourth, an overview on recommender systems upon which we rely to better guide unskilled facilitators of group activities in their processes' definitions and techniques' choices. It is important though to mention that some of the main definitions are further reinforced by the motivations and state of the art in the following chapters.

II.1. Introduction

Clarifying the encapsulations and differences that exist between various concepts that we use in this work is the main objective of the definitions and introductive notes of this chapter. Crowdsourcing technologies are the revolutionary source of data that opened the door to many new social, collaborative, and data-centred industrial and scientific fields. To make use of the data generated and collected thanks to (i) the citizen-science normalized collection protocols, (ii) the opportunistic data denormalized collection protocols and (iii) the VGI platforms, defining the technical limitations of each category is therefore a preliminary necessity. These limitations have been the focus of many works in the literature that covered mainly the data quality issues and their

drawbacks in comparison with the traditionally collected data done by qualified and experienced employees (Aitamurto et al. 2011; Allahbakhsh et al. 2013; Traunmueller et al. 2015). In fact, the balance between data quality and its collection and dissemination costs, is still a challenging research subject. However, solutions such as assistive guidelines and rigorous follow-up of participating data collectors' profiles etc. have encouraged the use of crowdsourced data for analysis purposes.

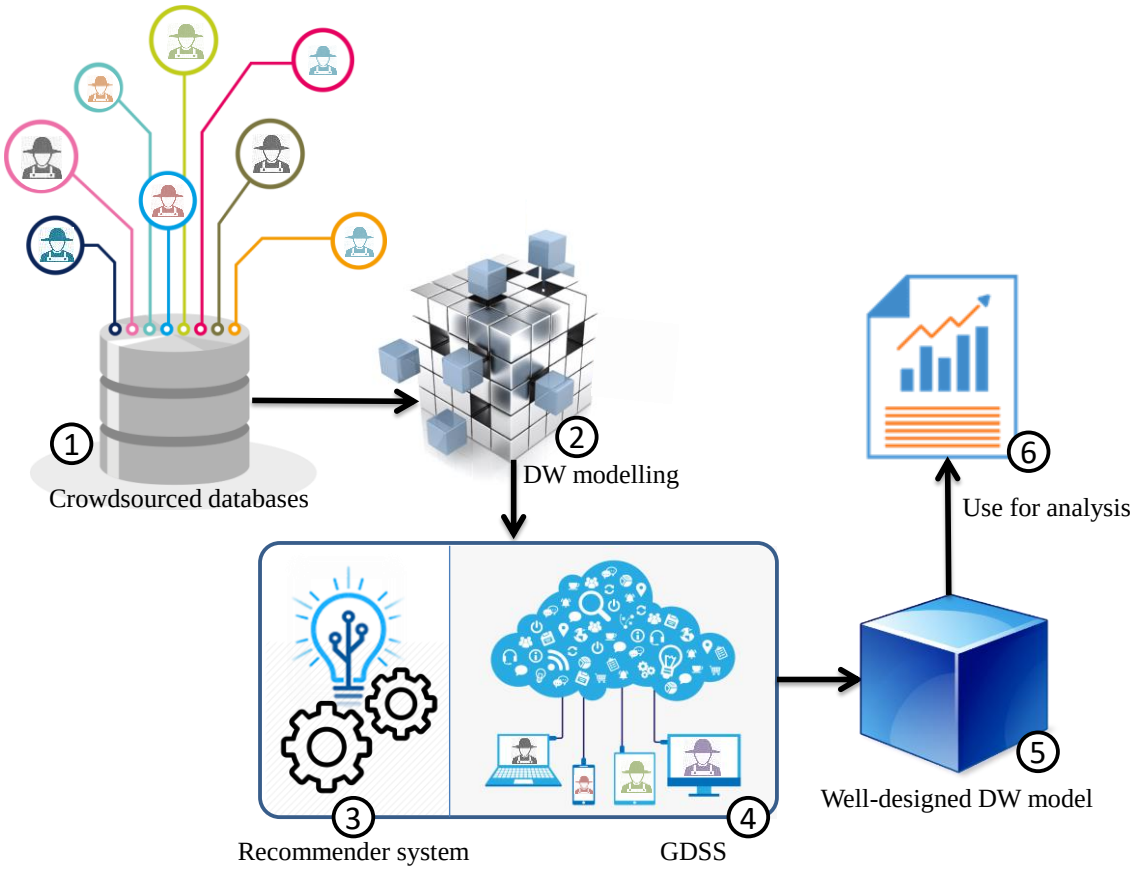


Figure II-1 Introduced concepts functioning sequence

This is where, in our general concepts' illustration, intervenes the OLAP technology that its role is to analyse large amounts of data for decision making purposes. Then, followed by GDSS technology overview which helps handling diverse visions of decision makers that are in our case VGI users, the reason why we introduce, as well, recommendation system's use, on which we rely later in this work to ease involving this specific category of users in collaborative data warehouse design. In Figure II-1, the functioning sequence of the concepts that we introduce in this chapter and use further in this work is illustrated where the numbered steps are:

1) Crowdsourced databases: the source of our data warehouse new design methodology.

2) DW modelling: defining data warehouse models for the crowdsourcing volunteers which we detail, in the remaining, its raised issues that we solve in steps 4.
3) Recommender system: assist inexperienced facilitators to use more effectively the GDSS.
4) GDSS: used to solve the conflictual design disagreements and inconsistencies in the data warehouse models proposed by volunteers in step 2.
5) A well designed DW model that can be deployed and used for analysis.
6) Use of the final system for analysis requests.

II.2. Crowdsourcing

In this section we introduce crowdsourcing phenomena from our work's perspective. As we present in Figure II-2, The technological base of these trending data collection methods is the Web 2.0 technological progress.

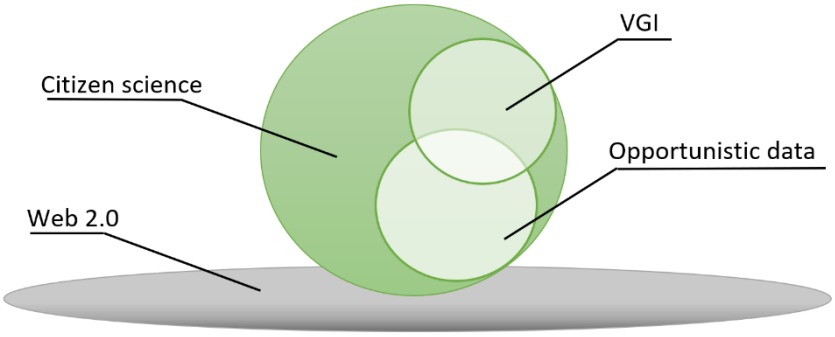


Figure II-2 Crowdsourcing technologies

We show here, what advantages have Web 2.0 brought to the data science community? where citizen science, VGI and opportunistic data overlap? And what are the main differences and advantages of each? To have a general understanding, we do that by covering, First, web 2.0 that offers the technical support allowing for new data paradigms to appear. Second, citizen science as the foundation of a new generation of data collection techniques. Third, opportunistic data as a technical adaptation of the technology use on the field. And eventually, VGI data that is also a trending result of the crowdsourcing concepts with the web 2.0 internet reshape.

II.2.1. Web 2.0

As it has been a question of controversy since its appearance around 2004, the term ‘web 2.0’ wasn’t accepted by all IT professionals and business specialists. Nonetheless, the term, designating a wide range of technology solutions business strategies and IT technical paradigms, have gotten its acknowledgement within both scientific and industrial communities ever since. Web 2.0 can also be defined in comparison with its predecessor ‘web 1.0’ that offers less dynamism and interactivity with web content in general terms.

More than that, according to (Murugesan 2007) Web 2.0 allows more than simply offering interactive use and various web content accessibility levels. It gives designers and developers more flexibility with more responsive interfaces. It tolerates and promotes collaboratively creating contents, which enables as well the re(use) of the cross-platform web services and their creation. Also, it allows the social networks functionalities and content creation that led to the raise of data collection interests. With that understanding, and according to(Constantinides and Fountain 2008), some examples of Web 2.0 application can be categorized into different categories:

- Facebook.com, Instagram.com, Snapchat.com, etc. as Social networks allowing personal content sharing via a customizable platform.
- Python.org, ubuntuforums.org, gaia.com, etc. as Forums that enable sharing ideas about common interests.
- Youtube.com, vimeo.com, soundcloud.com, etc. as Content communities offering platforms of sharing a particular type of information.
- Google.com, yahoo.com, bing.com, etc. as Content aggregators that use rich site summary (RSS) techniques to allow customized web content access.
- Tumblr.com, blogger.com, huffingtonpost.com, etc. as Blogs that are web logs or journals making use of multimedia content and recently Podcasts i.e. video or audio content available for stream or download.

The largeness of what the scope of web 2.0 can englobe, is the obvious explanation of the ambiguity that the term can raise especially when to be adopted by businesses and IT specialists that precision and concreteness of technical used language makes an important aspect of their daily used terminology. To summarize then, we can say that the trend of changes that emerged as web 2.0 are basically about:

- Engagement of users in the development of new and incremental technical solutions and not only considering them as for their customers’ nature. Which leads to reaching smaller communities with narrower needs with not necessarily huge products.
- Centring the focus on service-based and open-source web applications to free the technology of the ‘software as a product’ vision with all its expenses and difficulties both

for the users as consumers and the developers as producers, to the new generation of 'user-based web services' with its technical simplicity and user-friendliness aspects.

II.2.2. Citizen science

"Citizen Science" is a term that have been first chosen in the US by Rick Bonney to refer to the participation of the public to the engaging science communication projects in 1995 (Bonney et al. 2009) and at the same time in the UK by Alan Irwin that promoted the necessity of scientific openness and science policy processes openness to the large public referring to it by concepts of scientific citizenship (Mowat 2011). These two definitions have both influenced the scientific perspectives ever since. However, we here talk only about the former i.e. Bonney's definition that deals with the public's engagement not with policies' data publicity. With that understanding of "Citizen science", many projects in different disciplines have been designed and focusing on roles given to volunteer participants to, in best cases, give a shared benefit that keeps the continuity and the motivation of the amateurs to carry on delivering data, performing an analytical task, doing observation etc. (Silvertown 2009).

As the early citizen science projects were limitedly accessible to few privileged participants, it now has become available to all due to the unprecedented spread of communication technologies and to the emergence of new suitable application fields of high scientific interest such as invasive and endangered species, climate change phenomena, biodiversity conservation, ecology and water quality monitoring etc. Thus, that larger range of volunteers' profiles has led the scientists of data analysis communities to deal with doubts about the collected data reliability levels. More precisely, the representativity of the data samples collected by unspecialized participants and its accuracy related issues. As the data collection might be based on an observation protocol defined by the scientists or an explicit sampling design to assist the data collector / observer, this, therefore, is another level of data quality limitation that is worthy of consideration.

II.2.3. Opportunistic data

Opportunistic data is citizen science data collected without standardized field protocol and without explicit sampling design (Strien et al. 2013) Which, obviously, raises several problematic data quality issues and reliability points that are mainly, along with others:

- Incomplete reported data because of biased reporting.
- Erroneous reported data because of biased interpretation.

— Partially or completely changed data over time leading to contradictious reports on the same phenomenon.
— Selective misleading data observation whilst absence of restrictions to guide participant's interests.

Notwithstanding, the exploitation of opportunistic data is still very valuable to data scientists to make use of the huge irreplaceable amounts of produced data in its communities. In addition to that, opportunistically collected data is getting increased with the facilitating internet portals, with the spread of mobile phones in possession of amateurs of nature and biodiversity/ ecology aware people and with the lack of standardization of data collection norms even in standardized portals.

In order to solve these issues, many data quality improvement approaches have been proposed. To cite some, we can mention (Molinari-Jobin et al. 2018), that propose data collection procedures to be tested and validated as simple and reliable assisting assets. Standardization of monitoring protocols that are validated in realistic circumstances by professionals (Lin et al. 2017). Also, an important factor that has not been the focus of many studies as said (Newman et al. 2010) in their survey, which is the user-friendliness of the used tools that have been the work of (Palaghias 2017) for example.

II.2.4. Volunteered geographic information (VGI)

With the same interesting technological advancements that we have previously introduced, one of other trending phenomena that are interesting IT communities in the same context, is volunteered geographic information (VGI). It can be defined as the set of tools allowing individual volunteers to collect, create and disseminate geographic data. Also called Georeferenced crowdsourced data, VGI have been first coined by Michael Goodchild in 2007 (Goodchild 2007) to refer to the geographic data generated collectively by the voluntary participation of private and non-specialized citizens to collaborative platforms such as Wikimapia[1] or OpenStreetMap[2]. With that definition, VGI is also seen as an invaluable asset to the geographic data consumption that has rapidly risen with the web 2.0 and the geo-localization services that have emerged consequently. VGI is also to be distinguished from the geographic information systems (GIS) that are systems designed to capture, save and use spatial data, however, its use is restricted by its nature to experts and with a

[1] www.wikimapia.org
[2] www.openstreetmap.org

highly normalized functionalities that are the only available set of tools offered by steadily designed processes.

Google maps and WAZE are also widely used VGI platforms that rely on a structured geographic representation concepts and data enquiry forms to delimit the unspecialized users' participation inf the data collection, manipulation, description and presentation. With these semi-assisted software models, developers of VGI platforms tended in most cases to reduce the geographic information reliability weaknesses impacts. This reliability limitation is the same issue of all the crowdsourced data that came to existence with the web 2.0 technologies and for which, similar correction approaches have been put in place. Moreover, the VGI data have its additional critical factors that are forming a further level of restrictiveness. Confidentiality of oneself or neighbour's data may be a better example of that issue (Goodchild 2008), where the data is locally public, but, its dissemination for the larger public might be a question of personal data privacy violation.

II.3. Online Analytical Processing (OLAP)

A clear definition of OLAP is conceived in contrast with On-Line Transaction Processing OLTP (Tan 2005). The OLTP is known to refer to operational or transactional databases that do the daily data operations such as INSERT, DELETE, UPDATE, etc. whereas, OLAP is designed to offer an analysis-oriented system that allows complicated and read only queries, that necessitate fundamental alternance of the in-place data sources to reach the needed performance permitting so. OLAP technology allows powerful analysis capabilities offered by its multidimensional data representation modelling. It allows various supportive data formulations that are basically outputted as complex measures i.e. calculations that are based on unbounded abstraction levels to offer meaningful statistical analysis with flexible aggregations and dynamic visual representation. Also, business trends analysis and hidden economical phenomena discovery are, among others, some main use cornerstones of OLAP applications. Especially in business intelligence i.e. the first and main prosperity field of data warehousing and OLAP technologies, the use of OLAP has become a usual practice to tackle some common issues like data insights reporting, strategic planning, budgeting prioritization, performance improvement, etc.

OLAP datasets have to encompass historical data collected over at least a few years to allow, for example, a business growth assessment and sales or market expansion improvement evaluation and, therefore, foresights. An enterprise that wants to analyse its data must face the challenging task of putting it all together whilst it comes from various sources and in different formats. DW technology

offers this advantageous data sets transformation to allow fast and complex calculations and data queries which is the essence of quicker decision making.

The result of a DW project, that allows its exploitation by OLAP applications, is conceptually composed of two types of tables:

— Dimensions: many analysis axes, each of which is composed of at least one hierarchy of many levels. For example in Figure II-3, we have the “Time” dimension that is composed of three hierarchies: (I) day → week → month → year, (II) day → month → year, (III) day → season → year, where “day”, “week”, “month”, “season” and “year” are levels of abstraction.

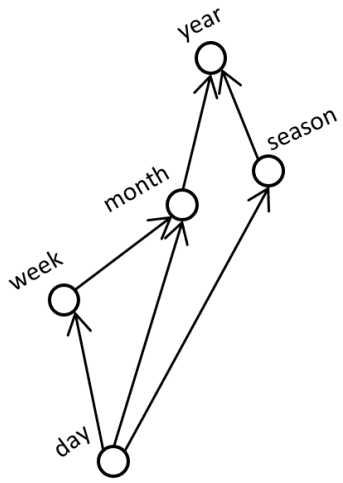


Figure II-3 Dimension "Time" design Example

This representation allows the navigation between levels of details of each hierarchy whilst aggregation is automatically done by the OLAP system. An example of a dimension table is shown in Table II-1,where we have the levels: date_y (year), date_m (month), date_d (day) and season. Having this representation, the configuration of different hierarchies permits aggregations without having to join other tables when running queries due to the denormalized design unlike operational databases constrained by normal forms rules.

Table II-1 Dimension "Time" table example

date_id [PK] integer	date_y integer	date_m character varying	date_d date	saison character varying
1	2010	2010-01	2010-01-01	Winter
2	2010	2010-01	2010-01-02	Winter
3	2010	2010-01	2010-01-04	Summer

— Fact(s): it is where subjects of analysis along with measures allow us to analyse them with reference to dimension tables. For example, in Table II-2 every line represents a subject of analysis which, in this case, is a product. A line can be read as for example "The quantity sold out of **210** stocked items of the product with *Product_id '**2**'* in the shop located in *Location_id '**12**'* at the date with *Time_id '**5**'*, is **128**".

Table II-2 Fact «Sales » table example

Product_id [PK] integer	Time_id integer	Location_id integer	Sold_qty integer	Stocked_qty integer
2	5	12	128	210
3	5	11	10	300
6	3	12	88	390

Hence, to have a properly designed OLAP system, the available data must be reformulated in order to allow executing complicated queries with an acceptable performance. This process is called ETL i.e. standing for Extraction, Transformation and Loading, that must be done prior to the requirements elicitation and multidimensinal modelling. Many OLAP projects rely, in their intended analyses, on several heterogeneous data sources, which needs an ETL that is crucial and time consuming, unlike in our case study which relies only on crowdsourced databases that we introduce in the following chapter. A thorough literature review of data warehouse requirements engineering, and design approaches is presented in this work since these two phases represent where our contribution resides as described in the chapter 1.

II.4. Group decision support systems (GDSS)

Group support systems (GSS) were defined by (Jr et al. 1996) as *"interactive computer-based environments which support concerted and coordinated team effort towards completion of joint tasks".* According to (Briggs et al. 2003; Helquist et al. 2008; Jr et al. 1996), the use of GSS have explicit and implicit advantages that must be understood by the responsible leaders to reach the best of their potential team-oriented productivity. The explicit benefits are mainly but not exclusively, a better problem definition, more group cohesiveness, a higher number of solutions with better quality, and stronger team commitment to those solutions' adoption and implementation. The implicit ones include a remarkable reduction of staff's engagement time to reach final decisions and budget savings thanks to the boosted productivity.

The original purpose of Group Support Systems, also called Group Decision Support Systems (GDSS), is also including the exploitation of opportunities that information technology tools can offer to support group work. Many studies have evaluated GDSS and showed that they can improve the productivity by increasing information flow between participants, by generating a more objective evaluation of information, by improving synergy inside the group, by reducing time, etc. (Fernandez et al. 2011; Jr et al. 1996; Vreede 2014; de Vreede and Briggs 2018). All these studies highlighted that the efficiency of using GDSS depends strongly on the facilitator role. Group facilitation is defined as a process in which a person, who is considered as trustworthy by all members of the group, intervenes to help improve the way they identify and solve problems and make decisions (Schwarz 2002). Unfortunately, professional facilitators are difficult to hold in position by their organizations and their disappearance therefore, often entail the abandon of using GSS (Briggs et al. 2003). For that reason, recommender systems might be an important asset when the facilitator is not fulfilling these highly expertise requirements, thus, our aim of assisting his/her use of GDSS by recommendation of the choices of decision processes and collaborative techniques.

II.5. Recommender systems

When a system is suggesting many various items to its users, the challenge of targeting a likely convenient set of propositions to each of them has been the focus of many commercial and academic research approaches (Shani and Gunawardana 2011). The recommendation or recommender systems are electronic and sophisticated versions of the social behaviour of recommending what a person presumes interesting to another. Thus, following the same logic, commercial online platforms are suggesting products to users when they buy or add an item to their shopping basket. A set of books of the same author, video games provided by the same company, videos uploaded by the same channel, friends from common communities etc. and many other examples that every today's internet user is exposed to. As the link in these examples is simple to deduce, other more

complicated recommendations are facing the challenges of audience conversion (Michelle et al. 2017) using advanced multi-criteria-based approaches. The focus then, have covered among others, users' categorization into different profiles to allow simulating similar behavioural habits (Chen et al. 2015). Also, users' satisfaction reports and interactive contents such as rating items and liking them, is recorded and used as additional criteria to better filter what to suggest to whom (Candillier et al. 2009). Recommender systems are usually classified into three types (Candillier et al. 2009):

1) Content-based filtering i.e. using elements' qualitative descriptions to build thematic classification of user profiles based on user preferences. For example, for a books' store it would give replies such as" *user likes politics and doesn't like romance*". Which means that for an article to be recommended to a user, its description must be in accordance with his/her profile. For further details on the used techniques and approaches, many works have used content-based filtering in their proposed methodologies. For example, the works of (Meteren 2000; Vanetti et al. 2011) propose two different approaches, the first for content collection restrictively from user positive feedback, and the second, relying on machine learning classification to enforce messages filtering in online social networks.

2) Collaborative filtering, where the prediction is based on the user's appreciations of items that he/she rated. Based on which, the notion of users' neighbourhood is created according to the similarity of their appreciations. Likewise, products' neighbourhood is as well created, which allows by combining both, to filter whichever product to whoever user. The work of (Herlocker et al. 2000), gives a detailed explanation of the collaborative filtering, since they assume necessary explaining what industrial solutions do not explicit, in order to improve the automatic collaborative filtering systems' acceptance. They as well analyse experimental data to confirm the proposed explanations. For more specifications on collaborative filtering techniques and algorithms, the work of (Schafer et al. 2007) covers thoroughly the concept in both its theoretical and practical details.

3) Hybrid filtering that combines both collaborative and content-based filtering methods. In order to improve the efficiency and accuracy of the recommendation process, the hybrid filtering is implemented to overcome some limitations that are associated with the separate use of content-based or collaborative filtering. For instance, the cold start problem is due to the fact that in collaborative filtering, The recommendations of the items are based on the history of user preferences in a way that new users will have to rate enough items to allow the system counting and recording their preferences and therefore offering a precise recommendation (Lü et al. 2012). Another problem that hybrid processes tend to solve is the overspecialization of content-based systems that fail to diversify the recommendations when suggesting items with a very narrowed characterization rather than giving new convenient

propositions to the user (Jain et al. 2015; Lü et al. 2012). The hybrid filtering is meant also to solve the sparsity problem, which is caused by the lack of knowledge about available items. In spite of the number of their purchases that might be high, when those items have an almost empty history caused by a very limited interaction of users with most of which, the recommendations will be misleading to a minor set that has been rated. Proposing a combination of the advantageous parameters of each technique to handle these issues and others like them (Jain et al. 2015), is what makes the effectiveness of hybrid recommender systems (Thorat et al. 2015).

Another approach that has been also adopted is the question-based recommendation as used by (Kung et al. 2003; Palma et al. 2012; Pavlic et al. 2014) for design patterns' recommendation. It relies on a set of questions addressed to potential beginner developers in an interactive session to orient their choices of the design pattern that is appropriate for each type of programming project. Other works have also used similar approaches i.e. the question-based recommendation, but for different objectives, such as (Psaraftis et al. 2018) that recommend a privacy model to apply an optimum anonymization of the data stored in relational databases to reduce information loss when confidential data is handed out to be analysed by different interested parties. Since it has been used successfully in these cases, where complications related to the lack of initial users' profiling and initialization of ratings of proposed items are not problematic, we have chosen the question-based approach for our GDSS. Its use consists of recommending an appropriate group activity process that we detail its functioning and parameters in chapter 7.

II.6. Conclusion

Crowdsourcing is the invaluable source of massive and highly important data that motivated a lot of technical solutions nowadays to even exist. In this chapter we briefly introduced the data types and some general definitions of concepts that these trending techniques came with. Also, we introduced OLAP systems with a basic example explaining the operations of its main components since the design of a specific type of data warehouse systems is our main contribution in this work. In addition, GDSS and recommender systems, were as well defined with a highlight of the fundamentals of their use and added value. For the GDSS, it also makes an important part of our contribution that will be detailed in the following chapters all with the recommender system that we propose for GDSS user-experience improvement purposes. With that introduction to the main concepts that we use in our work, we can start pointing to the detailed questions that motivated our contributions, to handle in precision the related works from the literature with an oriented view and a focus narrowed down to the specific issue that we tackle.

Chapter III

CASE STUDY AND MOTIVATIONAL ASPECTS

Summary

After having the main concepts of crowdsourcing, OLAP systems, GDSS systems and recommendation techniques, scoped in the previous chapter, in this chapter we illustrate the motivating aspects that encourage our work proposal. We start by detailing the case study settings of the VGI4BIO project within which our experiments and tests will take place. To do so, in the first section we introduce the project and its working parameters and environment as well as the underlying facilities that suit our scientific objectives. Next, in the second section we illustrate our motivations of analysing crowdsourced data. In the third section, we define, using a set of criteria, the parameters of differentiation between classical and volunteer users. Eventually, we cover the data warehouse critical success factors in classical working environments in order to better define the specific parameters of our design methodology and accordingly its evaluation later in this work.

III.1.Introduction

To have a clear view on our motivating aspects, here, we straightforwardly answer the questions: why the development of DW systems fed with crowdsourced data is interesting? What are the differences and specificities that have citizen science volunteers against enterprise employed users, so that the design approach gets them accounted? And what are the critical success factors that we need to be aware of while dealing with this new kind of system? Answering these questions is an introduction to the following chapter where we detail a state-of-the-art that we view limitedly from the angle of our context; not in a general manner that might be exhaustive to the reader and covering

a vast range of literature which is not necessarily contributing to our work nor inspiring for our proposal.

III.2. VGI4BIO project

VGI4BIO[3] is a research project financed by the French national agency of research (ANR) that aims to engage volunteer users and to define a set of data centred methods for the analysis of farmland biodiversity indicators. In this project, the farmland represents the areas dominated by agriculture, and includes cultivated areas, pastures and cropland/natural vegetation mosaics. In this context, we mobilize two VGI databases, namely:

- Faune-Aquitaine[4]: Biolovision database, which is fed and maintained by the league of birds' protection – LPO for "Ligue pour la Protection des Oiseaux" in French.
- and OAB[5]: Agricultural Biodiversity Observatory – for "Observatoire Agricole de la Biodiversité" in French.

This, in order to build OLAP applications to analyse farmland biodiversity indicators. Faune-Aquitaine and OAB have 7682 and 1500 volunteers, respectively, who are crowdsourcing data. Among the possible users interested in analysing these data, we have identified many volunteers belonging to diverse categories. For instance, farmers who are interested in analysing biodiversity data in relation with their farming practices, environmental non-governmental organizations needing to visualize biodiversity trends, and French public and private organizations (Regional Direction of Environment and Housing – DREAL, Chambre d'Agriculture, etc.).

Whilst participatory sciences have already successfully demonstrated their interest in different fields of applications as we have mentioned in the previous chapter, the French Ministry of Agriculture has set up the OAB to study the impacts of agriculture on biodiversity with large spatial-temporal data sets.

The OAB is based on the voluntary and "free" contribution of farmers throughout France. The OAB started in 2009 to respond to a proven lack of indicators for monitoring the state of biodiversity in agriculture. Four protocols are currently proposed concerning taxa chosen for their link with agriculture: "Solitary bee nesting boxes", "Butterfly transects", "Earthworms" and "Terrestrial invertebrate plates". The OAB aims to document the impact of agricultural practices on biodiversity. It is based on voluntary contributions from farmers. More than 400 farmers have participated in the observatory since 2011 and have collected more than 500,000 observations.

[3] www.VGI4BIO.fr
[4] www.faune-aquitaine.org
[5] www.vigienature.fr

Since 2007, the LPO has set up a web interface for the collection of data concerning fauna (birds, butterflies, dragonflies, mammals, etc.). More than 5 million observations performed by more than 9000 observers are stored in a Faune-Aquitaine database. These geo-referenced data are invaluable for describing and understanding biodiversity and thus allowing for better consideration in human activities.

Having that project as a case study is such an opportunity that allows to empirically test designing data warehouses for crowdsourced data. Furthermore, involving volunteers from Faune-Aquitaine and OAB in designing the possible DW schemata used to analyse their agro-biodiversity data, and as well to develop a methodology to deal with several problematic issues that this will raise, is what makes the validation environment of our work.

III.3.Why data warehouse systems for Crowdsourced data?

With the proliferation of the Web2.0 technologies that changed categorically the ways of data collection and representation, new analysis and exploration of the called "crowdsourced data" i.e. vast amounts of data provided by citizens to websites and online databases (See et al. 2013), has emerged. Such involvement of citizens in scientific research, or 'citizen science', has proven its effectiveness in several situations such as in (Clery 2011; Khatib et al. 2011; Nayar 2009; Miller-Rushing et al. 2012). The nature of crowdsourced data implies issues such as accuracy, large volumes, data credibility and heterogeneity, etc., that data would potentially contain due to the differences in terms of reliability between volunteers and official agencies' employees in traditional cases. However, crowdsourced data brought a new powerful type of infrastructure that allows the collection, synthetization, verification and redistribution of data through databases, geo-location technologies and mobile devices. In (Herrera et al. 2015), authors emphasize on the valuable contribution that can BI systems have by adequately processing the VGI data that has become in recent years huge due to the unprecedented growth of geographical information crowdsourcing.

Other works have also addressed warehousing crowdsourced data such as (Bimonte et al. 2014) that proposes a quality-oriented framework to do so. They build their approach of analysing VGI data using Spatial OLAP, among others, on the benefits that crowdsourced data has shown in managing environmental risks and crises in situations such as the Haiti Hurricane where citizens have voluntarily uploaded geographic data to OpenStreetMap (Haklay and Weber 2008) after the earthquake (Zook et al. 2012). Indeed, many application domains have become in an indispensable need to analyse their crowdsourced data. For example, after the concept of internet of things (IoT) had risen up during the last few years, big amounts of data are being accumulated from various sources like cars tracking devices, tickets printing machines, induction loops, tollways collection

systems, etc. (Flanagin and Metzger 2008). According to (Gusmini et al. 2017), considering this high level of complexity that have several real-life interesting scenarios, the involvement of individuals to crowdsource data about ongoing events is then the most reasonable technique. Otherwise, the costs and the efforts that would cover in the same circumstances the unexpected events will be, if even possible especially in human resources terms, unaffordable. As the technological equipment is no longer an issue nowadays with the wide availability of smart and wearable devices (e.g. Smartphones, smartwatches...), the improvement of the ongoing events' understanding and behavioural patterns' discovering are key assets to a better decision-making that uses this integration of crowdsourced data within the DWs (Gusmini et al. 2017).

Wherefore, the nature of crowdsourced data as huge and extremely valuable from an analytical perspective lies at the heart of DW decision-making supportiveness, which is its main aspect that all (Bimonte et al. 2014; Gusmini et al. 2017; Herrera et al. 2015) have emphasized on. Also, since users are the core of succeeding such a task (Chen et al. 2000), we put therefore in the following more emphasis on the differences that may be distinguished between crowdsourcing volunteer users and traditional DW organizations' employee users, which gives a further asset to our proposals.

III.4.Crowdsourcing versus Enterprise users: what differences?

In (Bimonte et al. 2014), authors compared crowdsourcing and DW users and cited, in spite of the different influences on data quality, that the unfamiliarity of volunteer users with OLAP tools is higher than in the case of enterprises' users, which is defined in Table III-1 by the criterion "knowledge of the DW fundamentals".

Table III-1 Volunteer versus enterprises' users

Criteria	**Involved users**	
	Volunteers	**Employed**
Knowledge of DW fundamentals	None / Very low	Low / Medium
Involvement in the overall BI project	Partial	Full
Geographical distribution	Very high	Very low
Understanding of the project goals	Low / Medium	Medium / High
Possibility of reaching unified vision	Very low	High
Proficiency in the subject matter	Medium	High / Very high
Availability to elicitation sessions	Low	High / Very high
Number	Very high	Very low

They also mentioned that the large number and the variety of backgrounds that crowdsourcing users have, makes it much more complicated to collaborate and to discuss, what would be simpler in the case of limited number of involved and experienced participants when it comes to employed users. Users of VGI systems are according to (Goodchild 2007) a huge number of internet users that collaborate to collect geographic data in a distributed manner. They are in fact, very often, biased and with specific interests that, by using citizen science systems, intend to share information and collaborate with other users (Fischer 2012), and not to commit themselves to a fully engaging process such as DW design. Thus, making crowdsourcing users unifying their vision about a common goal when participating to DW design is very complicated compared to what can a group of enterprise's users reach after sessions of training such as those qualified by (Hwang et al. 2004) as a critical success factor, which reflects in our comparison the criterion "Possibility of reaching unified vision". Furthermore, whether employed users are trained or not, the fact that they belong to the same enterprise makes it obvious that their knowledge of the application field is, at least, guaranteed to be over a required minimum of proficiency. By this token, in (Vaisman and Zimányi 2014), DW design engaged users are defined to be usually profiled as business owners, managers or employees and then, experienced and aware of the strategic goals, which is illustrated in Table III-1 by the criterion "Proficiency in the subject matter". Consequently, in terms of "availability to elicitation sessions" employed users are highly available to participate in the RE step of the design process, which is a sensitive step that can affect the success of the system if it is not correctly accomplished. On the other hand, citizen science users are very limitedly available due to their nature of non-engagement i.e. "involvement in the overall BI project" and "geographical distribution" of their collaboration (Bimonte et al. 2014; Flanagin and Metzger 2008).

We consider in the remaining, the differences between crowdsourcing users and employed users that we have defined here, in order to better classify the selected design methodologies in the literature as well as to adequately qualify the functionalities of elicitation techniques and methods to better suit volunteer users' specificities. In the following section we overview the DW critical success factors that play an important role in DW systems' goals accomplishment that therefore might differ considerably with our special case of relying on citizen science data as the unique data source.

III.5.Critical success factors of data warehouse systems

Assuming the nature of business intelligence (BI) systems' implementation as complex, costly and resource intensive, the attention in the literature have been drawn to the critical success factors (CSFs) for BI systems implementation in order to produce assessment frameworks or guidelines

relying on which implementers would have a clearer understanding of the factors affecting their system's construction.

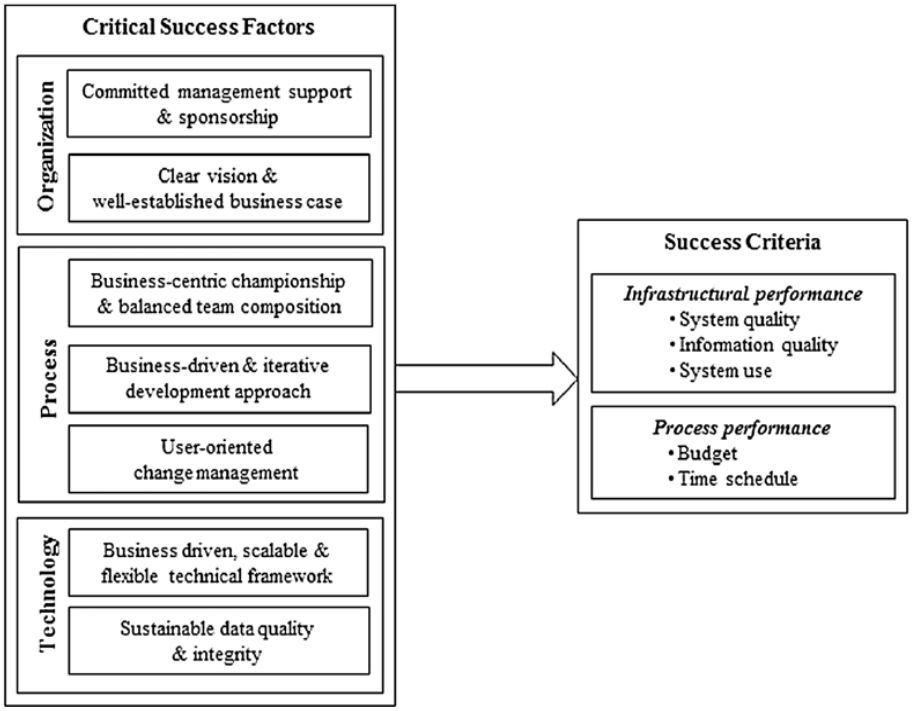


Figure III-1 Critical success factor from Yeoh and Koronios 2010

In the work of (Yeoh and Popovič 2016), a CSFs framework was detailed and used in analysing seven large organizations. In Figure III-1, authors of (Yeoh and Koronios 2010) defined the set of CSFs reused by (Yeoh and Popovič 2016) in their set of experimentations that concluded with the following:

— Enterprises that had an initial well-established business case, clear business vision, committed management support and a business-side sponsorship, are the more likely to succeed in implementing BI systems that are cross-functional and business-driven which is the need of most enterprises.
— Successful implementations were those that made organizational CSFs the cornerstone based on which they addressed the process.
— On the other hand, the case that failed had focused primarily on the technological side and neglected the organization's core requirements.

— One of the firms has failed in its BI system implementation because of a non-clear definition of requirements and business needs. These early phase business issues made them continue the system implementation with multiple versions of the same data.

By their study of the processes, organizational and technological factors that influence the implementation of BI systems, Authors (Yeoh and Popovič 2016) confirm that a well understood, significantly addressed and correctly prioritized CSFs are keys of BI systems implementation success.

Nevertheless, there are no commonly agreed upon success criteria to asset the BI systems implementation nor to adopt its appropriate architecture (Ariyachandra and Watson 2010). However, in (Ariyachandra and Watson 2010) eleven factors have been enumerated and considered as those affecting the architecture selection decision and claimed that the most important among them are information interdependence between organizational units, strategic view of the DW, and upper management's information needs. (Hwang et al. 2004) emphasized that the top management is the most important factor among others in the context of DWs adoption in the banking sector. Contrariwise, (Hwang and Xu 2007) deduced after empirical study that the top management involvement is insignificant and only considerable as an indirect CSF. Though, such contradictory conclusions show us how assessing DW systems might be complicated and affirms the fact that DW systems' design is a complex task that must be done with carefulness and awareness of the multitude of dependencies that every system might have.

In addition, other aspects of success factors have been evoked by other works such as the major role of organizational information centres, i.e. units that train and support users engaged in DW activities in the work of (Chen et al. 2000) that studied users' satisfaction with DWs by surveying 42 case studies. They concluded their findings by pointing to the fact that DW systems are quite new to many organizations, which implies the necessity of users' training, so they develop an awareness about the data residing in the DW.

The assessment of DW's success reveals, additionally to the common factors such as the ease of use, good quality and rapidly retrievable data, better decisions or productivity improvement, it reveals also some uncommon factors that might rise only in specific or unusual contexts. Therefore, the consideration of these specificities, whose influence can cause the system's failure if not properly handled, is mandatory. Accordingly, a variety of criteria can be defined to asset effective and efficient DW systems' implementation, especially when it comes to crowdsourced data. Furthermore, in the following chapters we detail our choice of using group support systems to perform the requirements engineering task for collaborative DW design.

III.6.Conclusion

In this chapter, we expressed the motivating aspects that explain our use of crowdsourced data with DW systems and the critical success factors that must be considered with consciousness along with the differences that exist between citizen science and enterprises users. We also presented the VGI4BIO project that allows us an experimental case study within which we will be able to validate our design approach that we illustrate in following Chapters. Next chapter covers existing works related to different aspects that we consider in order to effectively involve crowdsourcing users in the DW design.

Chapter IV

STATE OF THE ART

Summary

In this chapter we start by investigating the general aspects of requirements engineering in section IV.22, and data warehouse design in section 3. This is done with a general perspective to allow a better understanding of the detailed review that we illustrate in the fourth section. After having established this basis about the existing works from these two stages that we are scoping i.e. RE and DW design, we discuss in section 4 based on a set of features that we define considerably to formalize our solution, the interesting aspects of the selected works. We detail afterwards in subsections 4.2 and 4.3, the specificities of every approach of requirements engineering and design that we cite in Table IV-1. In subsection 4.4, we highlight some additional interesting findings from the literature that we consider relevant to our proposal. And finally, we detail in subsection 4.5 our motivations of using GDSS for RE in DW systems.

IV.1. Introduction

The motivational aspects that we illustrated in the previous chapter provided general answers to the feasibility of our collaborative DW approach. The main idea of this chapter is thereby to accordingly investigate what has been done in the literature even if the existing works do not cover precisely the raised point. Doing that by analysing the existing literature means that we tend to (i) reuse what might directly solve a limitation that we consider crucial for our work, (ii) define new solutions for what have not been covered yet and (iii) rely on what others' experiences have led to conclude with which in more or less similar situations.

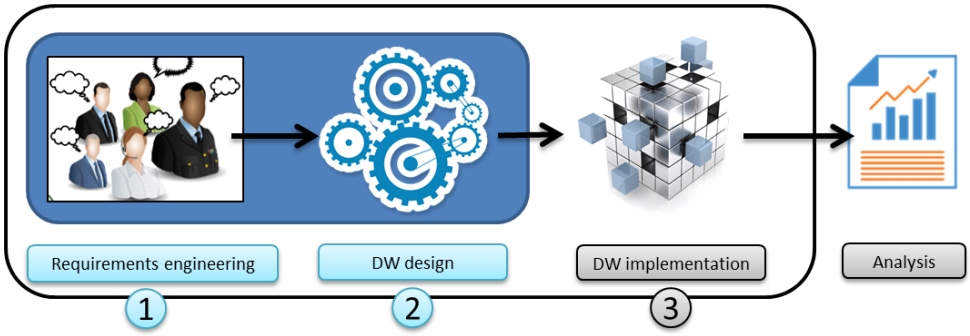


Figure IV-1 Data warehousing cycle steps

Several DW design methodologies have been proposed in the literature in various attempts to reduce the complexity of the crucial and meticulous task of efficaciously designing DWs and to normalize the most of its recurrent and time-consuming requirements engineering (RE), modelling and deployment phases. In order to deal with the specificity of designing DWs collaboratively by engaging citizen science volunteers in the design process, in this chapter we limit our interest to the first two stages of the data warehousing cycle i.e. requirements engineering and DW design; first and second steps in Figure IV-1.

IV.2. Requirements engineering from a general perspective

Requirements elicitation is the first of the four stages of requirements engineering process (RE). The other steps are analysis, specification and validation. In (Zowghi and Coulin 2005) the authors defined it as "*the process of seeking, uncovering, acquiring, and elaborating requirements for computer-based systems*". In traditional information systems, after the identification of the stakeholders and the different sources of requirements, the use of the selected techniques, approaches and tools starts the elicitation of the core requirements based on the needs of stakeholders and especially the system users. More precisely, activities of the requirements elicitation process can be divided into five different basic types as shown in Figure IV-2.

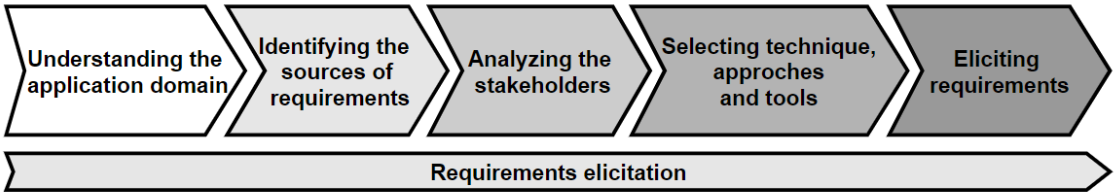


Figure IV-2 Requirements elicitation steps (Zowghi and Coulin 2005)

The first activity is understanding the application domain i.e. an initial step towards starting requirements elicitation is to get a sufficient knowledge of the application domain i.e. the real-world environment for which the system is conceived to reside and operate in its circumstances (Lopez-Nores et al. 2006). The different aspects of environment needs must be exhaustively covered during this step, along with the potential constraints that may influence the system, while focusing on issues to solve and key goals to attain. The second step is identifying the sources of requirements i.e. among others, stakeholders represent the most important requirements source of the system. However, requirements are often available in a variety of sources and represented in different formats. Domain experts and system's current users can also be a source of information describing user needs and system problems (Lopez-Nores et al. 2006; Mishra et al. 2008). The third activity is analysing the stakeholder's i.e. to involve all the people that are affected by the system, whether they are external or internal groups and/or individuals (Mishra et al. 2008). In the literature, the potential project stakeholders that might be consulted during the requirements elicitation are investigated (Agarwal and Tanniru 1990). The fourth is selecting the techniques, approaches and tools to use. Generally, the requirements elicitation technique, as being crucial for the elicitation process, needs to be chosen based on the specific context of the project (Agarwal and Tanniru 1990; Mishra et al. 2008). Eventually the fifth activity, which is eliciting the requirements from stakeholders and other sources i.e. after the identification of the stakeholders and the different sources of requirements, the use of the selected techniques, approaches and tools starts the elicitation of the core requirements based on the needs of stakeholders and especially the system users. During the elicitation, the future processes that the system needs to perform in order to reach the main objectives of the business, must be determined (Lopez-Nores et al. 2006; Mishra et al. 2008).

Usually the requirements elicitation is an incremental task performed over multiple sessions in an iterative way in order to increase the detailing levels and to consider all the related influencing aspects illustrated in Figure IV-2 (Zowghi and Coulin 2005). Cost and time constraints are in fact the factors that often determine whether the elicitation step has been completed or not, and not by achieving a required level of quality or completeness. Typically, the resulting output of this process is diagrammatic representations and natural language definitions of the set of requirements with other information like identification of the sources, stakeholders, priorities, and rationales (Mishra et al. 2008).

Furthermore, in order to perform the requirements elicitation effectively, analysts must significantly play the fundamental role of facilitator during work sessions ensuring all the while giving sufficient

opportunity to contributors and making them feel confident and comfortable (Zowghi and Coulin 2005). In addition to recording answers of the asked questions, they must also be able to assist and guide participants to cover the relevant bunch of requirements' details, constraints and dependencies for the sake of obtaining complete, consistent and correct requirements information (Lopez-Nores et al. 2006; Mishra et al. 2008). Consequently, as it is a main aspect of our work to study the DW systems' design by involving citizen science users in its whole process, in the following section we present a literature review in the scope of the participation of many users with heterogeneous business goals to the RE phase.

IV.3. Data warehouse design from general perspective

To design a data warehouse system, there are three modelling levels that must be distinguished and consequently accomplished:

- Conceptual modelling: It is the highest level of design in which we represent the different relationships between the different entities of a data cube. Its objectives are not to tackle any implementation dependencies nor to think of the data types or technical attributes of the entities to consider. It comes straight after the requirements definition step in order to represent, understand and overview the concepts of the model at hand.
- Logical modelling: Adds the logical details to the conceptual model by specifying the primary and foreign keys, by breaking down what was generically defined in the conceptual level, so it becomes architecturally suitable for any implementation technology, and resolve calculation, many-to-many and denormalization issues.
- Physical modelling: The lower level of design in which the implementation technology is to be chosen and the model is to specifically define all the entities, relationships, data types and constraints of the database according to the DBMS. This is the model that will be followed literally during the implementation and which contains, in addition to the logical level, all the needed technical specifications that might differ from a technology to another.

In this work, we perform the logical and the physical design phases in our implementation step, however, the focus is mainly put in the requirements engineering and conceptual design steps. Even though these modelling levels are agreed upon in the data warehouse literature, existing works have defined different approaches in relation to "where to start from" and "what to consider more or first". More precisely, whether to start from stakeholder's requirements and limit, therefore, the considered data to only those needs, or to start from the available data and just deliver all the

possible analysis subjects. The differences that led to these approaches are detailed in the following subsections.

IV.3.1. Data-driven approach

Also called supply-driven, the data-driven approach of DW design consists of only analysing the organizational data which is completely different from the classical systems' development that have a requirements-oriented life cycle.

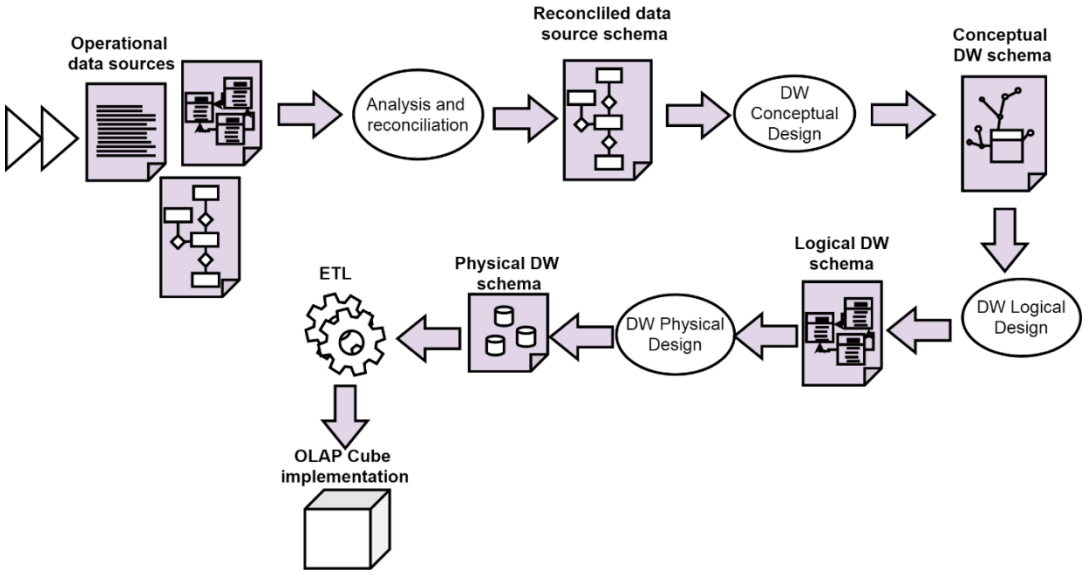


Figure IV-3 Data-Driven DW design approach

In Figure IV-3, an illustration of this approach's steps is depicted. It starts with analysing and reconciling data from operational data sources, then realizing the three phases of design i.e. conceptual, logical and physical, and finally executing the ETL and implementing the DW. According to (Inmon 1996), requirements are to be understood by users that analyse the querying results after the decisional system's population. Authors of (Golfarelli and Rizzi 2009) cited three success prerequisites for the data-driven approach:

- An available, or achievable in reasonable price or in short terms, in-depth-knowledge of the data sources.
- A good level of normalization in data source schemata.
- Data source schemata should not be too complex.

However, while this approach was, if applicable, recommended as the most suitable in terms of rapidity e.g. (Golfarelli and Rizzi 2009), it remains a high potential waste of resources by handling large unneeded data structures. Moreover, it's raising the system's complexity and ignoring the users' expectations of what the system should offer them, and therefore discourage their involvement in the project (Gardner 1998).

IV.3.2. Requirements-driven approach

Also called goal-driven or demand-driven, this approach is based exclusively on the users' requirements. It is following the same principle of software engineering of limiting the interest to what the final users require. However, the additional complexity in the case of data warehouse design comes from the necessity of dealing with the data availability in the implementation phase, and the paradox of its consideration to a better analytical needs' definition.

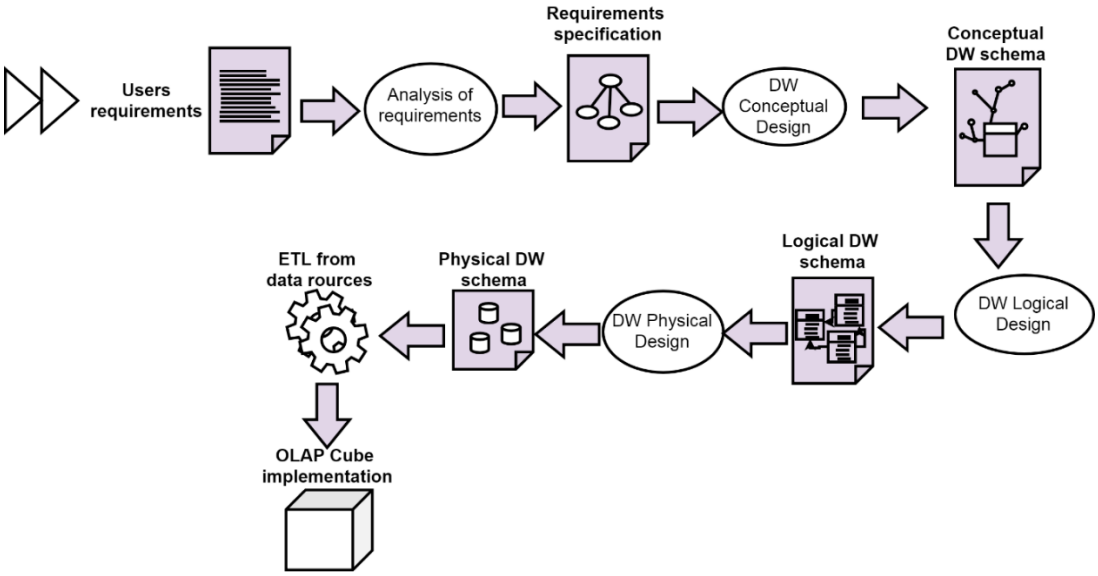


Figure IV-4 Requirements-Driven DW design approach

The process of this approach, as shown in Figure IV-4, starts with requirements elicitation, then specification, and after being turned onto physical schema, comes eventually their reconciliation with the data sources in the ETL step to load the cube accordingly. It's considered and recommended by many specialists as the most convenient approach e.g. (Salinesi and Gam 2006). In spite of the fact that it's time consuming, yet, it still delivers DWs that are more consistent in terms of users' expectations fulfilment and high-quality results. Contrary to this vision, (Golfarelli and Rizzi 2009) consider it as the most difficult to be performed because of the quick obsoleteness of its results since the high potentiality of having it reflecting personal viewpoints of participants

with different perspectives that might also be inadequate with the organization's culture or routines. However, this approach is still the unique choice when analysing the data sources is complex, for example, due to a legacy systems representation that makes its exploration and normalization an unrecommended task. For the sake of clarity, it's important to mention that there may be in some approaches such as (Guo et al. 2006), a differentiation between goal-driven, as based on the overall long-term goals of the organization, and requirements-driven, as based on the specific needs of the involved users.

IV.3.3. Mixed approach

Also called hybrid, the mixed approach takes advantage of both the easy conditions of the data-driven approach and the guarantees of the requirements-driven (Golfarelli and Rizzi 2009).

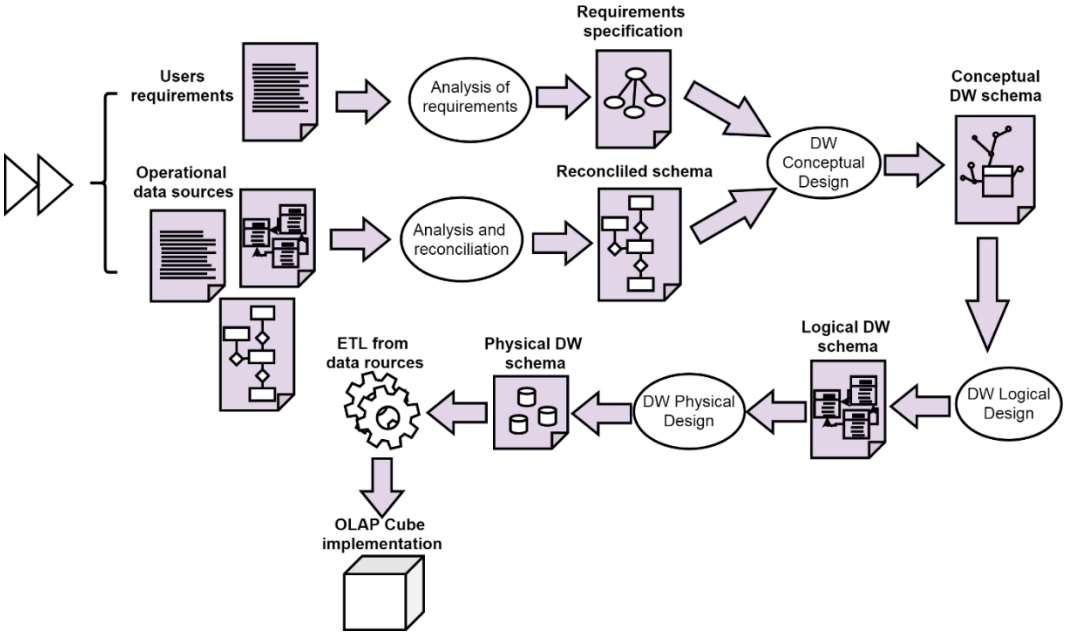


Figure IV-5 Hybrid DW design approach

It proposes a combination of the data-driven and requirements-driven paradigms in order to design the DW from data sources with an oriented structures' selection limited to user's requirements. (Romero and Abelló 2009; Tria et al. 2018) differentiate two types of mixed-approaches, the interleaved that performs both demand-driven and supply-driven stages in parallel using the feedbacks retrieved from each stage during all along the process to narrow down the ongoing reconciliation, whereas the sequential hybrid approaches perform independently the demand-driven and supply-driven stages to conciliate afterwards their outcomes. A general illustration of this approach is in Figure IV-5 that shows that the conceptual design is not done unless a reconciliation between requirements' specifications and the schema conceived from data sources is performed.

IV.4. Related work

After the general idea that we could get in the previous sections of this chapter, we review in this section some DW requirements engineering and design methodologies that we consider relevant to our research. We don't suggest by any means that this is an exhaustive literature panorama about both phases, however it is our hope that it would cover the most important features that we define in detail in section IV.4.1.

IV.4.1. DW design features for literature review

In order to better define a set of features for a DW design methodologies' review that goes in line with our specific context of the collaborative DW design methodology that we detail in the following chapter, we consider here the differences between crowdsourcing volunteers and BI usual decision-makers, that we presented in the previous chapter (Table III-1). Thereby, we defined the following features that we use to fill the Table IV-1 that summarizes the design approaches accordingly:

1) ***Handling divergent requirements*** i.e. considering and reformatting inconsistencies and contradictions in definitions because of volunteer users' medium proficiency in the subject matter, low, or at most, medium level of understanding vis-à-vis the global project's objectives, and an unlikeliness to reach a unified vision.

2) ***Handling requirements rejection*** i.e. in the case of unawareness or misconceptions in requirements, the rejection of some or a whole of a manifested requirement is considered because of the users' none or very low knowledge of DW fundamentals, medium proficiency in the subject matter, and low or medium understanding of the global project's objectives. For example, a level introduced by one volunteer is not considered as useful for the analysis of naturalist data.

Moreover, based on Table III-1 from the previous chapter, it is also difficult for volunteers to validate the quality of the provided DW schemata (completeness, level of details' abstraction, relevance to analysis, etc.). For example, in regard to questions such as, is the schema complete? Are there any missing dimensions? etc. which implies that the elicitation methodology must:

3) Be accompanied by some semi-structured interviews that allow guiding volunteers to identify the multidimensional schema problems in terms of semantic quality (i.e. ***Quality***

based elicitation). The issues associated with data distribution and normalization are left to the DW experts since they depend on the DW design structures.

4) The elicitation methodology must also be supported by some tools that allow to correct (modify, delete, add) the DW schema elements that might turn out erroneously defined (i.e. ***Correction actions***). For example, the temporal level 'week' could be removed since it is not explicative of any natural phenomena.

Other important features that we underline for the incrementality of multiple models' revision and risk-based iteration of collaborative design, are related to the availability in terms of employment in the project and ability to participate in all the steps of the DW's development cycle. For example, in our case study some volunteers cannot dedicate more than two-three meetings of two-three hours to exchange with DW experts in order to define their schemata, and sometimes even if their elicitation step is not finished, they do prefer to end these meetings (or they are obliged to) due to professional constraints. Moreover, the DW schemata elements defined by volunteers can be numerous and different as we previously detailed. Also, the volunteers cannot all be totally trustworthy which makes it important to prioritize their definitions in the DW requirements elicitation, in order to proceed decreasingly in terms of relevance, to the DW schemata elements' validation. Thus, the "Incrementality and risk-based iteration" principle must:

5) Prioritize DW schema elements (i.e. ***Prioritization***) based on the profile of the volunteer. For example, the volunteer seems novice to the agrobiodiversity domain application, and he/she is the only one defining the "users" dimension. Therefore, this dimension must not be considered as a priority in the DW requirements elicitation and design steps.

6) Provide an additional elicitation step with more engaged volunteers (i.e. ***Committers involvement***) to finalize missing elements and solve requirements conflicts. For example, a group of skilled volunteers can be involved in an additional and final DW design refinement step.

Furthermore, due to the huge number of volunteers that might have many different needs when designing DW systems for citizen science communities, providing an implementation for each proposed model is unrealistic because of its high human, temporal and financial costs. This leads to the "Prototypes and automated transformation" principle that must:

7) Reduce as much as possible the process's time from elicitation to prototypes' evaluation (i.e. ***Early rapid prototyping***). For example, in our case study we have 15 volunteers that each of them has defined three to five DW schemata, which is very time-consuming if we do so following any of the existing design methodologies.

The "formal and light documentation" principle that must:

8) Be based on a ***Simple elicitation formalism*** that corresponds to the low expertise level of most participants. For example, in our case study, some OAB volunteers are even not comfortable with uploading their data using a simple web page.

DW user-centric design methodologies, as described previously in this chapter, are Goal driven and Hybrid, which makes these two design paradigms the only adaptable ones to the "collaborative DW" design that we propose. These methodologies use as input a formal representation of decision-makers requirements, though, in our context an automatic translation of elicited requirements into a multidimensional schema is necessary to prove the feasibility of the DW implementation. In other terms:

9) *Must rely on a* ***Goal driven or hybrid methodologies*** that use a mechanism allowing the elicited requirements to be automatically translated into feasible DW schemata.

Finally, the 'distributed time/space' character of volunteers' participation means the consideration of the difficulty that some users have to participate in elicitation sessions due to a temporal or a geographic reachability limitation.

10) *Must, therefore, adopt* ***Web and asynchronous tools***. For example, farmers and ecology researchers don't have the same availability during the day.

To limit the size of the table for readability, we abbreviated the above detailed features as follows:

— DW design principle: *User involvement (**UI**)*:
  - *Handling Divergent Requirements (**HDR**)*
  - *Handling Requirements' Rejection (**HRR**)*
  - *Quality Based Elicitation (**QBE**)*
  - *Correction Actions (**CA**)*
— DW design principle: *Incrementality & Risk-Based Iteration (I&**RBI**):*
  - *Prioritization (**P**)*
  - *Committers Involvement (**CI**)*
— DW design principle: *Prototypes & Automated Transformation (**P&AT**):*
  - *Early Rapid Prototyping (**ERP**)*
— DW design principle: *Formal & Light Documentation (**F&LD**):*
  - *Simple Elicitation Formalism (**SEF**)*
— *Distributed Time/Space (**DTS**)*
— *Design Approach (**DA**)*

Table IV-1 Reviewed papers from literature

Related work	UI				I&RBI		P&AT	F&LD	DTS	DA
	HDR	HRR	QBE	CA	P	CI	P&AT	SEF		
(Bonifati et al. 2001)	Manually	-	-	-	-	-	-	Interviews	-	Hybrid
(Winter and Strauch 2003)	-	-	-	-	Yes	-	-	-	-	Req-driven
(Paim and de Castro 2003)	Review sessions, Prototyping	Review sessions, Prototyping	-	-	-	-	-	Interviews, Workshops, Scenarios	-	-
(Nabli et al. 2005)	-	-	-	-	-	-	-	2D sheets	-	Req-driven
(Guo et al. 2006)	-	-	-	-	-	-	-	Interviews	-	Hybrid
(Salinesi and Gam 2006)	Map formalism	-	-	Yes	-	-	-	Map Formalism	-	Req-driven
(Giorgini et al. 2008; Giorgini et al. 2005)	-	Basic operation DW-tool	-	-	-	-	-	Interviews, TROPOS	-	Req-driven/ Hybrid
(Prakash and Gosain 2008)	-	-	-	-	-	-	-	GDI model	-	-
(Jukic and Nicholas 2011)	-	-	-	-	-	-	-	Interviews, Questionnaires, Feedbacks	Yes	-
(Romero and Abelló 2010)	-	-	-	-	-	-	-	Filtering functions	-	Req-driven
(Cravero Leal et al. 2013)	-	-	-	-	-	-	-	-	-	Req-driven
(Khouri et al. 2014a)	Semantic ontologies, Pivot model	-	-	-	-	-	-	-	-	Req-driven
(Kumar and Thareja 2014)	Review sessions	-	-	-	-	-	-	Interviews, Workshops, Prototyping, Use cases, GDI, DWARF …	-	-
(Tria et al. 2015)	-	-	-	-	-	-	-	-	-	Hybrid
(Elamin et al. 2017)	-	-	-	Yes	-	-	-	NL	-	Req-driven
(Nasiri et al. 2017)	-	-	-	-	-	-	-	Guidelines	-	-
(Ren et al. 2018a)	Semantic ontologies	-	-	-	-	-	-	-	-	Req-driven
(Sakka et al. 2018)	Automatic	GDSS	Yes	Yes	Yes	Yes	Yes	Interviews + 2D sheets	Yes	Req-driven

We would like to mention that in the following two sections i.e. IV.4.2 and IV.4.3, some of the selected works are mentioned more than once but described differently since we put the focus on a different view at each time.

IV.4.2. Requirements Engineering in DW literature

Earlier in this chapter, we overviewed the elicitation of requirements from a general perspective in order to better orient our literature review in this section. Generally, during the elicitation, the future processes that the system needs to perform in order to reach the main objectives of the business must be determined (Zowghi and Coulin 2005). Authors of (Holten 2002; Prakash and Gosain 2008; Stroh et al. 2011; Tria et al. 2018; Vassiliadis 2000) consider, among others, RE in the context of data warehouse systems as a hard and critical step to perform. Thus, many of these systems have

failed when they did not give the RE phase its real importance by proceeding to design and ETL steps with incomplete or inconsistent elicited requirements. Hence, in DW projects, many works have emphasized their DW design methodologies on the RE step in both requirements-driven and hybrid approaches.

In the literature, many works have centred their focus on the RE phase of DW systems design and proposed different frameworks and techniques in order to accomplish this crucial task that the success of the system itself afterwards relies on which. For example, authors of (Paim and de Castro 2003) have limited their interest only to RE step. They adapted in this work a traditional requirements' engineering approach to define requirements and manage the data warehouse. Most importantly, in the third step of their methodology, called DWARF, review sessions and prototyping are used as techniques to validate the specifications, which may remain containing some pitfalls i.e. requirements overlapping and similarities that must be re-specified to fit with the requirements. Similarly**,** (Prakash and Gosain 2008) focused on the RE part of the DW design cycle and relied on the broad organizational goals to do so. For that, they used Request For Information-response (RFI-response) as informational scenarios technique to guide decision-makers through the elicitation process that outputs a Goal-Decision-Information (GDI) model schema. In (Jukic and Nicholas 2011) authors have addressed, with a framework of requirements' definition and collection, the problem of DWs failure caused by inconsistencies and inadequateness in requirements. However, what is new in this approach is that it covers the issue of involved users' limited availability during the requirements elicitation. Such availability limitations can be due to problems in the company, political issues, low-level business-end sponsorship for the project, geographically dispersed stakeholders etc. To do so, they divide the requirements definition and collection team into two groups, one has a DW analytical requirements role and the other a DW details role. The team charged with DW analytical requirements have, additionally to interviews, the responsibility to rely on other means to deduce requirements such as reviewing available records and past interviews, questionnaires addressed to all or some users, users' feedback on existing DW examples etc.

With the same objective of handling RE phase, the work of (Kumar and Thareja 2014) illustrates a requirements engineering framework that supports the implementation of DW systems incrementally and iteratively, offering definite and verifiable defined requirements. This framework allows for requirements management that improves the consideration of users' perceptions and their harmony. The proposed activities are similar to those adopted by most DW requirements engineering methods. However, additional efforts have been focused on eliciting and managing requirements, the two steps that are often ignored by other works as authors claimed. In (Cravero

Leal et al. 2013)**,** authors consider that their paper is the first that investigates requirements that use strategic business activities with a known technique of business analysis, while other researches were based on objectives' model. It is based on a set of guidelines that allow designers to stick with the global business strategy. Using this approach, designers must become more assured that early requirements are those that are needed by the business and lead them towards well-designed schemata that realize the users' strategic needs.

With a similar awareness of the multiple abstraction levels that might be considered, (Salinesi and Gam 2006) illustrated their requirement-driven approach called CADWA that starts by decision-makers' requirements elicitation. In order to solve the multiplicity of different users' interpretations of the same requirement, they organize them into four levels of abstraction i.e. Organization business plan (BP), Decision-maker macro BP, Decision-maker micro BP and Action plan. They also classify users by activity and represent them by a user that assures achieving the BP and verifying consistency with other groups' macro BPs. In (Winter and Strauch 2003), authors review existing data warehouse approaches and based on their findings of a four-year project with large service sector companies. They propose in general outlines a methodology that covers the entire process of identifying requirements for DW users, matching them with information supply, assessing and homogenizing them, prioritizing those of them that are unsatisfied and specifying formally the results at the end. Although, the design methodology is still based only on reviewed literature and some components have been applied in actual DW projects.

Moreover, other works have used more formal techniques to deal with semantic and syntactic issues in RE. For instance (Khouri et al. 2014a), propose a demand-driven approach that covers all the DW design phases. Focusing on the problem of data heterogeneity, they illustrate an extension of ontologies' use to solve syntactical and semantical requirements conflicts that emerge by integrating data elicited using various formalisms.

Another ontology-based approach is the works of (Ren et al. 2018a)**.** It proposes a dimensional modelling method of ontology based medical DW considering the characteristics of medical data sources and business requirements. It covers the optimization of requirements analysis process and effective elimination of semantic heterogeneity in both business requirements and data sources. The proposed framework is composed of four steps: First, building medical ontology from heterogeneous data sources. Second, transforming ontology into potential facts, dimensions and measures. Third, use objective oriented DW requirements analysis to obtain medical requirements effectively. Fourth, comparing the requirements model's concept with the multidimensional concept and generating the final model based on specific rules.

Another important aspect that we focus on its relevance to our context is automation. In (Tria et al. 2015)**,** in order to build Business Intelligence systems for academic organizations, a design process is described based on a mixed-driven methodology. The proposed methodology is largely automatic and relies on an ontology-based approach to integrate different data sources. The empirical application of the DW design process is dedicated to analysing the main factors by which, importance and quality level of Universities such as research and didactics quality, might be affected.

Also, (Nasiri et al. 2017) propose a Goal oriented RE framework for DW systems called RE4DW that is composed of two modelling components. One is the context modelling that identifies the organizational data required to design a DW supporting Key Performance Indicators 'KPI' for monitoring purposes, using i* framework. The second is data modelling that uses a MD model to assure the appropriate data structuring. What is important to retain in accordance with our premise, is that this framework assists its use by an iterative guidance.

KPIs have also been used by (Guo et al. 2006)**.**They introduced a data modelling methodology of DW integrating goal-driven, data-driven and user-driven approaches. The first is the Goal-driven step, which produces a set of subjects and KPIs of main business fields. Next, the Data-driven step generates a subject oriented enterprise data schema. Then the third, which consists of a User-driven step, that represents as measures and dimensions all the analytical requirements. Finally comes the combination of the triple-driven outcomes. The methodology is supposed to improve completeness, structuration and superposition of DW's data models.

Among the employed techniques, some are more dedicated to inexperienced users than the others. Such as in (Nabli et al. 2005)**,** authors adopted the use of pivot tables for the requirements definition in their automatic schemes generation approach that uses later on a set of algebraic operators to fuse them into data marts. Another hybrid approach that covers the elicitation phase by applying the Tropos methodology (Bresciani et al. 2004) is in the works of (Giorgini et al. 2008; Giorgini et al. 2005)**.** They cited only in their 2005's version the idea of refinement; a final step to rearrange the fact schemas in order to fit better the users' needs. However, in the 2008's version, a java swing based CASE tool "DW-Tool" was introduced and its inclusion of the basic refinement operations was mentioned. (Elamin et al. 2017) standardized the formulation of users' goals using natural language queries and then a matrix filled with which, to avoid redundancies in elicited requirements. In each step, they relied on heuristics-based algorithms to stick with the three key properties that they have defined i.e. completeness, normalization and correctness, for requirements normalization and schemes' generation steps.

With a more generic vision, (Bonifati et al. 2001) detailed a semi-automatic methodology that covers all the design steps. The first steps were a top-down stage that makes an elicitation and

consolidation of user requirements. However, they do not detail the requirements elicitation layer of the methodology. (Romero and Abelló 2010) also say that the elicitation step is out of the scope of their work. Despite this, they focus their validation on a SQL queries representation of requirements defined by skilled employees from relational data sources. It is done using an algorithm that checks a set of semantic constraints, that by satisfying them, the query is considered as meaningful.

To summarize, common interests, that practically all DW design methodologies covering RE stage share, are requirements consistency, unambiguity in semantic and users' perception terms and exhaustiveness in its coverage of users' needs. For that, as we have illustrated in this section, many aspects are possibly interesting from this perspective to improve the RE in DW systems whether they are leading to a better understanding of the subject matter, data sources identification, users' classification or choice of the most convenient techniques, methods and tools to perform the elicitation correspondingly. That being so, DW design methodologies consider the resulting definitions differently based on various design paradigms. In the next section, we highlight approaches conceived according to design paradigms with their illustrations and related literature.

IV.4.3. DW design approaches from literature

The works selected in Table IV-1 include requirement-driven and hybrid methodologies that fall in the scope of our interest since both of which cover the RE stage which we overviewed its different aspects in the previous section.(Nabli et al. 2005) proposed a requirement-driven approach and focused on the fusion of the generated schemata after the use of pivot tables i.e. two-dimensional tabular format for the elicitation stage, from which the dimensional model is retrieved and reconciled later on with the data sources. The same in (Cravero Leal et al. 2013)**,** where authors have based their DW design on a requirements-driven approach that consists of following strategic business highlights to keep an alignment between organizational goals and DW objectives. However, in this work, emphasis has been placed on the clarification phase of the importance it attaches to the failure of many systems due to the inability to achieve initial objectives.

Others have taken the demand-driven paradigm from different point of view in order to solve the common problems of requirements and data-source correspondence such as (Romero and Abelló 2010) that have introduced a requirements-based approach with automated processing and analysis of end-users needs with a focus on data of interest to them, unlike to data-driven approaches. Additionally, the special feature of this methodology over requirements-based methods is that it allows suggestions of multidimensional components similar to those that users initially queried.

This way, its validation step provides meaningful schemas representation of the system. Similarly, based on pure theoretical findings from literature research and experience in the field**,** (Winter and Strauch 2003)**,** outlined a Demand-driven approach that focuses on requirements definition step. Then, they follow it by a phase of data schema modelling using an appropriate data model, and eventually, a schema evaluation step that verifies end users' satisfaction with the resulting schemas that are usually developed by specialists**.**

For the same improvement aim, and as we have mentioned earlier that requirements can be taken from different levels of abstractions, works such as (Salinesi and Gam 2006) introduced a requirement-driven approach that takes the design vision from a goal-based angle. It focuses mainly on requirements' classifications to organize hierarchically the different levels of goals that motivate the DW implementation and so, yield better the consistency of the deliverable with its future users' expectations.

As automation is still a problem that faces all complicated design methodologies that have always differences in their processes regarding the application circumstances, many works have automated them like (Elamin et al. 2017). They introduce in this work a requirements-driven approach that generates multidimensional schemas from elicited requirements automatically following heuristic rules in order to guarantee the coherence of the extracted elements with the defined needs.

Another important issue that has been well studied in the literature is the semantic and syntactic integration of requirements. In their work**,** (Khouri et al. 2014b) have illustrated a demand-driven approach that, after eliciting requirements from different heterogeneous sources, makes conceptual, logical and physical designs to provide a DW schema that integrates these various requirements. Likewise, (Ren et al. 2018b), solve the requirements heterogeneity problem with a requirements-driven approach that builds its core on an ontology in the context of medical DWs conceptual modelling. Also, (Tria et al. 2015) use ontologies, as an integration of various data sources mechanisms, to handle the requirements definition step in their automatic and hybrid approach for academic organizations' DWs. Which makes as a hybrid methodology, as they argue, an additional conformity confirmation from data sources in parallel with the requirements formulation process.

Hybrid approaches are those that start with requirements definition, and in parallel, derive from data sources what reconciles the elicited needs with the available data. Hence, many hybrid approaches in the literature do not detail their RE step and focus on the other steps. Among others, (Bonifati et al. 2001) presented a semi-automatic method identifying and designing data marts. Three basic steps can be identified; first a top-down step that makes an elicitation and consolidation of user requirements, then a bottom-up extraction of candidate data marts from conceptual schemas of the information system and finally a reconciliation between ideal and candidate data marts.

In a hierarchical structuration of requirements in organizational contexts, (Guo et al. 2006) also proposed a hybrid methodology that aims to mixing user-driven, goal-driven and data-driven approaches in order to improve data completeness, users' satisfaction about the resulting system, structuration, and layered data modelling in terms of design. The same for (Giorgini et al. 2008; Giorgini et al. 2005) that presented a hybrid methodology of design that, however, can be also used in only demand-driven design approaches. After using the Tropos methodology (Bresciani et al. 2004) for requirements elicitation as we mentioned in the previous section, they propose a mixed design framework that consists of three steps, first mapping the requirements, then constructing the hierarchies and finally refining the resulting conceptual schema.
In the next section, we highlight some important findings that, in contrast with the features of Table IV-1, make a theoretical background to our work.

IV.4.4. Highlights on the selected works

In Table IV-1, we could identify some very important aspects that only a few works have mentioned due to the vision and the context of each. In addition, as we have detailed the set of features that we defined and that have led us to be highlighting some interesting proposals and practices that support our perspective of this chapter, which is assessing the existing literature that handles the specificities arising with crowdsourcing users' engagement in designing DWs.

For example, what is important in (Bonifati et al. 2001) for our interest is that they resolve conflicts, which is a potential issue in our context, by eliminating some goals after being formulated, during an interaction with the involved people. However, there were no dedicated techniques nor tools introduced for that matter and it's only applied on high-level goals and not on multidimensional concepts to fit with the application domain.
Also, interestingly, the work of (Paim and de Castro 2003) was to our knowledge, the only design methodology that rely in its elicitation phase on the intervention of external reviewers to support defects detection. This idea of external reviewers is close to the role of data steward (Kimball 2013), which is different than our concept of committers' resolution of conflicts based on their subject matter's mastery and the definitions' consistency from an analytical relevance's point of view, not a data-sources' coherence one (detailed in next chapter). Nonetheless, while data stewards are traditionally data governors that assure data quality i.e. definitions and conformity between cross business units' data marts over data sources (Rifaie et al. 2008; Sammon and Finnegan 2000), the only cited criterion in this work about external reviewers is their unbiased view and no tools or techniques were mentioned. Review sessions was as well briefly mentioned as the used technique

by (Kumar and Thareja 2014) in order to accomplish the requirements validation step that involves all the participant parties.

It is also important to mention that (Giorgini et al. 2008; Giorgini et al. 2005) did not introduce the refinement step to handle consistent requirements' rejection in case of non-relevance to the analytical needs, but it is only reshaping the users' specifications since they initially focused only on early requirements i.e. high-level objectives.

In (Khouri et al. 2014b), the semantic and syntactic conflicts, that they use ontologies to solve, are out of our interest's scope since we have a unique protocol-based data source and multiple users with diverse objectives and not as in their case, many heterogeneous data sources and a common strategic objective of the system.

For its simplicity as an elicitation formalism that fits well with the unfamiliarity that have crowdsourcing users with DW concepts, pivot tables used by (Nabli et al. 2005) is a formalism that we adopted by our use of ProtOLAP that (Bimonte et al. 2013b) have defined as "a tool-assisted fast prototyping methodology that enables quick and reliable test and validation of data warehouse schemata in situations where data supply is collected on users' demand and users' ICT skills are minimal".

In (Salinesi and Gam 2006), after classifying users by activities, they represent them by one user for each group, who assures achieving the business plan and verifying consistency with other groups' macro business plans as well as guaranteeing the coherence with the higher-level strategic goal. That concept of representative users is interesting in the way it simplifies incoherencies management and requirements validation. Notwithstanding, this approach remains classifying requirements by goals' hierarchies in an assumption that a unified vision of a unique organization exists, which is already problematic in the citizen science community's participation.

In a more formalized way, (Elamin et al. 2017) eliminated redundancies in requirements after defining them in natural language by the means of a matrix of requirements. However, it remains a very time-consuming formalism especially with availability and vision's unity limitations.

Moreover, among the covered works, only (Jukic and Nicholas 2011) addressed the issue of involved users' limited availability during the requirements definition step. Such availability limitations can be due to problems in the company, political issues, low-level business-end sponsorship for the project, geographically dispersed stakeholders etc. To solve this problem, they charge a team with DW analytical requirements that have the responsibility of relying on other means to deduce requirements such as reviewing available records and past interviews, questionnaires addressed to all or some users, users' feedback on existing DW examples etc. This

approach is also only interesting when other normalized and formalized documentations and protocols exist already with a clear strategic orientation within an organization and not with crowdsourced data that have diverse interpreters and associations.

IV.5. Group perspective for DW RE

Group Support Systems (GSS) showed a promising potential to be used for the requirements' elicitation since the early 90s (Carmel et al. 1993; McGoff et al. 1990). Authors of (Evans et al. 1997) propose the replacement of the term "requirements" and its associated processes by "decisions" with a decision process to change the nature of the engineering task of computer-based systems from unilateral to mutual.

(Ruhe 2002) defined a methodology called Software Engineering Decision Support (SEDS) that gives a set of guidelines to cover the complete software engineering lifecycle, including handling the various stakeholders' perspectives and expectations that might complicate the definition of right and complete requirements. Also, (Aurum and Wohlin 2003) proposed an approach of illustration of a decision-making model in RE process models in a way that it helps to commonly formulate vocabulary and to improve the manageability of the RE process in order to enhance the organizational learning process. The same in the work of (Konaté et al. 2014), where the emphasis was on the collaborative aspect of RE. Their approach is mainly focused on the collaborative requirements elicitation using Thinklets (Briggs et al. 2001) and a web based GSS. (Regnell et al. 2001) mentioned that Requirements can be differently defined as the results of stakeholders' decisions of what software's functionalities and quality are to be constructed.

In (Tuunanen 2003), a review of the existing literature that formulates and covers the gap of the wide-audience end-users involvement in RE was made. Also, an example of GSS based requirements elicitation and negotiation methodology is the work of (Briggs and Grünbacher 2002). Named EasyWinWin, their GSS-supported methodology is meant to reduce the complexity of requirements' establishment by taking advantage of the cognitive load's reduction that the GSS offers. Consequently, GSS proved an acknowledged improvement when they were deployed in RE processes in different areas.

Exploiting data for analytical purposes cannot be effectively done unless it's cautiously considering stakeholders' requirements. Stakeholders in traditional DW systems are business owners, organization's managers or company's employees who are the potential users of the system i.e. decision makers (Vaisman and Zimányi 2014). In consequence, the high potentiality of having conflictual needs when involving citizen science users in RE for DW, makes it mandatory to

manage and negotiate those conflicts in order to reach a consensual set of coherent and consistent requirements that a successful DW can then be designed accordingly.

Thus, and as it was effectively a key asset of RE in other areas, our new approach of DW requirements elicitation is based on the use of GDSS to handle the divergent requirements as shown in Figure IV-6.

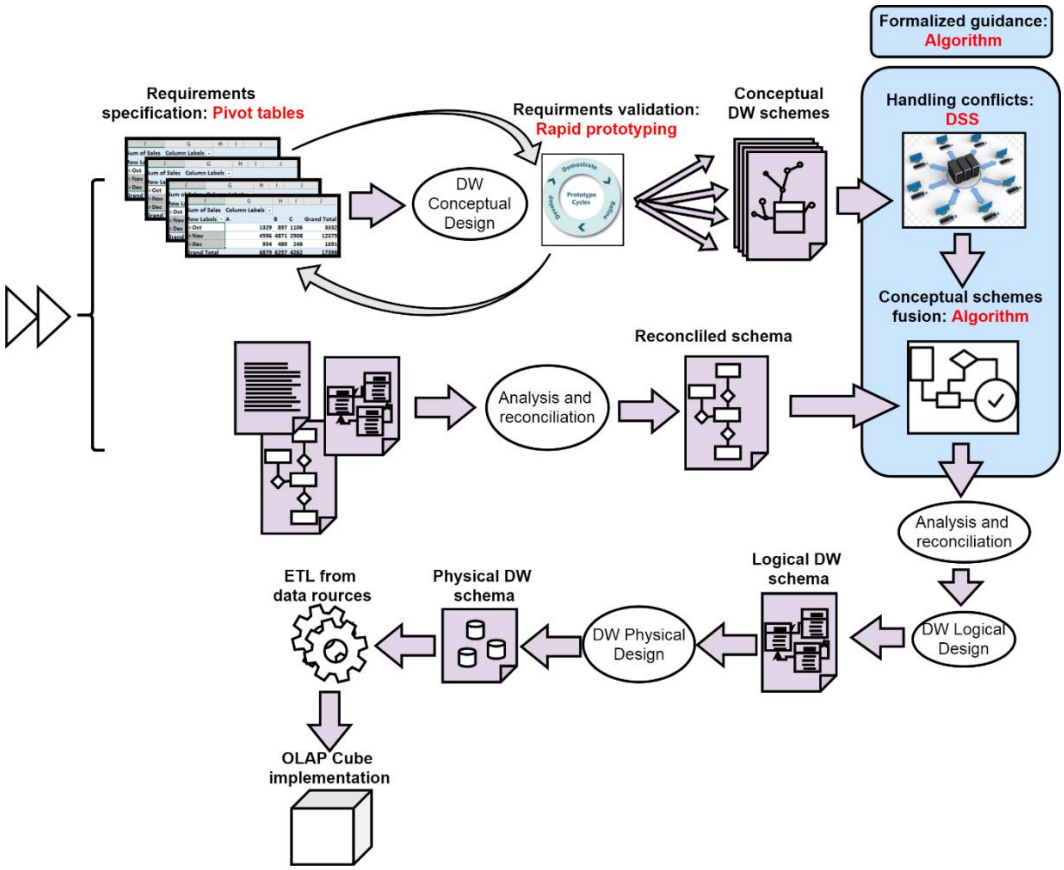


Figure IV-6 Hybrid approach for Collaborative DW design

In reflection with the features of Table IV-1, our use of GDSS will be in the step of handling requirements expressed by several users as well as handling the rejection and personalization of multidimensional conceptual components. In addition, GDSS as an asynchronous web-based tool allows overcoming the users' geo-temporal distribution constraint as well as the involvement limitation issue since it supports different types of flexible and structured meeting and group activities' techniques. Furthermore, to reduce the requirements elicitation time and to formally guide the unskilled participants, we propose in our methodology (detailed in next chapter) a semi-automatic assisting step that formalizes the elicitation i.e. formalized guidance and fuses the individually defined requirements i.e. conceptual schemes fusion. Hence, for the "simple elicitation

formalism", we only adopt an existing solution proposed by (Bimonte et al. 2013b; Nabli et al. 2005) i.e. pivot tables formalism. To the best of our knowledge, our methodology is the first that uses GDSS in RE for DW design.

IV.6. Conclusion

In this chapter, we have introduced, from the RE in DW design perspective, an overview on the existing approaches in the literature. The selection of the studied works was done with a set of features that go in line with our proposal parameters in order to build an empirical background of proceeding. We first assessed the existing works from RE and general DW design angles. Then we detailed our works' selection features and we discussed the interesting findings that we proposed eventually, based on which, our solution of adopting GDSS for the RE stage in DW systems that was also used for RE in other systems. This literature review makes, as well, a theoretical support for the collaborative DW design approach that we detail in the next chapter.

Chapter V

COLLABORATIVE DATA WAREHOUSE DESIGN

Summary

In this chapter we define our collaborative DW design methodology which is composed of two main steps: (i) 'Requirements elicitation and modelling' that aims at collecting the requirements of each volunteer, translating them into models and validating them with rapid prototypes. And (ii) 'Solving subject matter issues of requirements' in which we provide a models' fusion algorithm and two approaches of collaborative resolution of requirements conflicts. We do so in order to reduce the potentially unrealistic number of predefined models that would most probably contain redundancies and application domain related pitfalls which therefore require a further merging and refinements. Afterwards, we suggest an implementation environment for our methodology. This is the same that we further use with a real case study to execute all its steps and validate with which one of the two proposed collaborative methods while the second requires additional group techniques that we provide and test later in Chapter VII.

V.1. Introduction

As we have substantiated in the previous chapter, using GDSS when a crucial collaborative task is tackled by a group of stakeholders is a safer solution that applies, as in other collaborative application areas, to requirements engineering. However, in our case of designing data warehouse systems with the involvement of volunteers, this cannot be guaranteed to be always applicable since the participation of users has no clearly identifiable unified objectives. For that reason, we consider in our methodology both cases of having collective and individual participation to the requirements elicitation phase. After that is done, the collaborative aspect of the second group engaging step that we suggest i.e. collaborative resolution of requirements conflicts, is much more straightforward and clearly executable with GDSS group processes. With that in mind and in addition to the phase of

fusing the elicited models, we also focus in our approach on the evaluation criteria and the steps of the collaborative processes that we define for our methodology.

V.2. Collaborative data warehouse design methodology

The methodology that we propose considers the specificities of the volunteer users that we previously discussed in chapters III and IV. With this type of users' participation, many limitations that imply their involvement in DW requirements engineering and design phases can be summarized as follows:

- Too many potentially interested users: the implementation of a DW for every participant wouldn't be possible in terms of time and financial costs.
- Very limited knowledge of DW fundamentals: They are amateurs, biologists, nature scientists, etc. that don't know the architecture of an OLAP system, the terminology used by experts, the database basics etc.
- No involvement in the overall project: in terms of availability and commitment that is necessary in data warehouse usual design contexts and that is not what volunteers can guarantee, especially when it comes to collaborative participation in many different sessions.
- Geographically distributed: that adds another limitation to the important aspect of communication for a collaborative design to be successfully accomplished.
- Limited vision of the project objectives: Even though the project might be flexible on its main goals in regard with what the citizen science community would define iteratively, there is no obligation nor motive for any individual to commit to what does not personally interest him of other's requirements.
- Difficulty of reaching unified vision: Conflicts and disagreements are harder to manage with volunteers that have diverse profiles, experiences, interests, scientific backgrounds etc.
- Limited proficiency of the subject matter: Even in the subject matter that makes actually an interesting field to the volunteer, there is no guarantee of his/her qualifications which mean by consequence, his/her requirements' pertinence is not always reliable.
- Difficulty of participation to elicitation sessions: Since we perform a collaborative elicitation to solve some of the previous issues and to increase the volunteers' motivation to use the outcomes of the project, a number of elicitation sessions must be

collectively attended, which is also problematic because of the availability constraints that volunteers have.

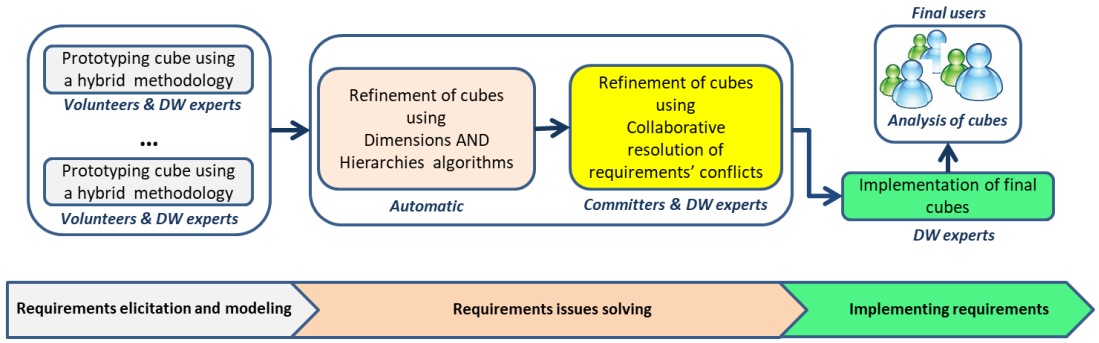


Figure V-1 Collaborative DW design methodology

A general illustration of our methodology is shown in Figure V-1. It is composed of the following three steps:

1) Volunteers express their requirements in a simple elicitation formalism e.g. natural language, pivot tables, GDSS brainstorming Thinklets etc. This is done mainly in terms of the measures and dimensions of analysis that they estimate important to their analysis. For that, we adopt an iterative and rapid prototyping process that allows DW experts, who are usually not qualified in the specific field of the application subject, to understand the requirements during the elicitation sessions. The main idea is to allow the volunteers to express separately their requirements that will be then translated into multidimensional conceptual models.

2) These DW models are fused using an algorithm that we detail next in this chapter. Then they are submitted to a group of particular volunteers that we refer to as **committers**, who are fully involved in the project and highly experienced with the crowdsourced data. In (Kimball 2013), authors emphasized on the necessity of data stewardship to solve issues related to the lack of users' experience in queries specification and data ownership or sensitivity problems encountered by organizations during the DW implementation process. This role is to be distinguished from what does our step of committers' resolution of conflicts that we count on their subject matter mastery and their definitions consistency from an analytical relevance point of view, while data stewards focus only on the data-sources coherence aspects. Thus, committers decide whether to implement crowdsourced requirements (i.e. multidimensional prototyped models) of volunteers or not, according to their expertise to judge the relevance of the requirements. Also, in case of imprecise, incomplete or ambiguously defined components, they can rectify, modify or complete

multidimensional elements and rename or change appellations to what they see more semantically appropriate.

3) After that, the DW expert designers implement the models agreed upon by the committers that will be eventually made available to all users i.e. the large public that the project leaders authorize sharing data with, where they can visualize, explore and analyse the data.

In the following subsections we describe the functioning of the methodology presented in Figure V-1 in detail. It is important to bear in mind though, that the methodology participates in the first two steps while the implementation step doesn't make any limitation since it's left to the DW experts to realize following the traditional DW techniques with no volunteers' implication.

V.2.1. Requirements elicitation and modelling

This step is composed of two phases: the first is the requirements elicitation i.e. elicitation sessions with volunteers, and the second is the translation of their requirements into valid multidimensional prototype models for the next step of refinement and validation. We detail in this section our proposal for these two phases as well as some technical solutions suitably.

V.2.1.1. Requirements elicitation

While doing this step with volunteers and in order to elicit requirements for many of them without losing a lot of time, especially with their voluntary availability limitations as previously substantiated, we use for that a simple elicitation formalism that, at the same time, offers an automatic generation of DW prototype schemata. As we mentioned in the previous chapter prior to the literature review, among the numerous available elicitation techniques, we have chosen to adopt "pivot tables" formalism followed by a "rapid prototyping" step which is the proposition of (Bimonte et al. 2013b; Nabli et al. 2005). We precisely propose two possible scenarios for that:

— As shown in Figure V-2, in the case where it is not possible to create volunteer groups, using the ProtOLAP methodology and tool (Bimonte et al. 2013b) that allows conducting interviews and workshops where, during the meetings with DW designers, the volunteers define their analysis requirements in natural language and with word or excel documents. After that, using the protOLAP tool to generate automatically the DW schemes' SQL codes for the rapid prototyping phase.

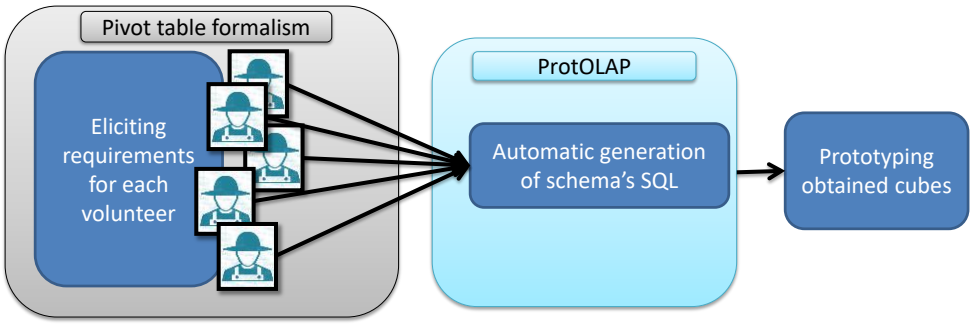


Figure V-2 First elicitation scenario: with only individual participation

— As shown in Figure V-3, where volunteers are working in groups, using GDSS brainstorming techniques for a collaborative definition of requirements. Then the DW designers will prepare the DW schemes' SQL codes for the rapid prototyping phase either manually, if the groups have converged to define only few models or using automatic tools such as ProtOLAP otherwise.

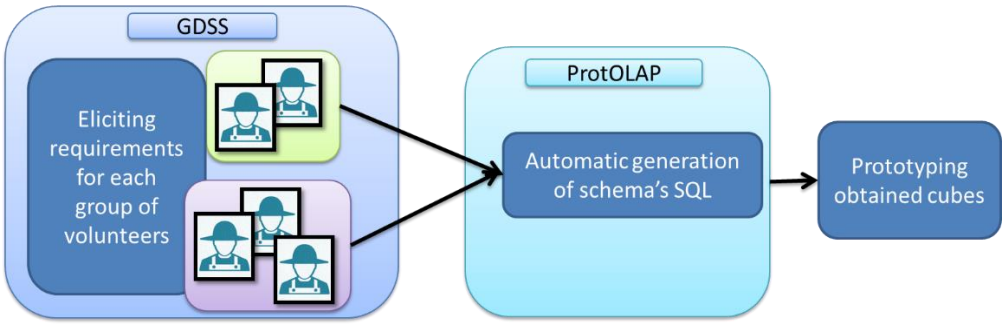


Figure V-3 Second elicitation scenario: with groups' participation

The creation of volunteer groups depends on their initial motives of participating in the project. It might be in some cases as well proposed by the designer that, by questioning them about their analysis interests, would be able to recommend a collaborative elicitation session to those that show openness to do so. Thus, the DW designers associate to each model a goal specification given by its definer which will be as well used later at the Collaborative resolution of requirements conflicts step. This first phase will in both scenarios output a SQL code of DW schemes, corresponding to a set of conceptual models that the designers must use for prototyping.

V.2.1.2. modelling and validation

This step takes as input the requirements of each volunteer or group of volunteers, and outputs a set of multidimensional models that are validated on data sources.

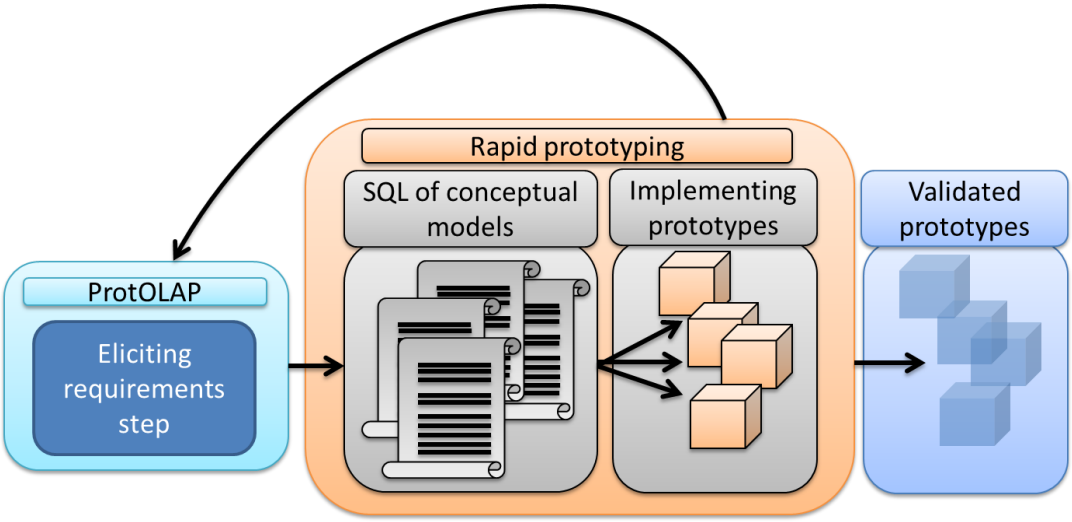


Figure V-4 Iterative rapid prototyping of elicited cubes

To do this, as shown in Figure V-4, the designers will use the conceptual models that have been defined in the form of DW schemes' SQL during the elicitation phase, validate them with the data sources, prepare a prototype for each of them and eventually confirm their validity with volunteers. For that, the designers must do a reconciliation step following a data-driven or a hybrid methodology as we detailed in the previous chapter.

In order to rapidly prepare the prototypes, having at hand the requirements that were transformed into SQL schemes, the designers must execute the SQL in a DBMS and fill the resulting cubes with some fictive information i.e. facts and measures. In this stage it is senseless to start ETL loads since it is time consuming, which we highly avoid because we are still in the prototyping stage. Also, even if the prototyped model gets validated by its definer(s) since the first attempt, it will be anyway modified later in the following step of the methodology where pertinence and correctness issues are implying further refinements. Moreover, the volunteers know very well the datasets since they have already used or alimented it in the data collection process. This means that they can define some measures that are likely retrievable from the data sources, which eases to the DW designers the reconciliation of the defined requirements on the available datasets. Finally, to reduce as much as possible the semantic issues during the elicitation phase, the DW experts might recommend the

reuse of the same technical appellations that have been already defined by other volunteers in their validated models. This can be done by keeping a track of all the defined multidimensional elements in a global repository that shares all the used technical terms for every specified requirement.

After doing this task of rapid prototyping iteratively until reaching the satisfactory definitions of users' requirements by allowing them during each validation meeting to use and explore the resulting cubes, they actually get more and more familiar with the OLAP cubes exploration. This will also help them to better conceive, in the next step of refinements, the potential impacts of any modifications that they might suggest on the resulting output.

V.2.2. Solving subject matter issues of requirements

The previous step gives a set of DW conceptual models that were validated by volunteers after the rapid prototyping phase. Although, the higher the number of participating users gets the more models this step will give, which, up to a certain extent, becomes unrealistic to have them all implemented at the end. In addition to that, the contents of the models might be problematic, even in the case where the DW designers have succeeded to regroup volunteers that initially expressed common analysis interests. More importantly, there is the necessity of validating and refining the relevance of the defined multidimensional cubes' elements from the subject matter view. We do this because of what we previously described of the residing possibility that volunteers define wrongfully understood, erroneously described or misleading representations of the analysis subjects. Therefore, at this level both volunteers might be satisfied by the resulting prototypes and the DW designers by the DW models that are loadable from the data source, while indeed, there is subject matter inconsistencies and redundancies that the resulting models are likely to contain. Hence, in this section we define an algorithm that merges the models that we consider fusible where they incorporate common measures i.e. regrouping of a set of aggregators, for example 'Birds_abundance' as measure for which we have 'min', 'max', and 'sum' as aggregators, which means that common analysis goals are separately expressed. Then we follow that step by a further refinement where intervene the committers to validate and readapt the resulting fused models.

V.2.2.1. Fusion of prototyped models

The fusion of models in this phase is different from what data marts integration works are tackling in the literature for two main reasons. First, the centralization of the data in almost a unique source i.e. observatory crowdsourcing database. Our models are relying upon such undistributed data

sources in our case of working with citizen science data for example or in similar contexts, whether the data collection is done opportunistically or following a standardized protocol. This guarantees that the requirements defined in all the different elicited models will remain retrievable from data sources after the fusion. And at the same time, it assures that, when merging level attributes of a non-conformed common dimension, the resulting dimension's construction is coherent from a measures' calculation point of view, unlike the main motivations of works such as (Cabibbo and Torlone 2005; Kwakye 2011) that work with heterogeneous data marts. For the pertinence of the generated cubes' elements from a pure analysis of the subject matter view, we perform a final refinement of the resulting models in the next step. The second reason is the fact that in our case we are still in the design process. Which means that we still have a certain amount of flexibility in relation with the ETL phase that we are, however, preceding with a session of the resulting cubes' evaluation and revision that will confirm the consistency and relevance of the definitions.

The different multidimensional elements, that a DW model is composed of, are illustrated in Figure V-5. A cube is the global element that contains all the other components. It is to be thought of as the DW schema that is composed of at least one or many dimensions i.e. axes of analysis and measures i.e. the properties on which calculations such as sum, minimum, maximum, average etc. are to be made. Then, every dimension is composed of at least one or many hierarchies which is a structure of organizing the data in aggregation levels. For example, we can have a hierarchy for the 'location' dimension that is composed of levels from the 'county' level to the 'region' level to the 'country' level. We represented as well in Figure V-5, a dependency relationship that has the measures with dimensions because of the fact that every measure depends on the representation of some dimensions so it can be calculated. For example, for the indicator that analyses the 'average abundance of birds by region', the 'sum' function must be executed on the measure 'abundance' prior to dividing it by the count of facts to calculate the average. That measure depends, in this case, on the existence of the 'region' level in the 'location' dimension to answer acceptably to this indicator's requests. Otherwise, it will be considered as a problem in the design that we should as well consider when fusing models, and later on during the collective refinement step.

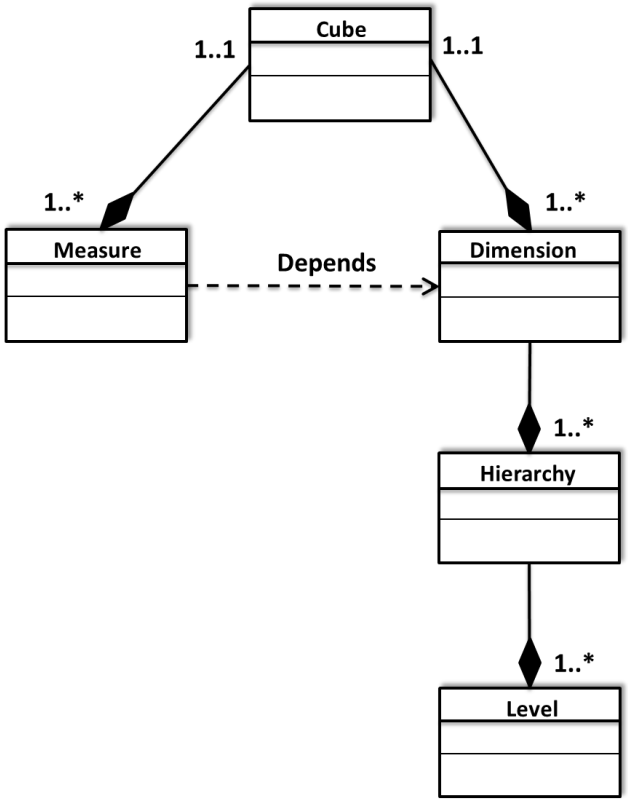


Figure V-5 metamodel of conceptual DW model

We here define three categories of multidimensional elements' issues that are going to be treated consequently by the fusion algorithm that we describe below, for the first two of them, and for the third by the next step of the refinement of the originated models:

— Differences: The same element has been defined in different models but with a dissimilar representation.
— Similarities: The same element has been defined in different models with the exact same representation.
— Conflicts: The element that has been defined and validated in the prototyping phase with the data sources, but that is misleading or likely to cause wrong interpretations in relation with the subject matter.

To solve the issues of *Differences* and *Similarities* of requirements, we define the algorithm 1 to merge the models.

Algorithm 1 Dimensions_fusion

```
Input: all cubes C //the set of cubes to fuse
Output: FinalCubes //a set of fused cubes
1: M = Measures of C;
2: For each m in M do
3: Generate new cube called FusionCube;
4: Add m to FusionCube;
5: Let CommonDims = Common dimensions of cubes with m;
6: Let NonCommonDims = Non-Common dimensions of cubes with m;
7: Add CommonDims and NonCommonDims to FusionCube;
8: Let NonConformedDims = Non-Conformed dimensions of cubes with m;
9:   For each d of NonConformedDims do
10:    Let H = hierarchies of d;
11:    d = Hierarchies_fusion(H);
12:    Add d to FusionCube;
13:  Endfor
14: Add FusionCube to FinalCubes;
15: Endfor
16: For each set of cubes having Common measures Cs do
17:   Add All Measures of Cs to all their originated cubes of FinalCubes
18: Endfor
19: return Cs
```

The dimensions fusion algorithm takes as input the set of the prototyped cubes' models and generates as output a set of fused ones. This is done by running through all the measures of all cubes combined and creating a new cube for each measure with its dependent dimensions as we illustrated in Figure V-5 by the measures and dimensions relationship. For that, we categorize the dimensions into three categories:

- Non common: dimensions that have been defined only in one model, which means that no shared measure has any of these dimensions as common between the cubes to which it belongs.
- Common: dimensions that have been similarly defined in different models to which the measure belongs.
- Common but not conformed: dimensions that are defined in different models to which the measure belongs, but differently, which means that their hierarchies and levels are dissimilarly represented.

Then, for each measure the algorithm puts in a new model the common and the non-common dimensions and merges the hierarchies of those that are common but not conformed using the hierarchies' fusion algorithm. The aggregators defined for each measure are all considered in the fusion and verified later in the step of collaborative refinement of the resulting models.

Algorithm 2 Hierarchies_fusion

```
Input: hierarchies h1, …, hn // set of hierarchies to fuse
Output: Dimension d // the resulting dimension
1: G = Union(h1, …, hn);
2: V = all Bottoms of G;
4:   If  size(V) > 1 do
5:     choose vBottom among V
6:    forEach node in V  do
7:       createEdge (G, vBottom, node);
8:    Endfor
9:  Endif
10: d = ∅;
11:  ForEach path in G do
12:      add path to d
13:  Endfor
14: return d
```

The algorithm 2 merges all the hierarchies' levels in one oriented graph, and then finds all the possible paths from the leaf i.e. the highest level, to the root i.e. the lowest level. When the graph has multiple bottom leaves, the DW expert must choose one to be considered as the finest granularity of the dimension to return. For example, in Figure V-7, the level "day" was considered as the lowest level (lower than the level "decade") of the enriched hierarchy of the dimension "Time".

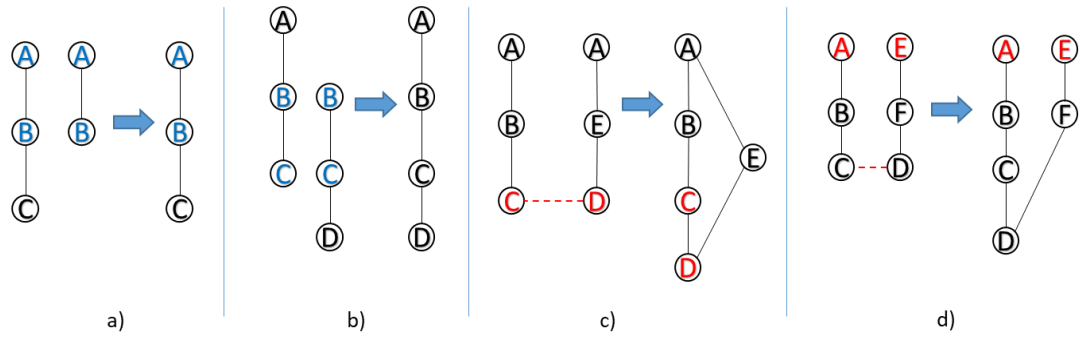


Figure V-6 Hierarchies fusion possibilities

In Figure V-6, we have four possible fusion cases that we consider treating in the hierarchies' fusion, algorithm 2:

1. In the case where a hierarchy is included in another, we use the including one:

— A→B→C.

2. If only a part of a hierarchy is included in another, we merge the two hierarchies:

— A→B→C→D.

3. If the lowest levels are not the same, one of them is considered as a child of the other, so we keep both hierarchies with unified lowest level:

— A→B→C→D.

— A→E→D.

4. If the top level is not the same, multiple hierarchies with unique lowest level are to be given:

— A→B→C→D.

— E→F→D.

These paths are the enriched hierarchies of the returned dimension. Following algorithm 2, an example of multiple hierarchies of the 'Time' dimension is shown in Figure V-7, where we have multiple paths generated at the end, and represented in the resulting dimension.

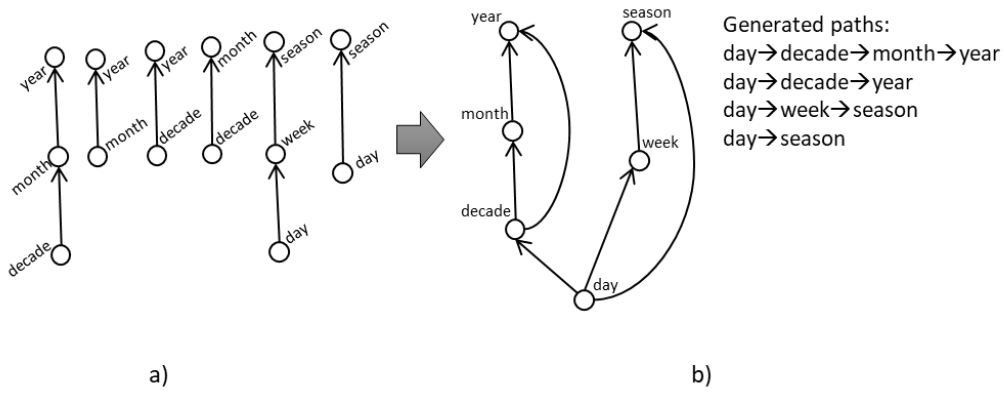


Figure V-7 Example of hierarchies' fusion of the 'Time' dimension

After the fusion of the hierarchies of the non-conformed common dimensions, they are added to the fusion cube, the same as the common and non-common ones. The final step of algorithm 1 consists of adding, for each issued fused cube, all the measures of its original fused models and that have not been shared amongst them. At this level, the DW designer saves also the indication that has been initially given by the volunteer or group of volunteers that have defined each model so that every fused cube keeps a description of its main analysis objectives.

A detailed example of the fusion is illustrated in Figure V-8 where three initial models defined respectively to analyse birds' abundance, behaviors and presence. The fusion of the cubes is performed by means of the algorithm 1. The two first models, namely 'Abundance' and 'Behaviour' have one common measure which is 'Abundance' highlighted in green, for which the algorithm has

started by creating the 'F1' model. Then the only one common dimension, 'Species' which is circled in green, and the non-common dimensions, 'Behaviour' and 'User' that are circled in red, are all together added to the model 'F1'. After that the algorithm 2 is called with the common but non-conformed dimensions 'Date' and 'Location', circled in blue in Figure V-8, to fuse their hierarchies. The dimensions with fused hierarchies are after that also added to the 'F1' model. Eventually, all the non-shared measures of all the cubes that have participated in the generation of the model 'F1', in this case measure 'Mortality' and 'Behaviour', are added to its measures list. The same logic is followed with the models 'Behaviour' and 'Presence' that have in common the 'Behaviour' measure, and which have given the model 'F2' in Figure V-5.

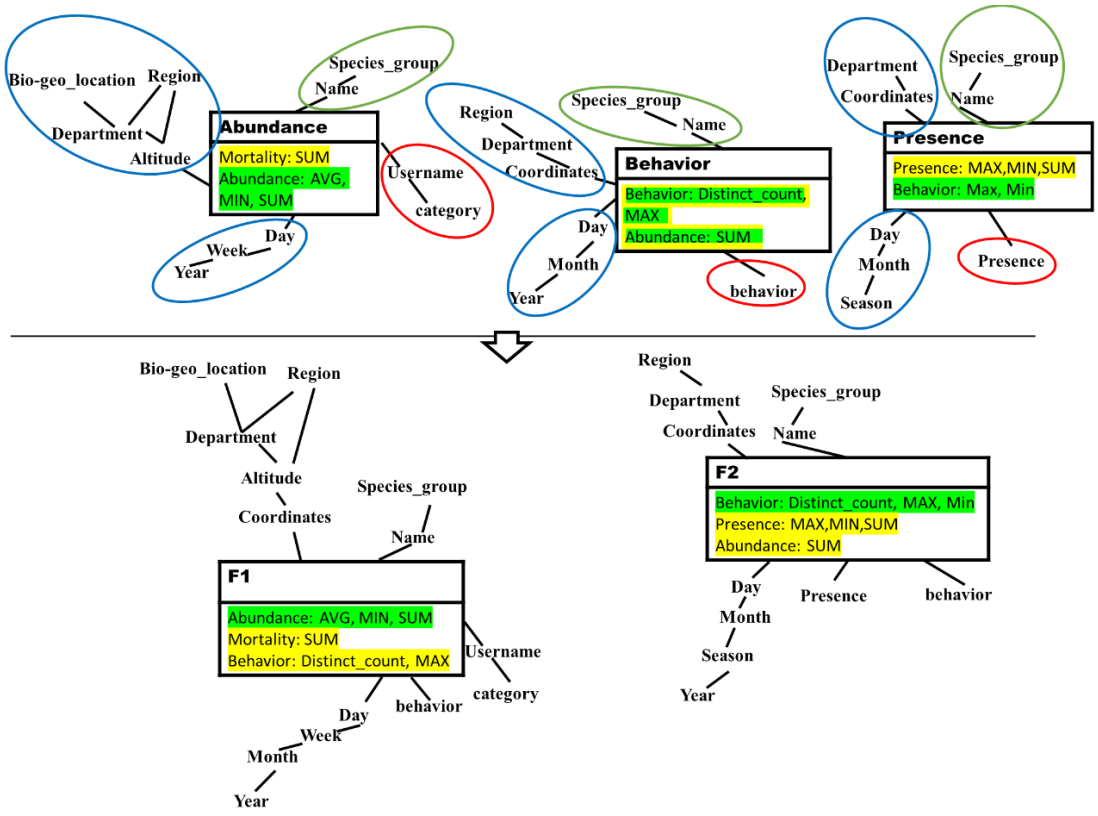


Figure V-8 Fusion example: VGI4BIO cubes

We can imagine in the case of too many models defined in the first step that their fusion would remarkably reduce their numbers, and consequently, the amount of time and effort that would require the implementation phase. Also, since the fusion is based on the measures' sharing among models, it is only adding coherent additional analysis possibility to each of the predefined measures. For example, in the resulting model 'F2' the new combination between presence and behaviour might be considered as an interesting analysis subject by the users that have defined both the 'Behaviour' and 'Presence' in initial models. More importantly, it is realizable from a data

warehouse design view, which is not always the case where data marts are fused, or additional elements are integrated after the validation of requirements realizability on the available data sources. This is so because if a measure exists in all the initial models, its aggregators' calculation is essentially feasible with all of these cubes' dimensions, since an initial verification has already been realized for the prototyping purpose. This implies as well, the possibility of introducing some new calculated measures that can exploit the newly available combinations of measures such as for example a ratio of birds' mortality by behaviour in the model 'F1'. It is also important to mention at this stage, that we do not consider the fusion of the resulting models, since we believe it very likely, especially when the number of initial models is important, to produce too many incoherent combinations of measures with dimensions that their analysis results are meaningless. For example, if we merge the models 'F1' and 'F2' based on the shared measures that were not there before the fusion, new feasible but senseless queries are going to appear. For instance, 'The presence of birds by the observer users' is not of any use since the 'Presence' measure is defined to see whether certain migrant species are present during each season and the user that collected the data is only considerable to check the category of users that have been capable of observing the mortality occurrences of species that usually fly in certain altitude.

As we have mentioned earlier, the algorithms 1 and 2 solve the issues related to the *differences* and *similarities* detected among the elicited models. The remaining limitation that the automatic fusion of models does not tackle, is the irrelevance of the defined multidimensional elements in regard with the subject matter, that we referred to as *conflicts*. This is handled by the collaborative resolution of requirement conflicts that we detail in the next subsection.

V.2.2.2. Collaborative resolution of requirements conflicts

For this step, we suggest two different methods: A simplified one that relies on a limited number of evaluation criteria to allow its execution with existing GDSS systems, and a profile-aware one that extends the evaluation criteria since we suggest a new GDSS solution that allows its execution.

V.2.2.2.1. Simplified collaborative method

The aim of this step is to solve the conflicts that are caused either by the limited knowledge or experience that characterize the volunteers on the one hand or engendered at the previous step of models' fusion that outputs larger models with potentially controversial newly generated elements on the other hand. We handle this by a collaborative refinement of the obtained models. The main

question that this step treats is "Are the multidimensional elements i.e. dimensions, hierarchies, levels and measures merged by the fusion of the prototyped models' step, consistent for the subject matter analysis purposes? In addition to the use of the GDSS in requirements elicitation scenario that we have described previously in the requirements engineering step of our methodology, our main use of its group techniques is done in this step of collaborative refinement that we illustrate in Figure V-9. The committers are a group of qualified users who are experienced with the original data sets and that are fully involved in the project. Their role is to participate in a group meeting in order to evaluate the contextual relevance of the DW conceptual models. Before starting the group meeting, the cube's main analysis objective that has been defined initially by the eliciting volunteer or that have resulted of the fusion step, is reviewed so every committer expresses a weight of his/her confidence based on what level of expertise does he/she has in this specific subject of analysis.

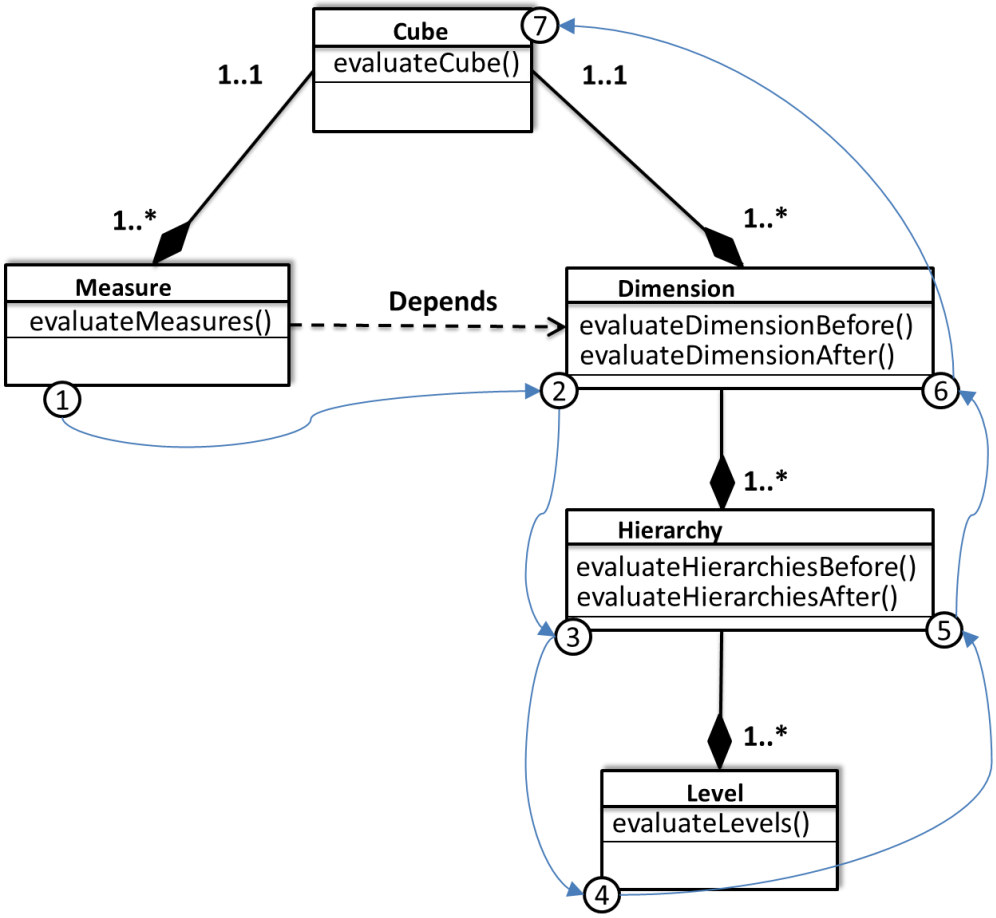


Figure V-9 Collaborative refinement of conceptual models

In Figure V-9, we propose a process of 7 steps in order to evaluate each of the cubes' elements in a logic that respects the composition of dimensions and the dependencies that the measures have with each dimension's correlated elements:

1) The committers start by evaluating the analysis relevance of the measures' aggregators in order to remove or correct those that they consider erroneously or partially defined by the inexperienced volunteers. For instance, the 'abundance' measure of some species must fulfil the acquisition protocol's constraint that requires performing the observation for a particular duration or distance e.g. 10 meters or beyond of butterflies' observation so it becomes considerable as useful, otherwise, it does not provide any biological significative information.

2) Afterwards, in the case where all the measures have been removed in the first step, there is no need to continue the process at all, the cube is therefore removed, and the meeting is ended at this step. If at least one measure is kept, the committers evaluate the holistic relevance of each dimension in order to rectify or remove the useless dimensions. For example, in Figure V-10, knowing the user that has collected the data, which is the information given by the 'Users' dimension, has been considered as irrelevant for the analysis objectives. It is therefore completely removed without considering its hierarchies nor levels in the next steps. Only the hierarchies of the dimensions kept by the end of this step are evaluated in the next steps. It is also important to mention that the holistic measures i.e. measures that use holistic aggregation functions such as the distinct count (Kimball 2013), are removed when their dependent dimension is not kept since this type of measures' aggregator calculation becomes erroneous if it does not have access to its finest level of granularity. For other measures i.e. distributive and algebraic, the dimension removal does not affect the aggregators' calculation correctness since they can be aggregated on its 'All' member, and then reused for the other aggregations using for example using the materialized views (Quass 1998). Consequently, in our example of Figure V-10, if we remove the 'Behaviour' dimension, the holistic aggregator 'Distinct_count' of the 'Behaviour' measure must be removed as well.

3) The same in this step, if all the dimensions are removed, the meeting is ended, and the cube is rejected. Once all the dimensions are evaluated, the committers must, then, evaluate each retained dimension's hierarchies to check their completeness and the accuracy of its lowest level of granularity. For example, in Figure V-10 the committers considered that all the dimensions' hierarchies are well-defined except for the hierarchy {Coordinates→Altitude→Region} of the 'Location' dimension. Let us note that at this step the dimension is removed if all its hierarchies are eliminated.

4) In the fourth step, the committers evaluate the levels of all the hierarchies. The levels that are considered useless for the analysis objectives are removed and those that present potential ambiguous interpretations are corrected. For example, in Figure V-10 the level 'week' of the temporal dimension is considered useless since the committers know that there is no analysis protocol that relies on weekly reporting. Also, the level 'Species_group' has been renamed to 'Species_family' for more clarity.

5) After the evaluation of the levels of all the hierarchies, a second evaluation of the hierarchies that their levels have been modified or removed, is made in order to confirm that the richness of their dimension's analysis information i.e. completeness, is still acceptable.

6) The same logic is applied to every dimension as a whole of hierarchies that have been modified. In this step, they are evaluated to see whether the modifications of the hierarchies would affect the usefulness of the dimension, especially if the lowest level of details is changed. In this step, it is up to the DW designer to decide if the evaluation requires a demonstration with a prototype for clarity reasons, such as what proposed (Golfarelli and Rizzi 2011), or if it would be sufficient enough without it.

7) Finally, the committers must evaluate the usability of the cube with the modifications that have been performed on its components because of the fact that the number of the used dimensions affects the usability of the cube, and so the decision-making process. The committers decide whether to implement or not the resulting cube.

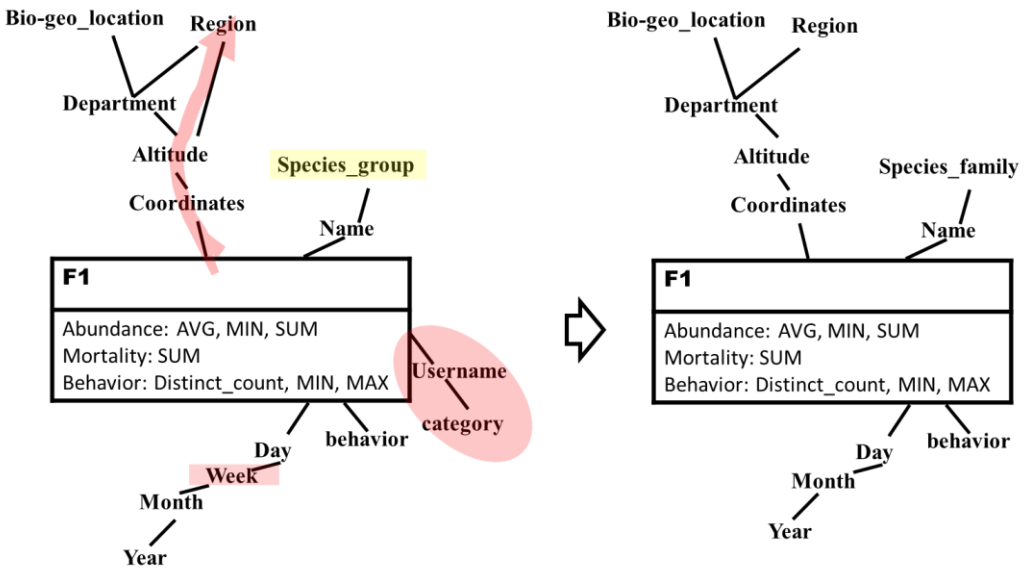


Figure V-10 Example of cube's refinement result

To execute this process, we use a GDSS that offers a ‘Direct vote’ and ‘Multicriteria’ decision making techniques (Zaraté et al. 2016). The steps of this process are summarized in Table V-1 with the criteria used for each group activity.

Table V-1 Steps of collaborative evaluation of conceptual models

Step	Group activity	Input	Output	Used technique	Criteria
-	Set Confidence Level	Analysis objective	Confidence levels	Parametrization step	Subject matter mastery
1	Evaluate Measures	Measures	Measures	Direct vote	Usefulness
2	Evaluate Dimensions Before	Dimensions	Ranked useful Dimensions	Multicriteria evaluation	Usefulness Lowest level accuracy
3	Evaluate Hierarchies Before	Dimension hierarchies	Ranked Dimensions	Direct vote	Usefulness
4	Evaluate Levels	Rich hierarchies	hierarchies	Direct vote	Usefulness
5	Evaluate Hierarchies After	hierarchies	hierarchies	Direct vote	Richness
6	Evaluate Dimensions After	Dimension	Dimension	Direct vote	Usability
7	Evaluate Cube	Cube	Final cube	Direct vote	Usefulness

The committers confidence level defined for the cube according to their skills in its subject matter is going to be used when aggregating the participants’ preferences. It will prioritize the choices of committers that are more specialized than the others. For example, a participant who is an expert specialized in ornithology, would have his/her confidence level for the cube that analyses the birds’ behaviour, set to a higher level than what an ecologist would have when evaluating the same cube together. After the parametrization step, as shown in Table V-1, the group processes used for each step are either ‘Direct vote’ where only one criterion is used for the evaluation e.g. ’usefulness‘ for steps 1,3,4 and 7 of Table V-1, or ‘Multicriteria’ where more than one criterion is necessary for the evaluation e.g. ‘usefulness’ and ‘Lowest level accuracy’ for step 2 of Table V-1. An example of this process’s use is detailed in V.4.

As this method requires a separate meeting for each step, which would make its execution time too long in case we extend the evaluation criteria for further precision, we define in the next subsection the profile-aware method that we developed a specific tool allowing its quicker execution.

V.2.2.2.2. Profile-aware collaborative method

In this section we extend the step ‘collaborative resolution of requirements conflicts’ of our methodology so it can be used by both volunteer users in the case of crowdsourcing projects and by employed users in the case of enterprise projects. We do so by considering the differences

between users engaged in DW projects whose involvement differs significantly from volunteers to employees as we detailed previously in Table III-1 of section III.4. We also use a more specific definition of our evaluation criteria that have been first revisited by (Bimonte et al. 2020) in their extension of our requirements elicitation methodology (Bimonte et al. 2018).

Listed in Table V-2, these criteria are an extension of our criteria of evaluation defined in Table V-1 i.e. usefulness, lowest level accuracy, richness and usability, and they are acquired based on quality attributes of databases (Batini and Scannapieco 2006).

Table V-2 Evaluation criteria extended by coauthors in "(Bimonte et al. 2020)"

-	Criterion	Applies to	Meaning
1	Completeness	all sets of elements	All necessary concepts have been modeled.
2	Precision	Measure, Dimension	The element is represented with sufficient abstraction.
3	Relevance	Measure, Level	The element is useful for analysis.
4	Minimality	all sets of elements	No redundancies.
5	Consistency	Measure, Hierarchy	Rules of application domain are respected.
6	Certainty	all single elements	No ambiguities in the chosen appellations.
7	Usability	The whole cube	The cube allows the sought analysis.
8	Confidentiality	Measure, Level	Causes no legal, privacy or confidentiality issues.

In addition to that, we suggest using a measure of user trustworthiness such as what propose (Fogliaroni et al. 2018) for VGI crowdsourcing data or the work of (Green and Howe 2011) that defines and substantiates the trustworthiness measure from a larger perspective. We do that because we consider using only the **expertise** in the subject matter i.e. the criterion that we have used in Table V-1 for our simplified approach, not precise enough to properly qualify the users' definitions. Hence, we can associate to each multidimensional element an attribute of its definer's **trustworthiness** and use it to keep a track of the participant's credibility, reliability, reputation, etc. This will give us a more precise hint on the quality of the element during the evaluation sessions rather than if we limited it to a more or less subjective claim of having enough knowledge about an application domain that might be not always true, in case of a biased committer's self-orientation for example. Thus, if the data acquisition platform already has a reputation or profile qualification of its volunteers, which is usually the case (Degrossi et al. 2017), we can consider the user reputation feature. It, for example, may be defined as the ratio of the volunteer's unrejected data entries' number to his/her total entries' number. It is indeed a fundamental practice in crowdsourcing platforms' data quality assessment (Daniel et al. 2018), that we can rely on to consider more

cautiously the definitions of less trustworthy users and to move forward without evaluation with those defined by highly trustworthy users. (Bishr and Janowicz 2010) say that the reliability of definitions increases with that of its definer's profile, whence the appellation of our second method 'profile-aware' is derived. In the case of applying our methodology with crowdsourced data where the user reputation values are available, the trustworthiness attribute can be calculated using one of the approaches of (Bishr and Janowicz 2010; Fogliaroni et al. 2018), for example. Whereas, either if the data crowdsourcing platform does not provide user reputation values or if the methodology is used in an enterprise project where individual reputation of employed users is not questionable, it would be calculated without considering the volunteer reputation variable as suggests the work of (Green and Howe 2011). Furthermore, although integrating these attributes to our approach seems promising, either in the fusion algorithm or in the GDSS configuration, we, however, have chosen to move forward with a simpler attribute i.e. **attractiveness,** that replaces the trustworthiness formula since it is still out of the scope of our contributions in this work. Thus, the attractiveness attribute can be calculated as the ratio of element's occurrences to the total number of defined models in the initial set of validated prototypes before the fusion, the higher the better. For example, if the level 'year' has been defined in 4 models out of 5, its attractiveness would be 80%, higher than a threshold value that the facilitator of the group meeting must define before the evaluation, let's say 50%, and therefore it will be considered as valid without further evaluation, while if its attractiveness were lower than the threshold, this would mean that it shall be evaluated. The process that we suggest using the extended criteria of Table V-2, and this attractiveness attribute is shown in Figure V-11.

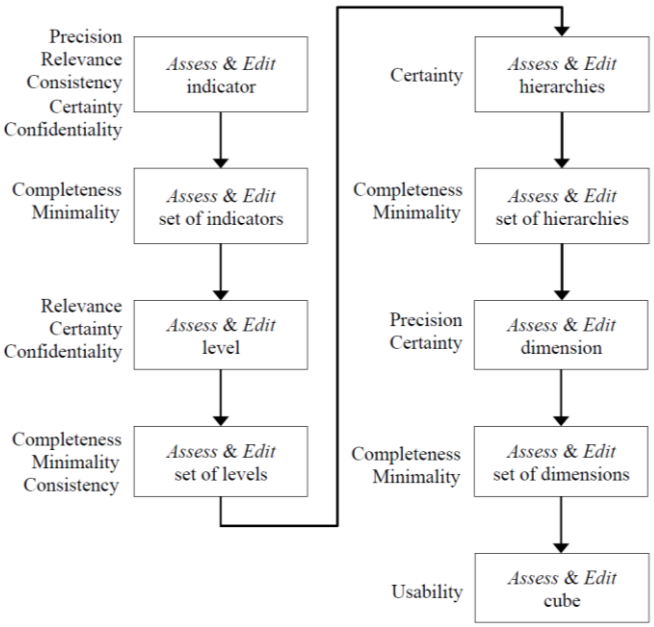


Figure V-11 Profile-aware collaborative refinement

Despite the fact that using multidimensional elements attractiveness and the extended criteria of evaluation makes our profile-aware method more meticulous in treating the conflictual definitions, it however has the drawback of being too long if executed using any existing GDSS. This because, for each step of evaluating an element against more than one criterion, we avoid aggregating the evaluation results in order to keep a separate track of each criterion's feedback and therefore be able to tackle every criterion's limitation by itself even though in similar GDSS situations this would imply using a Multicriteria process. Furthermore, this would necessitate a meeting for each element that has a low attractiveness using each associated criterion and with a weighted sum calculation method, which would consequently require an unrealistic number of group processes for example using the 'Direct vote' process proposed in GRUS, i.e. the GDSS that we use with our simplified method and that we conduct with which a set of experiments detailed in the following chapter. To solve these issues, we propose in the Chapter VII a new GDSS that we developed to solve the limitations reported in the next chapter, and that in which we offer as well a new Thinklet i.e. a concept of group activity that we define in detail in VII.2, that using which we do only one meeting for our profile-aware group process. In addition to that, the Thinklet that we propose allows dynamic configurations that handle the two previously mentioned application cases i.e. with

crowdsourcing or enterprise users, as well as with the two possible user profiles i.e. trustworthy or not.

V.3. Implementation environment and used technology

In this section we describe our implementation of the methodology in (i) its phase of 'requirements elicitation and validation on data' i.e. in subsection **Semi-structured interviews and validation** and (ii) its phase 'fusion of prototyped models' i.e. in subsection **fusion algorithm implementation**. For the phase 'collaborative resolution of requirements conflicts', we validate in this chapter the simplified method with an exisiting GDSS called GRUS — a web-based group support system that can be used to organize collaborative meetings (Camilleri and Zaraté 2019) that we describe thouroughly in Chapter VI— and, since we rely for its execution on a new GDSS, we test the profile-aware method in Chapter VII after introducing its new execution environment.

V.3.1. Semi-structured interviews and validation

As we have mentioned in section V.2.1.1, we adopt the 'pivot table' as the formalism allowing the volunteers to express their analysis needs. To elicit the requirements, we provide volunteers with a simple example of a pivot table defined using an Excel file that contains measures and dimension members organized into hierarchies. Then, we ask each of them to provide his/her needs in a similar representation using some sample data that they know. An example of a pivot table defined using Excel is shown in Table V-3.

Table V-3 Volunteer requirements elicited in Excel pivot table example

species	location			date		presence
Paridae	ARA			1978	40-1978	1,00
	ARA	puy	montagne	1978	40-1978	1,00
Sturnidae	ARA			1978	3-1978	1,00
					40-1978	0,00
	ARA	puy	mer	1978	3-1978	1,00
					40-1978	0,00

In this example, the volunteer has proposed one measure aggregated with the sum, and three dimensions (species, location, and date). It only shows a preliminary interaction of the methodology which does not represent the final DW prototype that the volunteer will move forward with. To

help users to propose well-defined pivot tables, we do a semi-structured interview for each pivot table's definition, while the DW expert checks at each iteration the retrievability of the requirements from the data sources. This semi-structured interview is composed of 2 phases: (i) For each dimension, the DW expert asks the volunteer to validate or modify his/her hierarchy, (ii) Then, the measures and their aggregation functions are questioned. An example of the questions that DW expert must be asking during the semi-structured interview about the pivot table of Table V-3 and that would help the volunteers to improve their definitions of hierarchies might be: "*For the column location, do you think that there is a need to have a coarser level grouping region into biological locations for example?*". Another example of questions that might be used to identify complex DW structures such as non-strict hierarchies (Pedersen et al. 2001) is: "*Do you think that a region can have several biological locations?*". This exchange about the XSL[6] file between DW experts and volunteers as decision-makers is also important in the case where the volunteer has difficulties with defining well-formed multidimensional elements that require a DW expert's intervention to correct any logical errors in the pivot tables. Another important property of the pivot table formalism's use is, as described by (Nabli et al. 2005), that they can be formally translated into DW models.

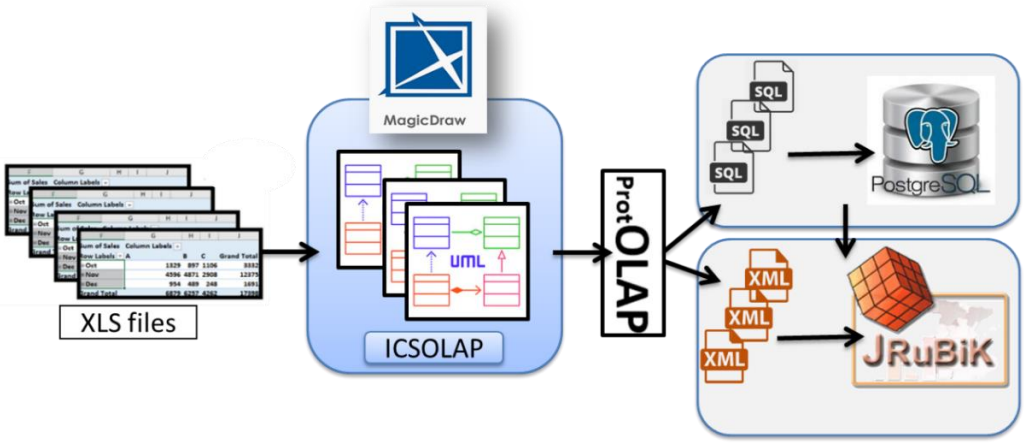
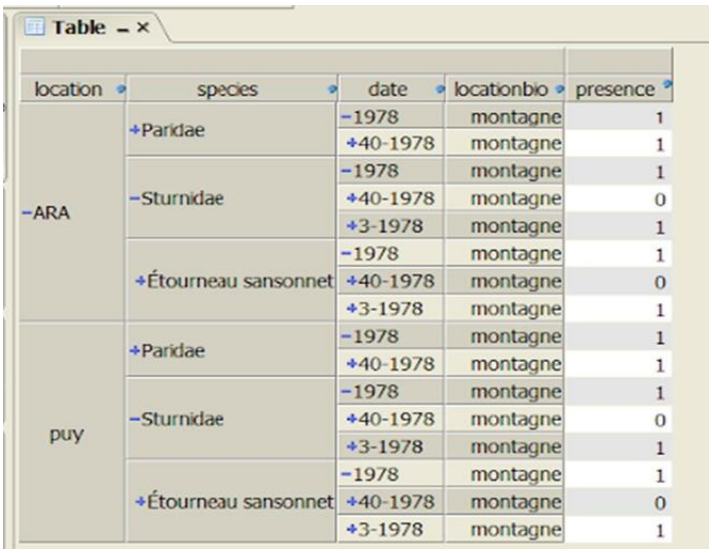


Figure V-12 The technical solutions used in our requirements elicitation implementation

As illustrated in Figure V-12, this is done by DW experts using the ProtOLAP tool (Bimonte et al. 2013b). ProtOLAP takes as input a UML model defined using the ICSOLAP(Bimonte et al. 2013a) which is a UML profile for OLAP/SOLAP modelling implemented in the CASE[7] tool MagicDraw[8].

[6] Stands for eXtensible Stylesheet Language, a styling language for XML.
[7] Stands for Computer Aided Software Engineering tools.
[8] www.nomagic.com/products/magicdraw

It automatically creates the SQL scripts for PostgreSQL i.e. tables' creation and data insertion SQL codes, as well as XML configuration files for JRubik[9] i.e. the open source OLAP client that uses Mondrian[10] as OLAP server. ICSOLAP has also the advantage of allowing the creation of a semantic repository for the terms that have been previously used by others in order to decrees as possible the semantic divergence especially in cases where we can tell that the analysis objectives are closely similar and relying on the same source of data. This allows therefore the pivot tables to be translated into UML models, which *is the specification formalism of requirements*, then to a DW prototype that can be created using PostgreSQL and JRubik for the validation with volunteers. An example of an implemented model queried in JRubik is shown in Figure V-13.



Table

location	species	date	locationbio	presence
−ARA	+Paridae	−1978	montagne	1
		+40-1978	montagne	1
	−Sturnidae	−1978	montagne	1
		+40-1978	montagne	0
		+3-1978	montagne	1
	+Étourneau sansonnet	−1978	montagne	1
		+40-1978	montagne	0
		+3-1978	montagne	1
puy	+Paridae	−1978	montagne	1
		+40-1978	montagne	1
	−Sturnidae	−1978	montagne	1
		+40-1978	montagne	0
		+3-1978	montagne	1
	+Étourneau sansonnet	−1978	montagne	1
		+40-1978	montagne	0
		+3-1978	montagne	1

Figure V-13 Example of JRubik prototype

This process has two important properties. Firstly, allowing generating a rapid prototype that is shown to the volunteer to validate its content and that it fits well with the intended analysis. And secondly, it allows the DW expert to validate the availability of the necessary datasets and, using existing OLAP servers and DBMSs, the implementation feasibility of complex DW models such as complex hierarchies, facts-dimensions relationships, etc. (Pedersen et al. 2001). This is important since indeed sometimes due to implementation issues, the pivot tables can be translated into quite different resulting models. For example, to avoid the non-strict problem related to introducing a 'biological region' level to the 'location' dimension suggested in the pivot table of Table V-3, the DW expert might decide to create a separate dimension for this level. Once the prototype is implemented, it is presented to the volunteer(s) that validate it before applying the fusion step. If

[9] rubik.sourceforge.net/jrubik/intro.html
[10] mondrian.pentaho.org

not validated, volunteers are asked to update the pivot tables in question and the validation step is iteratively applied until reaching the satisfactory model that will be altered after the fusion with the other models in the further steps of our methodology.

V.3.2. Fusion algorithm implementation

The fusion algorithms i.e. algorithm 1 that merges the models and algorithm 2 that merges the non-conformed common dimensions' hierarchies, that we detailed in section V.2.2.1, have been implemented using java as a programming language. For that, we have used the open source Eclipse IDE[11] that allows frameworks integration and with the Apache maven build automation tool[12] for its interoperability, automated builds and pre-defined packaging, etc. This would allow an easier integration of its features in case of future extension such as within the upcoming version of ProtOLAP that such an automation technique would be suitably adaptable with its functionalities. For the Algorithm 2 of section V.2.2.1, we have used the jgraphT[13] java library that allows structured manipulation of graphs assisted by predefined algorithms. The use of jgraphT framework for the hierarchies' fusion has allowed us to optimize the code since it offers various graph iterators, algorithms such as pathfinders, pre-defined standard graph theory objects such as 'edge' and 'vertex' etc. We have as well automated the execution of the originated application, except for the choice of the lowest level in the case where there are two independent ones then the user is asked to choose the correct order as explained in section V.2.2.1.

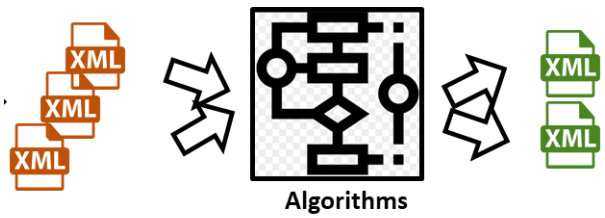


Figure V-14 Data flow of fusion algorithms

[11] www.eclipse.org
[12] maven.apache.org
[13] jgrapht.org

As illustrated in Figure V-14, the fusion java application takes as input the validated prototypes' xml configuration files that have been generated by the ProtOLAP tool. It merges them following the algorithms 1 and 2 detailed in section V.2.2.1 into new xml configuration files ready to be evaluated during the 'collaborative resolution of requirements conflicts' step. We have also considered an automatic importation of these files into the new GDSS platform that we propose later in this work, Chapter VII. A detailed example of this application's use with our VGI4BIO project's case study LPO is as well shown in V.4. Please see appendix A where the class diagram of the project is available for further code details.

V.4. Validation with the a VGI4BIO use case

In this section we detail the use of our methodology to design a DW with real volunteer users. The cube has been designed within the working environment that we have described in the previous section and implemented with some additional technologies, mainly for the ETL and the spatial data integration that we describe hereafter. For the collaborative resolution of the requirement conflicts, we have used the simplified method since for the profile-aware method we have not yet had at hand the necessary collaborative tool for its execution which we detail in Chapter VII its implementation and tests.

V.4.1. LPO involved volunteers

In the context of the VGI4BIO project, as we have introduced in III.2, we have mobilized 11 volunteers from those that have participated in more than 5 million observations stored in the Faune-Aquitaine database called 'Biolovision' and that is fed and maintained by the league of birds' protection – LPO for "Ligue pour la Protection des Oiseaux" in French. The list of the participant volunteers and their main analysis objectives is shown in Table V-4.

Table V-4 Volunteers of LPO database

-	Volunteer affiliation	interest	Defined model	Prototyping sessions
1	LPO Aquitaine	Analysis of Atlas codes	Atlas-behaviour	2
2	LPO Aquitaine	Phenology stages of insects	Phenology-stages	2
3	LPO Aquitaine	Climate change impacts	Mortality/abundance	2
4	Farmer	Sensible species presence	Species-sensibility	3
5	Farmer	presence of species in communes	Abundance/commune	3
6	LPO Aquitaine	Species diversity	Presence/time	3
7	DREAL[14]	Presence of species by location	Presence/location	3
8	LPO Aquitaine	Abundance by geographic coordinates	Spatial-abundance	3
9	LPO Aquitaine	Observation by geographic coordinates	Spatial-observations	3
10	OAFS[15]	species' habitat	Observation/habitat	2
11	LPO Aquitaine	Threatened species	Diversity	2

The volunteers that have participated in this project were, as shown in Table V-4, from 4 different affiliations (2 farmers, 7 LPO Aquitaine association members, 1 member of DREAL and 1 member of OAFS) and with various analysis objectives. We do not provide more information about the volunteers' profiles for privacy reasons.

V.4.2. LPO's RE phase

With each volunteer, we have organized at least two sessions of 1-2 hours each. The first session was about an introduction to data warehousing and OLAP systems' advantages that is done by videoconference. During this introductory session, we have explained to each user what advantageous queries and, hard to request otherwise, analysis combinations OLAP systems allow. For that we have used another already deployed DW system which analyses agriculture related issues in order to accentuate the prominent use of OLAP technology by demonstrating its usefulness in a field similar to the volunteers' central interest. Afterwards, we have done in a second videoconference with each user, 1 or 2 requirements elicitation sessions using the

[14] Abbreviation for Regional Management for Environment, Housing and Development (Direction Régionale Environnement Aménagement Logement)- occitanie.developpement-durable.gouv.fr/

[15] Abbreviation for Aquitaine Wildlife Observatory (Observatoire Aquitain de la Faune Sauvage) si-faune.oafs.fr/

working environment that we have detailed in V.3.1. The resulting conceptual models after the validation are illustrated in Figure V-15.

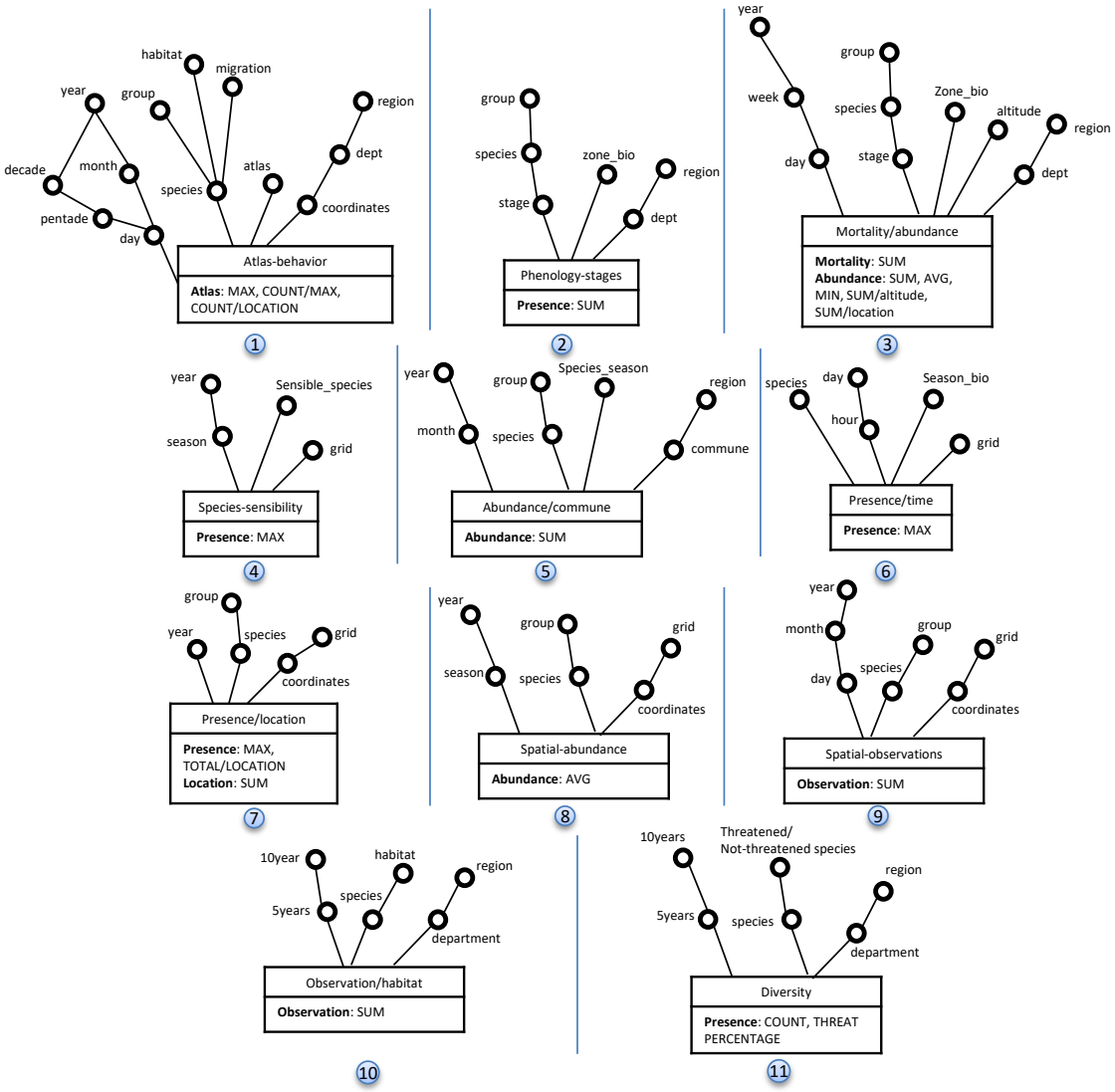


Figure V-15 Conceptual models for LPO project prototypes

The verification of data retrievability from the sources was not complicated in this project, which is probably the case with most observatory data, since it relies on a unique data source i.e. the observatory database, and with data that have been collected according to a normalized protocol even though its recommendations might not be always respected. This has been accomplished in a negligible time since the only unavailable information that we have had needed an external source

to complete it was the atlas codes i.e. birds' behaviour signification codes, and that we have retrieved from the complementary symbols' explanation list available at[16].

V.4.3. LPO Models' fusion

We have fused the models using the fusion algorithm whose description and implementation were detailed earlier in this chapter, respectively in V.2.2.1 and V.3.2. This step has reduced the models' number from 11 to 4 as illustrated in Figure V-16. The originated model F1 has been kept in its original state as in Figure V-15 since it only contains an unshared measure which is 'Atlas'. However, the three other models are newly generated by the algorithm as follows:

- The model F2 is the fusion of the models 2,4,6,7 and 11 of Figure V-15 that all share the measure 'Presence'.
- The model F3 is the fusion of the models 3,5 and 8 of Figure V-15 that all share the measure 'Abundance'.
- The model F4 is the fusion of the models 9 and 10 of Figure V-15 that share the measure 'Observation'.

[16] www.faune-tarn-aveyron.org/index.php?m_id=41

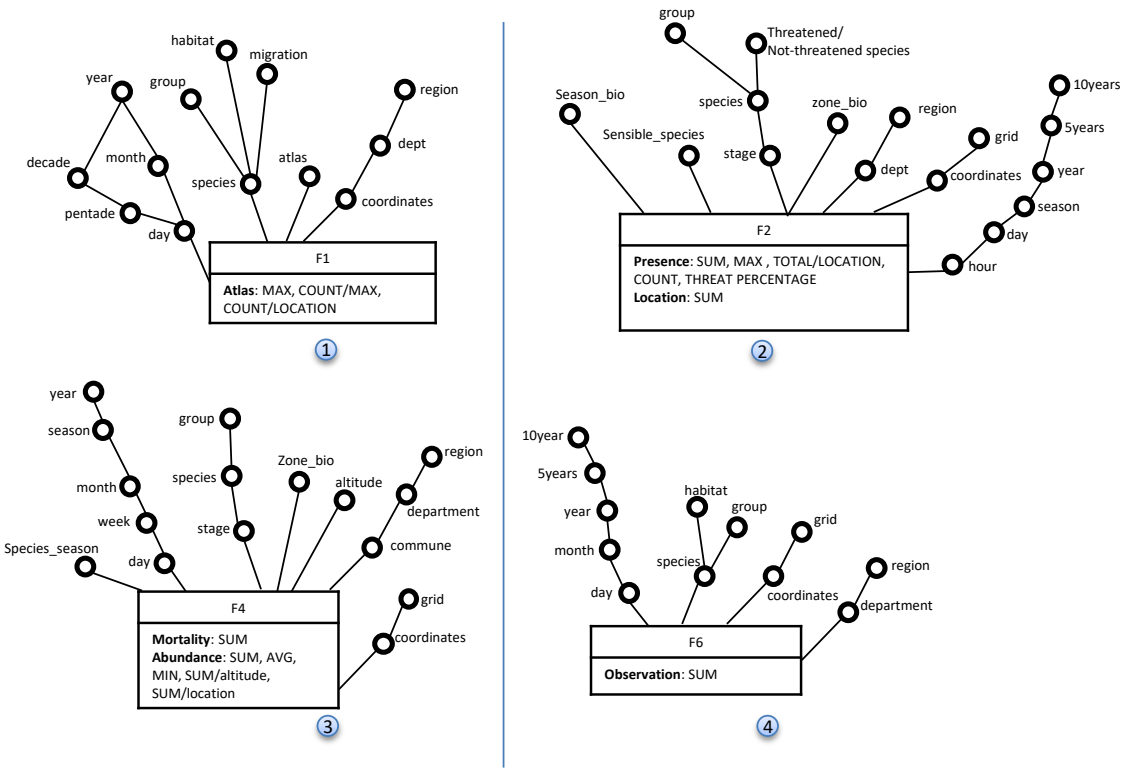


Figure V-16 LPO fused models

It is obvious that the generated models contain, in addition to some possible imprecise or ambiguous initial definitions of the volunteers, some newly generated questionable multidimensional elements that must be validated by knowledgeable subject matter's specialists. At this level we have initiated the preparation of the GDSS use for the refinement step. We have started by preparing some prototypes with these 4 models and invited some biologists and ornithologists amongst the VGI4BIO project's partners to participate in the group meeting(s).

V.4.4. Use of the GDSS to refine LPO's fused models

Prior to the GDSS meeting(s), we have done a videoconference with three experts that accepted to participate in the evaluation sessions during which we have presented to them the 4 prototypes issued from the fusion step. These three committers are experts from the LPO Aquitaine, DREAL and AgroParisTech[17] that know very well the data sources, since they have used them either before or during the VGI4BIO project, and who have respectively the profiles: ecologist, environmental manager and agricultural engineer. After the videoconference that introduced the committers to the

[17] Paris institute of technology for life, food and environmental sciences.

intended work, we have decided to run a first experiment with a straw model inspired by the volunteers' definitions in order to test the performance of the methodology with the GDSS that we use. This choice has been made to serve as a pre-meeting middle-of-the-road solution (Wong 2010) in order to allow building a better understanding and to avoid treating the at hand real models biasedly if no least common ground is pre-built. The straw model that we have used has been defined by an LPO Aquitaine volunteer as shown in Figure V-17a and has given the model of Figure V-17b after the use of the GDSS system GRUS for an evaluation.

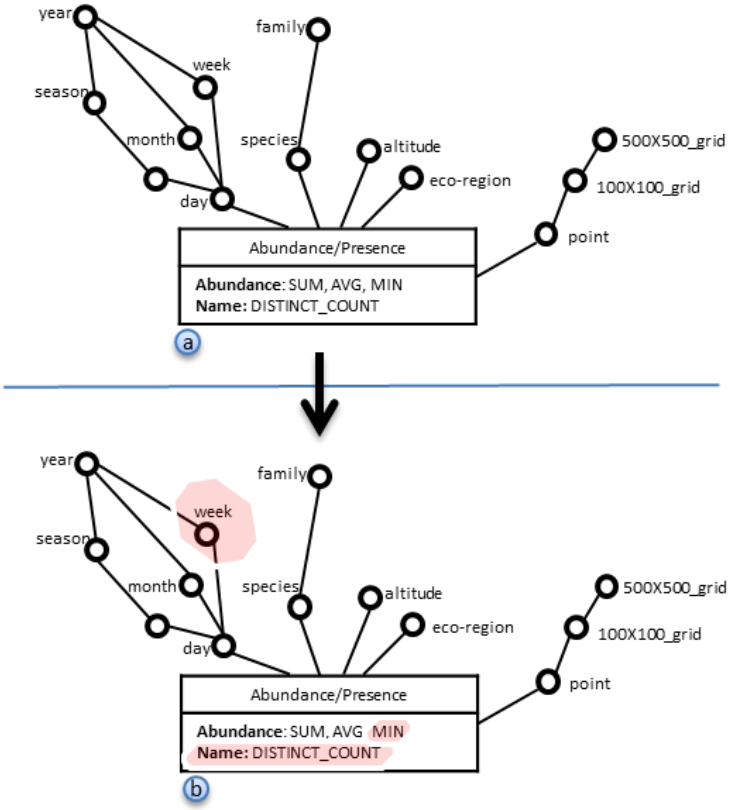


Figure V-17 Straw model of LPO

This evaluation has been done using the simplified collaborative method that we have detailed in V.2.2.2.1 and with only authorizing unsatisfactory multidimensional elements removal whereas the modification and creation of new elements actions will be as well permitted when working with the fusion issued models. The "Evaluate Measures" and "Evaluate Levels" group activities have eliminated respectively the "Abundance:MIN" and "Name:DISTINCT_COUNT" {Measure:AGGREGATOR} couples and the "week" level from the temporal dimension as shown in Figure V-17b. The committers have eventually evaluated the cube and considered it useful for

its wanted analysis. The details of the group activities “Evaluate Measures” of the ‘Abundance’ measure and “Evaluate Levels” of the temporal dimension ‘Date’ are presented respectively in Table V-5 and Table V-6.

Table V-5 Measure ‘Abundance’ evaluation results

{Measure:Aggregator} /Committer	Comiitter1	Committer2	Committer3	Result	Ranking
Abundance:SUM	5	3	5	37	1
Abundance:AVG	4	4	4	36	2
Abundance:MIN	3	1	5	25	3

Table V-6 Dimension 'Date' evaluation results

Dimension	Level	Committer1	Committer2	Committer3
Date	Day	Yes	Yes	No
	Month	Yes	Yes	Yes
	Week	Yes	No	No
	Season	Yes	Yes	Yes
	Year	Yes	Yes	Yes

The Table V-5 displays the ranking values for each committer and the classification in a decreasing order where the maximum possible value is 75 points and the threshold of measures’ acceptance that has been defined by the facilitator and agreed upon by the committers was set to 41% i.e. 30 out of 75 points at least. In Table V-6, we have used a majority vote which explains the removal of the level ‘week’ from the ‘Date’ dimension. A screenshot of the group meeting done with GRUS is shown in Figure V-18.

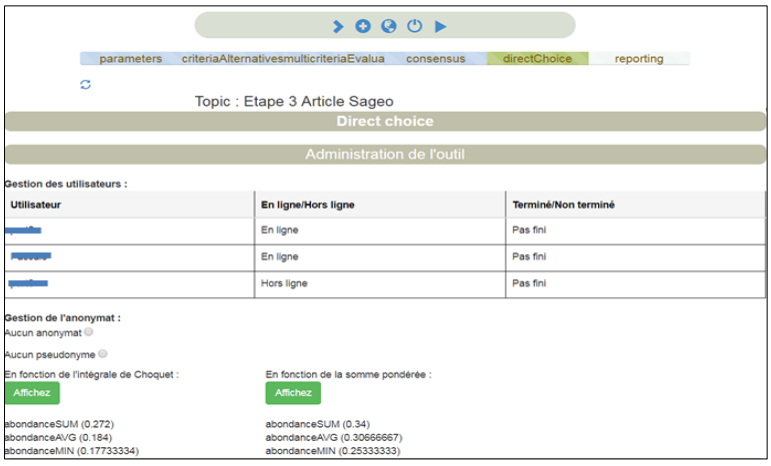


Figure V-18 GRUS experiment meeting screenshot

The total duration of the entire collaborative process was 2 hours and it was assisted by videoconference to allow communication during the meeting. The participants' feedback on the methodology was positive since they have expressed their willingness to participate in future evaluation meetings and considered both the methodology and the GRUS system helpful in organizing and therefore evaluating the components of an OLAP cube. On the other hand, they brought to light the need to make modifications on the cube's elements which will be authorized in the future meetings with the LPO fused model.

After this experiment of the methodology with GRUS — the GDSS that we had at hand at this level and that is still under development thus the tests that we conducted with which, detailed later in the following chapter, in order to better tackle the technical issues related to its well-functioning — we continued the refinement of the 4 fusion resulted models manually. Nevertheless, the methodology has been the same and the group of committers that have used GRUS with the similar straw model were also those with whom we have accomplished the evaluations that we have finished in three videoconference-assisted workshops. The committers have had more liberty on these models since, as mentioned earlier, all edit, add and delete actions were permitted. They have decided, after the presentation of the 4 models that we have done in a videoconference prior to the first test, to implement only one model analysing both the 'Atlas' and 'Abundance' measures which they have confirmed that a considerable correlation exists between the two of them. These two measures were retrieved from two separate models i.e. 1 and 3 of Figure V-16. For the models 2 and 4, they have used them to complete and refine the dimensions. The resulting model is illustrated in Figure V-19 where 8 dimensions have been validated:

1) Estimation: newly introduced by committers, contains only one level and available in the data sources as a recommended entry when collecting the data to mention the level of certainty of the entered counts and can have 4 possible values: 'Minimum', 'Exact-value', 'Estimation' and 'Not-counted'.
2) Date: Kept as in the model 1, with an additional level that comes from the model 3 i.e. 'season', and contains two hierarchies:
    i. day→month→year;
    ii. day→pentade→decade→season→year.
3) Species: kept as in the model 1, with an additional hierarchy that contains another upper level to the level 'species' and which comes from the model 2 i.e. 'sensible_species', which resulted 4 hierarchies:
    i. species→group;

ii. species→habitat;
iii. species→migration;
iv. species→sensibility.

4) Atlas: kept as in the model 1.
5) Location: kept as in model 1.
6) Altitude: kept as in model 3, but with a two-level hierarchy: altitude_class_10→ altitude_class_100, which divides the altitudes into two classes that the committers suggested for a more friendly exploration.
7) Grid: kept from the model 3, but with one only newly suggested hierarchy i.e. 05X_grid→ 10X_grid, that the committers consider better for special visualization.
8) Soil: newly introduced by committers, contains only one level and publicly accessible at18. It contains CLC data i.e. stands for CORINE Land Cover, that classifies biophysically the use of 39 European countries' lands e.g. arable land, forests, heterogeneous agricultural areas, etc.

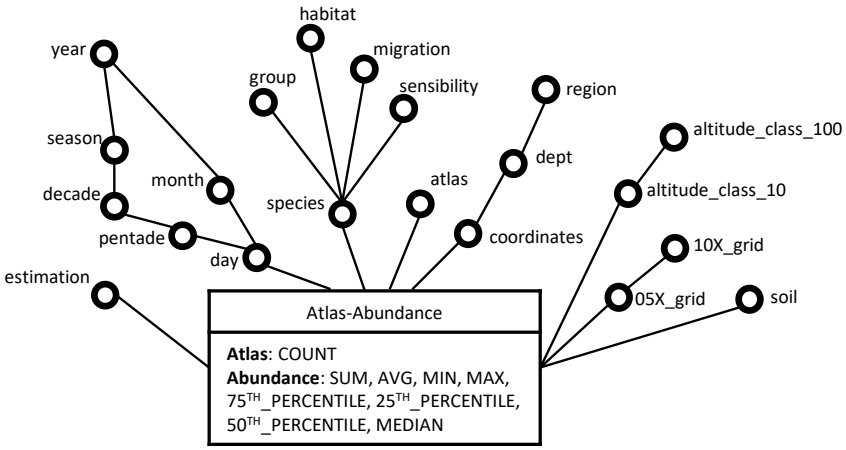


Figure V-19 Final LPO model after refinement

For the measures, the committers have validated the 'Atlas' of the model 1 but only with the 'COUNT' aggregator while the other measures have been considered irrelevant for the analysis and therefore removed. Also, the measure 'Abundance' of model 3 has been kept but only with the 'SUM', 'AVG' and 'MAX' aggregators while the rest of them have been removed as well. The committers have introduced 4 new aggregators for the measure 'Abundance': '75TH_PERCENTILE', '25TH_PERCENTILE', '50TH_PERCENTILE' and 'MEDIAN' that they

[18] https://www.data.gouv.fr/en/datasets/corine-land-cover-occupation-des-sols-en-france/

consider necessary when an in-depth analysis of the abundance is sought-after. Eventually, we have implemented this model using open source solutions:

- Postgres database loaded with over 3 million facts.
- Saiku[19] as an OLAP client with the Mondrian OLAP server.
- Talend open studio[20] which is a data integration software that we have used for the ETL.
- Qgis[21] which is a geographic information system that allowed us executing some advanced geographic operations, especially for the CLC data that we have needed for the soil types and the 5- and 10-kilometers grids for the 'Grid' dimension.
- PostGis[22] which is an extension of Postgres that allows creating and handling geographic data types.

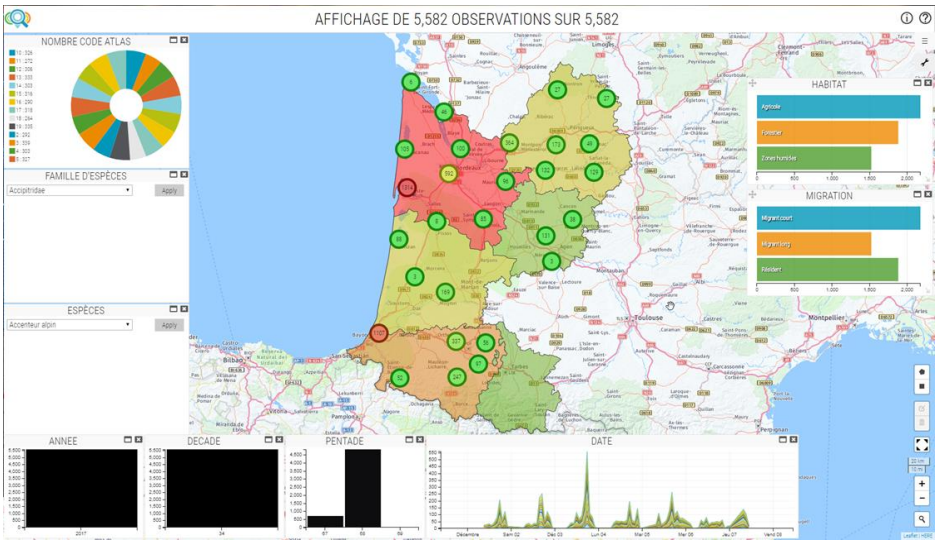


Figure V-20 Example of spatial query of 'Atlas:COUNT' measure of the LPO implemented cube

In Figure V-20, we show an example of a spatial query of the measure 'Atlas' aggregated by 'COUNT' run by a SOLAP client solution provided by GEOSYSTEMS France[23], one of the VGI4BIO project partners. The model has been validated and started being used, in a first step, by

[19] https://www.meteorite.bi/products/saiku/
[20] https://www.talend.com/products/talend-open-studio/
[21] https://qgis.org/en/site/
[22] https://postgis.net/
[23] https://www.geosystems.fr/services/cartes-dynamiques

'LPO Aquitaine' researchers and project leaders before being publicly deployed for the large interested community by the end of the VGI4BIO project i.e. fall of 2021.

V.5. Conclusion

In this chapter, we have introduced our new collaborative DW design methodology that allows involving volunteers in the definition of analysis needs if data is issued from crowdsourcing and that can be as well adopted in an enterprise context for a further elaborated and assisted collaboration. After investigating the DW design approaches available in literature in the previous chapter, we have decided to use a GDSS system both in collaborative elicitation of requirements and in solving the conflictual issues after merging the initial models. The methodology proposes two methods to handle the conflicts where the first is implemented and validated in this chapter using the VGI4BIO's LPO biodiversity case study, while the second will be tested in Chapter VII after introducing an implementation of a tool that allows its execution.

Chapter VI

EXPERIMENTS OF THE GDSS USER-EXPERIENCE

Summary

In this chapter we start by giving an overview on the GDSS that we used to conduct a set of experiments. The objective of these assessment sessions is to evaluate the user-experience satisfaction on behalf of unspecialized participants using the techniques that are universally used by group systems, mainly, for voting and for collective evaluation. Then, using the resulting conclusions to knowingly implement a new GDSS tool, introduced in the next chapter, we tend to solve the reported limitations in this new GDSS tool. Also, as we have proposed a methodology of DW design in the previous chapter that relies on a specific use of GDSS advantages in its step of solving conflictual designing details, we also evaluate the techniques used for that matter in order to adopt those that can improve the performance of the methodology's execution.

VI.1. Introduction

GDSS are platforms of collaborative technology that intend to increase decision makers' cohesiveness by the means of computational support (Adla et al. 2011). They offer a variety of group techniques where the interactive and friendly user interfaces are mandatorily critical to guarantee, among the participating users, a shared understanding and an accessibility to an at-hand problem's parameters. Although the GDSS activities are always run by a facilitator, a knowledgeable mediator of the group meetings who is highly qualified in decision making techniques, the context in which we use this technology has an additional specificity. This is due to the fact that we involve people with a different profile than the traditional users that are usually those who have authority, knowledge, privilege or influence. While in our case of eliciting DW

requirements or for the collaborative resolution of conflicts, the users are volunteers with no confirmed knowledge in most cases for the former, and with non-managerial qualifications and specialties as for the latter. In addition to that, even if the facilitator knows very well the necessary group techniques to use in these cases, he/she will need to know as well the DW design fundamentals at least. Having in mind these challenging requirements, we run a set of experiments that focus on the users' satisfaction and interactivity. More precisely, the objective of these tests is to assess the limitations related mainly to the user-experience and to the aspects of interaction with the user interface. These experiments have been done in the context of the RUC-APS project[24] i.e. stands for: Enhancing and implementing Knowledge based ICT solutions within high Risk and Uncertain Conditions for Agriculture Production Systems, that allowed us working with real volunteer users in an empirical case study. It is important to mention that even though the system that we use is not the main subject of the intended work, the gathered feedback is of a major importance for its improvement in regard with the related issues. This is why our reporting strategy has been centred on some generic aspects that we describe hereunder in this chapter.

VI.2. The GRUS System

GRUS (GRoUp Support) is a web-based group support system that can be used to organize collaborative meetings in both synchronous and asynchronous modes (Camilleri and Zaraté 2019). In its synchronous mode, all participants are connected to the system at the same time, while in the asynchronous, they can do so at different times. It is also possible to use GRUS in a mixed mode, synchronously and asynchronously at different steps of the process. With GRUS, users can also join sessions in distributed and non-distributed ways, i.e. in the same meeting room or distantly. The only requirements are the internet connection and the access granted by the meeting's facilitator if the session is a private one.

A user of GRUS can participate parallelly in several meetings. He/she is allowed to either be a facilitator that leads the execution of the meeting's process, or a simple user if he/she has been invited to take part in a team led by another user. A facilitator of a collaborative process can always participate in all its activities.

The system proposes several collaborative tools, the main ones are:

— Electronic brainstorming: allows participants to submit contributions i.e. ideas, to the group.

[24] www.ruc-aps.eu/ funded by the European Union under their funding scheme H2020-MSCA-RISE-2015.

— Clustering: the facilitator defines a set of clusters and puts items inside of them in order to categorize the defined ideas.
— Vote: A class of tools that refers to voting procedure.
— Multicriteria evaluation: Allows users to evaluate alternatives according to a set of criteria.
— Consensus: That displays statistics on the multicriteria evaluation outcomes in order to ease building a consensus about the decision to make.
— Miscellaneous: reporting i.e. automatic report generation, feedback i.e. questionnaires for the meeting quality evaluation, conclusion i.e. for conclusions of the meeting integration.

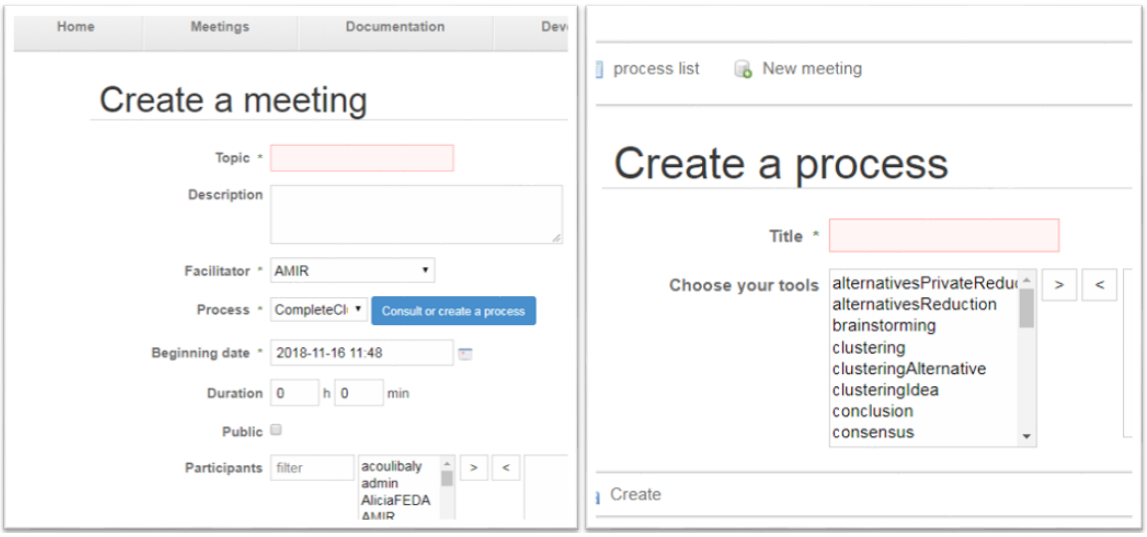


Figure VI-1 Meeting and process creation in GRUS

A GRUS session is composed of two general stages: the meeting creation and the participation. As shown in Figure VI-1, for the meeting creation, a user defines the topic of the meeting, assigns the role of facilitation to one of the users that he/she invites as well in this same step, choses the collaborative process to follow, and indicates the beginning date, time, and the whole duration of the meeting. The users can use one of the available predefined processes, or, if they do not find a process that corresponds to their needs, they can create a new one. The meeting is carried out in the second stage where the facilitator starts the meeting and leads the participants throughout the process.

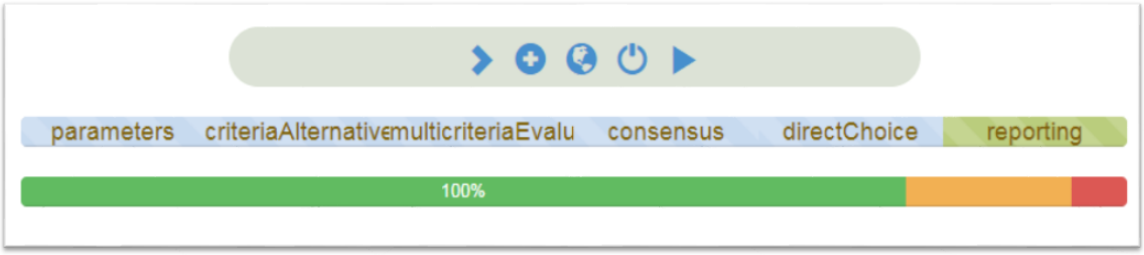


Figure VI-2 Facilitation toolbar

The facilitator has a toolbar to manage the meeting, as in Figure VI-2. Thanks to this toolbar, he/she can add or remove participants, go to the next collaborative tool, mark as ended a finished step, modify the group process, and set an end to the meeting. The other participants do not have this toolbar and they only follow the sequence of steps that the chosen process allows.

VI.3. Three-fold usability evaluation of GRUS

In a previous work, (Grigera et al. 2018) ran a three-fold usability evaluation on GRUS, as a representative software system to support collaborative decision making. The evaluation included user tests with volunteers, a heuristic evaluation i.e. manual inspection without users, and an automated diagnosis supported by a usability service named Kobold (Grigera et al. 2017). The main aim of the evaluation was to find out the usability issues, not only of GRUS in particular, but also to any other GDSS product in a more general way. The motivation behind this evaluation was to understand why, in spite of the existence of many different GDSS systems available for the agricultural field, the adoption rates are so marginally low. According to (Rose et al. 2016; Rossi et al. 2012), usability is one of the main factors for this lack of adoption. Being a particular setting for evaluating usability, especially given the collaborative component that is a key asset in such systems, the evaluation was designed with the aforementioned three different techniques. The motivation behind this approach was to maximize the coverage of the usability issues that are likely to be easier captured with different techniques. For instance, the automated diagnosis was expected to catch the issues overlooked by experts, while the heuristic inspection could give the experts the chance to more likely detect the issues that user tests could not cover since the tasks are designed for end-users to follow a somewhat fixed path.

We ran the tests with (Grigera et al. 2018) approach in the context of decision-making in the specific scenario of tomato production in the green belt of La Plata city in Argentina. For the user tests that involve real volunteers, we have designed tasks that are mainly related to the different alternatives that producers face at the time of planting or harvesting. In the different tests that we have run, some users were sharing the same physical space, and others were connected by video calls. The

automated tests were run simultaneously with the user tests, since the automated tool that we used requires capturing real users' interaction in order to produce a list of usability issues. More details on the preparation and use of Kobold can be found in (Grigera et al. 2018; Grigera et al. 2017). After the experiment, we detected a total of 15 issues, with some overlaps between the three different techniques of evaluation. The issues detected in the experiments were consistent with the literature conclusions of (Rose et al. 2016; Rossi et al. 2012) and most serious and repeated issues were connected to two general problems:

— Excess of information, or bloated GUIs: one representative example was the "overloaded report" for the decision-making process. This was actually an issue that has been captured by all three techniques. Other issues related to this general problem were "complex GUI in multicriteria features selection", and also "redundant controls and terminology".
— Lack of awareness in the collaborative process: many issues were related to the collaborative nature of the software combined with the linear process. Volunteers were frequently confused about what the next action is, or where the other participants were standing.

Many of the 15 reported issues were related to the system's learnability. However, after running into the problematical UIs for the first time, volunteers have shown an improvement in dealing with it.

VI.4. Protocol of experimentation

The main goal of conducting experiments using GRUS is to show how a web-based GDSS system can be supportive to a group involved in collective decision-making while being non-used to technology-based solutions to make critical decisions. For that, we have put in place a protocol of experimentation that is based on a two-stage approach as illustrated in Figure VI-3. The first stage consists of preparing the users for the meeting and the second is the execution of the test. Following this protocol allows us to re-implement it for each experimentation session to guarantee having the same circumstances in which the reported results would be generated. Thus, the resulting feedback and evaluations of the system i.e. comments, questionnaires, etc. would be comparable since the same surveying elements can be evaluated against each other to produce a better reporting strategy.

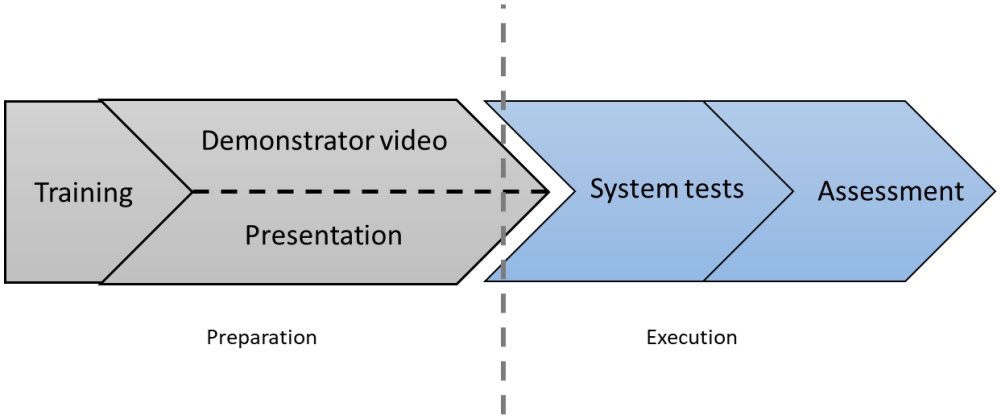


Figure VI-3 GRUS experimentation protocol

Another important aspect of having such a standardized protocol is to consolidate the evaluation of the user-experience and the satisfaction about the system's outcomes. This will also help to get more trustworthy conclusions about what and how to enhance in GDSS system's features and functionalities to better suit their use by non-IT skilled users.

VI.4.1. Preparation Stage

According to (Sutcliffe and Ryan 1998), one of the four techniques of the SCRAM, the method that they propose for requirements elicitation and validation, is providing a designed artefact that users can react to, like using prototypes to conduct concept demonstrator sessions. What is presented in a demonstrator session is called a demonstrator script and its nature can vary. (Røkke et al. 2011) mention that it could be a prototype-simulation or even a prospective design and that the session can be interactive, either with the participants using the system or simply by a presentation showing how it works. In both cases, what is important is triggering a debate to get feedback from the participants and to observe their reactions. (Sutcliffe and Ryan 1998) also state that the demonstrator has limited functionality and interactivity, and that it is intended to illustrate a typical user task. It runs "as a script" that illustrates "a scenario of typical user actions with effects mimicked by the designer". (Maguire 2010) mention video-prototyping as well as an alternative to demonstration and to show the concepts behind the system.

In our case, as we run the experiments in different sessions explained by different presenters when needed, and with participants living in different countries i.e. the UK, Argentina, Chile, France, and Spain, we have chosen to present the use of the tool in a training session that is a combination of a presentation of the GDSS use and advantages with a scenario recorded in a video in both English

and Spanish. This will allow us to avoid having a bias caused by the different circumstances of the training sessions like different operating systems, browsers, team configurations, presenters' precision, the amount of the provided details, possible mistakes while demonstrating each step, etc. We generated the demonstrator script with the goal of getting feedback from the users and to familiarize them with the interactivity offered by the system before they would have to actually use it to solve a different problem.

The demonstrator script shown in Figure VI-4 was designed to show the fundamental use of a GDSS in order to make a multicriteria decision in the field of Agriculture with participants who are experts in the application field of the problem. As GRUS allows facilitators to use predefined processes, we chose to show its usage with the same process that we will follow for the execution of the experiment stage. It consists of a multicriteria evaluation process which involves the six following steps:

1) Parametrizing the meeting
2) Defining the criteria and alternatives
3) Multicriteria evaluation
4) Consensus building
5) Direct choice
6) Reporting

We have illustrated these 6 steps differently, as shown in Figure VI-4, for a better understanding when presented to users.

Figure VI-4 The process of the presented example

The demonstrator script starts with an introduction where an explanation of the scenario is presented. It is set up in the context of five greenhouse leaders from a farm who need to agree on how much stems per plant they should use for the next crop. The farmers have this doubt because it is a known assumption in their community that increasing the number of stems to 3 or 4 increases the yield significantly without compromising the fruit quality (Candian et al. 2017). However, such practice was proven efficient only with certain soil and weather conditions, and with a different kind of mini-tomato seeds. Such differences are presented in the introductive presentation that we

make. We also refer to the five participants by avatars for further references in the screen-recorded sessions for the video.

After presenting the scenario, the video shows how the five users solved the problem using GRUS. The video was divided into sections that are separated with a progress graph to indicate which is the following step to demonstrate as shown in screenshot 1 of Figure VI-5 while screenshots 2 and 3 show respectively an explanation of the problem to solve and the participating practitioners.

Figure VI-5 Screenshots from the GRUS use video

The first section involved only the participation of one user who acted as a facilitator as shown in screenshot 4 of Figure VI-5. Then for the rest, the actions of the different users are presented sequentially as for example the participation of one of the 5 users shown in screenshot 5 of Figure VI-5. At the end, the facilitator explains the calculation results and how to interpret them for decision-making as shown in screenshot 6 of Figure VI-5. For the session recordings we have defined recording guidelines, so all the participants recorded the video under the same settings:

recording in mp4 format with a high resolution (720p onwards), full-screen mode, 30 fps, disabling the audio input and enabling the recording of the pointer. The software used to produce the video was Kdenlive[25], an open-source multi-track video editor. The resultant video is publicly available on Youtube[26].

VI.4.2. Execution stage

For availability and time management reasons, the experiments that we did are done synchronously and with a collocated attendance. This necessitates a setup of the following elements which must be available at the meeting room:

- Two or more decision-makers with a computer for each of them.
- An Internet connection, which is mandatory since the system is a web-based application that is available online.
- A facilitator to manage the meeting, preferably the same person in all experiments or at least facilitators that have equal levels of proficiency in using the system's functionalities and in explaining its tools and techniques.
- A shared screen or a video projector to share the facilitator's screen when demonstrations are needed.

Before starting the experiment, every participant must have a user account on GRUS, if not, a new one needs to be created. After having all the participants logged in, the facilitator will create the new meeting with the confirmation of everyone on the parameters of the process in relation with their availability and the problem to solve. Then he/she invites all the decision makers to join the meeting that is accessible from the list of meetings available in the meetings page.

After joining the meeting, the participants must follow the facilitator instructions to properly complete all the following steps, which differs according to the chosen or newly defined process. When the decision to be made is dependent on multiple influencing factors that must be considered to have a precise evaluation of the different available possibilities, the multicriteria process is the one to be followed to accomplish such a meeting. To do so, GRUS proposes a predefined MCDM process that consists of:

[25] kdenlive.org/
[26] youtu.be/jkn7XhNK8hU

- Parametrization of the meeting, i.e. title, description, stakeholders' weights, evaluation scale.
- Brainstorming engaging all participants to define collectively the set of criteria and alternatives.
- Individual preferences matrix of Criteria/alternatives to be evaluated against one another and that needs to be done by everyone separately using the defined evaluation scale.
- Consensus building step is the one during which the facilitator shows and explains the resulting calculations and leads the interpretation process to build up a final common decision.
- Decision to be made after having the consensus about what is, based on the supportive results given by the system, the most likely to be held as a better alternative, what might be suitable or feasible in the impossibility of applying the first chosen one or set of elements and what are the eliminated alternatives that had a non-encouraging score during the multicriteria individual preferences step.
- At the end of the meeting, an automatically generated report would be downloadable containing all the parameters and results that have been used and produced during the test.

Finally, a questionnaire should be given to participants to get it filled after the meeting. It must collect feedback about satisfaction levels and propositions in relation to the efficiency of the training on the meeting execution and the system's usefulness. Mainly, the questionnaire can be altered with what the facilitator thinks would improve the quality of the collected answers. However, it must still evaluate and impel criticizing the user-experience. The questionnaire needs to conclude with users' recommendations on four global aspects:

- The level of complication that characterizes the system and its use.
- The friendliness of the user interface and what to improve in its interactivity.
- The usefulness of the training for a pre-built understanding.
- The role of the facilitator and its impact on the final decision.

An example questionnaire is available in appendix C.

VI.4.3. Experiments

As illustrated in Table VI-1, we have run six experiments using the defined protocol. In order to illustrate the execution in different situations and with various profiles amongst the participating teams, we detail, in this section, only two of these experiments while more of which have been cited with further specifics in our work (Sakka et al. 2019). As the general purpose of these sessions is

evaluating the user-experience aspects, we here report only its related users' feedback after describing the two experiments' parameters and outcomes.

Table VI-1 The experiments run with GRUS in RUC-APS project

Experiment	Date	Institution	Number of participants	Tested Process	Problem to solve
1	20/06/2018	FEDACOVA[27]/ Spain	7	Multicriteria	Improving egg production
2	28/06/2018	FEDACOVA/ Spain	5	Multicriteria	Use of GMOs [28]
3	12/07/2018	FEDACOVA/ Spain	4	Direct vote	Meat packaging for international shipments
4	23/08/2018	La Plata university[29]/ Argentina	5	Multicriteria	Allocation of crops for university's nursery
5	13/09/2018	INIA La Cruz[30]/ Chile	9	Multicriteria	Research topics to be prioritized by the institution
6	17/06/2019	IFA[31]/ The UK	2	Multicriteria	natural or industrial fertilizers?

As shown in Table VI-1, we have followed the experimentation protocol that consists of using the Multicriteria process except for the third experiment when we have used the direct vote process since we had the opportunity of doing more than one session with the same team.

VI.4.3.1. A first experiment of GRUS

Our first experiment was held in the Faculty of Agrarian and Forestry Sciences in the national university of La Plata in Argentina with the participation of 5 decision makers: two researchers in agronomy, one researcher in computer science and two master's students. The used process was defined by the facilitator after consulting the participants about their matter of uncertainty as shown in the 'step a' of the Figure VI-6. The meeting was meant to support a decision about allocating the amount of each under-study crop inside the university's nursery. Because of the advanced knowledge and valuable experience in agronomy of the two specialized researchers, they took higher weights in the parametrization step than the other members. In the brainstorming as shown

[27] Business Federation of Agri-food of the Valencian Community (www.fedacova.org).
[28] GMOs stands for genetically modified organisms.
[29] LIFA, Research Center of the Faculty of Informatics - National University of La Plata - Scientific Research Commission of the Buenos Aires Province.
[30] INIA, The Institute of Agricultural Research of Chile - The state agency for agricultural research and development (www.inia.cl).
[31] IFA, Innovation For Agriculture - A consortium of English agricultural societies disseminating related latest developments in new science & technology(www.innovationforagriculture.org.uk).

in the ‘step b’ of Figure VI-6, participants easily defined the set of criteria and alternatives since they understood well the operability of the system during the training session a few days before. Then in the ‘step c,’ the MC-evaluation gave the same ranking with the two available calculation methods i.e. Choquet’s integral and weighted sum (Grabisch et al. 2000).

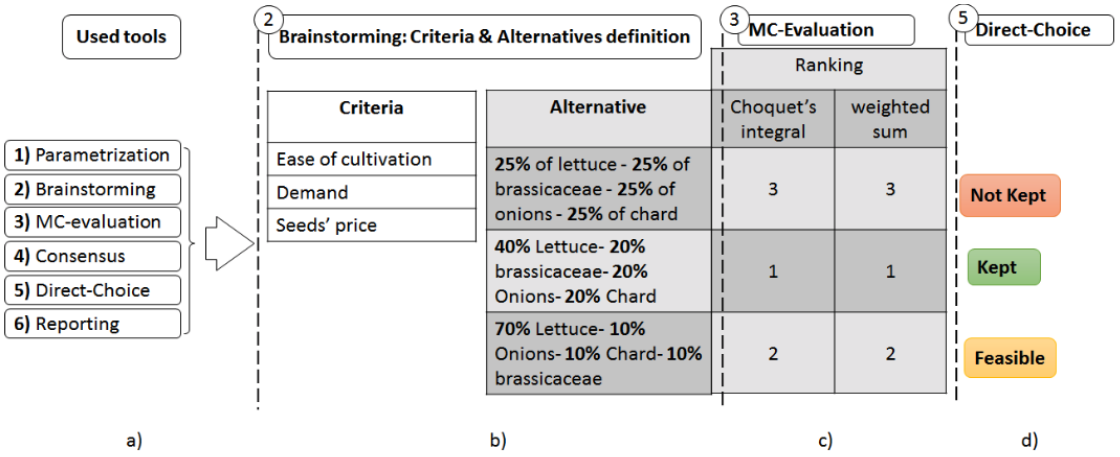


Figure VI-6 Process of GRUS' first experiment

Finally, and after discussion, the decision makers have reached in a consensual manner the final choice that consisted of keeping the second alternative, to consider as feasible the third and to not keep the first one as illustrated in the ‘step d’ of Figure VI-6.

VI.4.3.2. A second experiment of GRUS

With a group of researchers and administrative staff of the main agricultural research institution in Chile, i.e. INIA for “Instituto de Investigaciones Agropecuarias”, and after a training session where we introduced the use of the GDSS with a presentation followed by the video-demonstrator, the participants have collectively decided about the most important topics that they need to have them prioritized by the institution for the next year working plan.

The meeting was reached by 9 decision makers with different administrative positions and scientific backgrounds that defined together the topic and the tools to use. After discussing the level of knowledge and the influence of each one on the final decision, different weights were given according to everyone’s proficiency level during the parametrization step. Afterwards, as shown in Figure VI-7, the participants have defined a set of criteria i.e. Social impact, Regional economic contribution, Regional centre skills and Climate change and alternatives i.e. Crop physiology,

Irrigation research, Pollination research, Computer science application in agriculture, Agricultural ecology, Tree fruit research and Horticultural research.

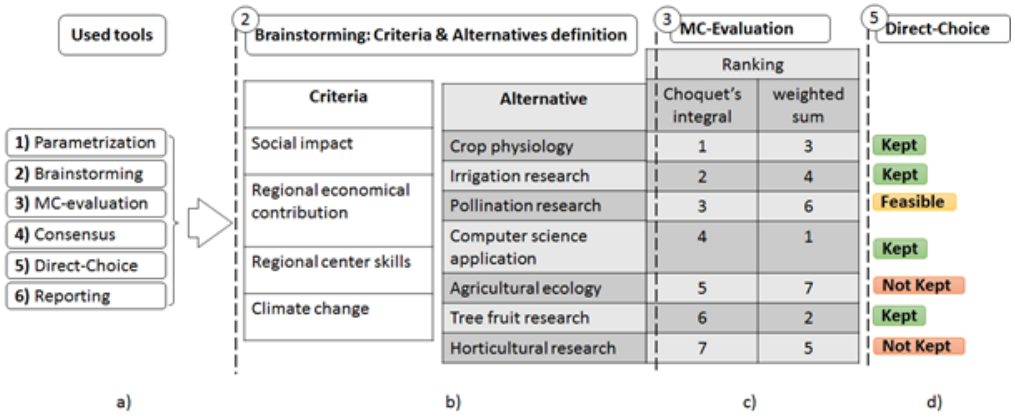


Figure VI-7 Process of GRUS' second experiment

Next, they gave separately their personal preferences that have been collected and ranked based on Choquet's Integral and weighted sum, and finally took the decision to consider both rankings by keeping the first two alternatives of each calculation method, to consider the 3rd elements as feasible and to not keep the rest as shown in the 'step d' of Figure VI-7.

VI.4.3.3. Users' feedback

After conducting the experiments, all the users have filled after each session a questionnaire that focuses on the aspects that we detailed previously in this chapter. Generally, the users were satisfied with the features of the system and they appreciated the assets that it offers. They have also considered that the use of a GDSS system in making their decisions, especially in uncertain circumstances, would most likely simplify the problem perception and help in building a consensual or a majoritarian view. In addition to that, they consensually expressed their satisfaction about the training session and said that it gave them an essential introduction to the system that without which, it could have been more complicated for them to define correctly the problem and to give consistently their preferences. Likewise, the video was considered very helpful and friendly for the introduction of the GDSS use.

Some improvements from the user-experience point of view were proposed as well, such as the revision of the matrix of preferences' presentation in the process of multicriteria evaluation that over 70% of the users had difficulties with and thought that it might be hard to understand by non-IT experienced stakeholders. This has been the case with 4 of the participants of the 3 experiments that have been run with FEDACOVA, as shown Table VI-1 that have participated in all the 3 sessions and that have found the voting process easier and faster to use than the multicriteria one. The questionnaires' feedback has further shown us that the system is too complex to be used by real deciders and that some parts of its processes must be hidden depending on the current step in order to reduce the visual load on the users while they are supposed to keep all their focus on the matter of the group activity. Besides, many critiques have been manifested in relation with the interactivity of the interfaces. For instance, the fact that the navigation between the different steps is not clear and that it depends too much on the facilitator's instructions not on a fluid passage that assists the participants dynamically throughout the meeting. Another example of these raised interactivity issues is that users are doing their definitions in an almost separated way, which needs to be improved to help build the positive interinfluence in cases of brainstorming or ideas revision activities. This has also been raised when the users were supposed to brainstorm ideas while one of them has mistakenly clicked the submit button, whereas he/she should have waited for the others to finish so they move on together. Such coordination issues need to be addressed by certain restrictions on the used tools' functionalities as well as by more explicit oral interactions between decision makers when using electronic communication techniques, as pointed out more by than 50% of the participants. In the same way, and since the GRUS system is still under development, many programming bugs have been detected and therefore influenced the interactions and raised the complexity of the system's use. The presentation of the results at the end was as well clearer and more understandable to almost all the users in the web presentation more than in the downloaded report.

Although the level of satisfaction about the usefulness and the added value of the system was high, expressed by nearly 90% of the participants, over 20% of them thought that whether only facilitation or one training session would suffice for a correct use of the system, and have emphasized on the probable complexity that they might face in case where they try to use such GDSS without having an expert to play the facilitator role.

VI.5. Conclusion

In this chapter we have presented a protocol of experimentation to evaluate the use of GDSS systems in order to spot the aspects to tackle in our new system that we introduce in the following chapter. Carrying out these experiments in different countries with different users has shown us that

learning to use these collaborative systems' fundamentals is mandatory in order to obtain satisfactory feedback from the end users, regardless of the users' profiles or the working environments. The carried experiments have highlighted weaknesses, problems and areas for improvement, even though more investigation of our results might reveal more factors that will help to understand how the training could influence the eventual decisions, which is out of our actual work's scope.

Chapter VII

GROUDA: A NEW GDSS SYSTEM

Summary

In this chapter, we develop some existing Thinklets with interactivity related improvements as well as a new DW design Thinklet. We start therefore by defining the Thinklets and the classification that we use for them in the remaining. Next, we detail the general architecture of our new GDSS where we illustrate the design followed to offer a dynamic and user-friendly logic of group meetings creation and execution. To allow novice facilitators and DW designers that have no experience with the crucial facilitation task, we also build a question-based recommender engine that assists the collaborative processes' creation. Then, we describe the implemented Thinklets with a focus on the new features of interactivity and user-experience improvement. Eventually, we detail a set of tests that we carried out with our new GDSS in order to validate its main functionalities and to assess its intended improvements for both generic group activities and DW design, either in the context of volunteers' engagement or with classical collaborative design as detailed in V.2.2.2.2.

VII.1. Introduction

After conducting the set of experimentations briefed in the previous chapter, we developed a new GDSS system inspired by its conclusions. In this chapter we introduce our new system that we rely in its general architecture on the concept of Thinklets proposed by (Briggs and Vreede 2009) to solve the problem of expert facilitator's availability by structuring the collaborative processes' construction and therefore their reproducibility. We called it GROUDA standing for GROUp Decision & DAta-warehouse, since we have also created one Thinklet for the specific need of using the GDSS to enhance the collaborative design of DW systems. For high-value recurring tasks, the

collaborative engineering proposes to transfer facilitation skills to participants (Briggs et al. 2003). In our case, to allow participants that are novice to facilitation and to avoid maintaining professional facilitators, we propose Thinklets-based collaborative processes whose creation is assisted by a recommendation engine. As any other software application, the user experience is an important aspect for its adoption and success. Improving the user experience could constitute an interesting leverage for promoting the use of GDSS by unusual communities, the reason for which we use the outcomes of the experimentations that we have carried out with volunteer users and with a GDSS that had some user-friendliness limitations. Those of which that we consider generic i.e. mainly user-experience related such as the friendliness of the interfaces and the ease of interactivity, are the centric improvements that we are focusing on their amelioration in GROUDA that we detail its architecture and implementation in this chapter.

VII.2. The Thinklets concept:

The Thinklets concept has been first defined in 2001 by (Briggs et al. 2001) as a named package of group activity that builds a predictable and repeatable pattern of collaboration among people working towards a goal. Every Thinklet is composed of three group activity stimuli that are described independently of the technological support of its implemented solution. (i) The tool which is the hardware and software technology used so the Thinklet definition allows reproducibility, (ii) the configuration to give a precise parametrization amongst the various possible combinations of the used tool settings, and (iii) the script that leads the sequential progress of the activity execution. As detailed in (Briggs et al. 2003), a collaborative process is a set of Thinklet-based group activities that each of which is defined by the tool to use, its configuration, and the script to follow for a precise and correct execution.

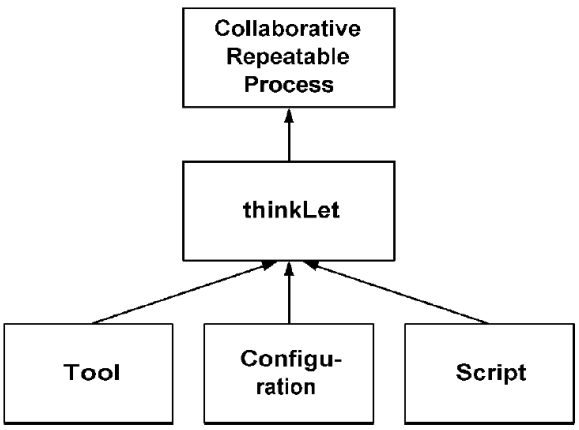


Figure VII-1 Thinklets as collaborative building blocks from (Briggs et al. 2003)

This concept has been followed to create and test many Thinklets that (Kolfschoten et al. 2004) have categorized into five types of collaboration patterns:

1) Divergence: Moving from the state of having few concepts to the state of having more of which.
2) Convergence: Moving from the state of having many concepts to having few of which to put more emphasis on those that are worthy of future attention and improvement of their understanding by this reduction.
3) Organization: increase and build an understanding of the relationships that exist between the concepts.
4) Evaluation: increase and build an understanding of the values of concepts defined to move towards a goal.
5) Building consensus: Move towards more agreement about the intended courses of actions among the participating stakeholders.

Although, (Kolfschoten et al. 2004) have also presented other classifications i.e. by outcomes and by group process phases, they have mentioned that the classification by patterns is still the most effective one. This classification organizes the flow of group meetings and eases the differentiation between the objectives of each activity in a way that allows a better understanding and therefore operability of each separately conceived pattern. We use this classification but with a slight change that consists of regrouping the 'Convergence' and 'Organization' classes into one class that we call 'Clustering'. This merge encompasses all the categorization, convergence, selection and reduction of the defined ideas, filtering and limiting the number of definitions etc. Hence, and for a more friendly terminology, in our implementation of the Thinklet-based GDSS that we introduce in this chapter, we refer to Thinklets' categories by 'Brainstorming', instead of 'Divergence', 'Clustering' as we have explained above, 'Evaluation' and 'Consensus'. Also, we use the term 'technique' instead of 'Thinklet' to avoid confusing novice facilitators with the used appellations.

VII.3. General architecture

GROUDA (GROUp Decision & DAta-warehouse) is a Group Decision Support System (GDSS) which allows groups of decision-makers to organize a variety of Thinklet-based group activities. We describe the Thinklets that we have implemented further on in this chapter. GROUDA is a web application developed in Python with the Django 2.2.2 Framework and a PostgreSQL database, that allows the availability of its services on the Internet. This decentralization of the system allows

groups of users to remotely join the collective meetings handled by the system. It is also possible to use some of the offered techniques asynchronously, allowing flexibility in terms of users' availability. The general principle is to dynamically create a process consisting of one or many steps which will be followed during the meetings by a group of participants led by a facilitator. The system's architecture is designed to allow a high level of flexibility and personalization of its GDSS functionalities. This, because GROUDA is built with the principle of allowing dynamic creation of meetings' processes using the process creator.

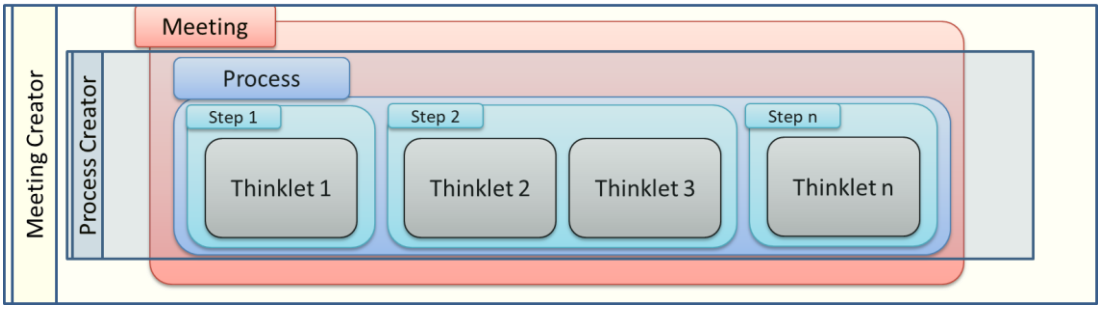


Figure VII-2 Design of process and meeting creators

In Figure VII-2, we illustrate the concepts of meetings and processes creators. A process is a set of steps and a step contains at least one Thinklet. The meetings are created by the facilitator in the meeting creator where he/she chooses necessarily one process, either of the already created processes, or by creating a new adapted one if he/she does not find an adequate previously defined process. He/she also defines the problem around which the group activity will be held and invites the participants that will join him during the execution of the meeting later on. Both process and meeting creators allow flexible management, for each user, of his/her definitions and according to his/her permissions that are granted by the system admin. Mainly, the facilitators have accounts with more permissions and have access to the Django admin site that allows all add, delete and modify actions on all the system's models.

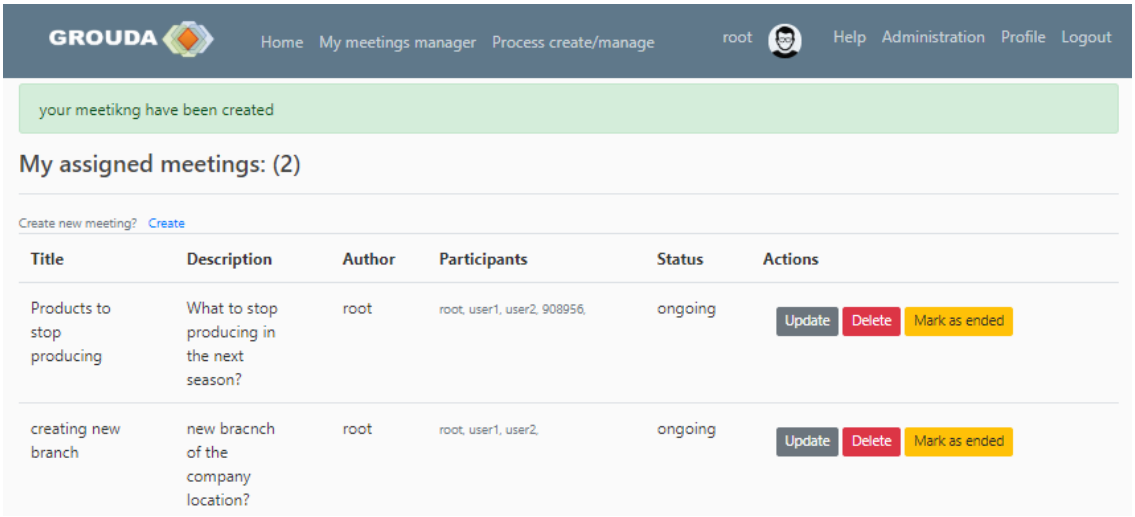


Figure VII-3 Meetings manager

The meetings manager shows to each facilitator the list of the meetings that he/she has created with their details. For each meeting he/she can perform update, delete or mark as ended actions as shown in Figure VII-3. He/she can also create a new meeting which will orient him to the meeting creator as shown in Figure VII-4.

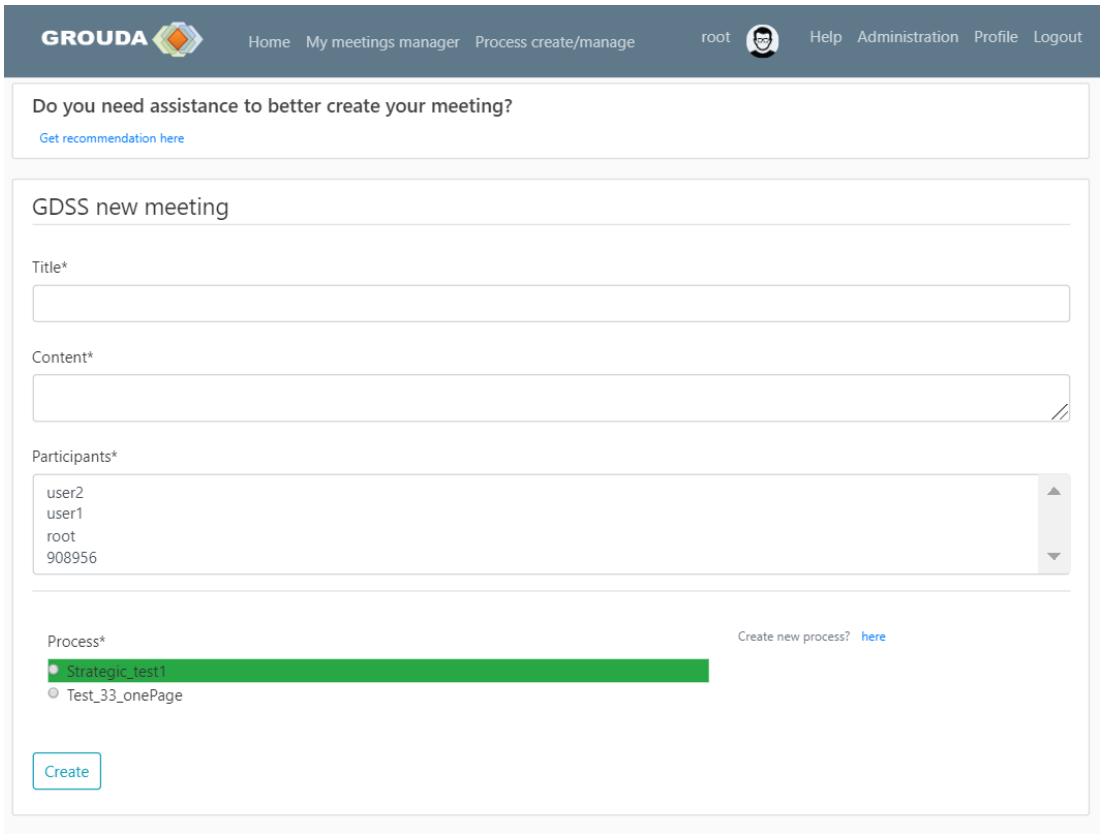


Figure VII-4 Meeting creator

If the facilitators cannot find the process that they are looking for in the processes list, they can create a new one by clicking on the ‘create a new process’ link or directly from the ‘Process create/manage’ tab that leads directly to the process creator as shown in Figure VII-5. In the process creator, the facilitators can add as many steps as they want where in each step, they must select a Thinklet type and then pick the Thinklet that they are looking for. In the bottom of the process creation form they have a list of the available processes with a detailed display and where the only permitted action is delete. We did not allow other processes management actions to avoid misleading other users of the system that might choose again a process only for its name while its content has been altered. This will cause them to lose more time discovering that it is not the correct one during the meeting execution, than if they create a new process when a previously used one has been deleted by its owner. Nonetheless, modifications are still allowed to the users that have access to the admin site.

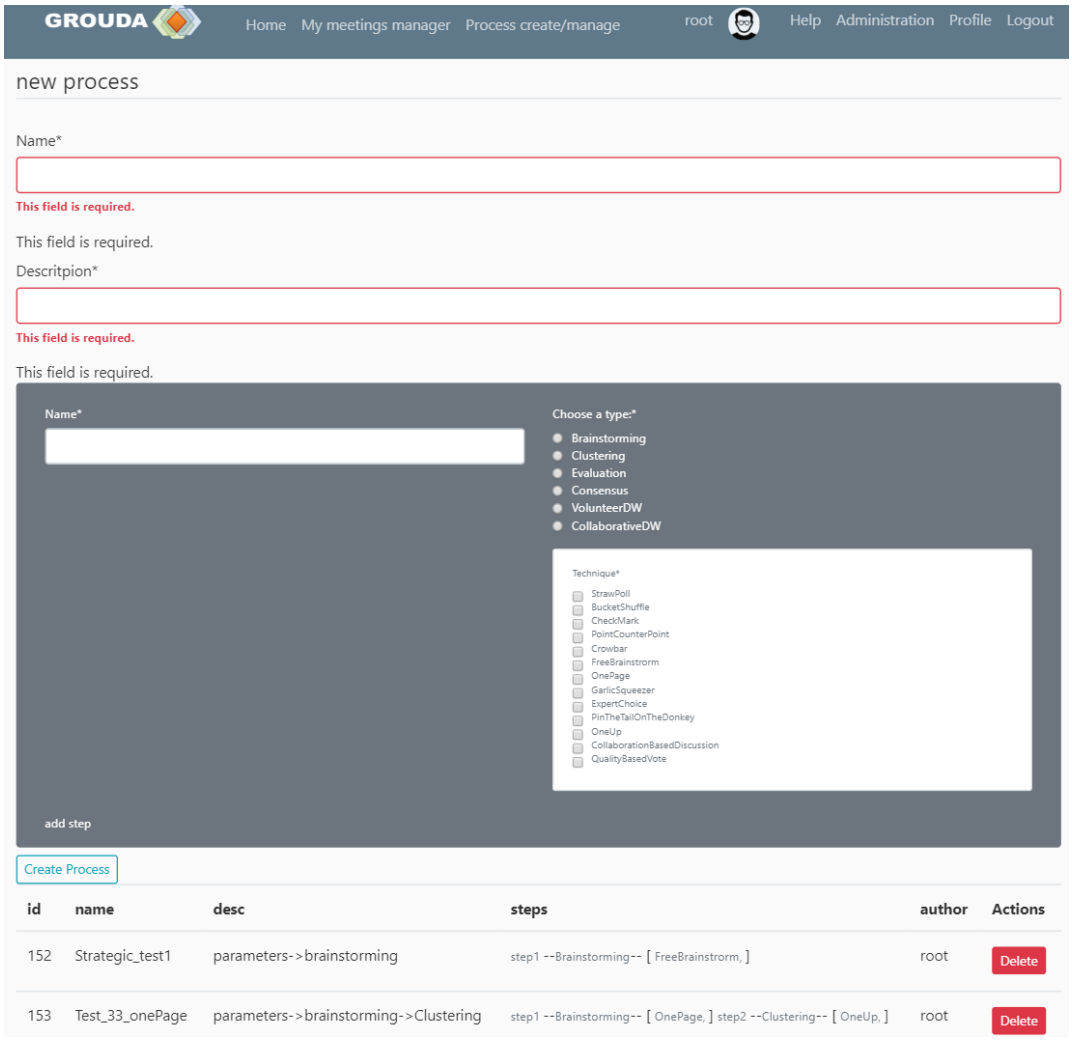


Figure VII-5 Process creator

As the knowledge of the Thinklets concept is only supposed to be a qualification of GDSS expert facilitators, we introduced a question-based recommender engine to GROUDA in order to help novice facilitators in their processes' creation. As shown in Figure VII-6, the recommender engine layer assists the facilitator in the creation of a new meeting's process. It is accessible from the meeting creator at the top of the screen in Figure VII-4, when the user finds it hard or unclear to define his/her process steps and Thinklets. The functioning of the two illustrated levels of recommendation is detailed in the next section.

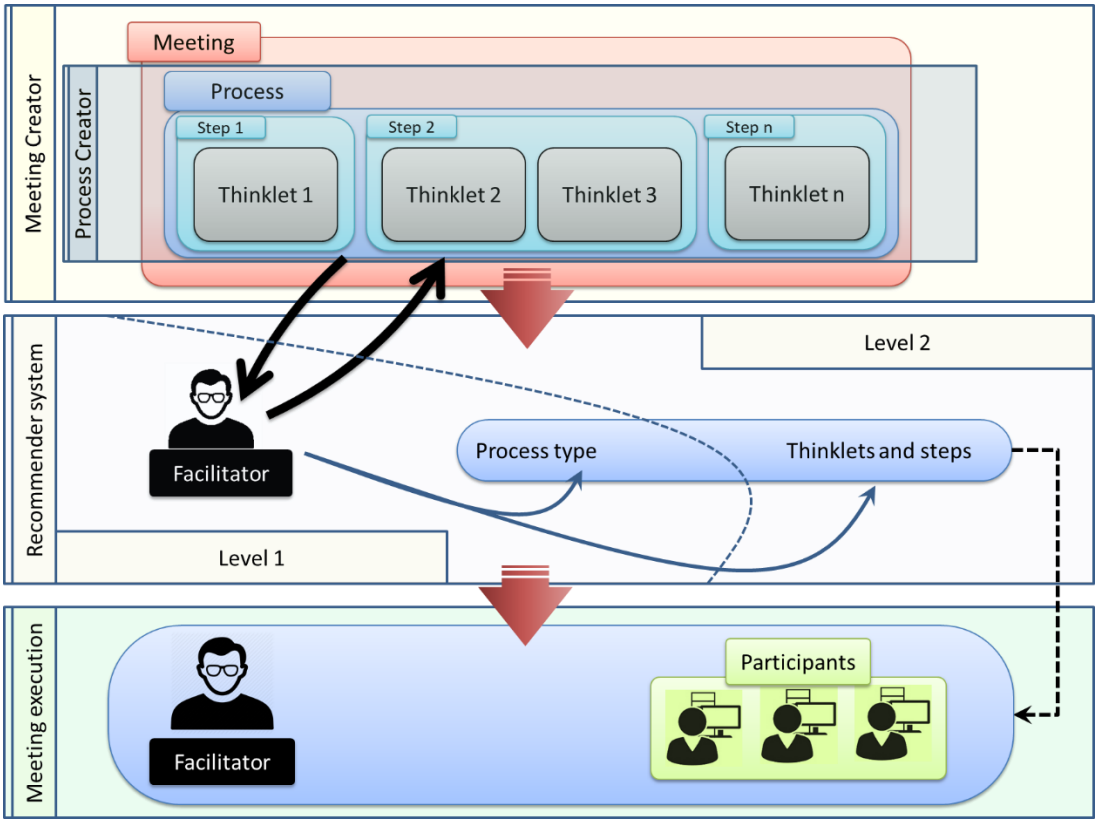


Figure VII-6 General architecture of GROUDA

For the meetings' execution, which is shown in Figure VII-6 as the third and final layer of use of GROUDA, we have put all the meetings that users are invited to join in the home page as it is the main service offered by the system. This is shown in Figure VII-7 where the list of the ongoing meetings, that the logged user is invited to join, is shown with their details and with two possible actions. The first is the 'join' action that redirects the facilitator to the meeting's parameterization step then to the first step of the process, and the invited participants directly to the first step of the

process. And the second is the 'decline' action that allows removing the logged user's name from the participants list, which removes the meeting from his/her ongoing meetings list.

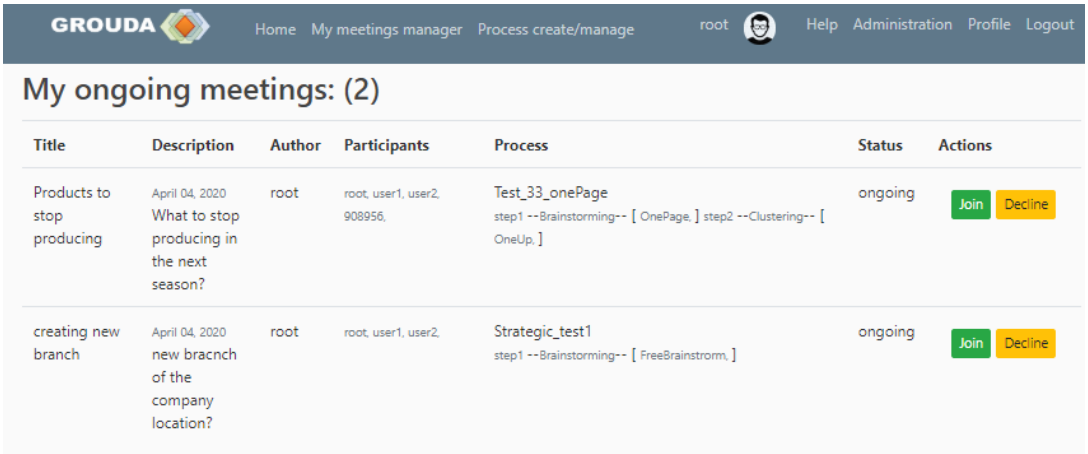


Figure VII-7 GROUDA home page: the ongoing meetings list

We don't describe in this chapter all the functionalities of GROUDA that we consider generic, such as the login, sign up, user profile and help. Also, the use of the Django framework has offered us the admin site where all the basic actions on the defined models are easily executable due to the friendliness of the user interface. This permits the admins of the system to perform all actions on all the defined data without needing to access the DBMS i.e. the database management system, PostgreSQL in our implementation.

VII.4. Recommender engine

The objective of the recommender engine that GROUDA proposes, is to offer guidance to novice facilitators to better build, when needed, a process for each meeting that they define. This is possible since it suggests the best suited composition of the process to use in two levels of recommendation as shown in Figure VII-8. In the first level, a set of questions are asked to the facilitator in order to define the process type that suits his/her intended general objectives. For this, we have defined 4 types of processes. The first 3 types: strategic, operational and tactical have been defined according to the organizational management filed as detailed thoroughly by (Far et al. 1998). In general terms, strategic planification is about the global long-term objectives, tactical is about more structured and

therefore more critical mid-term planification, and operational is the day-to-day tasks through which the most detailed and structured executive operations are tackled. Decision-making works have widely adopted this classification such as by (Fountas et al. 2006; Harrington and Ottenbacher 2009). We describe below, from our perspective, these 3 types of processes as well as a fourth one that we have chosen to put separately from the others for ease-of-use reasons even though it fits perfectly in the operational category i.e. DW design:

1) Strategic: when the group needs to conduct a brainstorming-based activity to gather propositions, to maximize choices or to allow people to diverge ideas, in order to enrich the strategic view of the group.
2) Operational: when a team needs to take a decision that's based on whether a multicriteria or a direct choice evaluation.
3) Tactical: when it's more an argumentation activity, which we don't consider in this version of GROUDA since it is more related to negotiation than it is to decision making. Although the PointCounterPoint Thinklet, as defined by (Briggs and Vreede 2009), can be adapted for a negotiation activity since its role is to bring a meeting's members to a common ground after a bad confliction by allowing them to debate and argue against each other.
4) DW design: when the decision consists of evaluating a DW schema, we suggest a new Thinklet that can be setup according to the DW basics' understanding of the group of participating committers and to the application domain expertise of the facilitator. We called this Thinklet CollaborativeDW, that we assist its use, when creating a process for the meeting, by the recommendation of 4 different employments. We detail this in the next section along with its different use cases.

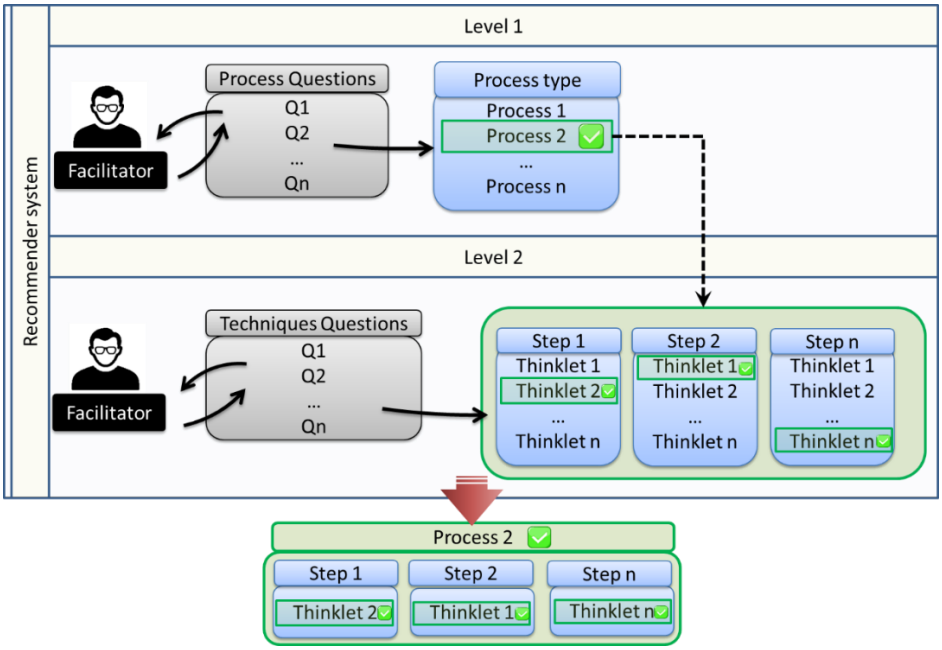


Figure VII-8 Recommender system of GROUDA

After suggesting the appropriate process type, there are two possible assistive actions by which the system can orient the users. Either by getting them back directly to the meeting creation form where they will find the recommended process highlighted and put on top of the processes' list when the process is already created and does not require a construction of its steps, or by orienting them to the second level of recommendation as shown in Figure VII-8. An example of the second level of recommendation is shown in the screenshot of Figure VII-9.

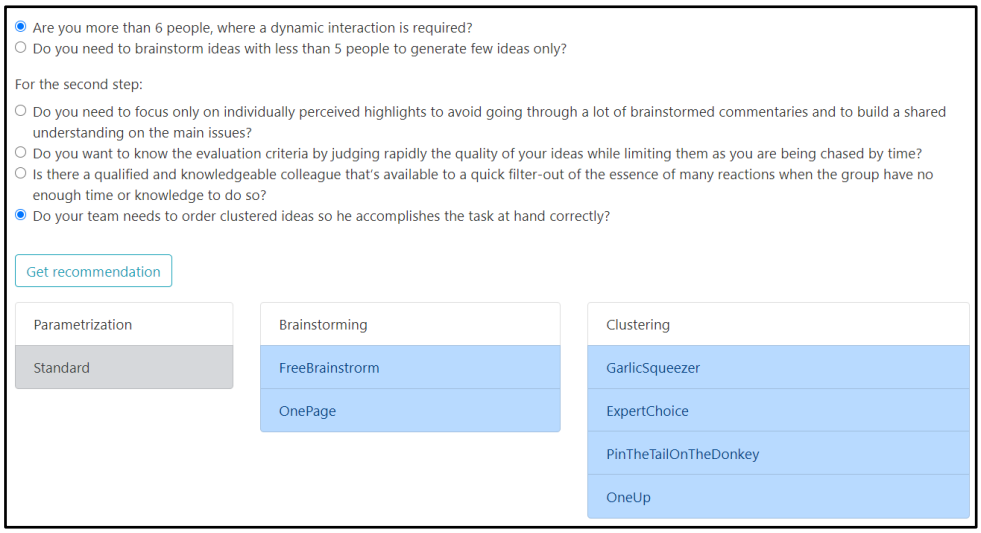


Figure VII-9 Example of the second level of recommendation questions

For the second level of recommendation, based on the meeting's circumstances, a set of questions that are related to the technical needs for each step are asked to the facilitator. Based on his/her answers, the Thinklets are recommended to be used in each process' step. After that, the facilitator will be redirected to the process creator where he/she will find the process creation form prefilled by the Thinklets that the system recommends. In this step, it is not mandatory to accept all the recommended parameters and the facilitator will always have the choice of personalizing the process construction manually for an ad-hoc use of the tool.

VII.5. Implemented Thinklets

Although the definitions of existing Thinklets only specify the generic and essential functionalities of the appropriate tool accompanied by its clearly detailed configuration for reproducible usage, the tool itself might not be always as suitable as its use would suffice to handle the specificity of the sought activity (Kipp 2016; Schwabe et al. 2016; Twinomurinzi et al. 2008). Thus, we have implemented some of the existing Thinklets with further adopted functionalities that we believe will improve the results of their use. In this section we explain briefly the Thinklets that are well known in the literature with more emphasis on the new aspects. Hence, the focus of our implementation of the existing Thinklets is mainly put on improving the user experience and interactivity aspects that we have identified in the previous chapter and that we believe will improve the quality of the collaborative meetings either for the second scenario of 'requirements elicitation' or for the 'collaborative resolution of requirements conflicts' of our methodology proposed in Chapter V.

VII.5.1.1. FreeBrainstorm Thinklet

As it has been introduced by (Briggs and Vreede 2009), FreeBrainstorm is a Thinklet of brainstorming that is useful to diverge quickly when the patterns of thinking are comfortable, to push participants farther out of their personal boundaries and to motivate a generation of new ideas. It helps to eliminate information overload during brainstorming in teams of 6 or more people by allowing an anonymous criticism of everyone's ideas. It allows therewith, in dynamic interaction, the members with narrow views to see the big picture and to quickly create a shared vision with the team. What makes the new functionalities that we introduce in this implementation of FreeBrainstorm is the interactive interface:

— We give the facilitator the right to authorize whether to allow or not the users to modify the defined ideas of others directly on the electronic page.
— The facilitator is the only one that sees the list of all the elements since the electronic page is supposed to show only three elements on each turn to avoid confusing the users with an overwhelming number of ideas displayed on the same screen as shown in step 1 of Figure VII-10.
— The 3D flipping card animation of the electronic page helps understanding the dynamism of the random ideas' selection in each turn, step 2 of Figure VII-10.
— When the electronic page flips to the 3 randomly selected ideas' side, we have created a three-state button that when clicked in its initial state, it allows the editing of the chosen idea and deactivates all the other ideas to avoid editing more than one on each flip as zoomed at in step 3 of Figure VII-10. Then, after it has been clicked for the first time, the button becomes a two-state button that allows mentioning whether the current interaction with the idea at hand is to be seen as a positive ,if the 'Agree' state is selected, or as negative if the 'Disagree' is selected.

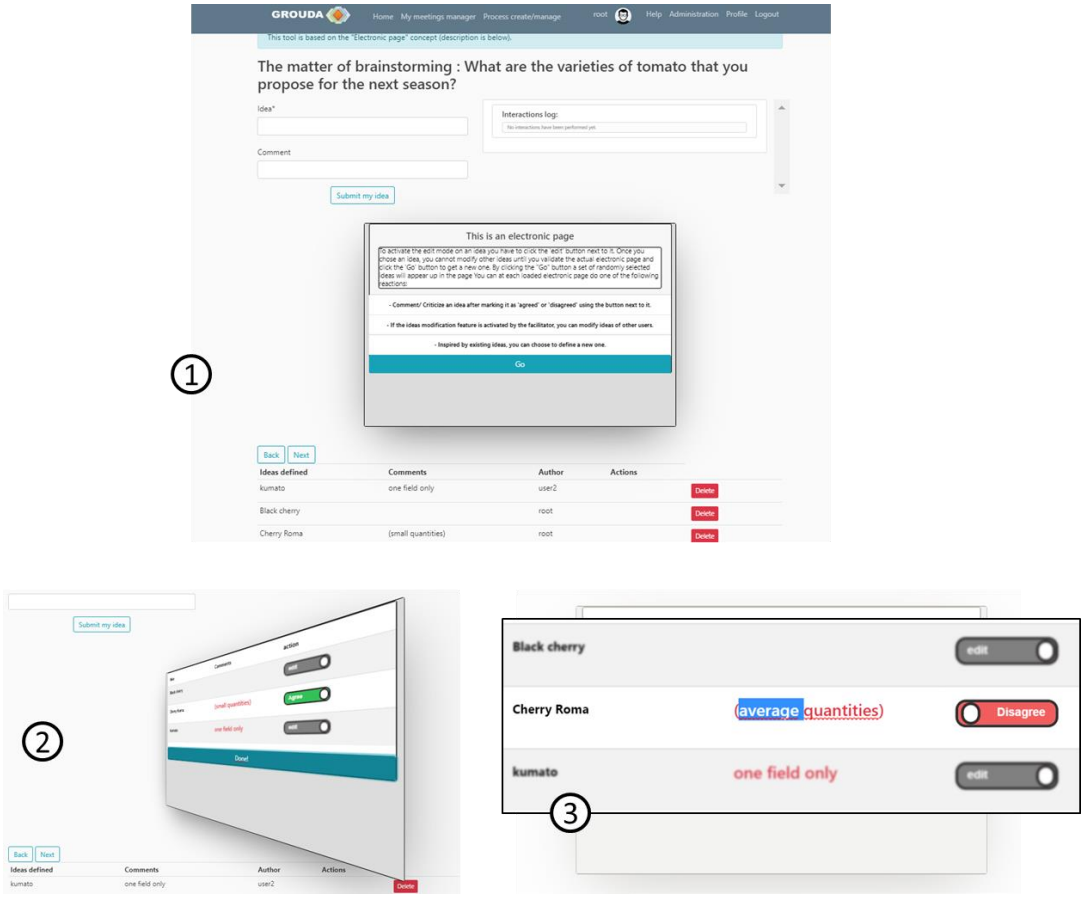


Figure VII-10 New FreeBrainstorm implementation example

In addition to that, we have implemented a tool that we use with FreeBrainstorm and other interaction based Thinklets for a live log of interactions between the practitioners. As shown in Figure VII-11, this is the interaction log that displays, in the case of the FreeBrainstorm, the editing of ideas and the comment i.e. includes critiques, propositions, etc. in different colours according to the chosen state when using the electronic page.

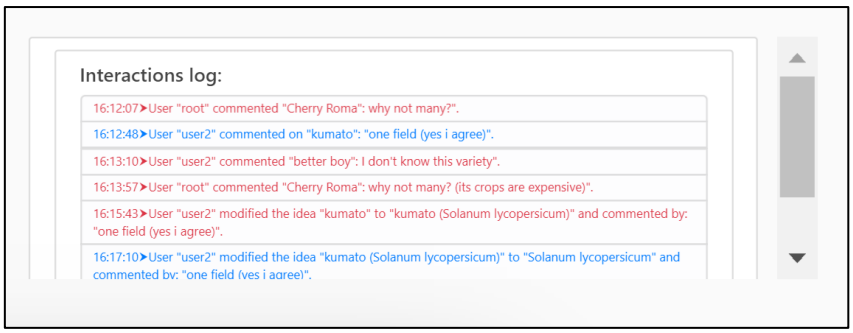


Figure VII-11 The interactions log of GROUDA

The interactions log is used for two reasons, first, to give the users a sense of a live interaction that helps them to have a visibility of the raised points and to keep them committed to the course of the meeting. Second, it will be saved with the final report of the meeting to keep a track of the evolution and elaboration of the final results for further analyses or improvements, even if the ideas that the users have interacted with initially have been narrowed down to a reduced set in the following convergence steps of the collaborative process.

VII.5.2.OnePage Thinklet

The OnePage Thinklet as defined by (Briggs and Vreede 2009), is to be used to generate a few comments on one topic in the same page at the same time. It is the one to choose when there are less than 6 practitioners or when 6 or more will brainstorm for less than 10 minutes. In addition, where it is unlikely that there will be too many interactions with the topic at hand and to support the back-channel communication between the team members. It is not the appropriate Thinklet to choose when too many ideas are expected or wanted to be generated, in this case the FreeBrainstorm, for example, would be helpful instead. It also does not fit for more than one topic at a time or for many people working on a topic until reaching its limits. As our concentration is being on the interactive aspect's improvement, we have implemented the OnePage Thinklet with some additional functionalities.

The same as with the FreeBrainstorm Thinklet, we also used the interactions log with the OnePage Thinklet but without colour variations since we do not have the three-state button that mentions, once activated by a first click, whether the comment is positive or negative. Without overwhelming the display, we have put on the same screen all the necessary actions for the work to be done by all the users on a single page. As in screen 1 of Figure VII-12, clicking on the comment section of any idea of the list highlights the line and displays the cursor so the user can enter his/her comments, critiques, etc. This activates the 'Save update' button to users so they submit their updates. The 'delete' button is shown to users only next to the ideas that they have defined as shown in screen 2 of Figure VII-12. If the facilitator authorizes the ideas of others' editing, the click on the idea would activate its section, the same as for the comments.

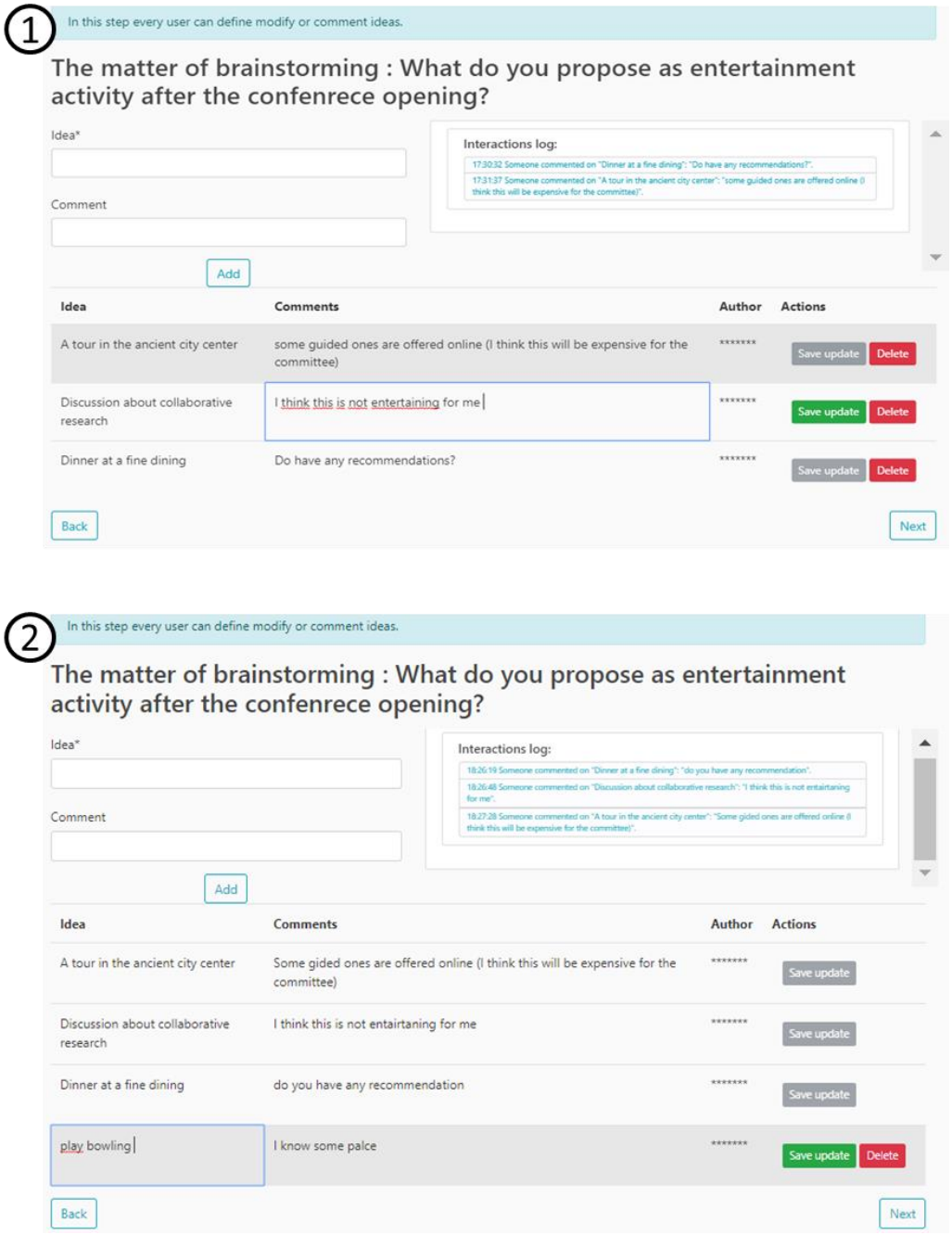


Figure VII-12 New OnePage Thinklet implementation

To avoid distracting one user when another submits a modification on an idea that he/she is currently editing, we have ensured that there will be no updates of the list of ideas unless the user deletes or saves his/her current modification, otherwise, even if other actions on the data have been carried out, we keep the screen fixed so he/she focuses on the holistic current state of the available ideas. The actions performed on the example shown in Figure VII-12 are anonymously displayed in the interactions log as well as in the ideas list since the anonymity option has been selected by

the facilitator in the parametrization step. Even though, it is possible for the facilitator to get back and update his/her choices of the 3 main settings of the OnePage Thinklet that are the question of the brainstorming, the authorization of others' ideas editing and the anonymity of the interactions. This is feasible without losing the definitions at hand if the facilitator estimates during the meeting that it will improve the group cohesion or help raise the inspirations that some influential team members might have on the others when their actions are revealed to the rest of the team.

VII.5.3.Multicriteria Thinklet

As multicriteria evaluation is a fundamental asset that the GDSS systems offer, we have implemented the Multicriteria Thinklet so that it encompasses both the brainstorming of criteria and alternatives followed by the evaluation step. It starts with the criteria' brainstorming on a shared screen where all the participants are allowed to enter new criteria and to delete others' ones after discussion as illustrates the step 1 of Figure VII-13. We don't allow any modifications or interactions with the defined criteria to avoid losing too much time and to encourage the decision-makers to talk their propositions through. Additionally, the oral discussion of the criteria before entering them so that everyone gets to know at least what they are about, has been one of the raised points that we have observed the effects of its lack during the experimentations of GRUS as we have concluded in the previous chapter. The same approach is followed for the alternatives' definition in the following step as shown in step 2 of Figure VII-13. In this step, we display the list of criteria in the middle of the interface, so it keeps being under the attention of everyone while thinking and discussing the alternatives to propose. At this level, we have considered the possibility of navigating backward to change the list of the defined criteria that might be needed after having raised new ideas. To avoid distracting the participants' concentration and to guarantee the completion of every step by the whole group, the backward and forward navigation buttons are only displayed to the facilitator. Next, as shown in step 3 of Figure VII-13, every user is asked to define his/her individual preferences by filling the two-dimensional matrix to evaluate the alternatives against the criteria. Since the calculation methods are out of our work's scope, we have used the weighted sum model that is well known and widely acknowledged for a simple and consistent multicriteria decision making (Tsurkov 2001).

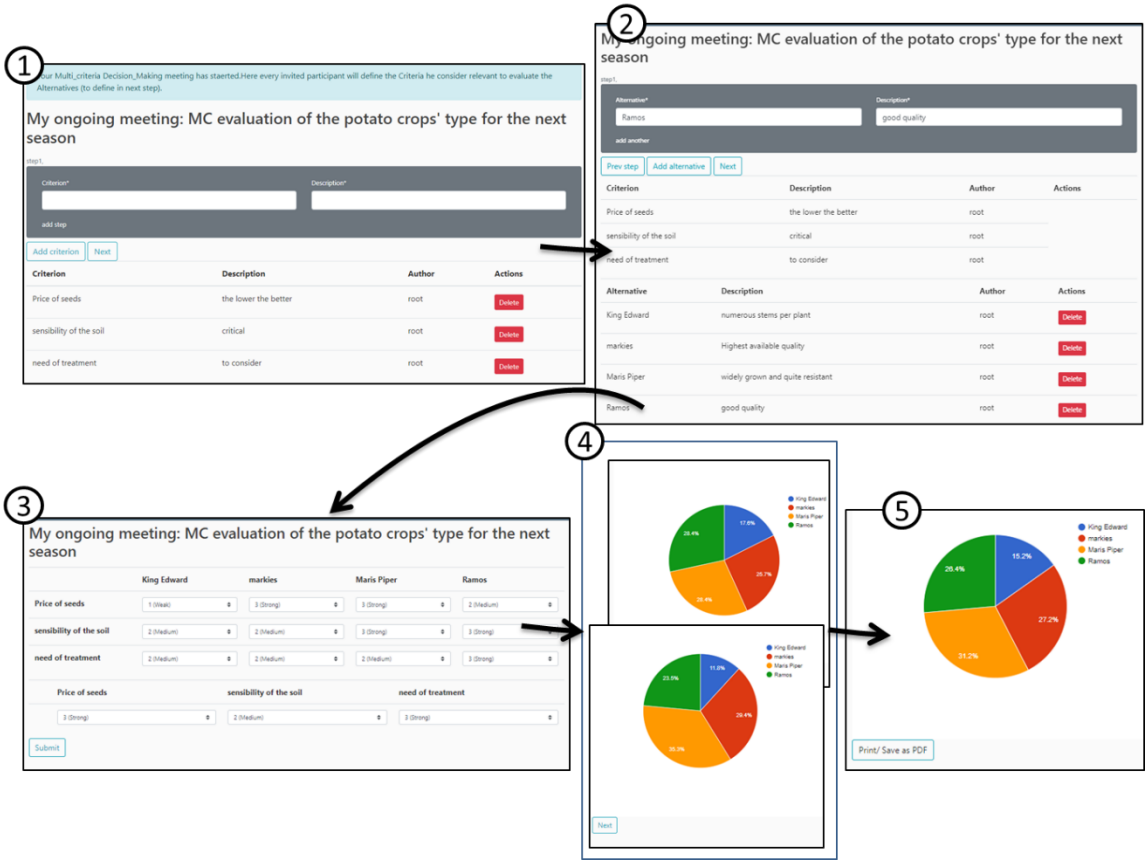


Figure VII-13 Steps of the Multicriteria Thinklet implemented in GROUDA

We have used a 1-3-point scale, 1 for 'weak', 2 for 'medium' and 3 for 'strong'. In this step the users should also evaluate the criteria by giving them importance weights that will be used for the calculation, as displayed under the evaluation matrix in step 3 of Figure VII-13. Once the users finish the evaluations, every one of them can go to the next step without waiting for the rest of the group, where he/she will see the calculation results of his/her individual definitions plotted in a pie chart. As shown in step 4 of Figure VII-13, the individual results' visualization permits the users to separately see what everyone's own evaluation has given, which will help him figure out the differences between the whole calculation results and his/her own in the next step. Finally, as in step 5 of Figure VII-13, The final aggregated results of all users are depicted in a pie chart that gets automatically updated with every user's submit. This final interface contains a button that allows downloading a recapitulative report with all the meeting's details. This evaluation Thinklet can be used separately for a multicriteria decision making, or for a direct vote by setting it up with a unique criterion of evaluation, when the alternatives' brainstorming does not require a prior exhaustive or

tricky interactive collaboration amongst stakeholders, otherwise, it must be precedented by a brainstorming Thinklet such as FreeBrainstorm or OnePage. It also allows, as we detail in the next section, skipping the alternatives' definition step when its use is invoked by another Thinklet such as OneUp.

VII.5.4.OneUp Thinklet

The OneUp Thinklet is a convergence Thinklet that has been defined by (Briggs and Vreede 2009) as one of many that can be chosen to converge on high quality brainstormed mass of ideas under time pressure. It helps surfacing the criteria of qualification of ideas since it relies on an initial collective prioritization, then a multicriteria evaluation that serves to clarify a murky problem's parameters for a further evaluation or organization. It consists of having the facilitator preparing a multicriteria evaluation of a set of defined items, usually defined during a prior brainstorming activity, while he/she asks the team members to watch a common screen that displays the list of ideas and to suggest orally what they believe most important among them, or to suggest new better ones, for a further evaluation that should be handled by either a multicriteria Thinklet or by a simple discussion. When proposing an idea, the user is asked to explain his/her choice, so the facilitator uses his/her argument to prepare the criteria list for the evaluation. Our implementation of the OneUp Thinklet is only based on a multicriteria evaluation that follows a brainstorming activity using the Multicriteria Thinklet that we have introduced in the previous section. For that, after finishing the brainstorming, we give the facilitator access to the criteria definition step of the Multicriteria Thinklet using the button 'prepare Multicriteria evaluation' that only him/her has it on his/her screen such as in the first screen of step 1 in Figure VII-14. In the meanwhile, staying blocked in the defined list screen as shown in the second screen of step 1 Figure VII-14, all the participants will start telling him in an oral discussion what they think better for evaluating them as criteria. While having the list of the brainstormed ideas displayed in the criteria definition page unlike when configuring the Multicriteria Thinklet for a standalone use, the facilitator enters the orally agreed proposed criteria as shown in step 2 of Figure VII-14. Then, he/she goes to the alternatives definition step of the Multicriteria Thinklet, step 3 of Figure VII-14, where the system will preload the ideas that the group have previously defined during the brainstorming activity in the alternatives list..

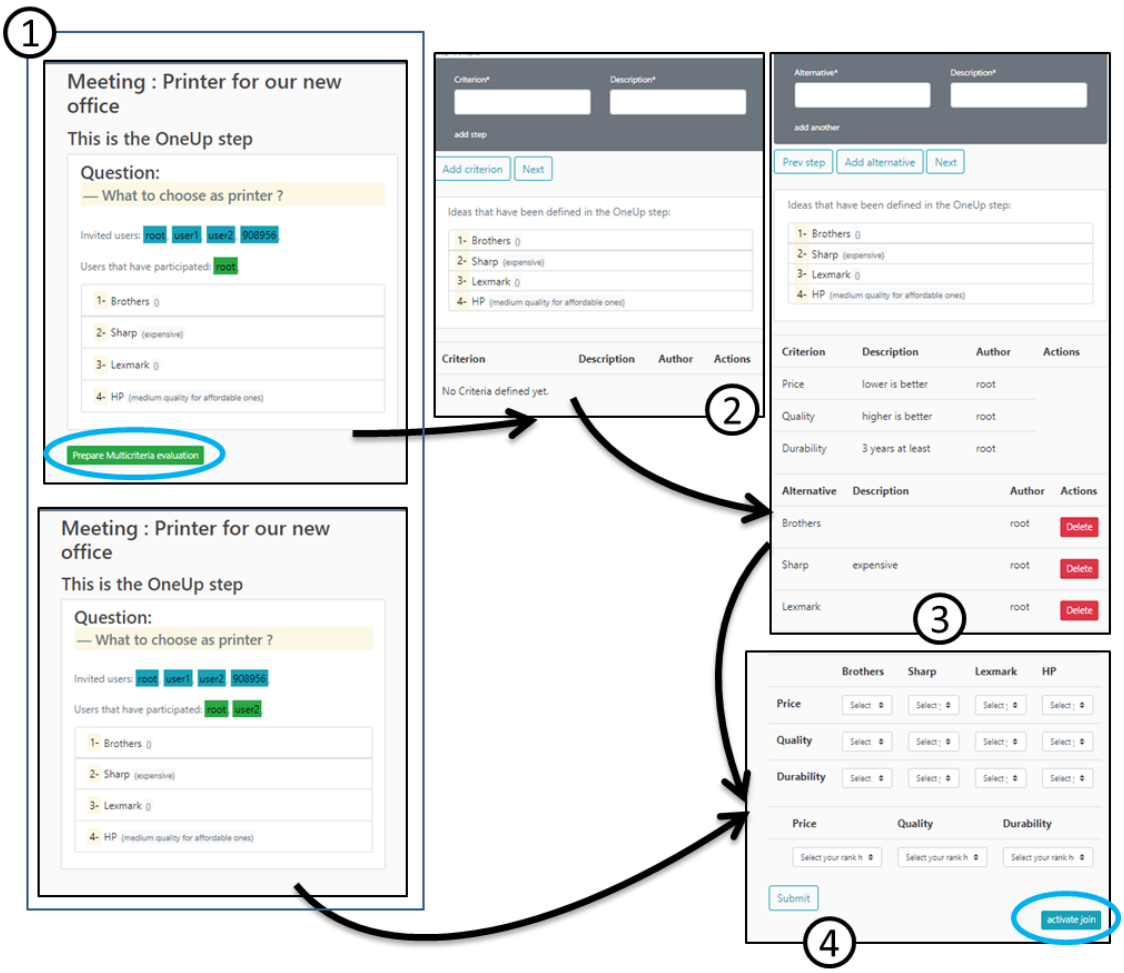


Figure VII-14 OneUp Thinklet steps in GROUDA

If there are no newly raised alternatives that the users have come up with while reflecting on those already defined during the criteria definition step, the facilitator will authorize the participants to join him directly in the following step i.e. the two-dimensional evaluation matrix using the 'activate join' button that is only shown to him when using the Multicriteria Thinklet with OneUp as shown in step 4 of Figure VII-14. The results of the OneUp Thinklet are eventually calculated and plotted in a pie chart, the same as for the Multicriteria Thinklet that we have detailed previously.

VII.5.5.Pin-The-Tail-On-The-Donkey Thinklet

The clustering Pin-The-Tail-On-The-Donkey Thinklet has been defined by (Briggs and Vreede 2009) in order to permit users of reducing, when they have a lot of definitions, their number to those that are worthy of further attention. It consists of allowing the practitioners to pin the definitions

that they consider key for the problem at hand, so they reduce the large number of the initial items that they cannot go through all of which one by one. Thus, the Pin-The-Tail-On-The-Donkey leads to a reduced set of ideas or commentaries that the group is willing to go on with to a plenary discussion where more explanations might be given by the users to assure their considerations.

We have implemented this Thinklet in GROUDA with the same perspective of improving the user interface friendliness and group members interactivity aspects. To do so, we have created a two-state pin that we display next to each element of the defined ideas list. The pins are initially put to the disabled state i.e. in grey colour, that gets changed to the active state i.e. in green colour, when clicked to signify that its correspondent idea is of a key interest, as shown in step 1 of Figure VII-15. At the same step, we have given the facilitator the right to authorize the number of allowed pins i.e. set to 3 as the default value, that the participants cannot use more than which when selecting the ideas from the list. Then without waiting for all the users to finish submitting their choices, as shown in step 2 of Figure VII-15, each user can go to the following screen where he/she sees all the filtered selections aggregated and displayed in a list that shows only the chosen ones and that gets updated with each other user's submit. Finally, the screen where the discussion takes place as shown in step 3 of Figure VII-15, shows a recapitulation of the meeting. It contains the initial list of definitions before the pinning activities, the list of the filtered ideas with the name of each idea's definer when the anonymity isn't required, the number of pins that it got and with an empty checkbox next to it so the facilitator can select it as finally kept after the discussion. Also, to keep the discussion aware of all the interactions that led to the current definitions, mainly during the brainstorming activity, we also load the interactions log that will display the list of all the modifications and comments.

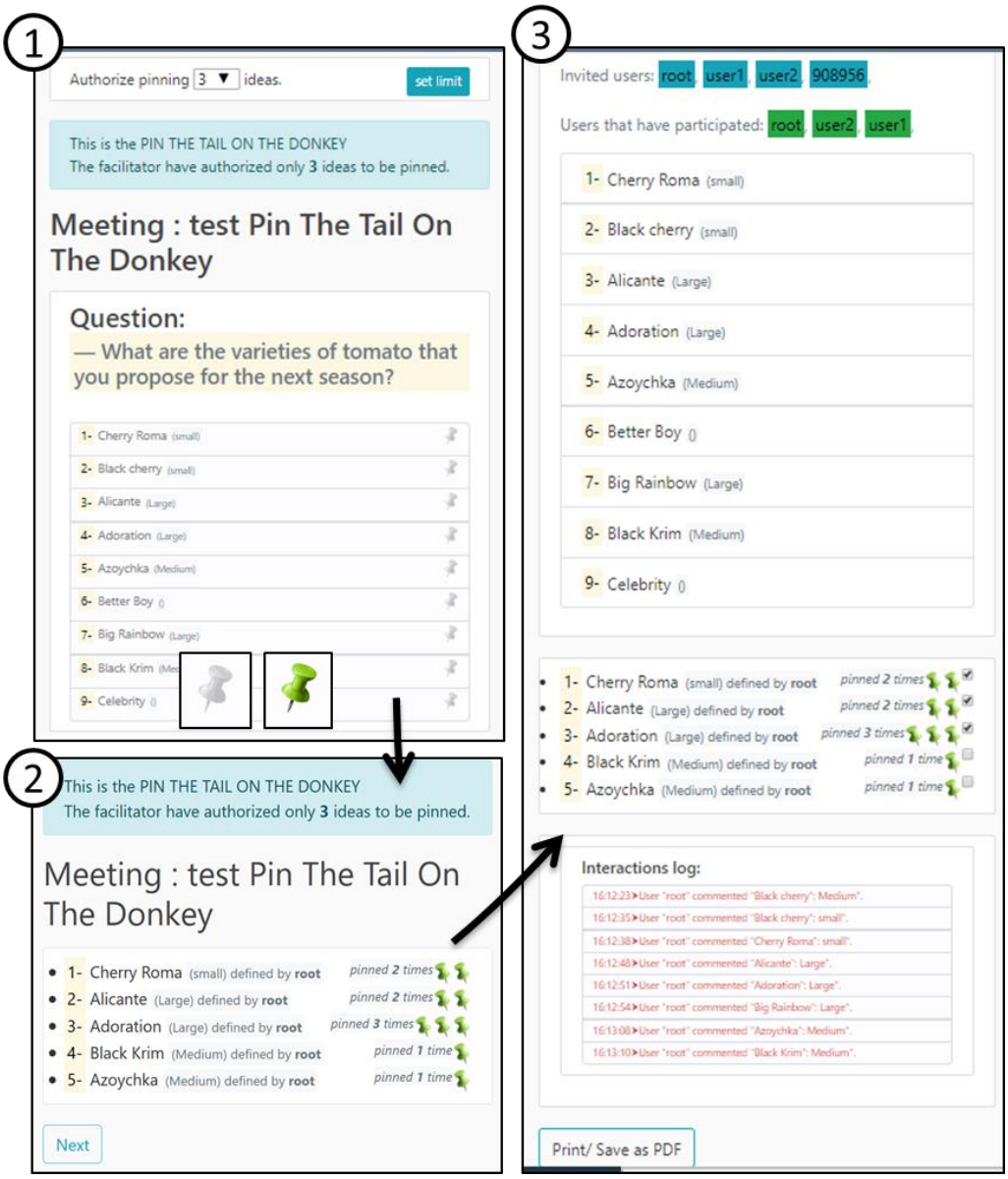


Figure VII-15 Pin-The-Tail-On-The-Donkey Thinklet step in GROUDA

As in all other meetings of GROUDA, the final step contains the meeting report generation button that allows printing or downloading a final recapitulative report. However, since the Pin-The-Tail-On-The-Donkey is a clustering Thinklet, it can be used for a preliminary organization of alternatives that are question of further evaluation activity such as Multicriteria, in this case its resulting set of well agreed upon elements will be the input for the following group activities without generation of report.

VII.5.6.CollaborativeDW Thinklet

The **CollaborativeDW** Thinklet has four possible usage parametrization's combinations that are illustrated in the process flow diagram of Figure VII-16. Using this Thinklet can be an asset for both participating users' profiles i.e. crowdsourcing volunteer or enterprise employee, for which we have detailed the differences that we considered in our methodology in Table III-1 of Chapter III as well as by the two possible 'collaborative resolution of conflicts' methods that we proposed in V.2.2.2.

The first recommendation question is: Do you, as a facilitator, consider yourself as an expert of the application domain of the meeting? If the answer to this question is a 'yes', there will be no need of using the attractiveness attribute that we have defined previously in V.2.2.2.2. Otherwise, the attractiveness attribute will be extracted from the configuration file, and will appear next to each element as shown in Figure VII-18, to help the facilitator with a hint on its validity likeliness. Then for the second question: Do the committers know the basics of data warehouse systems use? If this is the case, a discussion followed by a selection of the elements to modify later by the facilitator is proposed without using the evaluation criteria of section V.2.2.2.2 of Chapter 5. Otherwise, an evaluation assisted by the evaluation criteria of the elements will be performed.

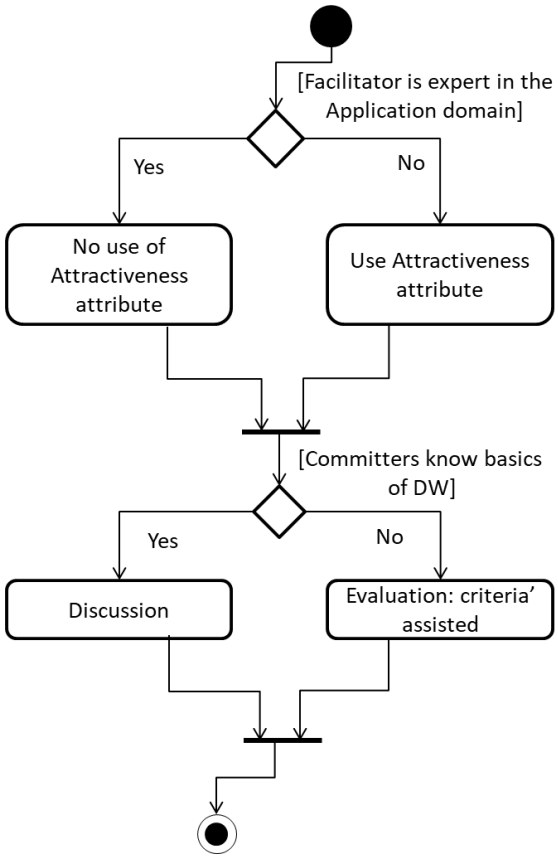


Figure VII-16 Process flow diagram of CollaborativeDW Thinklet parametrization

The CollaborativeDW Thinklet allows the automatic importation of the multidimensional elements of the cube schema at hand. In the current version of GROUDA, the importation is only allowed from Mondrian xml configuration files.

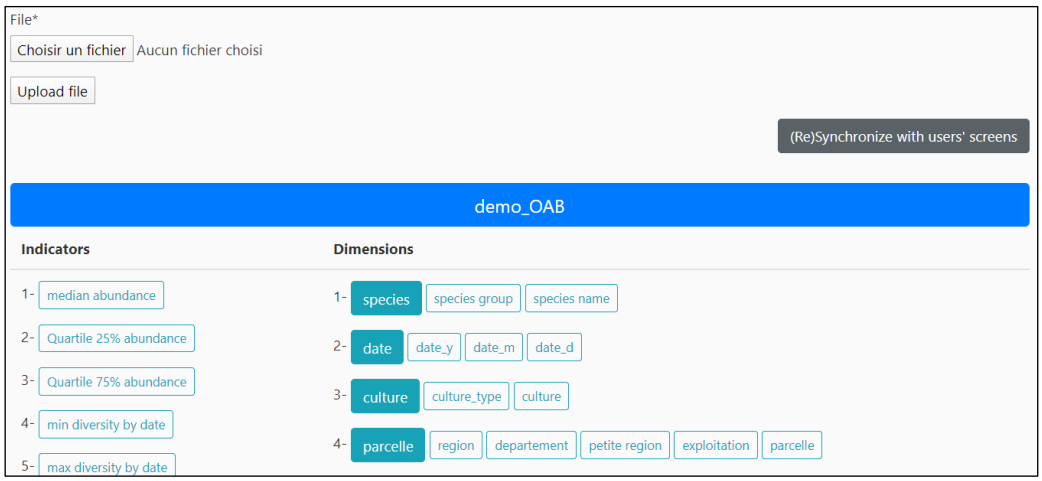


Figure VII-17 CollaborativeDW Thinklet configured for discussion and without Threshold values

As shown in Figure VII-17, the CollaborativeDW is based on one screen when it is configured for a collective discussion about the schema elements. If it is configured for an evaluation, it is based on a navigation between two screens: the first for the schema importation and display as in Figure VII-18, as well as for the results of each accomplished evaluation display such as in Figure VII-20. The second screen is for the navigation to a chosen element evaluation as in Figure VII-19.

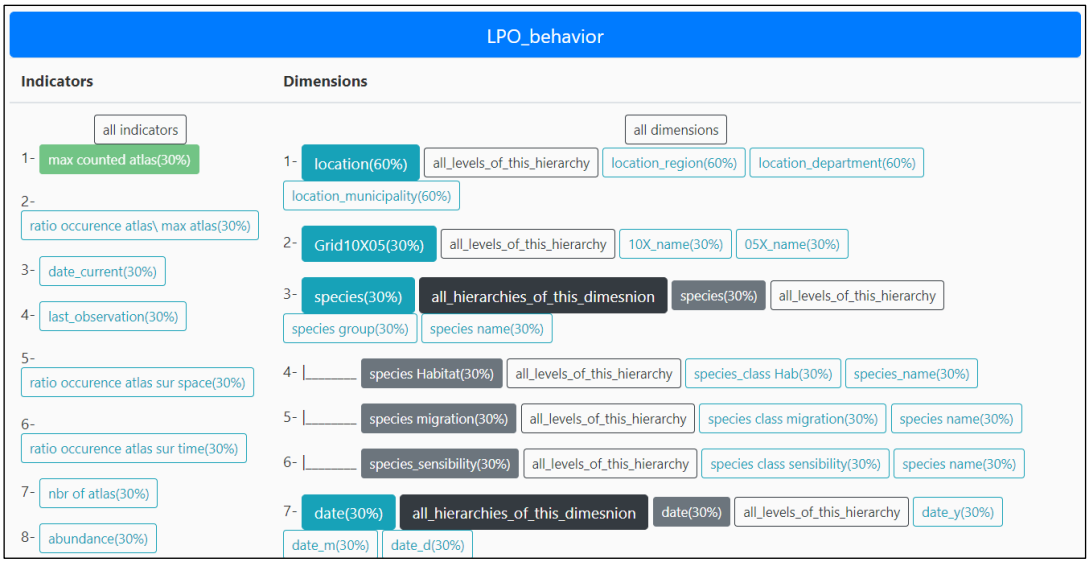


Figure VII-18 CollaborativeDW Thinklet configured for vote and with Threshold values

This Thinklet should be chosen among the generic evaluation Thinklets that are available in GROUDA in order to evaluate a data warehouse design either if the committers have a sufficient understanding of the basics of DW use or not. This way, they can evaluate the schema elements without the evaluation criteria' assisted vote when they are familiar with the use of DW systems or get led by the DW expert in a further voting step.

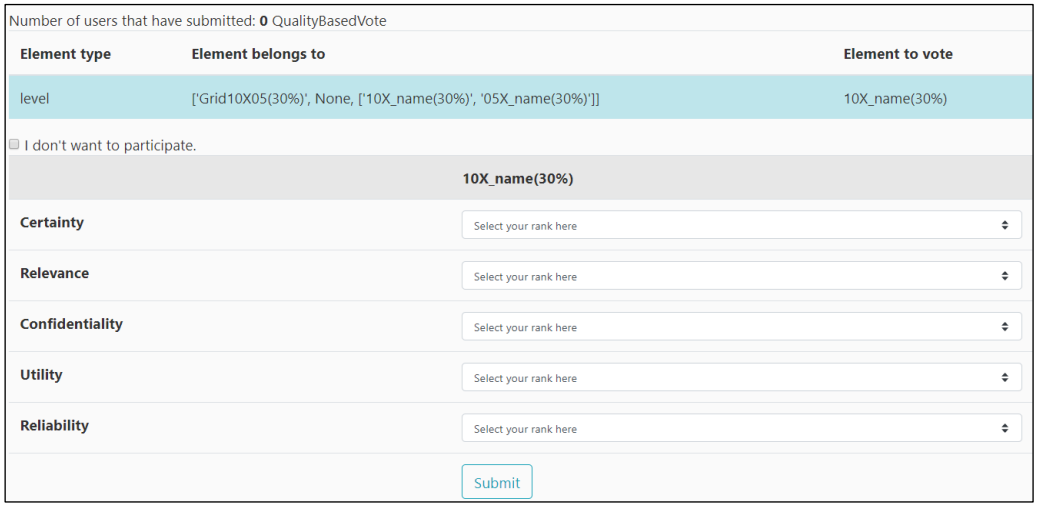


Figure VII-19 Example of a voting screen of the CollaborativeDW Thinklet

When the committers are familiar with DW basics, the facilitator of this Thinklet, as an expert of the DW design, explains the displayed schema components, invites the committers to discuss and notes their suggestions so he/she modifies them later on in the schema design. It shouldn't be used for a fast discussion if the committers don't know at least the fundamentals of using a DW, such as understanding the differences between dimensions, hierarchies, levels etc.

For the case where the committers don't have the least knowledge of the DW fundamentals, the same screen of this Thinklet is used with an additional functionality that allows voting the multidimensional elements when clicking on each of them. As shown in Figure VII-18, in this case, additional elements are added to the imported schema elements, so that the voting procedure suits the evaluation criteria that we have defined earlier Table V-2. Then, when the facilitator clicks on an element, the voting screen will be automatically loaded on all participants' screens. This will help the group members to keep focused on each currently chosen item as well as staying synchronized when following up the facilitator's choices among the many displayed components. An example of the voting screen is shown in Figure VII-19, where a description is always shown to remind the element's name, its type, and to what upper level component it belongs. In the example shown in Figure VII-19, every participant evaluates the level "10x_name" on the basis of the five criteria that we have defined for 'level' elements and submits the results. Only the facilitator has the button 'next' that allows to bring back automatically and at the same time all the team together to the first page where, in addition to the initially displayed elements that they can continue to evaluate, they can as well see the results of their accomplished evaluations as shown in Figure VII-20.

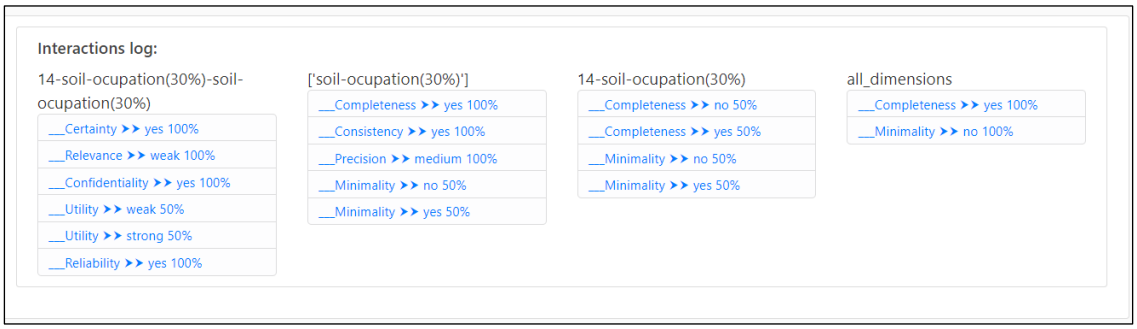


Figure VII-20 Example of evaluation results display

As we have mentioned in V.2.2.2.2, this implementation of the CollaborativeDW Thinklet is a solution that is dedicated to solving the complex issues with the extended approach of requirements conflicts' resolution. Its execution with the extended, but too detailed, criteria list would cause

unreasonably long collaborative sessions if used with a real conflictual model with any classical GDSS. We can hence summarize the advantageous functionalities of this Thinklet in three main points:

1) Automatic importation of schemas that:
   - Saves the time of entering by hand many elements.
   - Reads directly the Mondrian xml configuration file.
   - Extracts from the xml and displays, when configured for, the attractiveness attributes next to their corresponding elements.
   - Generates, when configured for, buttons regrouping the sets of elements detailed in Table V-2.
2) An all-in-one meeting that:
   - Allows evaluating large cubes, which is the case with the fused ones of our methodology, in a reasonable time.
   - Is minimized to a limited number of steps and pages to ease reducing the configuration and the preparation cognitive loads.
   - Allows compacted results' display at the same schema elements' navigation page, which simplifies following-up with the meeting's advancement and having a precise and well-presented report for later-on schema updates.
3) Flexibility:
   - All the navigation is controlled by the facilitator to avoid synchronization issues.
   - Unbounded navigation logic to allow the facilitator moving forward and backward, with all the participating team, without being forcibly restricted by our methodological recommended order of elements' evaluation.
   - Allowing users to skip participating to evaluate elements that they feel unable to give correct answers about their features i.e. a checkbox putting, when checked, all the criteria ranks selection list to 'NA-Not applicable', as in Figure VII-19.
   - A meeting can be flexibly reported without losing its data nor parameters or ended without finishing all the evaluation, in case where the remaining elements are considered unproblematic for example.

VII.5.6.1. Validation with the VGI4BIO use case

In order to test this Thinklet and validate its usability, we have run 3 tests with different committers that treated separately the same model illustrated in Figure VII-21. This is a straw model that we

have defined to resemble the volunteers' elicited models of the OAB database i.e. Agricultural Biodiversity Observatory – for "Observatoire Agricole de la Biodiversité" in French, and that we have, on purpose, left in which some imprecise definitions that we want to assess the effectiveness of the tool in easing their detection.

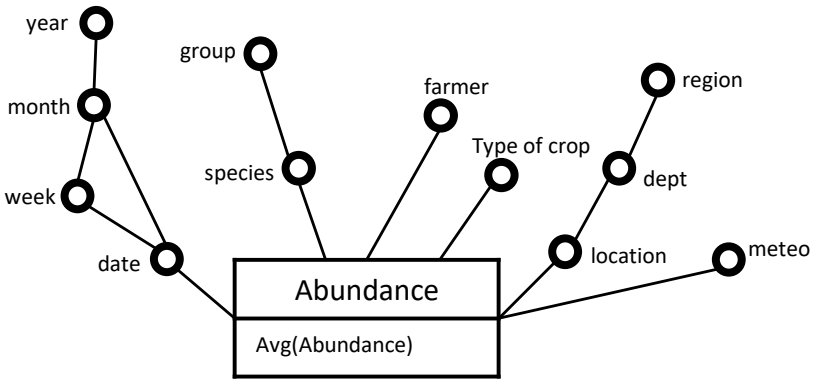


Figure VII-21 Cube schema for agro-biodiversity analyses relative to OAB database

This model contains 6 dimensions and 1 measure as shown in the conceptual model of Figure VII-21 and that is loaded by the CollaborativeDW Thinklet as in Figure VII-22. We have used in these tests the profile-aware evaluation method defined in V.2.2.2.2 with the attractiveness values that we have manually prepared for which in the XML configuration file.

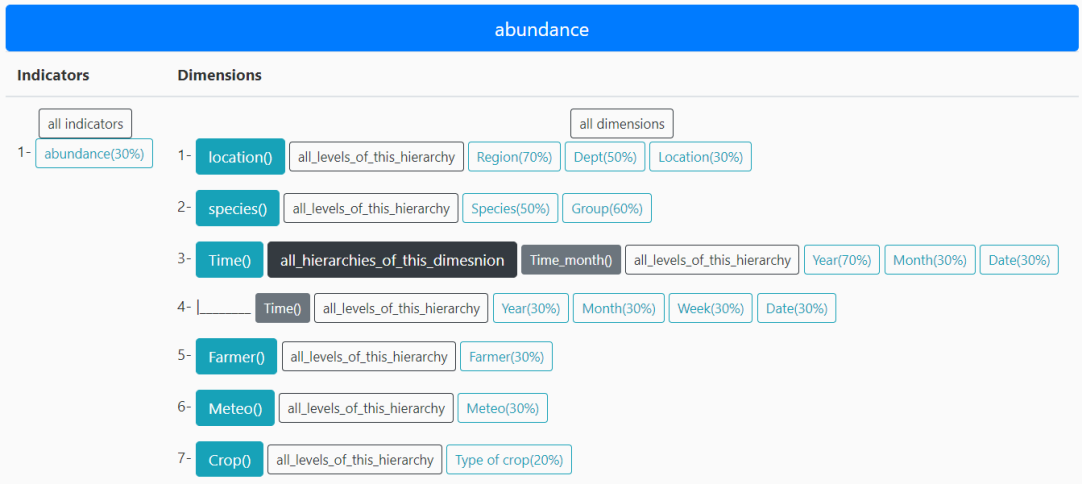


Figure VII-22 OAB's straw model imported by CollaborativeDW

The attractiveness values are defined only for the levels and measures elements with a threshold of 50%, which means that only 3 elements are not going to be evaluated:

- The level 'Region' of the dimension 'Location' that has 70%;
- The level 'Group' of the dimension 'Species' that has 60%;
- The level 'Year' of the hierarchy 'Time_month' of the dimension 'Time' that has 70%.

VII.5.6.1.1. OAB model tests' results

In order to evaluate the effectiveness of the CollaborativeDW Thinklet with the process defined in Figure V-11 for the profile-aware refinement method, three tests have been run in three different sessions with two DW expert facilitators and three groups of committers that we introduce in Table VII-1 with their affiliations and expertise. All the 7 committers know the OAB data due to prior participation to the OAB crowdsourcing project or to the VGI4BIO project during DW elicitation and data cleansing or transformation tasks. It is important to mention that committers 5,6 and 7 who are affiliated to MNHN[32], are very familiar with the data since they work on the OAB project.

Table VII-1 Committers' groups of the conducted tests with OAB model

-	Committer	Affiliation	Expertise
Test 1	Committer 1	Chambre D'agriculture Allier[33]	Environmental researcher
	Committer 2	Symbiose Allier[34]	Environmental engineer
Test 2	Committer 3	AgroParisTech	Agronomist
	Committer 4	Chambre D'agriculture France[35]	Biodiversity specialist
Test 3	Committer 5	MNHN	Ecological researcher
	Committer 6	MNHN	Agronomist, Project coordinator
	Committer 7	MNHN	OAB project manager

We have succeeded in executing the collaborative process using GROUDA and a videoconference with the three groups in 1, 2 and 1.25 hours respectively. After The meetings we have compared the suggested modifications accompanied by the evaluation results in order to better assess the effectiveness of detecting the existing pitfalls along with a questionnaire that has been filled by all the committers with questions about their satisfaction levels (available in appendix B). The pre-existing issues and their suggested corrections are illustrated in Table VII-2.

[32] https://www.mnhn.fr/en
[33] https://extranet-allier.chambres-agriculture.fr/
[34] https://symbioseallier.fr/
[35] https://chambres-agriculture.fr/

Table VII-2 OAB model's pre-existing issues and the suggested solutions by the 3 committer teams.

-	issue	Test	Solution (if any)
1	The hierarchy 'Time' is included in the second hierarchy 'Time_month' therefore useless.	1	Detected and the name of the 'Time' hierarchy was found imprecise if the two hierarchies are to be kept.
		2	Detected and the hierarchy 'Time_month' was considered useless.
		3	Detected and the hierarchy 'Time_month' was considered useless. New levels were also suggested for the 'Time' hierarchy, so it becomes: 'day_date'→'pentade'→'decade'→'week'→'month'→'season'→'year'.
2	The level 'week' of the hierarchy 'Time' is controversial since some experts consider it imprecise for analysis.	1	Detected and the removal of the level 'week' was suggested.
		2	Detected and the removal of the level 'week' was suggested.
		3	Detected and considered as useful for unspecialized users after discussion.
3	The name of the level 'Location' of the 'Location' dimension is ambiguous.	1	Detected and 'coordinates' was suggested instead but with caution since it is a confidential information.
		2	Detected and 'county' was suggested instead since the coordinates reveal a confidential information.
		3	Detected and 'coordinates' was suggested instead but with caution since it is a confidential information. The same risk of revealing the confidential information i.e. the precise parcel of land, was pointed out for the level 'Dept' of the same hierarchy where parcels become identifiable for departments that contain 3 or fewer of which. The replacement of the name of the 'Dept' level by 'Department' was also suggested.
4	The order of levels of the hierarchy 'Species' is inverted.	1	Not detected.
		2	Detected and the removal of the level 'Species' was suggested while considered very technical to know for most insect species. More precise appellation was proposed as well: 'Species family' instead of 'Group'.
		3	Detected and inversion was suggested.
5	The hierarchy 'Crop' might be incomplete.	1	Detected without suggestions.
		2	Detected and an upper level 'Type of crop' was suggested.
		3	Detected and an upper level 'Type of crop' was suggested.
6	The dimension 'Farmer' is confidential and useless for the abundance analysis.	1	Detected and removal of the dimension was suggested.
		2	Detected and removal of the dimension was suggested.
		3	Detected and replacing farmers' names by their ids to hide their identities was suggested. The usefulness for the analysis was discussed and considered as possibly useful when wanting to identify the active/ trustworthy observers.
7	The absence of the important dimension 'Agricultural practice'.	1	Detected and 'Agricultural practice' dimension was proposed.
		2	Detected and 'Technical practice' dimension was proposed.
		3	Detected and 'Agricultural practice' dimension was proposed.
8	The absence of more useful aggregations of abundance such as 'MIN', 'MAX' etc.	1	Detected and aggregators 'MIN', 'MAX' and 'MEDIAN' suggested.
		2	Detected and aggregators 'MIN' and 'MAX' suggested.
		3	Detected and aggregators 'MIN', 'MAX', 'MEDIAN', '25^{TH}_PERCENTILE' and '75^{TH}_PERCENTILE' were suggested for the measure 'Abundance' and a 'Diversity' measure with aggregators 'AVG', 'MIN', 'MAX' and 'MEDIAN' was also proposed.

The success in resolving the design issues has been remarkable since almost all three committer teams have identified and suggested an adequate correction for almost all the 8 pre-existing pitfalls as detailed in Table VII-2. Only the first team has not detected the issue 4 i.e. The order of levels of the hierarchy 'Species' is inverted, and detected, but did not had a suggestion in mind by the moment for, the issue 5 i.e. The hierarchy 'Crop' might be incomplete. Nonetheless, with the proposed modifications, all the three teams have expressed their satisfaction about the fact that the resulting models have become implementable and usable for an effective analysis of the OAB abundance unlike the initial state of the treated model. It has been also obvious to us that the third group of committers has had the most effective and precise contribution which is most probably

because of their high level of specialization with the data, since they are actually employed by the MNHN i.e. the VGI4BIO project partner that leads the OAB data collection and monitoring project.

VII.5.6.1.2. Users' feedback on the CollaborativeDW Thinklet

Right after each test, the participants have answered the 14 questions of the questionnaire of appendix B that we have defined in order to better assess their satisfaction levels with the CollaborativeDW Thinklet and the DW design results. The first 11 answers were delivered in the 5-point Likert scale (Nadler et al. 2015) which allows a neutral midpoint and two nuances for positive and negative answers e.g. very dissatisfied, fairly dissatisfied, Neither satisfied nor dissatisfied, Fairly satisfied and Very satisfied. We have given a maximum space of 2 sentences for the remaining 3 open questions that asked mainly for general suggestions, if any. As described in the previous subsection, the results of the evaluation were very satisfying for us since all the three teams have succeeded in identifying and appropriately correcting the cube's issues i.e. 87.5% for the team 1 and 100% for the teams 2 and 3. However, their interactivity impressions, satisfaction with the user-experience and willingness to use the system in the future is what we focus on in this evaluation, so we concentrate more on the raised points when improving the collaborative process and the implemented tool. We have listed in Table VII-3 11 metrics that are associated each to one of the 11 first questions and that we rely on to conclude on their basis the further improvements.

Table VII-3 Evaluation metrics feedback

Negative		Neutral	Positive	
-2	-1	0	+1	+2

		Test 1		Test 2		Test 3		
		Committer						
-	Metric	**1**	**2**	**3**	**4**	**5**	**6**	**7**
1	Overall satisfaction	+1	+1	+1	+1	-1	0	/
2	Success in identifying conceptual errors	+1	+1	+1	+1	+1	+1	/
3	Success in identifying conflictual elements	+1	0	+2	0	+1	+1	/
4	Success in building group consensus	+1	+1	+1	+2	+1	+1	/
5	Improving understanding DW design	+1	+2	0	+1	0	+1	/
6	Complexity of the evaluation process	0	0	-1	-1	0	-1	/
7	User-friendliness of the interface	0	+1	0	+1	0	+1	/
8	Sufficiency of facilitation	0	-1	+2	+2	0	0	/
9	Understandability of the collaborative process	-1	-1	-1	/	+1	+1	/
10	Ease of following up with the process execution	-1	-1	-1	/	0	0	/
11	Willingness to reuse the system in the future	+1	+1	0	/	0	-1	/

As the overall experience can be judged satisfactory, the metrics that took, as illustrated in Table VII-3, the more critical feedback are 6, 9 and 10. This has also been orally expressed by more than 50% of the committers as well as pointed out in the questionnaire as the most complicated steps of

the tool that accompanied the metric 6. More precisely, the 8 evaluation criteria were hard to understand and have required a quite overwhelming reminder of their meanings done by the facilitator at each voting step. Some of which has remained confusing even after the explanation either for their unclear positive or negative aspects such as ‘Minimality’ that is, indeed, positive when the employed elements of a set are all required, which means that when satisfied the answer must be ‘yes’, or for their unnecessary use when a set does not contain more than one element such as ‘Consistency’ for level elements. These limitations have, as well, complicated the facilitation task and obliged the facilitator to skip some redundant steps to avoid repeating for example the discussion about a rejected dimension that gets evoked, unavoidably, when starting its levels’ evaluation. The general understanding of the collaborative process and the meanings of the evaluation criteria have shown an improvement over time. In fact, the hardest steps to follow-up with were the first 2 or 3 evaluations while after that the process starts to become iterative and thus clearer to an extent that allows putting more focus on the subject matter discussions rather than getting along with meeting steps. These results have allowed us to spot, for further improvements, a set of limitations in both the collaborative process and the CollaborativeDW Thinklet that we list as follows:

For the collaborative process:

- Too complicated and unexplained set of evaluation criteria that must be reviewed for simplification, if possible, and for replacement with detailed questions instead.
- Reconsidering the order of the steps to minimize as much as possible the repetitions and to avoid evaluating components before their containing composite elements.
- Preparing a prototype of the treated model so it can help visualizing examples in case committers mix-up similarly named multidimensional elements such as a dimension and its hierarchy, when a query would help seeing the usefulness of a measure, etc.

And for the CollaborativeDW Thinklet:

- Improving the visibility of each under evaluation element at each step so the current step becomes clearer to the user.
- Adding the possibility to comment the elements and to suggest editing actions i.e. add, delete and modify, in addition to its evaluation which has been done manually during the tests.
- Improving the display of the results so it becomes more readable for the DW designer and more understandable by the committers so they can notice the effectiveness of the process or point out errors even after the validation of their votes in case they changed opinion afterwards.

VII.6. Conclusion

In this chapter, we have introduced a new GDSS system i.e. GROUDA, that relies on the concept of Thinklets as building blocks of the collaborative processes. We have accomplished before its design a series of experiments that we have used GRUS i.e. another GDSS system that we have used with our collaborative DW design approach, to conduct them in order to identify the most critical aspects of using such a system outside of its classical environments. This has led us to identify a set of limitations that we proposed an enhanced implementation of certain Thinklets to solve them. We have also implemented a prototype of a recommender system to assist novice facilitators in constructing their processes. To use this new system with our case study, we have proposed a new Thinklet, namely CollaborativeDW, that allows the execution of the profile-aware method of the collaborative refinement step of our DW design approach. This Thinklet has, as well, been implemented and successfully tested with real users from the VGI4BIO project partners.

Chapter VIII

GENERAL CONCLUSION

VII.1. Summary of our research contribution

Our proposal of a new data warehouse design methodology that aims at exploiting crowdsourced data as a huge and valuable source of analytical information (Bimonte et al. 2014; Gusmini et al. 2017; Herrera et al. 2015) has necessitated a thorough investigation of the literature in order to contextualize its specific parameters in contrasts with those that have drawn existing approaches' considerations. We have tackled this, first, by introducing the general concepts of crowdsourcing, OLAP, GDSS, and recommender systems along with their encapsulations in order to have the required basis that allows addressing properly the literature review. Second, by illustrating the motivational aspects where we have introduced our empirical case study of the VGI4BIO project and answered the introductive questions that explained (i) the interest of developing data warehouses for crowdsourced data, (ii) the differences and specificities that have citizen science volunteers against enterprise employed users, the classical users engaged in data warehouse projects, so that our new design approach gets them into consideration, (iii) and the critical success factors that we need to be aware of while dealing with this new kind of system. Third, by defining a set of features that characterize our proposal and selecting accordingly the literature works in order to identify, for each step, either the most convenient existing approach or to suggest a new one that solves a newly raised issue. By this literature review, we have managed to spot some interesting ideas that have been applied or pointed out as potential promising practices in relation with our new context of engaging crowdsourcing volunteers in data warehouse design. In other terms, our methodology has to deal with the limitations of the volunteers' involvement that we had enumerated as:

- Too many volunteers: costly implementation in both time and resources if a DW is to be implemented for each user.
- Very limited knowledge of DW fundamentals: volunteers are data users or collectors with different backgrounds, hence the need for more explanatory and introductory effort than in classical projects.
- Not employed by the project: limited availability and unguaranteed commitment to a many sessions' participation.
- Geographically distributed: which represents another inconvenience for the collaborative work to be effectively accomplished.
- Limited vision of the project objectives: the volunteers are usually only interested in their individual goals.
- Difficulty of reaching unified vision: complexity of coping with conflicts.
- Limited proficiency of the application domain: volunteers' definitions are doubtable from the application domain point of view.
- Difficulty of participation to elicitation sessions: Limited time of availability, which requires an efficacious and flexible requirements' engineering management.

We were unable to find methodologies that are created to handle specifically the design of DW systems for crowdsourced data. However, when taken separately, some of the raised issues have been invoked in different contexts. We have found in the existing design methodologies some interesting approaches such as:

- (Bonifati et al. 2001) that suggest settling the formulated conflictual goals by eliminating those of them that are not agreed upon during an interaction with the involved stakeholders.
- (Paim and de Castro 2003) that rely on the intervention of external reviewers to support erroneous requirements detection.
- (Kumar and Thareja 2014) that also use review sessions in order to accomplish the requirements validation.
- (Giorgini et al. 2008; Giorgini et al. 2005) that defined a refinement step to reshape the users' specifications with a focus limited on early requirements i.e. high-level objectives.
- (Khouri et al. 2014b) that put in place an ontology to solve semantic and syntactic conflicts where working with many heterogeneous data sources and a common strategic objective.
- (Nabli et al. 2005) and (Bimonte et al. 2013b) that used the pivot table formalism for rapid prototyping of requirements.
- (Salinesi and Gam 2006) that proposed the concept of representative users to simplify managing the incoherencies and validating the requirements.

— (Elamin et al. 2017) that suggested using a matrix of requirements to eliminate the redundancies.
— (Jukic and Nicholas 2011) that addressed the limited availability of engaged users during the requirements elicitation and that solved this issue by assigning the task of deducing the missing DW analytical requirements from meta-data sources to a team of employees.

With that identified research opportunity, we have defined our methodology of collaborative DW design that is composed of two main steps: (i) Requirements elicitation and modelling and (ii) Solving subject matter issues of requirements. Another new aspect of our methodology, that we have not found in the existing literature, is the use of GDSS systems for DW requirements engineering, although it has been proven plausibly capable of managing the conflictual users' needs in other requirements engineering contexts. We have, therefore, defined in the first step of our methodology two scenarios of requirements engineering, using the ProtOLAP methodology that structures individual interviews and workshops when no collective objectives are identified and using GDSS's brainstorming techniques for a collective elicitation when groups of users with common objectives are identified. Next, we fuse in the second step the issued models based on their similarities with an algorithm that we have defined especially for our case of having probable analysis objectives intended by numerous unspecialized users in order to eliminate the redundancies and to merge the distinctly defined interdependent multidimensional elements. The resulted models will inevitably contain some new controversial components that are likely to be conceptually acceptable but with potential incoherence in relation with the application domain. To solve these conflicts, we also use the GDSS's collaborative techniques to assist the identification of the problematic elements, to raise the stakeholders' cohesion and to reduce the ill-structuration of the fused models by applying modifications proposed by experienced specialists of the application domain. We have structured this task with two methods that detail the collaborative process along with its criteria for each multidimensional element's evaluation. The first is the simplified collaborative approach that uses classical vote and multicriteria evaluation methods with a reduced number of criteria and group activities while the second , the profile-aware method, goes further in the detail of evaluation which necessitates a personalized GDSS tool that allows it to be executed in reasonable time. We have tested these two methods with our empirical case study, the first with an existing GDSS i.e. GRUS, and the second with a new GDSS that we have implemented after a set of user-experience experiments conducted with real practitioners of the RUC-APS project. For our new GDSS i.e. GROUDA, we have proposed an architecture that relies on the concept of Thinklets in composing collaborative processes, that improves the user-interface interactivity and that assists the unspecialized facilitators by a recommender system. In addition to some classical Thinklets, we have also implemented a new one i.e. CollaborativeDW, that can be used for a

collective assessment of the conceptual elements of any data warehouse model. This is possible thanks to its flexibility in executing the decision-making process. The elements on which to decide can be evaluated on general criteria and those in any chosen order. It suffices for this that the model to treat gets imported from a Mondrian xml i.e. the only format that the system accepts in its current version. This Thinklet allows the execution of the second collaborative resolution of requirements method of our DW design methodology that can be used either by volunteers or by enterprise users.

VII.2. Limitations of our research achievements and future research perspective

While we have focused our research on the problem of satisfying the volunteers engaged in the process of designing DW systems so it offers them a result which is comprehensible and useful for their various analytical needs, our methodology still comprises some limitations that need to be further addressed in order to confirm its efficacy and to widen its operability scope. These limitations are related to (i) our assumption of the data sources homogeneity, (ii) the incomplete usage of the methodology with the VGIO4BIO case study, (iii) the limited usability of the fusion algorithm if applied in an enterprise project and (iv) the previously identified user-experience limitations of the CollaborativeDW Thinklet and the collaborative profile-aware process that we have detailed in VII.5.6.1.2.

VII.2.1. Consideration of various data sources

Our assumption of the data sources homogeneity is due mainly to the advantageous centralization of data collected onto crowdsourcing platforms. However, even if it is less likely than with distributed data sources when working with organizational data, this cannot be always the case since, as we afterwards claimed the usability of our methodology with enterprise projects using the profile-aware collaborative method, external data may always be of interest depending on the specific needs of volunteers. Additionally, we have indeed needed some external data to complete the spatial dimension and the interpretations of Atlas codes with the LPO model. This have been verified and integrated without impacting the users' requirements which could have been problematic if it were related to other fundamental analytical requirements defined initially by users or rectified after the models' refinement. Taking this under consideration will require additional verifications during the elicitation and validation phase and an improvement of the fusion algorithm in order to manage the retrievability of compatible multidimensional elements from heterogeneous data sources such as what is suggested in the approaches of (Cabibbo and Torlone 2005; Kwakye 2011).

VII.2.2.More complete usage of the methodology

Although being used with a straw model that contained some pitfalls which has been eventually resolved, our methodology is still unclaimable to be fully functional nor totally consistent. This will require executing all its steps with empirical case studies such us the one offered by the VGI4BIO project with which we have applied as much as we could while we did not, however, foresee the time constraints that have prevented us from carrying out a holistic test strategy without impacting the progress of the other work packages. The phases that we have tested with the VGI4BIO project are:

- The first scenario of requirements engineering that relies on the pivot table formalism was thoroughly executed with the LPO model while the second, which uses GDSS brainstorming techniques, has not been tested due to technical issues that we have had with GRUS at the time. It is important to mention at this stage that our approach of using pivot tables for requirements engineering has been revisited and improved by our co-authors with the same case study of the VGI4BIO project (Bimonte et al. 2020).
- The fusion algorithm was partially tested with the LPO model and requires more validation of its approach which we believe it would be, anyway, altered after adding the data heterogeneity and the semantic appellations factors to the list of the issues at hand.
- For the two methods of the step collaborative resolution of requirements conflicts, the simplified method was executed once with a straw model and manually with the real case study while the second has been executed three times with a straw model and real users with the CollaborativeDW Thinklet of GROUDA.
- The rest of the implemented Thinklets have not been tested with the real case study of the RUC-APS project while it was planned but postponed because of the COVID-19 pandemic.

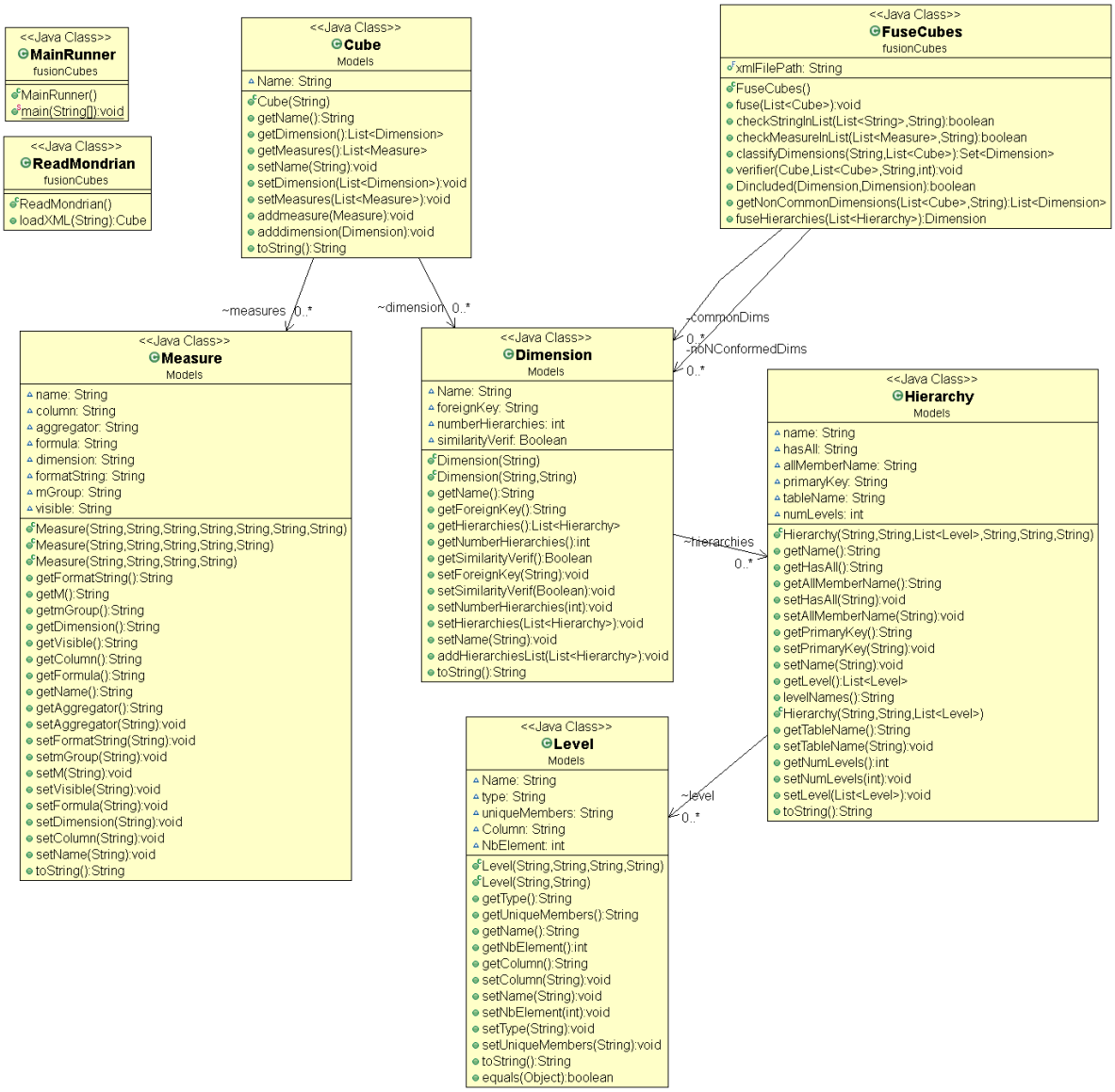
VII.2.3.The GROUDA platform

As aforementioned, the usability of the fusion algorithm is questionable if it is applied in an enterprise project since it does not consider the data heterogeneity which is very often a prerequisite with various departments, the usual in an enterprise working environment. However, this can always be achieved using other approaches while the GROUDA system is still, presumably, adequate in different phases of a collaborative design of DWs. This is because of its flexibility not only using the CollaborativeDW Thinklet but also with the other Thinklets that are meant to be

adaptable to organizational conflictual activities. Nevertheless, the implemented Thinklets still require a validation of the newly added user-friendliness functionalities. Another component of the GROUDA system that would be of an advantageous usefulness if both fluid functionality and appropriate design are proven suitable is the recommender system. The prototype that we have deployed for that matter has not as well been conveniently tested which keeps its success contingent on a further investigation and, potentially, more invested theoretical approach.

To conclude, we believe that we have tackled a research topic that lies on the intersection of many fields of technological research and that, by establishing our methodology of collaborative DW design, we have identified some promising articulative areas of scientific contribution that may lead to solve, or at least mitigate, the deficiencies of such multidisciplinary exploits. Although, we believe that reaching solid and acknowledgeable results would also require broadening the empirical and contextual circumstances' boundaries in order to confirm an independency vis-à-vis the application case study's domain. It was, in our case, the VGI4BIO and RUC-APS projects that were rather exceptional inaugural opportunities, but in no guaranteed way, an affirmative generalization proof for the configurations, the comprehensiveness by inexperienced users, the completeness and the clarity of the chosen communicative terminology etc. of our suggested approach.

Appendix A – The class diagram of Cubes' fusion java project



Appendix B – Questionnaire of GROUDA test

Please tick only one answer for each question:

- Overall, how satisfied are you with the system?
  - Very dissatisfied …….☐
  - Fairly dissatisfied…….☐
  - Neither satisfied nor dissatisfied…….☐
  - Fairly satisfied …….☐
  - Very satisfied……☐
- To what extent do you believe the system helped detecting design errors?
  - Very unhelpful …….☐
  - Fairly unhelpful …….☐
  - Neither helpful nor unhelpful ……..☐
  - Fairly helpful …….☐
  - Very helpful …….☐
- Did the system help identifying precisely the conflictual elements?
  - Very unhelpful …….☐
  - Fairly unhelpful …….☐
  - Neither helpful nor unhelpful ……..☐
  - Fairly helpful …….☐
  - Very helpful ……..☐
- After the use of the system, do you think that the next version of the DW will be more satisfying to all the group members?
  - Very unlikely …….☐
  - Fairly unlikely …….☐
  - Neither likely nor unlikely …….☐
  - Fairly likely …….☐
  - Very likely…….☐
- Do you think that this kind of activities help to understand more about DW design?
  - Very unhelpful …….☐
  - Fairly unhelpful …….☐
  - Neither helpful nor unhelpful ……..☐
  - Fairly helpful …….☐
  - Very helpful ……..☐
- Do you think that the system is too complicated to use? In which step(s)?
  - Very complicated …….☐
  - Fairly complicated…….☐
  - Neither simple nor complicated…….☐
  - Fairly simple …….☐
  - Very simple……..☐
  - Complicated steps: ……. (if any).
- How satisfied are you with the user interface friendliness?
  - Very dissatisfied …….☐
  - Fairly dissatisfied…….☐
  - Neither satisfied nor dissatisfied…….☐
  - Fairly satisfied …….☐
  - Very satisfied……☐
- Do you think that the facilitator role can be enough to build the necessary understanding of the system without any tutorials before the meeting?
  - Very unlikely …….☐
  - Fairly unlikely …….☐
  - Neither likely nor unlikely …….☐
  - Fairly likely …….☐
  - Very likely…….☐
- Do you think the DW evaluation process is easy to understand/comprehend?
  - Very complicated …….☐

- Fairly complicated……..……☐
- Neither simple nor complicated……..…☐
- Fairly simple ……..……..…☐
- Very simple…….☐

- Do you think the DW evaluation process is easy to implement?
  - Very complicated ……..……☐
  - Fairly complicated…….……☐
  - Neither simple nor complicated……..…☐
  - Fairly simple ……..……..…..…☐
  - Very simple……..…..……☐
- Would you like to reuse the system for another project?
  - Very unlikely ……..…….…..…☐
  - Fairly unlikely ……..……..☐
  - Neither likely nor unlikely ……..…..……..…☐
  - Fairly likely ……..……..……☐
  - Very likely……..…☐

- What are or are the <u>most</u> useful/beneficial steps in the DW design method? What for?
  .....
  .....

- What are or are the <u>least</u> useful/beneficial steps in the DW design method? What for?
  .....
  .....

- What would you propose as improvement of the system and/or the process?
  ……
  …….
  ……
  ……..

Appendix C – Example of GRUS' tests questionnaire

Evaluation of the system:

- Do you feel that the system helped creating the decision?
- Do you think that the system is too complicated?
- Is the user interface user friendly?
- What would you propose as improvement of the system?

Evaluation of the training against the experiment:

- Did the training help you to understand more the system?
- Do you think that the training helped you defining better the problem?
- Do you think that the facilitator role can be enough to build the necessary understanding of the system without any training before?
- Would one another experiment be enough for you to get more effectively used to the system?
- Would one another experiment on the same example give results that are more precise after having more understanding of the system?

Bibliography

1. Adla, A., Zarate, P., Soubie, J.-L.: A Proposal of Toolkit for GDSS Facilitators. Group Decis Negot. 20, 57–77 (2011). https://doi.org/10.1007/s10726-010-9204-8
2. Agarwal, R., Tanniru, M.R.: Knowledge Acquisition Using Structured Interviewing: An Empirical Investigation. Journal of Management Information Systems. 7, 123–140 (1990). https://doi.org/10.1080/07421222.1990.11517884
3. Aitamurto, T., Leiponen, A., Tee, R.: The Promise of Idea Crowdsourcing – Benefits, Contexts, Limitations. 30 (2011)
4. Allahbakhsh, M., Benatallah, B., Ignjatovic, A., Motahari-Nezhad, H.R., Bertino, E., Dustdar, S.: Quality Control in Crowdsourcing Systems: Issues and Directions. IEEE Internet Computing. 17, 76–81 (2013). https://doi.org/10.1109/MIC.2013.20
5. Ariyachandra, T., Watson, H.: Key organizational factors in data warehouse architecture selection. Decision Support Systems. 49, 200–212 (2010). https://doi.org/10.1016/j.dss.2010.02.006
6. Aurum, A., Wohlin, C.: The fundamental nature of requirements engineering activities as a decision-making process. Information and Software Technology. 45, 945–954 (2003). https://doi.org/10.1016/S0950-5849(03)00096-X
7. Batini, C., Scannapieco, M.: Data Quality: Concepts, Methodologies and Techniques. Springer Science & Business Media (2006)
8. Bimonte, Amir Sakka, Lucile Sautot: A New Methodology for Elicitation of DataWarehouse Requirements based on the Pivot Table Formalism - In: EDA 2018 vol. RNTI-B-14, 263-272 (2018)
9. Bimonte, S., Antonelli, L., Rizzi, S.: Requirements-driven data warehouse design based on enhanced pivot tables. Requirements Eng. (2020). https://doi.org/10.1007/s00766-020-00331-3
10. Bimonte, S., Boucelma, O., Machabert, O., Sellami, S.: From Volunteered Geographic Information to Volunteered Geographic OLAP: A VGI Data Quality-Based Approach. In: Murgante, B., Misra, S., Rocha, A.M.A.C., Torre, C., Rocha, J.G., Falcão, M.I., Taniar, D., Apduhan, B.O., and Gervasi, O. (eds.) Computational Science and Its Applications – ICCSA 2014. pp. 69–80. Springer International Publishing (2014)
11. Bimonte, S., Boulil, K., Pinet, F., Kang, M.-A.: Design of Complex Spatio-multidimensional Models with the ICSOLAP UML Profile - An Implementation in MagicDraw. In: ICEIS (2013)(a)
12. Bimonte, S., Edoh-Alove, É., Nazih, H., Kang, M.-A., Rizzi, S.: ProtOLAP: rapid OLAP prototyping with on-demand data supply. In: DOLAP (2013)(b)
13. Bishr, M., Janowicz, K.: Can we Trust Information ?-The Case of Volunteered Geographic Information. Presented at the (2010)
14. Bommarco, R., Kleijn, D., Potts, S.G.: Ecological intensification: harnessing ecosystem services for food security. Trends in Ecology & Evolution. 28, 230–238 (2013). https://doi.org/10.1016/j.tree.2012.10.012
15. Bonifati, A., Cattaneo, F., Ceri, S., Fuggetta, A., Paraboschi, S.: Designing data marts for data warehouses. ACM Trans. Softw. Eng. Methodol. 10, 452–483 (2001). https://doi.org/10.1145/384189.384190

16. Bonney, R., Ballard, H., Jordan, R., McCallie, E., Phillips, T., Shirk, J., Wilderman, C.C.: Public Participation in Scientific Research: Defining the Field and Assessing Its Potential for Informal Science Education. A CAISE Inquiry Group Report. (2009)
17. Bresciani, P., Perini, A., Giorgini, P., Giunchiglia, F., Mylopoulos, J.: Tropos: An Agent-Oriented Software Development Methodology. Autonomous Agents and Multi-Agent Systems. 8, 203–236 (2004). https://doi.org/10.1023/B:AGNT.0000018806.20944.ef
18. Briggs, R.O., Grünbacher, P.: EasyWinWin: Managing Complexity in Requirements Negotiation with GSS. In: HICSS (2002)
19. Briggs, R.O., Vreede, G.-D., Nunamaker, J.F., Tobey, D.: ThinkLets: achieving predictable, repeatable patterns of group interaction with group support systems (GSS). In: Proceedings of the 34th Annual Hawaii International Conference on System Sciences. pp. 9 pp.- (2001)
20. Briggs, R.O., Vreede, G.-J.D.: ThinkLets: Building Blocks for Concerted Collaboration. University of Nebraska, Center for Collaboration Science (2009)
21. Briggs, R.O., Vreede, G.-J.D., Jr, J.F.N.: Collaboration Engineering with ThinkLets to Pursue Sustained Success with Group Support Systems. Journal of Management Information Systems. 19, 31–64 (2003). https://doi.org/10.1080/07421222.2003.11045743
22. Cabibbo, L., Torlone, R.: Integrating Heterogeneous Multidimensional Databases. In: SSDBM (2005)
23. Camilleri, G., Zaraté, P.: A Group Multicriteria Approach. In: Kilgour, D.M. and Eden, C. (eds.) Handbook of Group Decision and Negotiation. pp. 1–26. Springer International Publishing, Cham (2019)
24. Candian, J.S., Martins, B.N.M., Cardoso, A.I.I., Evangelista, R.M., Fujita, E., Candian, J.S., Martins, B.N.M., Cardoso, A.I.I., Evangelista, R.M., Fujita, E.: Stem conduction systems effect on the production and quality of mini tomato under organic management. Bragantia. 76, 238–245 (2017). https://doi.org/10.1590/1678-4499.558
25. Candillier, L., Jack, K., Fessant, F., Meyer, F.: State-of-the-Art Recommender Systems, https://www.igi-global.com/chapter.aspx?ref=state-art-recommender-systems&titleid=6634
26. Carmel, E., Whitaker, R.D., George, J.F.: PD and Joint Application Design: A Transatlantic Comparison. Commun. ACM. 36, 40–48 (1993). https://doi.org/10.1145/153571.163265
27. Chen, C.-C., Huang, T.-C., Park, J.J., Yen, N.Y.: Real-time smartphone sensing and recommendations towards context-awareness shopping. Multimedia Systems. 21, 61–72 (2015). https://doi.org/10.1007/s00530-013-0348-7
28. Chen, L., Soliman, K.S., Mao, E., Frolick, M.N.: Measuring user satisfaction with data warehouses: an exploratory study. Information & Management. 37, 103–110 (2000). https://doi.org/10.1016/S0378-7206(99)00042-7
29. Clery, D.: Galaxy Zoo Volunteers Share Pain and Glory of Research. Science. 333, 173–175 (2011). https://doi.org/10.1126/science.333.6039.173
30. Constantinides, E., Fountain, S.J.: Web 2.0: Conceptual foundations and marketing issues. J Direct Data Digit Mark Pract. 9, 231–244 (2008). https://doi.org/10.1057/palgrave.dddmp.4350098
31. Cravero Leal, A., Mazón, J.N., Trujillo, J.: A business-oriented approach to data warehouse development. Ingeniería e Investigación. 33, 59–65 (2013)

32. Daniel, F., Kucherbaev, P., Cappiello, C., Benatallah, B., Allahbakhsh, M.: Quality Control in Crowdsourcing: A Survey of Quality Attributes, Assessment Techniques, and Assurance Actions. ACM Comput. Surv. 51, 7:1–7:40 (2018). https://doi.org/10.1145/3148148
33. Degrossi, L.C., Albuquerque, J.P. de, Rocha, R. dos S., Zipf, A.: A Framework of Quality Assessment Methods for Crowdsourced Geographic Information: a Systematic Literature Review. In: ISCRAM (2017)
34. Degrossi, L.C., Albuquerque, J.P. de, Rocha, R. dos S., Zipf, A.: A taxonomy of quality assessment methods for volunteered and crowdsourced geographic information. Trans. GIS. (2018). https://doi.org/10.1111/tgis.12329
35. Elamin, E., Alshomrani, S., Feki, J.: SSReq: A method for designing Star Schemas from decisional requirements. In: 2017 International Conference on Communication, Control, Computing and Electronics Engineering (ICCCCEE). pp. 1–7 (2017)
36. Evans, R., Park, S., Alberts, H.: Decisions not requirements: decision-centered engineering of computer-based systems. In: ECBS (1997)
37. Far, T.I.F., McNair, C.J., Vangermeersch, R.: Total Capacity Management: Optimizing at the Operational, Tactical, and Strategic Levels. CRC Press (1998)
38. Fernandez, A., Insfran, E., Abrahão, S.: Usability evaluation methods for the web: A systematic mapping study. Inf. Softw. Technol. 53, 789–817 (2011). https://doi.org/10.1016/j.infsof.2011.02.007
39. Fischer, F.: VGI as Big Data. A New but Delicate Geographic Data-Source. 46–47 (2012)
40. Flanagin, A.J., Metzger, M.J.: The credibility of volunteered geographic information. GeoJournal. 72, 137–148 (2008). https://doi.org/10.1007/s10708-008-9188-y
41. Fogliaroni, P., D'Antonio, F., Clementini, E.: Data trustworthiness and user reputation as indicators of VGI quality. Geo-spatial Information Science. 21, 213–233 (2018). https://doi.org/10.1080/10095020.2018.1496556
42. Fountas, S., Wulfsohn, D., Blackmore, B.S., Jacobsen, H.L., Pedersen, S.M.: A model of decision-making and information flows for information-intensive agriculture. Agricultural Systems. 87, 192–210 (2006). https://doi.org/10.1016/j.agsy.2004.12.003
43. Gardner, S.R.: Building the Data Warehouse. Commun. ACM. 41, 52–60 (1998). https://doi.org/10.1145/285070.285080
44. Giorgini, P., Rizzi, S., Garzetti, M.: Goal-oriented Requirement Analysis for Data Warehouse Design. In: Proceedings of the 8th ACM International Workshop on Data Warehousing and OLAP. pp. 47–56. ACM, New York, NY, USA (2005)
45. Giorgini, P., Rizzi, S., Garzetti, M.: GRAnD: A goal-oriented approach to requirement analysis in data warehouses. Decision Support Systems. 45, 4–21 (2008). https://doi.org/10.1016/j.dss.2006.12.001
46. Golfarelli, M., Mantovani, M., Ravaldi, F., Rizzi, S.: Lily: A Geo-Enhanced Library for Location Intelligence. In: Bellatreche, L. and Mohania, M.K. (eds.) Data Warehousing and Knowledge Discovery. pp. 72–83. Springer, Berlin, Heidelberg (2013)
47. Golfarelli, M., Rizzi, S.: Data Warehouse Design: Modern Principles and Methodologies. McGraw-Hill, Inc., New York, NY, USA (2009)
48. Golfarelli, M., Rizzi, S.: Data warehouse testing: A prototype-based methodology. Information and Software Technology. 53, 1183–1198 (2011). https://doi.org/10.1016/j.infsof.2011.04.002

49. Goodchild, M.F.: Citizens as sensors: the world of volunteered geography. GeoJournal. 69, 211–221 (2007). https://doi.org/10.1007/s10708-007-9111-y
50. Goodchild, M.F.: Commentary: whither VGI? GeoJournal. 72, 239–244 (2008)
51. Grabisch, M., Murofushi, T., Sugeno, M. eds: Fuzzy Measures and Integrals: Theory and Applications. Physica-Verlag Heidelberg (2000)
52. Green, C.H., Howe, A.P.: The Trusted Advisor Fieldbook: A Comprehensive Toolkit for Leading with Trust. John Wiley & Sons (2011)
53. Grigera, J., Garrido, A., Rossi, G.: Kobold: web usability as a service. In: Proceedings of the 32nd IEEE/ACM International Conference on Automated Software Engineering. pp. 990–995. IEEE Press, Urbana-Champaign, IL, USA (2017)
54. Grigera, J., Garrido, A., Zaraté, P., Camilleri, G., Fernández, A.: A Mixed Usability Evaluation on a Multi Criteria Group Decision Support System in Agriculture. In: Proceedings of the XIX International Conference on Human Computer Interaction. pp. 1–4. Association for Computing Machinery, Palma, Spain (2018)
55. Guo, Y., Tang, S., Tong, Y., Yang, D.: Triple-driven data modeling methodology in data warehousing: a case study. In: Song, I.-Y. and Vassiliadis, P. (eds.) DOLAP 2006, ACM 9th International Workshop on Data Warehousing and OLAP, Arlington, Virginia, USA, November 10, 2006, Proceedings. pp. 59–66. ACM (2006)
56. Gusmini, M., Jabeur, N., Karam, R., Melchiori, M., Renso, C.: Evaluating Reputation in VGI-Enabled Applications. In: EDBT/ICDT Workshops (2017)
57. Haklay, M., Weber, P.: OpenStreetMap: User-Generated Street Maps. IEEE Pervasive Computing. 7, 12–18 (2008). https://doi.org/10.1109/MPRV.2008.80
58. Harrington, R.J., Ottenbacher, M.C.: Decision-Making Tactics and Contextual Features: Strategic, Tactical and Operational Implications. International Journal of Hospitality & Tourism Administration. 10, 25–43 (2009). https://doi.org/10.1080/15256480802557259
59. Helquist, J.H., Kruse, J., Adkins, M.: Participant-Driven Collaborative Convergence. In: Proceedings of the 41st Annual Hawaii International Conference on System Sciences (HICSS 2008). pp. 20–20. IEEE, Waikoloa, HI, USA (2008)
60. Herlocker, J.L., Konstan, J.A., Riedl, J.: Explaining collaborative filtering recommendations. In: Proceedings of the 2000 ACM conference on Computer supported cooperative work. pp. 241–250. Association for Computing Machinery, Philadelphia, Pennsylvania, USA (2000)
61. Herrera, F., Sosa, R., Delgado, T.: GeoBI and Big VGI for Crime Analysis and Report. In: 2015 3rd International Conference on Future Internet of Things and Cloud. pp. 481–488 (2015)
62. Holten, R.: Specification of management views in information warehouse projects. Arbeitsberichte des Instituts für Wirtschaftsinformatik (2002)
63. Hwang, H.-G., Ku, C.-Y., Yen, D.C., Cheng, C.-C.: Critical factors influencing the adoption of data warehouse technology: a study of the banking industry in Taiwan. Decision Support Systems. 37, 1–21 (2004). https://doi.org/10.1016/S0167-9236(02)00191-4
64. Hwang, M.I., Xu, H.: The Effect of Implementation Factors on Data Warehousing Success : An Exploratory Study. Presented at the (2007)
65. Inmon, W.H.: Building the Data Warehouse (2Nd Ed.). John Wiley & Sons, Inc., New York, NY, USA (1996)

66. Jain, S., Grover, A., Thakur, P.S., Choudhary, S.K.: Trends, problems and solutions of recommender system. In: Communication Automation International Conference on Computing. pp. 955–958 (2015)
67. Jr, J.F.N., Briggs, R.O., Mittleman, D.D., Vogel, D.R., Pierre, B.A.: Lessons from a Dozen Years of Group Support Systems Research: A Discussion of Lab and Field Findings. Journal of Management Information Systems. 13, 163–207 (1996). https://doi.org/10.1080/07421222.1996.11518138
68. Jukic, N., Nicholas, J.: A Framework for Collecting and Defining Requirements for Data Warehousing Projects. CIT. Journal of Computing and Information Technology. 18, 377–384 (2011). https://doi.org/10.2498/cit.1001920
69. Khatib, F., DiMaio, F., Foldit Contenders Group, Foldit Void Crushers Group, Cooper, S., Kazmierczyk, M., Gilski, M., Krzywda, S., Zabranska, H., Pichova, I., Thompson, J., Popović, Z., Jaskolski, M., Baker, D.: Crystal structure of a monomeric retroviral protease solved by protein folding game players. Nature Structural & Molecular Biology. 18, 1175–1177 (2011). https://doi.org/10.1038/nsmb.2119
70. Khouri, S., Bellatreche, L., Jean, S., Ait-Ameur, Y.: Requirements Driven Data Warehouse Design: We Can Go Further. In: Margaria, T. and Steffen, B. (eds.) Leveraging Applications of Formal Methods, Verification and Validation. Specialized Techniques and Applications. pp. 588–603. Springer, Berlin, Heidelberg (2014)(a)
71. Khouri, S., Bellatreche, L., Jean, S., Ait-Ameur, Y.: Requirements Driven Data Warehouse Design: We Can Go Further. In: Margaria, T. and Steffen, B. (eds.) Leveraging Applications of Formal Methods, Verification and Validation. Specialized Techniques and Applications. pp. 588–603. Springer Berlin Heidelberg (2014)(b)
72. Kimball, R.: The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling, 3rd Edition. Wiley, Indianapolis, IN (2013)
73. Kipp, P.: Engineering Tool Supported Collaboration Processes for Web-based Platforms: ldea Elaboration in Virtual Ideation Communities. kassel university press GmbH (2016)
74. Kolfschoten, G.L., Briggs, R.O., Appelman, J.H., de Vreede, G.-J.: ThinkLets as Building Blocks for Collaboration Processes: A Further Conceptualization. In: de Vreede, G.-J., Guerrero, L.A., and Marín Raventós, G. (eds.) Groupware: Design, Implementation, and Use. pp. 137–152. Springer, Berlin, Heidelberg (2004)
75. Konaté, J., Sahraoui, A.E.K., Kolfschoten, G.L.: Collaborative Requirements Elicitation: A Process-Centred Approach. Group Decis Negot. 23, 847–877 (2014). https://doi.org/10.1007/s10726-013-9350-x
76. Kumar, V., Thareja, R.: Data Warehouse Requirements Engineering Dr. Presented at the (2014)
77. Kung, D.C., Bhambhani, H., Shah, R., Pancholi, G.: An Expert System for Suggesting Design Patterns — A Methodology and a Prototype. Presented at the (2003)
78. Kwakye, M.M.: A Practical Approach To Merging Multidimensional Data Models. Presented at the (2011)
79. Lin, Y.-P., Lin, W.-C., Lien, W.-Y., Anthony, J., Petway, J.R.: Identifying Reliable Opportunistic Data for Species Distribution Modeling: A Benchmark Data Optimization Approach. Environments. 4, 81 (2017). https://doi.org/10.3390/environments4040081
80. Lopez-Nores, M., Pazos-Arias, J.J., Garcia-Duque, J., Barragans-Martinez, B.: An agile approach to support incremental development of requirements specifications. In: Australian Software Engineering Conference (ASWEC'06). pp. 10 pp. – 18 (2006)

81. Lü, L., Medo, M., Yeung, C.H., Zhang, Y.-C., Zhang, Z.-K., Zhou, T.: Recommender Systems. Physics Reports. 519, 1–49 (2012). https://doi.org/10.1016/j.physrep.2012.02.006
82. Maguire, M.: User-Centred Requirements Handbook, /paper/User-Centred-Requirements-Handbook-Maguire/a5f7c1a6561a5b87ea34222414084a46b7d56cf8
83. McGoff, C., Hunt, A., Vogel, D., Nunamaker, J.: IBM's Experiences with GroupSystems. Interfaces. 20, 39–52 (1990)
84. Meteren, R. van: Using Content-Based Filtering for Recommendation. Presented at the (2000)
85. Michelle, C., Davis, C.H., Hardy, A.L., Hight, C.: Pleasure, disaffection, 'conversion' or rejection? The (limited) role of prefiguration in shaping audience engagement and response. International Journal of Cultural Studies. 20, 65–82 (2017). https://doi.org/10.1177/1367877915571407
86. Mishra, D., Mishra, A., Yazici, A.: Successful requirement elicitation by combining requirement engineering techniques. In: 2008 First International Conference on the Applications of Digital Information and Web Technologies (ICADIWT). pp. 258–263 (2008)
87. Molinari-Jobin, A., Kéry, M., Marboutin, E., Marucco, F., Zimmermann, F., Molinari, P., Frick, H., Fuxjäger, C., Wölfl, S., Bled, F., Breitenmoser-Würsten, C., Kos, I., Wölfl, M., Černe, R., Müller, O., Breitenmoser, U.: Mapping range dynamics from opportunistic data: spatiotemporal modelling of the lynx distribution in the Alps over 21 years. Animal Conservation. 21, 168–180 (2018). https://doi.org/10.1111/acv.12369
88. Mowat, H.: Alan Irwin, Citizen Science. Opticon1826. 6, (2011). https://doi.org/10.5334/opt.101109
89. Murugesan, S.: Understanding Web 2.0. IT Professional. 9, 34–41 (2007). https://doi.org/10.1109/MITP.2007.78
90. Nabli, A., Feki, J., Gargouri, F.: Automatic construction of multidimensional schema from OLAP requirements. The 3rd ACS/IEEE International Conference onComputer Systems and Applications, 2005. 28-NaN (2005). https://doi.org/10.1109/AICCSA.2005.1387025
91. Nadler, J.T., Weston, R., Voyles, E.C.: Stuck in the Middle: The Use and Interpretation of Mid-Points in Items on Questionnaires. The Journal of General Psychology. 142, 71–89 (2015). https://doi.org/10.1080/00221309.2014.994590
92. Nasiri, A., Ahmed, W., Wrembel, R., Zimányi, E.: Requirements Engineering for Data Warehouses (RE4DW): From Strategic Goals to Multidimensional Model. In: de Cesare, S. and Frank, U. (eds.) Advances in Conceptual Modeling. pp. 133–143. Springer International Publishing, Cham (2017)
93. Nayar, A.: Model predicts future deforestation. Nature. (2009). https://doi.org/10.1038/news.2009.1100
94. Newman, G., Zimmerman, D., Crall, A., Laituri, M., Graham, J., Stapel, L.: User-friendly web mapping: lessons from a citizen science website. International Journal of Geographical Information Science. 24, 1851–1869 (2010). https://doi.org/10.1080/13658816.2010.490532
95. Paim, F.R.S., de Castro, J.F.B.: DWARF: an approach for requirements definition and management of data warehouse systems. In: Proceedings. 11th IEEE International Requirements Engineering Conference, 2003. pp. 75–84 (2003)

96. Palaghias, N.: Opportunistic sensing platforms to interpret human behaviour., http://epubs.surrey.ac.uk/841529/, (2017)
97. Palma, F., Farzin, H., Guéhéneuc, Y.-G., Moha, N.: Recommendation system for design patterns in software development: An DPR overview. In: 2012 Third International Workshop on Recommendation Systems for Software Engineering (RSSE). pp. 1–5 (2012)
98. Pavlic, L., Podgorelec, V., Hericko, M.: A question-based design pattern advisement approach. Comput. Sci. Inf. Syst. (2014). https://doi.org/10.2298/CSIS130824025P
99. Pedersen, T.B., Jensen, C.S., Dyreson, C.E.: A foundation for capturing and querying complex multidimensional data. Inf. Syst. 26, 383–423 (2001). https://doi.org/10.1016/S0306-4379(01)00023-0
100. Prakash, N., Gosain, A.: An approach to engineering the requirements of data warehouses. Requir. Eng. 13, 49–72 (2008). https://doi.org/10.1007/s00766-007-0057-x
101. Princé, K., Moussus, J.-P., Jiguet, F.: Mixed effectiveness of French agri-environment schemes for nationwide farmland bird conservation. Agriculture, Ecosystems & Environment. 149, 74–79 (2012). https://doi.org/10.1016/j.agee.2011.11.021
102. Psaraftis, K., Anagnostopoulos, T., Ntalianis, K.S., Mastorakis, N.: Customized Recommendation System for Optimum Privacy Model Adoption. Presented at the (2018)
103. Quass, D.W.: Materialized views in data warehouses, (1998)
104. Regnell, B., Paech, B., Aurum, A., Wohlin, C., Dutoit, A., Dag, J.N.O.: Requirements Mean Decisions! – Research issues for understanding and supporting decision-making. In: in Requirements Engineering, First Swedish Conference on Software Engineering Research and Practise (SERP'01), October 25-26, Ronneby, Sweden [2] Kotonya and Sommerville (1998) Requirements Engingeering – Processes and Techniques. John Wiley & Sons (2001)
105. Régnier, C., Achaz, G., Lambert, A., Cowie, R.H., Bouchet, P., Fontaine, B.: Mass extinction in poorly known taxa. PNAS. 112, 7761–7766 (2015). https://doi.org/10.1073/pnas.1502350112
106. Ren, S., Wang, T., Lu, X.: Dimensional modeling of medical data warehouse based on ontology. In: 2018 IEEE 3rd International Conference on Big Data Analysis (ICBDA). pp. 144–149 (2018)(a)
107. Ren, S., Wang, T., Lu, X.: Dimensional modeling of medical data warehouse based on ontology. In: 2018 IEEE 3rd International Conference on Big Data Analysis (ICBDA). pp. 144–149 (2018)(b)
108. Rifaie, M., Kianmehr, K., Alhajj, R., Ridley, M.J.: Data warehouse architecture and design. 2008 IEEE International Conference on Information Reuse and Integration. 58–63 (2008). https://doi.org/10.1109/IRI.2008.4583005
109. Røkke, J.M., Muller, G., Pennotti, M.: Requirement elicitation and validation by prototyping and demonstrators: user interface development in the oil and gas industry. Syst. Res. Forum. 05, 89–108 (2011). https://doi.org/10.1142/S179396661100031X
110. Romero, O., Abelló, A.: A Survey of Multidimensional Modeling Methodologies. International Journal of Data Warehousing and Mining (IJDWM). 5, 1–23 (2009). https://doi.org/10.4018/jdwm.2009040101
111. Romero, O., Abelló, A.: Automatic validation of requirements to support multidimensional design. Data & Knowledge Engineering. 69, 917–942 (2010). https://doi.org/10.1016/j.datak.2010.03.006

112. Rose, D.C., Sutherland, W.J., Parker, C., Lobley, M., Winter, M., Morris, C., Twining, S., Ffoulkes, C., Amano, T., Dicks, L.V.: Decision support tools for agriculture: Towards effective design and delivery. Agricultural Systems. 149, 165–174 (2016)
113. Rossi, V., Caffi, T., Salinari, F.: Helping farmers face the increasing complexity of decision-making for crop protection. 1. 51, 457–479 (2012). https://doi.org/10.14601/Phytopathol_Mediterr-11038
114. Ruhe, G.: Software Engineering Decision Support – Methodology and Applications. Presented at the (2002)
115. Sakka, A., Bimonte, S., Sautot, L., Camilleri, G., Zaraté, P., Besnard, A.: A Volunteer Design Methodology of Data Warehouses. In: ER (2018)
116. Sakka, A., Bosetti, G., Grigera, J., Camilleri, G., Fernández, A., Zaraté, P., Bimonte, S., Sautot, L.: UX Challenges in GDSS: An Experience Report. In: Morais, D.C., Carreras, A., de Almeida, A.T., and Vetschera, R. (eds.) Group Decision and Negotiation: Behavior, Models, and Support. pp. 67–79. Springer International Publishing, Cham (2019)
117. Salinesi, C., Gam, I.: A Requirement-driven Approach for Designing Data Warehouses. Presented at the Requirements Engineering: Foundations for Software Quality (REFSQ'06) June 1 (2006)
118. Sammon, D., Finnegan, P.: The Ten Commandments of Data Warehousing. SIGMIS Database. 31, 82–91 (2000). https://doi.org/10.1145/506760.506767
119. Schafer, J.B., Frankowski, D., Herlocker, J., Sen, S.: Collaborative Filtering Recommender Systems. In: Brusilovsky, P., Kobsa, A., and Nejdl, W. (eds.) The Adaptive Web: Methods and Strategies of Web Personalization. pp. 291–324. Springer, Berlin, Heidelberg (2007)
120. Schwabe, G., Briggs, R.O., Giesbrecht, T.: Advancing Collaboration Engineering: New ThinkLets for Dyadic Problem Solving and an Application for Mobile Advisory Services. In: 2016 49th Hawaii International Conference on System Sciences (HICSS). pp. 787–796 (2016)
121. Schwarz, R.M.: The Skilled Facilitator: A Comprehensive Resource for Consultants, Facilitators, Managers, Trainers, and Coaches. John Wiley & Sons (2002)
122. See, L., Comber, A., Salk, C., Fritz, S., Velde, M. van der, Perger, C., Schill, C., McCallum, I., Kraxner, F., Obersteiner, M.: Comparing the Quality of Crowdsourced Data Contributed by Expert and Non-Experts. PLOS ONE. 8, e69958 (2013). https://doi.org/10.1371/journal.pone.0069958
123. Shani, G., Gunawardana, A.: Evaluating Recommendation Systems. In: Ricci, F., Rokach, L., Shapira, B., and Kantor, P.B. (eds.) Recommender Systems Handbook. pp. 257–297. Springer US, Boston, MA (2011)
124. Silvertown, J.: A new dawn for citizen science. Trends in Ecology & Evolution. 24, 467–471 (2009). https://doi.org/10.1016/j.tree.2009.03.017
125. Stefanovic, N., Jiawei Han, Koperski, K.: Object-based selective materialization for efficient implementation of spatial data cubes. IEEE Transactions on Knowledge and Data Engineering. 12, 938–958 (2000). https://doi.org/10.1109/69.895803
126. Strien, A.J. van, Swaay, C.A.M. van, Termaat, T.: Opportunistic citizen science data of animal species produce reliable estimates of distribution trends if analysed with occupancy models. Journal of Applied Ecology. 50, 1450–1458 (2013). https://doi.org/10.1111/1365-2664.12158

127. Stroh, F., Winter, R., Wortmann, F.: Method Support of Information Requirements Analysis for Analytical Information Systems State of the Art, Practice Requirements, and Research Agenda. Business & Information Systems Engineering. 3, 33–43 (2011)
128. Sutcliffe, A.G., Ryan, M.: Experience with SCRAM, a SCenario Requirements Analysis Method. In: Proceedings of the 3rd International Conference on Requirements Engineering: Putting Requirements Engineering to Practice. p. 164. IEEE Computer Society, USA (1998)
129. Tan, R.B.-N.: Onling Analytical Processing Systems, https://www.igi-global.com/chapter.aspx?ref=encyclopedia-data-warehousing-mining&titleid=10720
130. Thorat, P.B., Goudar, R.M., Barve, S.: Survey on Collaborative Filtering, Content-based Filtering and Hybrid Recommendation System. (2015). https://doi.org/10.5120/19308-0760
131. Traunmueller, M., Marshall, P., Capra, L.: Crowdsourcing Safety Perceptions of People: Opportunities and Limitations. In: Liu, T.-Y., Scollon, C.N., and Zhu, W. (eds.) Social Informatics. pp. 120–135. Springer International Publishing, Cham (2015)
132. Tria, F.D., Lefons, E., Tangorra, F.: Academic data warehouse design using a hybrid methodology. Comput. Sci. Inf. Syst. 12, 135–160 (2015). https://doi.org/10.2298/CSIS140325087D
133. Tria, F.D., Lefons, E., Tangorra, F.: A Framework for Evaluating Design Methodologies for Big Data Warehouses: Measurement of the Design Process. IJDWM. 14, 15–39 (2018). https://doi.org/10.4018/IJDWM.2018010102
134. Tsurkov, V.: Large-scale Optimization. Springer Science & Business Media (2001)
135. Tuunanen, T.: A New Perspective on Requirements Elicitation Methods. Journal of Information Technology Theory and Application (JITTA). 5, (2003)
136. Twinomurinzi, H., Phahlamohlaka, J., Ojo, R.B., Mahlangu, Z.N., Masanabo, L.: Assessing the quality of the 'TurnStormer' thinkLet as a Collaboration Engineering building block for the Implementation of the Promotion of Administrative Justice Act of South Africa. Presented at the (2008)
137. Vaisman, A., Zimányi, E.: Data Warehouse Systems: Design and Implementation. Springer-Verlag, Berlin Heidelberg (2014)
138. Vanetti, M., Binaghi, E., Carminati, B., Carullo, M., Ferrari, E.: Content-Based Filtering in On-Line Social Networks. In: Dimitrakakis, C., Gkoulalas-Divanis, A., Mitrokotsa, A., Verykios, V.S., and Saygin, Y. (eds.) Privacy and Security Issues in Data Mining and Machine Learning. pp. 127–140. Springer, Berlin, Heidelberg (2011)
139. Vassiliadis, P.: Gulliver in the land of data warehousing: practical experiences and observations of a researcher. In: DMDW (2000)
140. Vreede, G.J. de: Two case studies of achieving repeatable team performance through collaboration engineering. MIS Quarterly Executive. 13, 115–129 (2014)
141. de Vreede, G.-J., Briggs, R.: Collaboration Engineering: Reflections on 15 Years of Research & Practice. Presented at the January 3 (2018)
142. Winter, R., Strauch, B.: A Method for Demand-Driven Information Requirements Analysis in Data Warehousing Projects. In: Proceedings of the 36th Annual Hawaii International Conference on System Sciences (HICSS'03) - Track 8 - Volume 8. pp. 231.1–. IEEE Computer Society, Washington, DC, USA (2003)
143. Wong, Z.: Human Factors in Project Management: Concepts, Tools, and Techniques for Inspiring Teamwork and Motivation. John Wiley & Sons (2010)

144. Yeoh, W., Koronios, A.: Critical Success Factors for Business Intelligence Systems. Journal of Computer Information Systems. 50, 23–32 (2010). https://doi.org/10.1080/08874417.2010.11645404
145. Yeoh, W., Popovič, A.: Extending the understanding of critical success factors for implementing business intelligence systems. Journal of the Association for Information Science and Technology. 67, 134–147 (2016). https://doi.org/10.1002/asi.23366
146. Zaraté, P., Kilgour, D.M., Hipel, K.: Private or Common Criteria in a Multi-criteria Group Decision Support System: An Experiment. In: Yuizono, T., Ogata, H., Hoppe, U., and Vassileva, J. (eds.) Collaboration and Technology. pp. 1–12. Springer International Publishing, Cham (2016)
147. Zook, M., Graham, M., Shelton, T., Gorman, S.P.: Volunteered Geographic Information and Crowdsourcing Disaster Relief : A Case Study of the Haitian Earthquake. Presented at the (2012)
148. Zowghi, D., Coulin, C.: Requirements Elicitation: A Survey of Techniques, Approaches, and Tools. In: Aurum, A. and Wohlin, C. (eds.) Engineering and Managing Software Requirements. pp. 19–46. Springer, Berlin, Heidelberg (2005)