Volunteered multidimensional design to the test: the farmland biodiversity VGI4Bio Project's experiment.

S. Bimonte, S. Rizzi, L. Sautot, B. Fontaine

Volunteered multidimensional design to the test: The Farmland Biodiversity VGI4Bio project's experiment

Sandro Bimonte
Irstea, TSCF
Clermont Ferrand, France
sandro.bimonte@irstea.fr

Stefano Rizzi
DISI, Univ. of Bologna, Italy
Bologna, Italy
stefano.rizzi@unibo.it

Lucile Sautot
UMR TETIS, AgroParisTech, CNRS, CIRAD, IRSTEA
Montpellier, France
lucile.sautot@agroparistech.fr

Benoit Fontaine
MNHN, CESCO
Paris, France
benoit.fontaine@mnhn.fr

ABSTRACT

Moving volunteers of VGI (Volunteer Geographic Information) from passive data producers to active data analysts in the context of Data Warehouses (DWs) and OLAP systems is an open issue. Indeed, volunteers have particular features that make existing DW design methodologies inadequate. In this paper, using a real case study concerning the farmland biodiversity, we test the methodology proposed in [5], which enables volunteers to design DW schemes. The experiments aim at answering two research questions: (i) *How can volunteered design be streamlined with respect to the methodology described in [5]?*; (ii) *To what extent does the involvement of a large number of volunteers actually improve the cubes implemented?* Our experiments confirm the adequacy of the methodology proposed in [5], but they also reveal some important limitations. Among them, we identify possible conflicts among volunteers in the first steps of the design process. To address this issue we propose a solution based on social software engineering tools, and in particular Wiki systems.

1 INTRODUCTION

Volunteer Geographic Information (VGI) has been defined as "the mobilization of tools to create, assemble, and disseminate geographic data provided by volunteers" [28]. VGI has been successfully applied in several contexts such as urban, architectural, hazards, environmental, and traffic jam domains. In these contexts, crowd-sourced data is produced by amateurs and/or professional volunteers in order to collaboratively create useful datasets, which are then handled by community experts that provide analytic services to volunteers. However, it has already been proved that, when either volunteers are not fully involved in the analytic process or the services offered to them do not fit their needs, it becomes difficult to mobilize the volunteers to collect data [29]. Therefore, to increase the possible application domains of crowd-sourced data, there is a need for extending the role of volunteers from data producers to *active data consumers*, i.e., give them the possibility to impact the design of the analytical process.

Data warehouses (DWs) and OLAP are first-class citizens within Business Intelligence technologies, and enable an effective and friendly exploration and analysis of huge datasets [16]. Warehoused data are stored in databases according to the multidimensional paradigm, based on the concept of *cube*. A cube is focused on a subject of analysis, called *fact*, quantitatively described by a set of *measures*. Measures are analyzed according to *dimensions*, which are composed of hierarchical *levels*. Values of dimensions and levels are called *members*. Aggregation operators are applied to compute measure values when facts are analyzed at coarser levels. Finally, *derived measures* are calculated based on other measures and/or dimension members, while an *indicator* associates a measure with a specific aggregation operator.

DW and OLAP are promising tools for the analysis of VGI data, as shown in [3]. However, as mentioned above, to have volunteers closely involved in the decisional process, it is not sufficient to give them an OLAP front-end to access cubes designed by others: we should let them *design their own cubes*. This goal is not trivial, since volunteers are usually non-skilled from the ICT and OLAP points of view, and is the subject of our work.

The design of multidimensional cubes has been fully investigated in several papers [24]; the methodologies are classified into data-driven (i.e., the cube schema is derived from the data sources), requirement-driven (i.e., the cube schema is derived from the users' requirements), and mixed (i.e., data- and requirement-driven approaches are combined). Query-driven approaches are a subclass of requirement-driven approaches in which the cube schema is derived from the analytical queries and reports the users ask for. In all those classical approaches, decision-makers are a few OLAP-skilled users, and they are highly committed to the project. To best of our knowledge, only in [5] multidimensional design is investigated in the VGI context. In that article, the authors present a new methodological paradigm, which we will call *volunteered design*, that allows each group of volunteers to define its cube schemes, then a centralized approach is adopted where a DW expert solves the conflicts associated to different definitions of the same cube by several groups. The authors of [5] propose to involve volunteers with different social and professional skills and from different organizations in cube design, so that the analysis requirements obtained better represent the whole volunteers community. However, the methodology described in [5] presents some limitations: (i) conflicts among the volunteers can emerge also within a single group; (ii) the prototyping phase may require multiple iterations, which makes it very difficult for volunteers and DW experts to handle them over time; and (iii) DW experts can deal with several volunteers while prototyping different cubes at the same time, which makes communicating with them quite difficult and expensive.

In this paper we test volunteered design on a real case study concerning the analysis of agricultural biodiversity in the context of the French ANR project VGI4Bio[1]. In particular, we aim at answering two main research questions:

[1] www.VGI4bio.fr

(1) *How can volunteered design be streamlined with respect to the methodology described in [5]?*
(2) *To what extent does the involvement of a large number of volunteers in requirements analysis actually improve the cubes implemented from the point of view of the analyses they support?*

The paper is organized in the following way: the methodological framework of volunteered design is described in Section 2; the proof of concept for the volunteered design methodology is presented in Section 3 using the farmland biodiversity case study, together with the lessons learned from the case study; Section 4 presents the collaborative extension of the the methodology; Section 5 describes the related work; and Section 6 concludes the paper and discusses the future work.

2 AN OVERVIEW OF VOLUNTEERED DESIGN

In this section we recall the methodology proposed by [5] for involving volunteers in cube design.

The VGI context presents the following peculiarities:

- *Non-skilled users.* The volunteers are researchers in ecology, farmers, naturalists, and managers. They are non-skilled in the OLAP paradigm (i.e., they never used database and DW/OLAP technologies), therefore, as stated in [2], they do not know the proper technical terminology to express their analysis needs in terms of cube elements.
- *Limited time.* Most participants work on the project on a volunteer basis, therefore they cannot spend too much time in communicating with DW experts to express their analysis needs.
- *Large number of volunteers.* While DW experts handle the design of a cube, they also have to consider the cubes designed by several other volunteers.
- *Volunteer groups.* Volunteers can be organized in groups to define their requirements. Groups can be defined based on their members' organization and/or enterprise, social affinities, etc.

In this context, existing DW design methodologies are not appropriate since they implicitly assume that decision-makers are familiar with OLAP concepts, they are not numerous and fully employed in the DW project, and that there is no conflict about their analysis requirements. To fill this gap, [5] presents a new design methodology, sketched in Figure 1 using a UML activity diagram. In the first phase, using the ProtOLAP approach [2], volunteers and/or groups of volunteers separately communicate to DW experts their analysis requirements. In the second phase, committers solve conflicts between cube prototypes. The overall process can be more specifically described as follows.

(1) Volunteers express their requirements in natural language, mainly in terms of the indicators and dimensions they need (*Requirement Identification*). Remarkably, the iterative and rapid process adopted allows also DW experts not skilled in the specific application domain to understand the requirements during the interviews with users. In principle, other tools (e.g., brainstorming, workshops, scenarios, case studies) might be associated to natural language to make requirement elicitation more effective, but unfortunately the adoption of most of these tools to replace natural language is difficult in presence of unskilled OLAP users.

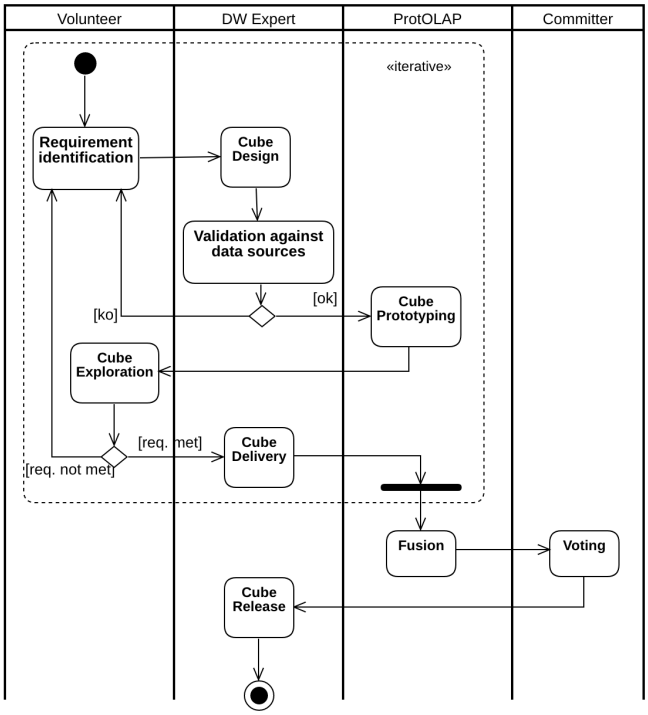

Figure 1: Methodological framework of volunteered design

(2) DW experts draw a draft conceptual schema with the ICSOLAP UML profile [7] (*Cube Design*).
(3) The prototyped DW schema is validated by DW experts against data sources using existing methodologies, e.g., [6] (*Validation against data sources*). If some pieces of data cannot be found on the data sources, then step (1) is executed again to tune the requirements.
(4) The draft schema is automatically implemented in a relational DBMS, deployed on an OLAP server, and filled with synthetic dimension members and randomly-generated measure values so as to create a cube prototype. The reason why synthetic data are used at this stage is that volunteers must be quickly enabled to "play" with the cubes to check that they are satisfactory, while implementing a real ETL process may take a lot of time.
(5) Volunteers explore the cube prototype using an OLAP client (*Cube Exploration*). If they do not agree with the way requirements were understood and implemented, another iteration is done; otherwise the cube is delivered for the next phase.
(6) A fusion algorithm is applied to all delivered cube prototypes to merge them into one or more cubes (*Fusion*).
(7) For each cube, the committers vote for each cube element to solve conflicts (*Voting*).
(8) The cube whose elements present a common agreement among committers is delivered and fed with real data through an ETL process designed and implemented by DW experts (*Cube Release*).

This methodology has been specifically conceived to support the VGI features previously described: (i) unskilled users, (ii) limited time, (iii) large number of users, and (iv) users groups. In

the rest of the paper, we experiment it in a real-world project to verify to what extent it actually supports these VGI features.

3 THE FARMLAND BIODIVERSITY CASE STUDY

In the context of the VGI4Bio project, we mobilize two VGI databases, namely Faune-Aquitaine (Biolovision database, LPO – Ligue pour la Protection des Oiseaux) and OAB (Observatoire Agricole de la Biodiversité), to build OLAP applications to analyze farmland biodiversity indicators. Faune-Aquitaine and OAB have 7682 and 1500 volunteers, respectively, who produce data. Among the possible users interested in analyzing these data, we have identified a large number of volunteers belonging to diverse categories: for instance, farmers who are interested in analyzing biodiversity data in relation with their farming daily practices, environmental NGOs needing to visualize biodiversity trends, and French public and private organizations (Regional Direction of Environment and Housing – DREAL, Chambre d'Agriculture, etc.).

An example of cube schema obtained from our data sources is called abundance, and can be used to analyze the abundance of individual species according to space, time, meteorological conditions, and altitude, in order to understand the impact of global changes on biodiversity (abundance increase or decrease, changes in range or phenology). The abundance cube schema is shown in Figure 2 using the ICSOLAP profile [7]. The cube has seven dimensions; for four of them (namely, Time, RespectOfMeteoParameters, Altitude, and Location), hierarchies are hidden in the figure inside UML packages for better readability. The cube also has one indicator that represents the sum of abundance (Sum(abundance)), and one derived measure that represents a spatial average of the abundance (abundance by location). Following the ICSOLAP profile, the Sum(abondance) indicator is not graphically connected to the fact, but it presents a tagged value (i.e., aggregatedAttribute) whose value is a measure of the cube. This cube has been implemented by two DW experts based on the analysis requirements provided by a group of three volunteers.

For the experiments conducted over these two datasets we have mobilized eight volunteers overall, five on the Faune-Aquitaine dataset and three on the OAB one. The three volunteers of the OAB dataset are organized into two groups, featuring one and two volunteers, respectively. The volunteers are all non-OLAP skilled users and they come from different organizations with different professional profiles (ecologists, farmers, and administration). Two DW experts are involved in the project, one 10-years-experienced engineer and one young engineer, who have been working half-day by week on the project for 7 months.

The following subsections focuse on the lessons learned during the most critical steps of the methodology (see Figure 1).

3.1 Cube prototyping

In order to assess the feasibility and the improvements brought by automatic prototyping, we have measured the time taken to implement the Faune-Aquitaine cubes with and without the ProtOLAP tool. We have not taken into account the time for designing the UML model since it is required in both cases. The average time for manual implementation (without ProtOLAP) is 2 hours, while with ProtOLAP it is only 5 to 10 minutes. This difference is due to the fact that in a manual implementation these kinds of errors frequently appear:

- SQL errors: wrong insert and create statements;
- Mondrian errors: wrong definition of Mondrian XML tags;
- SQL to Mondrian mapping errors: wrong usage of SQL tables and attributes in the Mondrian XML tags.

Another relevant difference between manual and automatic implementation is that the manual process is very tedious and long. Indeed, feeding the dimensions and fact tables with data, to offer volunteers to "play" with the cube and really understand the prototype, takes a lot of time.

Overall, our experiments confirm the feasibility and the benefits of the usage of ProtOLAP in our volunteered design context.

Learned lessons. From the experiments it appears that *ProtOLAP indeed makes prototyping more rapid, which allows non-OLAP skilled volunteers with limited time to participate to multidimensional design*. However, two limitations have been identified thanks to our case study:

- *Business Dictionary*. In the current implementation, the DW experts are in charge of matching the different terminologies used by different volunteers. This becomes too complex when the number of cubes increases, and the catalog of classes offered by the CASE tool used in ProtOLAP is not sufficient. For instance, differently from all other volunteers, a volunteer gave the name "cortege" to species groups. This means that this dimension level was considered different all along the process, introducing a misunderstanding, which in the end affected the quality of the designed cube and the duration of the project. So, *a unique and non ambiguous naming strategy is needed when dealing with a large number of volunteers*. Therefore, the implementation of Business Dictionary must be provided, for example using the Semantic of Business Vocabulary and Rules standard. A more flexible solution, that does not oblige DW experts and committers to create a Business Vocabulary, would be the adoption of a domain-specific ontology [27]; this will bring clear advantages only when such ontology already exists, since designing and implementing it from scratch would presumably take a lot of time and resources.
- *Derived measures*. ICSOLAP, as well as all the other DW conceptual models [7], supports an explicit definition of dimensions, facts, and aggregations in a non ambiguous way. However, in real projects, derived measures —such as the abundance ratio per location— are fundamental. Unfortunately, [7] does not support a conceptual representation of derived measures. This represents an important limitation since it is very difficult to express derived measures for volunteers, and DW experts will spend too much time to achieve a correct implementation in MDX and/or SQL. *This issue causes several misunderstandings that slow the prototyping phase down, so scaling up to a large number of volunteers might not be feasible*. Some works try to provide a high level representation of derived measures. For instance, [13] integrates a formal description of complex aggregations and derived measures into OLAP operators to explain the formula computation to decision-makers, but without automating their implementation. Only [19] uses a formal languages such as OCL to express complex aggregations and derived measures, and also proposes an MDX implementation. Therefore, this approach could be integrated in [7] with some possible extension of OCL

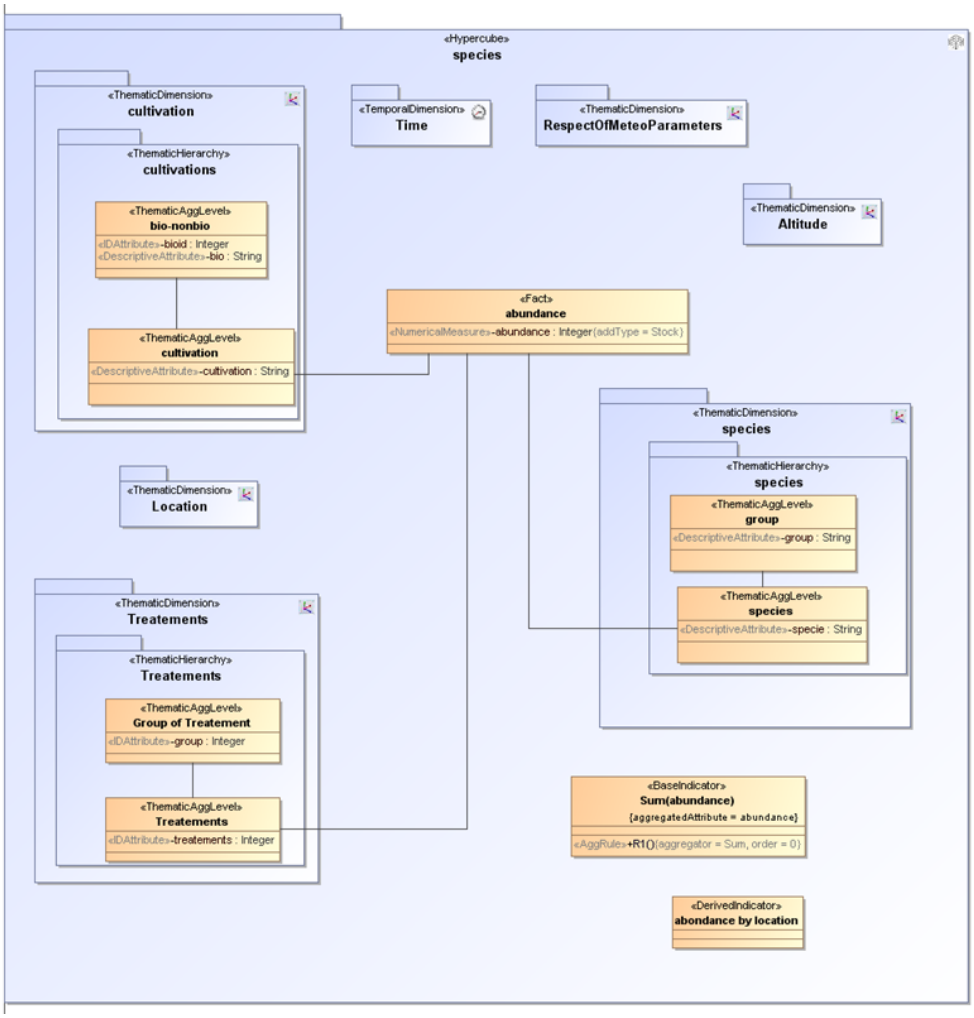

Figure 2: The abundance cube schema

statements to represent complex aggregations such as median and standard deviation as proposed in [9].

3.2 Cube exploration

In our experiments, each volunteer needs 3 to 5 meetings to obtain the final cube prototype, depending on the complexity of the cube. For example, 3 meetings were needed for the Faune-Aquitaine cubes, while more meetings were necessary for the OAB cubes since they have more dimensions and indicators than the Faune-Aquitaine ones. Each meeting took 2 hours on average.

Learned lessons. *The exploration step enables unskilled OLAP users to validate cube elements in a natural way since they are familiar with pivot tables.* However, this step still presents some problems:

- *Manual feeding.* Manually adding dimensional data in the feeding step has two main drawbacks. First, the process is tedious and long, which makes prototyping not so rapid. Secondly, when data is fed by DW experts the exploration step can be complicated for volunteers. Indeed, DW experts may use dimensional data that do not fit with real data or with the data used by volunteers. This may generate misunderstanding errors, i.e., make volunteers think that the implementation of dimensional elements is not correct. *This issue can negatively influence the ability of volunteers to validate the cubes, and slows this step down. Therefore, it could make this process unfeasible for a large number of users.*
- *Desktop OLAP client.* The OLAP client used in the ProtOLAP methodology is JRUbik. Though JRubik is a user-friendly desktop client, it must be configured by DW experts:
  - on the desktops of volunteers. This can be a complex task since a new configuration is needed for each new cube version, or it is simply impossible since DW experts may have no access to the volunteers' desktops for security reasons.
  - on the desktops of DW experts. This requires that volunteers access JRubik via a screen-sharing tool, and that DW experts are present during their exploration session. This option has been used in our case study.

  Overall, *the OLAP client desktop solution appears to be a technological barrier when several volunteers are involved in the project.*

- *Conflicts within the same group of volunteers.* Volunteers of the same group can have different visions of the same analysis need. For example, for the meteorological dimension, one volunteer may want to see all the details about rain, wind, temperature, etc. at the time when biodiversity data were collected, while another one may believe that just a Boolean attribute (stating whether the meteorological conditions defined by the data collection protocol were respected) will be enough. This issue remains open in the methodology, where conflicts within the same volunteers group are solved by them with exchanges that may cause misunderstandings. Therefore, conflicts within the same volunteers group negatively impact the quality of the designed cube and slows the overall methodology down.

3.3 Cube delivery

In this section, we study the impact of the involvement of several volunteers on the definition of requirements that are representative of the whole volunteers community. In particular, we take into account all the 11 cubes defined by 5 Faune-Aquitaine dataset volunteers.

To begin with, we have analyzed the number of dimensional elements, aggregations, and derived measures shared or not by them as shown in Figure 3, where each bar corresponds to one cube. Cubes have been designed in a temporal order; the first cube designed is c1 and the last one is c11.

The first cube (c1) defined by the first volunteer (User 1) contains only new elements, since User 1 expressed his needs first. The second cube (c2) defined by the second volunteer (User 2) contains four new elements and eight existing elements: User 2 has defined some new elements, but he has also used elements defined by User 1. In the same way, the following cubes share some existing elements, and they add new ones. After the definition of the first three cubes, the users just add a few new elements to define new cubes.

Figure 3 shows that, after the definition of the cube prototypes, the volunteers need just a few more elements to define new cubes, since thy share some elements. At the same time, each new volunteer brings some new elements and new analysis requirements.

In Figure 4, we represent the number of defined elements as a function of the number of defined cubes, and we propose an extrapolation of these data based on a logarithmic function. The extrapolation indicates that the number of needed elements increases at a slower pace than the number of new defined cubes. Therefore, according to these results, after the first prototypes, the definition of new cubes does not require the definition of many new elements.

Learned lessons. In our case study, a few volunteers were sufficient to define the core elements of the cubes; involving other volunteers improves the defined cubes but only in specific details. In other words, the methodology could effectively be used to discover analysis requirements representing the whole set of users of the Faune-Aquitaine dataset. The conception of cubes has thereby two interesting features:

- the elements commonly needed by the community of volunteers are rapidly identified by the group;
- each volunteer adds original elements and points of view.

It is also important to note that the prototyping phase is more and more rapid since volunteers share multidimensional elements, which confirms the feasibility of this step.

When the cube schemas produced respects the trend shown in Figure 4, the methodology can be applied with a large number of volunteers.

3.4 Voting

To test this phase, we used one cube and three committers with different profiles. They belong to LPO, DREAL, and AgroParisTech, and they are an ecologist, a manager, and an agronomy engineer, respectively. The cube they use concerns the Faune-Aquitaine dataset and contains 6 dimensions and 7 indicators; it is the result of the fusion step of 11 cube prototypes.

Conflict resolution took 2 hours, and led to the elimination of 3 indicators and one level. It relied on a video-conference system and was supported by the GRUS system as described in [5]. At the end of the meeting we asked the 3 committers their feedbacks on the process. They appreciated the methodology and found that it is suitable to deliver the final cubes. However, they pointed out a limitation: the methodology should allow them to modify the proposed cube by adding new elements.

Learned lessons. Conflict solving is well-suited for committers, the duration is convenient for a large number of cubes, but some effort should be done to allow committers to actively participate in the design phase.

4 COLLABORATIVE CUBE DESIGN

To overcome the previously described limitations, we have extended the methodology as follows.

4.1 Overview

To provide an effective answer to the research query *How can volunteer cubes be more effectively and efficiently designed?*, we extended [5] by adding a collaborative step, *Wiki Exchange*, to the ProtOLAP approach, aimed at streamlining volunteer-based design (Figure 5). In particular, this new step follows cube exploration and supports a collaborative validation of cubes; this is achieved by integrating the usage of a Wiki-based system into ProtOLAP.

Indeed, as previously mentioned, Wiki systems can be considered as powerful tools improving software engineering processes [17] by enabling the sharing of contents, a collaborative development, easy team communication, debugging, etc. Wikis are essentially content management systems, in the form of Web pages, which contain information that may be easily updated by their users using a simplified markup language.

In our context, associating a Wiki with an OLAP system enables (i) an easier communication between volunteers and DW experts, (ii) a faster resolution of conflicts among the volunteers of a group, and (iii) a simplified versioning of cubes during prototyping as described in the following.

The collaborative requirement validation step is implemented using a Wiki system that is integrated to the automatic and iterative implementation of the cube prototypes. Figure 6 shows, using the notation of a UML class diagram, the data model we have used to achieve this integration.

Since the prototyping approach is iterative, a cube can be associated to its previous versions. For example, the cube in

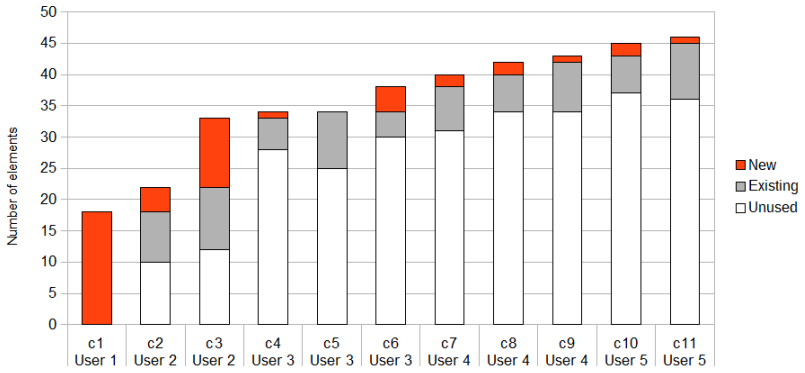

Figure 3: Definition of new elements and use of existing elements by user during the modeling process

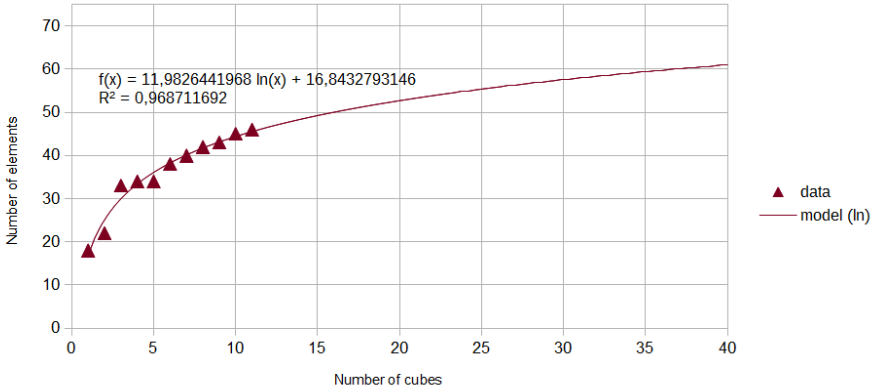

Figure 4: Extrapolation of the number of elements in function of the number of defined cubes

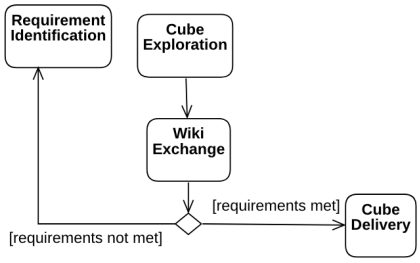

Figure 5: Collaborative extension

Figure 7 is the previous version of the one in Figure 2, where the Treatements hierarchy is defined using the Treatement level.

Volunteers explore the cube prototype using the pivot table provided by the OLAP client. Each pivot table is an OLAP query uniquely identified by its group-by set and measures; thus, a requirement corresponds to an OLAP query. When dealing with a problem about a requirement, two situations may arise:

- the problem cannot be solved due to technical DW/OLAP limits, or to data sources configuration;
- the problem can be solved, then a new version of the cube is implemented and proposed to the volunteers.

Solving a requirement problem may imply several explorations (i.e., OLAP queries) over several cube versions. We name *requirement tests* these OLAP queries. The collaborative step is then implemented by a Wiki page with a discussion associated to each OLAP query. A discussion can contain several comments. In this way, the DW experts and the volunteers of a group can easily communicate about the reasons of the problem and its possible solutions. Wiki discussions are organized in namespaces (i.e., directories) in the Wiki system according to the requirement test they refer to.

An example is shown in Figure 8, where a volunteer defines an OLAP query, then he states on the Wiki that the list of treatments is not sufficient and that he needs a coarser level describing the type of treatment.

At this time a requirement test about this problem is created (Figure 9). The DW experts take into account the comment and prototype a new cube with the new hierarchy. The volunteers explore the new cube (Figure 10) and create a new discussion to communicate to the DW experts that the hierarchy is now correct.

Note that, in Figure 9, the Wiki home page presents the list of solved and ongoing requirement tests. For example, for the requirement test “test5”, the Wiki page shows the two discussions associated with the two previously described OLAP queries.

Using a Wiki also provides another important benefit in the VGI context: *it enables an easier cooperation among the persons involved in design*. Indeed, volunteers and DW experts can provide their comments on the Wiki at any time and location, which is mandatory in such kind of projects where volunteers are geographically distributed and their agendas are not dependent from

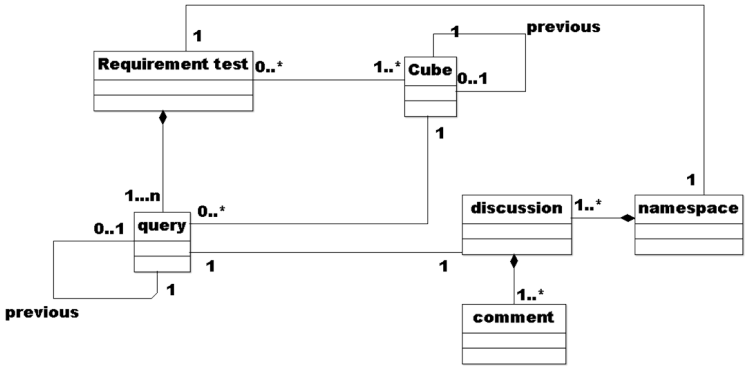

Figure 6: Data model for collaboration

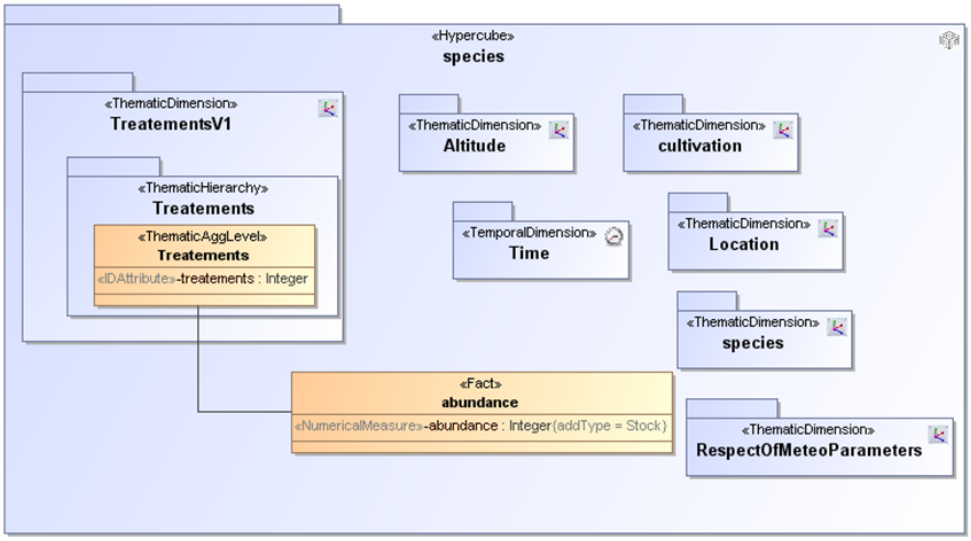

Figure 7: Previous version of the abundance cube of Figure 2

the DW project. This is possible since both the OLAP client and the Wiki system are web-based technologies.

Finally, both systems are simple and user-friendly, which allows non-ICT and non-OLAP skilled volunteers to easily use them.

4.2 Implementation

The target relational DBMSs used for data storage are Postgres and Oracle, while the OLAP server is Mondrian. The OLAP client used is JPivot, a web-based OLAP client offering tabular and graphical displays to visualize the results of OLAP queries. The Wiki system has been implemented with DocuWiki[2], deployed on an Apache server. A template of the Wiki discussion page is instantiated by means of a Java servlet each time an OLAP query is chosen by a volunteer. The name of the discussion page is generated using a unique ID build from the multidimensional elements of the OLAP query. Those elements are retrieved by a Java servlet that parses the XMLA[3] response of the OLAP Server Mondrian. Indeed, JPivot uses the XMLA web service to communicate with Mondrian.

An automatic implementation must also provide a complex configuration of the OLAP client and server inside the web server. This is not a simple task since several configuration files must be set up. This hand-made configuration is usually time-consuming since trivial errors can easily appear, slowing prototyping down. To overcome this limitation, we have developed a configuration wizard that allows a free-error and instantaneous XMLA configuration of Mondrian and JPivot, and Postgres JDBC connection. The wizard only needs the paths to some directories and files.

Finally, in order to overcome the problems related to manual feeding, we have extended the feeding step with a tool that allows volunteers to upload dimensional data using their own CSV files. Volunteers upload the files, then for each hierarchy level they choose the CSV column that contains the corresponding members. In this way each level of the dimension is fed with data that is familiar to the volunteers.

4.3 Experiments and validation

Validation concerns the requirements expressed about all the different elements of the cubes, and is carried out during a meeting

[2] www.docuWiki.net

[3] XMLA is a de facto standard for OLAP, which allows querying and visualizing cubes (docs.microsoft.com/en-us/bi-reference/xmla/xml-for-analysis-xmla-reference)

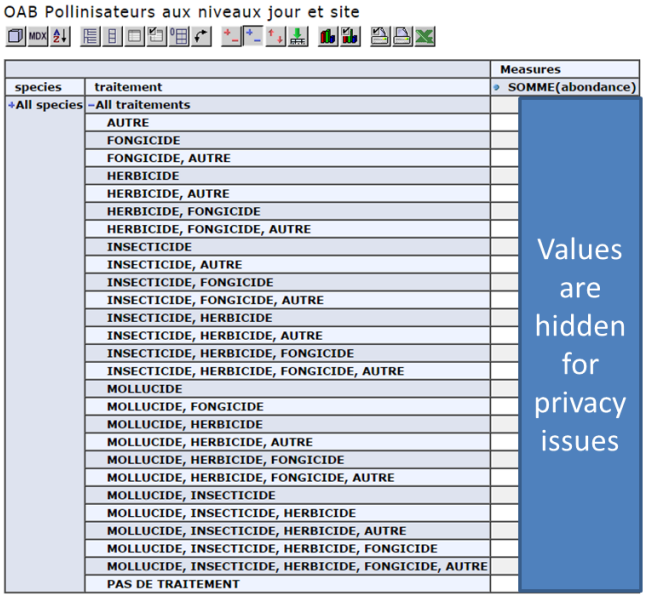

Figure 8: OLAP query with pivot table for the cube of Figure 7

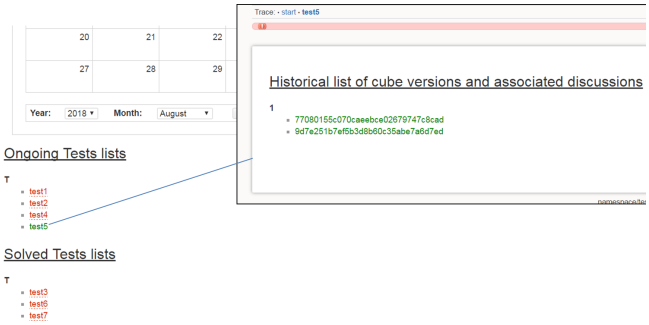

Figure 9: Organization of Wiki discussions

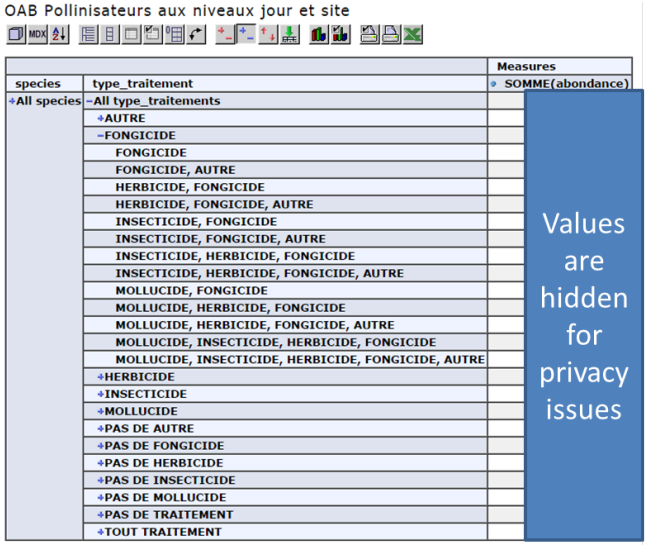

Figure 10: OLAP query with pivot table for the cube of Figure 2

in which the DW experts ask specific questions to the decision-makers. Note that, while the subsequent interactions between decision-makers and DW experts will be mediated by the Wiki, meetings are preferable for the first round of validation because decision makers are non-skilled in multidimensional modeling, so they typically need explanations and an expert guide for validation. The types of cube elements are listed below with the corresponding questions: :

- **Dimensional elements**: dimensions, hierarchies, and levels.
  - *Is this dimension useful?* For example, the temperature dimension can be excluded from the model since it is not relevant for the analysis of abundance.
  - *Does this hierarchy include all the necessary levels?* For example, is it not sufficient to have the class of a agricultural treatment without a level describing the active material.
  - *Is this level needed?* For example, the week level is not useful for a description of biodiversity temporal trends.
- **Data calculation**
  - *Is this aggregation operator useful?* In OLAP clients, the measures stored in the DW are always associated with aggregation operators in order to be visualized and analyzed by decision makers. Therefore, since the cube prototype is fed with data, the volunteers can validate the aggregation operator used. For example, for abundance, the MEDIANE operator is considered useful, but MIN and MAX are not.
  - *Does this derived measure correctly implement the defined requirement?* For example, the derived measure ratio of abundance by location was not correctly implemented in the two cube versions. It was calculated as the ratio between total abundance and the number of parcels, while volunteers wanted the ratio between total abundance and the total surface of the parcels.
- **Data**: *What constraints must cube data respect?* Source data can present some integrity constraints that must be taken into account when they are loaded in the cubes [7]. For example, volunteers do not want their cubes to include butterfly abundance data recorded outside a predefined itinerary.
- **Nomenclature**: *Are the names used for aggregations, derived measures, and dimensional elements well-defined?* For example, volunteers want to visualize in the OLAP client the sum of abundance as SUM(abundance) and not as sum abundance, since they find it more understandable.

To assess the benefits offered by the Wiki system, we have used it with one group composed of two volunteers for the OAB dataset. This group needed only three meetings to achieve the final prototype, differently from the other volunteer group of the OAB which needed six meetings. Therefore, it appears that the usage of Wiki reduces the number of needed meetings. Note that, in the general case, groups are composed by different persons with different analysis needs, so it is quite difficult to properly evaluate to what extent the results obtained by the different groups are the same. However, we mention that in our case study the cube schemes obtained by the two groups only differ in one dimension and one indicator. Figure 11 shows the number of discussions related to the elements of the cube that the group of volunteers created. We observe that they concern all the previously described requirements.

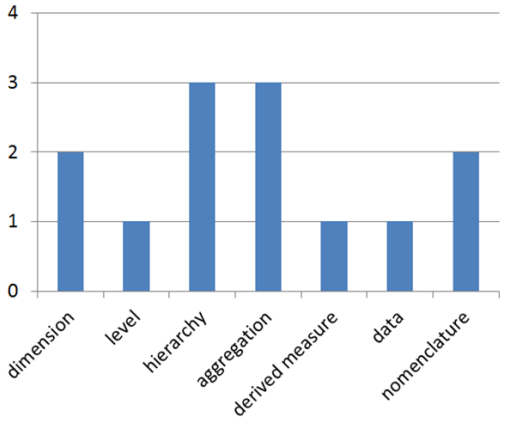

Figure 11: Number of requirements addressed with the Wiki by type of requirement

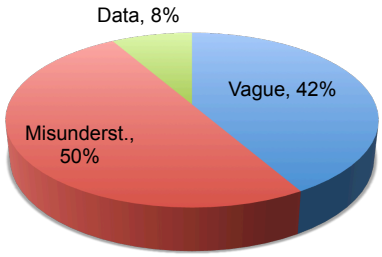

Figure 12: Error types

Finally, we have asked the DW experts to classify each discussion according to the cause of the error:

- "data" when the problem is related to the data fed in the cube;
- "misunderstanding", when the problem is that the DW experts have not well understood the requirement expressed by the volunteers; and
- "vague", when the error is due to the fact that the requirement has not been precisely defined by volunteers themselves.

From Figure 12 it appears that misunderstandings are the main class of problems, which confirms the need for a prototype-based methodology. A total of 13 errors were identified. We have also found that 63% of the discussions are used to solve a problem on a requirement without having to create a new cube version to be explored by volunteers.

Learned lessons. Experiments confirm that the easy communication support provided by the Wiki really enhances the exchange among DW experts and volunteers and avoids useless meetings. However, in our case study volunteers had to be trained in order to be able to use the Wiki system. Some volunteers did not use the Wiki since they considered it too complicated and they preferred the classical email exchange. This means that the Wiki interface must be made even more friendly to be adopted by the whole community of volunteers.

5 RELATED WORK

In this section we discuss related work concerning DW design methodologies (Section 5.1), testing DW (Section 5.2), and finally social tools for software engineering (Section 5.3).

5.1 Data warehouse design

Several methodologies have been developed in literature for the design of DWs [12, 24]; they can be grouped into three classes:

- *data-driven*, which analyze the data sources schemata to deduce numerical attributes that can be used as measures, and associated tables that can represent dimensions (such as [15]).
- *requirement-driven*, which derive a multidimensional schema from the analysis requirements that are formalized using ad-hoc or standard formalisms (e.g., [8, 22]). When requirements are expressed as analytical queries (SQL or MDX) and reports the users ask for, the term *query-driven* is used [25].
- *mixed*, which combine data- and requirement-driven approaches in that they validate the multidimensional schema derived from data sources over the analysis requirements (for example [6]).

Requirement-driven methodologies are based on the assumption that analytical requirements are well-defined by decision-makers; clearly, this gives a strong importance to the adoption of an effective elicitation requirement method. Requirements elicitation is the practice of collecting the requirements of a system from users, customers, and other stakeholders. Requirements elicitation is non-trivial. Requirements elicitation practices include interviews, questionnaires, user observation, workshops, brainstorming, use cases, role playing, and prototyping. Elicitation of requirements for DWs uses a set of tools (scenario, prototype, etc.) for helping DW experts to communicate with decision-makers [23]. To the best of our knowledge they also suppose that all the requirements expressed by decision-makers do not present any conflicts. In this context, only [5] studies conflicts among the requirements of different groups of decision-makers that have been already elicited and validated. However, [5] does not address conflicts that can emerge during the elicitation phase for a group of volunteers.

When decision-makers are non-skilled decision-makers, also rapid prototyping proved to be an effective tool to support requirement elicitation and design. Some works propose agile methodologies for the development of DWs using prototypes [14][2]; in particular, the usage of automatic prototype implementation from conceptual multidimensional schemes has been investigated.

5.2 Data warehouse testing

Classical DW development includes design phase, as described in the previous section, and then a functional and non-functional testing phase. Functional testing is devoted to finding out problems of requirements implementations. Some approaches have been developed to lead tests on DWs (e.g., [11, 20]); a survey is proposed in [14]. These methodologies are based on quantitative metrics to test the multidimensional schemata, the nomenclature, and the warehoused data [10, 26]. Besides these metrics, classical testing techniques have been also used, such as testing-in-the-small and stress tests [14]. However, to the best of our knowledge, no work propose a test phase done by decision-makers by using collaborative tools, differently from other computer science

domains where social tools have been quite successfully investigated, as described in the next subsection.

5.3 Social tools for software engineering

Social tools for software engineering (SSEs) are designed for sharing ideas, knowledge, and artifacts among groups and their members during software engineering processes. Two surveys can be found in [1, 18]. According to [1], SSEs can be considered as collaboration tools for web-based collaboration software. Web 2.0 fosters software engineering mainly using Wiki systems, for example [30] uses Wikis to enable collaborative programming. Moreover, it has been proved that Wikis are particularly useful in agile software development since they enable, among others, a fast and easy content creation [21].

To best of our knowledge only [4] proposes to associate an OLAP client to a Wiki system to allow decision-makers to share their analytical queries. The main difference with our work is that in [4] the Wiki system is not used in the design phase but for an already implemented DW, which does not require to handle cube versioning.

6 CONCLUSIONS AND FUTURE WORK

Changing VGI volunteers from passive data producers into active data analysts in the context of DWs and OLAP systems is an open issue. Indeed, volunteers have some peculiarities that make existing DW design methodologies inadequate: they are non-skilled in ICT and OLAP, they can dedicate to the project only a limited time, and they are a numerous. In this paper we used a real case study concerning a farmland biodiversity project to test the methodology proposed in [5], which allows volunteers to actively participate in the design of DW schemas.

Our experiments confirmed the relevance of the methodology proposed in [5], but also revealed some limitations. In particular, though this methodology allows volunteers to rapidly design draft multidimensional schemas, there are conflicts among volunteers in the first steps of design. To cope with this issue we have proposed a solution based on social software engineering tools, and in particular Wiki systems. Integrating a Wiki with an OLAP system allows for managing requirements validation over all the different versions of the elements of the cubes. All the remaining limitations pointed out in this paper represent our current work.

REFERENCES

[1] Navid Ahmadi, Mehdi Jazayeri, Francesco Lelli, and Sasa Nesic. 2008. A survey of social software engineering. In *Proc. ASE*. L'Aquila, Italy, 1–12.

[2] Sandro Bimonte, Elodie Edoh Alove, Hassan Nazih, Myoung Ah Kang, and Stefano Rizzi. 2013. ProtOLAP: rapid OLAP prototyping with on-demand data supply. In *Proce. DOLAP*. San Francisco, USA, 61–66.

[3] Sandro Bimonte, Omar Boucelma, Olivier Machabert, and Sana Sellami. 2014. A new Spatial OLAP approach for the analysis of Volunteered Geographic Information. *Computers, Environment and Urban Systems* 48 (2014), 111–123.

[4] Sandro Bimonte and Myhoung Ah Kang. 2013. WikOLAP Integration of Wiki and OLAP Systems. In *Encyclopedia of Business Analytics and Optimization*. 1–5.

[5] Sandro Bimonte, Amir Sakka, Lucile Sautot, Pascale Zarate, Guy Camilleri, and Aurelien Besnard. 2018. A Volunteer Design Methodology of Data Warehouses. In *Proc. ER, to appear*.

[6] A. Bonifati, F. Cattaneo, S. Ceri, A. Fuggetta, and S. Paraboschi. 2001. Designing data marts for data warehouses. *ACM Transactions on Software Engineering Methodologies* 10, 4 (2001), 452–483.

[7] Kamal Boulil, Sandro Bimonte, and François Pinet. 2015. Conceptual model for spatial data cubes: A UML profile and its automatic implementation. *Computer Standards & Interfaces* 38 (2015), 113–132.

[8] R. Bruckner, B. List, and J. Schiefer. 2001. Developing requirements for data warehouse systems with use cases. In *Proc. Americas Conf. on Information Systems*. 329–335.

[9] Jordi Cabot, Jose-Norberto Mazón, Jesús Pardillo, and Juan Trujillo. 2010. Specifying Aggregation Functions in Multidimensional Models with OCL. In *Proc. ER*. Vancouver, Canada, 419–432.

[10] C. Calero, M. Piattini, C. Pascual, and M. A. Serrano. 2001. Towards Data Warehouse Quality Metrics. In *Proc. DMDW*. Interlaken, Switzerland, 2.1–2.10.

[11] R. Cooper and S. Arbuckle. 2002. How to Thoroughly Test a Data Warehouse. In *Proc. STAREAST*. Orlando.

[12] A. Cravero and S. Sepúlveda. 2014. Multidimensional Design Paradigms for Data Warehouses: A Systematic Mapping Study. *Journal of Software Engineering and Applications* 7 (2014), 53–61.

[13] Claudia Diamantini, Domenico Potena, and Emanuele Storti. 2016. Extended drill-down operator: Digging into the structure of performance indicators. *Concurrency and Computation: Practice and Experience* 28, 15 (2016), 3948–3968.

[14] Matteo Golfarelli and Stefano Rizzi. 2011. Data warehouse testing: A prototype-based methodology. *Information & Software Technology* 53, 11 (2011), 1183–1198.

[15] B. Hüsemann, J. Lechtenbörger, and G. Vossen. 2000. Conceptual Data Warehouse Design. In *Proc. DMDW*. Stockholm, Sweden, 3–9.

[16] Ralph Kimball, Margy Ross, Joy Mundy, and Warren Thornthwaite. 2015. *The Kimball group reader: Relentlessly practical tools for data warehousing and business intelligence remastered collection*. John Wiley & Sons.

[17] Panagiotis Louridas. 2006. Using Wikis in Software Development. *IEEE Software* 23, 2 (2006), 88–91.

[18] Ioanna Lykourentzou, Foteini Dagka, Katerina Papadaki, Giorgos Lepouras, and Costas Vassilakis. 2012. Wikis in enterprise settings: a survey. *Enterprise IS* 6, 1 (2012), 1–53.

[19] Alejandro Maté, Juan Trujillo, and John Mylopoulos. 2017. Specification and derivation of key performance indicators for business analytics: A semantic approach. *DKE* 108 (2017), 30–49.

[20] A. Mookerjea and P. Malisetty. 2008. Data Warehouse/ETL Testing: Best Practices. In *Proc. Test (Test Excellence through Speed and Technology)*. New Delhi, India.

[21] Maria Paasivaara, Sandra Durasiewicz, and Casper Lassenius. 2009. Using Scrum in Distributed Agile Development: A Multiple Case Study. In *Proc. ICGSE*. Limerick, Ireland, 195–204.

[22] N. Prakash and A. Gosain. 2003. Requirements Driven Data Warehouse Development. In *Proc. CAiSE*.

[23] Naveen Prakash and Deepika Prakash. 2018. *Data Warehouse Requirements Engineering: A Decision Based Approach*. Springer.

[24] Oscar Romero and Alberto Abelló. 2009. A survey of multidimensional modeling methodologies. *International Journal of Data Warehousing and Mining* 5, 2 (2009), 1–23.

[25] Oscar Romero and Alberto Abelló. 2010. Automatic validation of requirements to support multidimensional design. *DKE* 69, 9 (2010), 917–942.

[26] M. Serrano, J. Trujillo, C. Calero, and M. Piattini. 2007. Metrics for data warehouse conceptual models understandability. *Information & Software Technology* 49, 8 (2007), 851–870.

[27] Vijayan Sugumaran and Veda C. Storey. 2002. Ontologies for conceptual modeling: their creation, use, and management. *DKE* 42, 3 (2002), 251–271.

[28] Daniel Sui, Sarah Elwood, and Michael Goodchild. 2012. *Crowdsourcing geo knowledge: volunteered geo information (VGI) in theory and practice*. Springer Science & Business Media.

[29] D. Wright, L. Underhill, M. Keene, and A. Knight. 2015. Understanding the motivations and satisfactions of volunteers to improve the effectiveness of citizen science programs. *Society & Natural Resources* 28 (2015), 1013–1029.

[30] Wenping Xiao, Chang Yan Chi, and Min Yang. 2007. On-line collaborative software development via wiki. In *Proc. Int. Symp. on Wikis*. Montreal, Canada, 177–183.