Requirements Engineering (2021) 26:43–65
https://doi.org/10.1007/s00766-020-00331-3

ORIGINAL ARTICLE

Requirements-driven data warehouse design based on enhanced pivot tables

Sandro Bimonte[1] · Leandro Antonelli[2] · Stefano Rizzi[3]

Received: 8 August 2019 / Accepted: 28 March 2020 / Published online: 11 April 2020

Abstract

The design of data warehouses (DWs) is based on both their data sources and users' requirements. The more closely the DW multidimensional schema reflects the stakeholders' needs, the more effectively they will make use of the DW content for their OLAP analyses. Thus, considerable attention has been given in the literature to DW requirements analysis, including requirements elicitation, specification and validation. Unfortunately, traditional approaches are based on complex formalisms that cannot be used with decision makers who have no previous experience with DWs and OLAP. This forces a sharp separation between elicitation and specification. To cope with this problem, we propose a new requirements analysis process where pivot tables, a well-known representation for multidimensional data often used by decision makers, are enhanced to be used both for elicitation and as a specification formalism. A pivot table is a two-dimensional spreadsheet that supports the analyses of multidimensional data by nesting several dimensions on the *x*- or *y*-axis and displaying data on multiple pages. The requirements analysis process we propose is iterative and relies on both unstructured and structured interviews; particular attention is given to enable the design of irregular multidimensional schemata, which are often present in real-world DWs but can hardly be understood by unskilled users. Finally, we validate our proposal using a real case study in the biodiversity domain.

Keywords Data warehouse · Requirements elicitation · Pivot tables · Multidimensional modeling

1 Introduction

Data warehouses (DWs) are databases specifically aimed at supporting decision makers in extracting and analyzing useful information from heterogeneous data sources, and they are widely used within both the academic and industry communities. More precisely, a DW is a subject-oriented, integrated, time-variant and non-volatile collection of data to support the decision-making process [29]. The data in a DW are organized in the form of *cubes* structured according to the multidimensional model. A cube is focused on a phenomenon of interest called *fact* (e.g., sales or invoices). The occurrences of a fact are called *events*; they are described by numerical *measures* (e.g., the quantity sold or the invoiced amount) and can be analyzed along *dimensions* (e.g., data, product and customer). Dimensions are typically described at different levels of granularity organized in aggregation *hierarchies* (e.g., customers can be aggregated by their city and by their state). Measures are aggregated along hierarchies using aggregation functions (e.g., sum, average, min and max). Cubes are normally analyzed through *OLAP* (On-Line Analytical Processing) tools using intuitive operators such as *roll-up* (which aggregates events), *drill-down* (which disaggregates events) and *slice-and-dice* (which selects subsets of events).

Nowadays, more and more data are made available thanks to new data acquisition systems and the proliferation of open data and social networks. On the one hand, the democratization of DW and OLAP tools represents an excellent opportunity for any organization/company to take advantage of these data. On the other hand, most small organizations/companies do not have experiences in DW projects, which represent a serious barrier in terms of human and material costs for the

✉ Sandro Bimonte
sandro.bimonte@inrae.fr

[1] INRAE, UR TSCF, Université Clermont, 9 Av. B. Pascal, 63178 Aubiere, France

[2] LIFIA, Facultad de Informatica, UNLP, 50 esq 120, La Plata, BsAs, Argentina

[3] DISI, Università di Bologna, V.le Risorgimento 2, 40136 Bologna, Italy

development of these projects. In this situation, users are really interested in using a DW for decision making, but their understanding of the multidimensional model and their OLAP skills are not sufficient to let them successfully lead the design of a DW.

Basically, two approaches can be followed to make multidimensional content available to decision makers. In *schema-on-write* approaches, source data are transformed, cleaned and loaded onto a physical repository (the DW); thus, they are forced to fit into a fixed multidimensional schema created at design time [29]. These approaches work perfectly well to satisfy the stationary needs of "traditional" decision makers. Conversely, in *schema-on-read* approaches, source data are left in their raw format; then, they are adapted to some (flexible and user-defined) multidimensional schema only at the time of accessing them for analyses [11, 15]. Schema-on-read approaches normally aim at satisfying the situational analysis needs of skilled users such as data scientists. In this paper, we will focus on users who are unskilled in DWs and OLAP, so we will assume that a schema-on-write approach is taken.

Several methodologies have been developed for the design of DWs in schema-on-write approaches [46]. They can be grouped into three classes: (1) *data-driven*, which analyze the data sources schemata to find numerical attributes that can be used as measures and discrete attributes that represent dimensions and hierarchies [17]; (2) *requirements-driven*, which create a multidimensional schema out of the user requirements formalized through ad hoc or standard formalisms [37, 42]; and (3) *mixed*, which combine data- and requirements-driven approaches mainly by validating the multidimensional schema derived from data sources over the user requirements [6, 16].

The more closely the DW multidimensional schema reflects the stakeholders' needs, the more effectively they will make use of the DW content for their OLAP analyses [[1, 56]. Besides, in some projects, a data-driven approach can hardly be followed because either data sources are too complex, or their structure is not known in advance, or there are too many possibly useful data sources. Thus, great attention has been given in the literature to requirements-driven approaches, which are typically structured into four steps: (1) requirements elicitation by means of interviews with decision makers; (2) requirements specification; (3) translation of the formal model of requirements into a multidimensional schema of one or more cubes; and (4) implementation of the DW.

Requirements elicitation is at the core of all requirements-driven approaches; it is the practice of collecting the requirements of a system from users, customers and other stakeholders and normally relies on techniques such as interviews, questionnaires, user observation, workshops, brainstorming, role-playing and prototyping [41]. Although several works deal with the specification and validation of DW requirements, as well as with the translation of requirements models into multidimensional schemata, only a few works focus on DW requirements elicitation [41, 42]. Interestingly, they recommend to use some classical elicitation techniques (e.g., interviews) coupled with some analyses formalisms for specification (e.g., use cases), but they do not explain precisely how to use them in the context of DW design. Besides, the classical elicitation techniques mentioned above are only adequate for DW- and OLAP-aware users.

To fill this gap, in [4] the authors introduced the idea of using pivot tables for requirements analysis to more effectively involve unskilled decision makers in DW design and proposed a preliminary sketch of a methodology that relies on this idea. In this paper, we refine and formalize the whole methodology and extend it by taking into account irregular multidimensional schemata, which are omnipresent in real-world DW projects. The conceptual foundations of our proposal are summarized in Fig. 1 using the data flow diagram notation. Our methodology relies on four iterative cycles: the first one, *goal modeling*, uses classical interviews to create a goal model for the DW; the second one, *pivot table modeling*, uses *enhanced pivot tables* (in the following, *e-pivot tables*) and semi-structured interviews to refine the goal model; the third one, *multidimensional modeling*, transforms the requirements specification into a set of multidimensional schemata for the DW; the fourth one, *implementation*, prototypes and deploys the DW. Both goal and pivot table modeling encompass requirements elicitation, specification and validation. With reference to our previous paper, in this paper we also (1) explain in detail how semi-structured interviews are made and how their questions are related to the quality of the final multidimensional schema, aimed at making our methodology repeatable and (2) assess our proposal through a set of experiments based on a real case study in the field of agriculture.

The remainder of the paper is organized as follows. Section 2 presents the main concepts of DW and OLAP and introduces the biodiversity case study related to the VG4bio project. Section 3 explains the motivation for our work. Our methodological proposal is presented in Sect. 4. Section 5 describes some experiments we have conducted to validate our methodology, and Sect. 6 describes the related work. The paper is concluded in Sect. 7.

2 Preliminaries

2.1 Background on multidimensional model and OLAP

As already mentioned, data in a DW are stored in cubes structured according to the multidimensional model. The

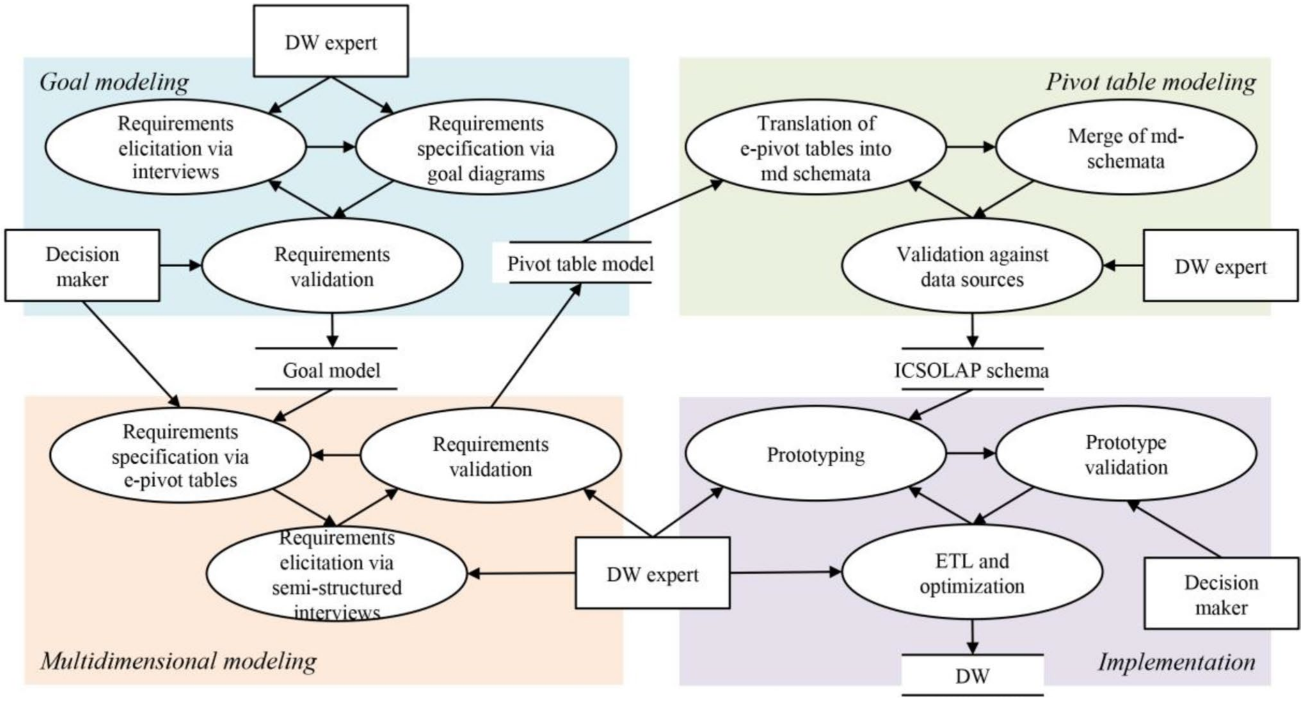

Goal modeling
DW expert
Requirements elicitation via interviews
Requirements specification via goal diagrams
Decision maker
Requirements validation
Goal model
Pivot table model
Pivot table modeling
Translation of e-pivot tables into md schemata
Merge of md-schemata
Validation against data sources
DW expert
ICSOLAP schema
Requirements specification via e-pivot tables
Requirements validation
Requirements elicitation via semi-structured interviews
DW expert
Multidimensional modeling
Prototyping
Prototype validation
ETL and optimization
Decision maker
DW
Implementation

Fig. 1 Methodology overview showing the four iterative cycles, the activities they encompass and the results they produce

basic concepts of the multidimensional model are facts, measures, dimensions and hierarchies. A *fact* is a phenomenon of interest (e.g., tweets); its occurrences, called *events*, are analyzed via some axes called *dimensions* (e.g., keyword, time and location). Each event is quantitatively described by one or more numerical *measures* (e.g., number of tweets). Dimensions are described at progressively coarser levels of spatial, temporal or thematic granularity to create aggregation hierarchies (e.g., city, region and country). The possible values of dimensions and levels are called *members* (e.g., Paris, Ile de France, France). An event is related to one member for each dimension; so, for instance, an event may state that 100 tweets were made in Paris at 9 am of today mentioning the keyword "Notre Dame."

Hierarchies allow exploring and analyzing data by aggregating measure values. Regular hierarchies are characterized by many-to-one relationships between the members of two levels, to form a balanced tree of members. A couple of a measure and an aggregation operator (e.g., number of tweets and sum) is called an *indicator*.

Cubes are explored and analyzed using OLAP tools, through which users can pose even complex queries intuitively. The most common OLAP operators are *roll-up* and *drill-down*, which let users navigate along hierarchies by aggregating and disaggregating measured values (e.g., view total sales revenues by store country and year), and *slice-and-dice*, which lets users select a subset of relevant events (e.g., view revenues for European stores only). The results of the queries obtained by applying OLAP operators are displayed by means of charts and pivot tables.

Pivot tables are a widely used representation of the results of OLAP queries [37, 44]. Basically, a classical pivot table is a two-dimensional spreadsheet with associated subtotals and totals that supports viewing multidimensional data by nesting multiple dimensions on the *x*- or *y*-axis and displaying data on multiple pages. Pivot tables permit to interactively select a subset of data and change the displayed level of aggregation.

2.2 Case study

The methodology we propose will be explained using a case study concerning the analysis of biodiversity data in the agricultural context, related to the French ANR project VGI4Bio. The VGI4Bio project aims at collecting crowdsourced data from farmers about the presence of some particular taxa (butterflies, worms, etc.) in their plots [5]. These data are integrated with the agronomic characterization of each plot, e.g., its crops and the usage of pesticides. The decision makers involved in the project belong to different domains and organizations: They are farmers, public managers, ecologists, etc. They have different skills and analysis goals, but all of them are non-skilled in OLAP and DWs.

An example of a multidimensional schema used in VGI4Bio is shown in Fig. 2 using the ICSOLAP UML Profile [7]. The «Hypercube» package stereotype defines a cube.

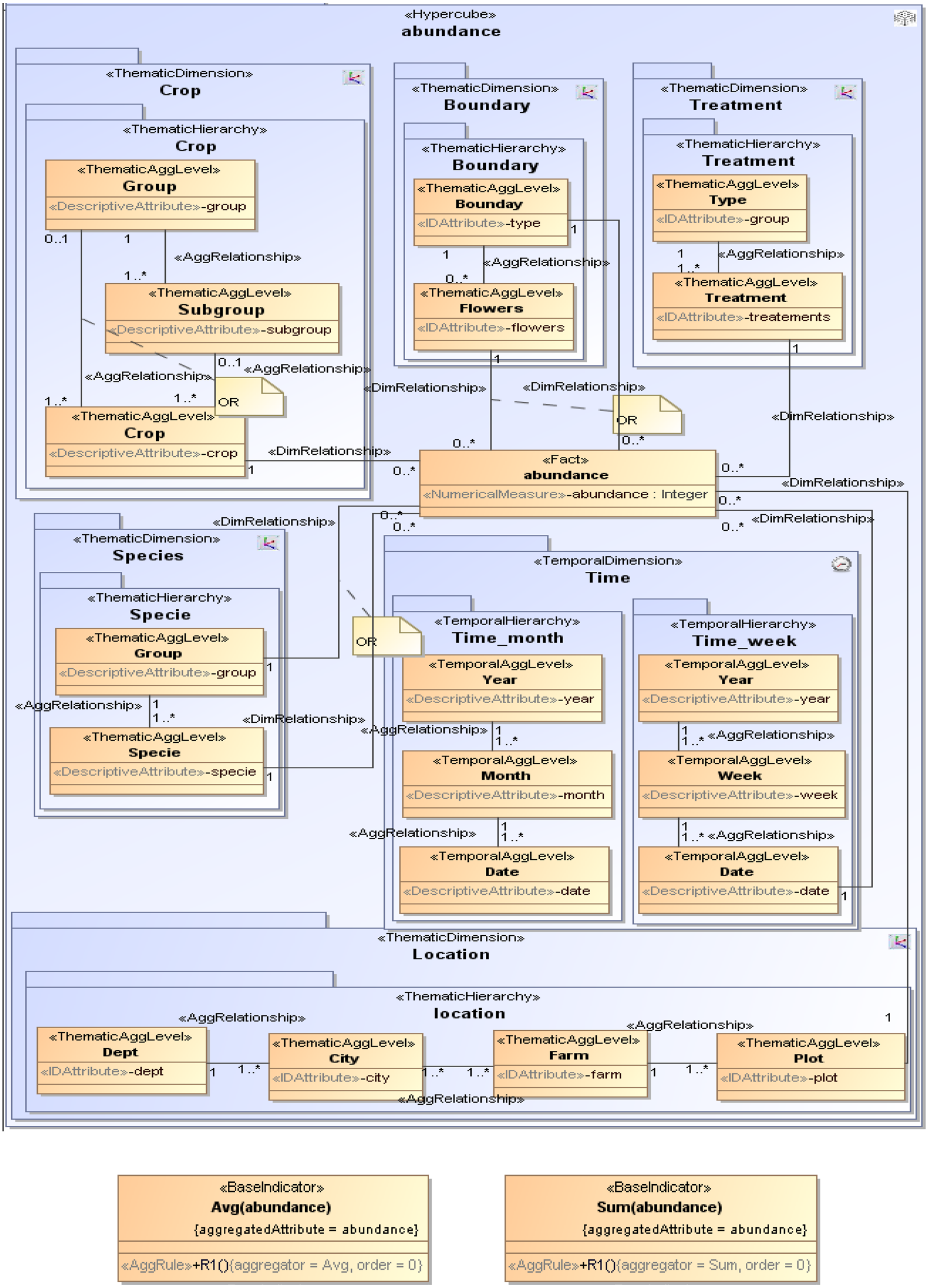
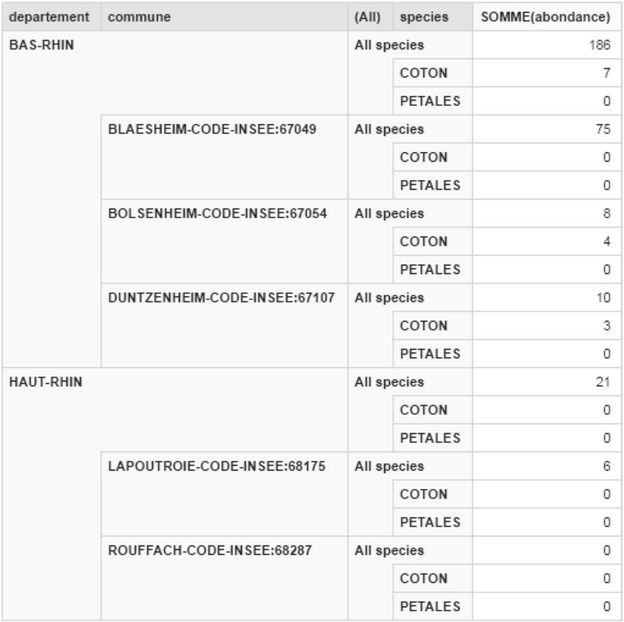


Fig. 2 ICSOLAP multidimensional schema for the analysis of biodiversity in the agricultural context

The «Fact» class stereotype describes the fact, and it can have «NumericalMeasures» and «DimRelationship» associations with 1-to-0..* multiplicity. The «NumericalMeasure» property stereotype defines a measure. The «Dimension» package stereotype represents a dimension. A dimension contains one or more hierarchies («Hierarchy» package stereotype). Different dimension types have been defined based on the type of data they represent. Thus, a dimension can be either thematic («ThematicDimension», non-temporal) or temporal («TemporalDimension»). The «Hierarchy» package stereotype contains several «AggLevel» levels, associated with the «AggRelationship» association with 1..*-to-1 multiplicity. Like dimensions, hierarchies can be thematic or temporal. The couple measure plus aggregation function is represented using a «BaseIndicator». A «BaseIndicator» defines a tagged value («aggregatedAttribute») that points to the measure of the fact, and a method called «AggregationRule» with the aggregation function used on that measure. In ICSOLAP, a hierarchy defines a unique path from the coarsest level to the finest one.

Specifically, the schema of Fig. 2 describes fact abundance and includes six dimensions: (1) location, which groups plots into farms, cities and departments; (2) crop, which represents the crop on the plot; (3) treatment, which groups the used pesticides by their type; (4) species, which represents the observed species; (5) boundary, which distinguishes between hedges (possibly with flowers) and walls; and finally the temporal dimension (6) time, on which two distinct hierarchies (date/month/year and date/week/year) are defined. The only measure is abundance and is aggregated with two aggregation functions: sum and average, so two «BaseIndicator» are defined. This multidimensional schema allows decision makers to answer queries like these: "Average abundance by species, department and city" (the result is shown in the pivot table of Fig. 3) and "Total abundance by treatment type, boundary and month."



departement	commune	(All)	species	SOMME(abondance)
BAS-RHIN		All species		186
			COTON	7
			PETALES	0
	BLAESHEIM-CODE-INSEE:67049	All species		75
			COTON	0
			PETALES	0
	BOLSENHEIM-CODE-INSEE:67054	All species		8
			COTON	4
			PETALES	0
	DUNTZENHEIM-CODE-INSEE:67107	All species		10
			COTON	3
			PETALES	0
HAUT-RHIN		All species		21
			COTON	0
			PETALES	0
	LAPOUTROIE-CODE-INSEE:68175	All species		6
			COTON	0
			PETALES	0
	ROUFFACH-CODE-INSEE:68287	All species		0
			COTON	0
			PETALES	0

Fig. 3 The pivot table resulting from a query on the schema of Fig. 2

3 Motivation

In this section, we describe the challenges we had to face in designing our methodology, which are mainly related to decision makers (Sect. 3.1), DW experts (Sect. 3.2) and multidimensional modeling (Sect. 3.3).

3.1 Decision makers

As already mentioned, decision makers often have little or no experience in OLAP and DW, and even little knowledge of information systems in general. For this kind of decision makers, who are not able to express their requirements in abstract terms, classical elicitation tools are not adequate. To understand how the cubes they are contributing to design will look like, these users need examples [3, 10, 24, 25]. They also have problems in understanding all the possibilities (and limitations) offered by OLAP analyses, so they can hardly express their business goals in multidimensional terms. In this situation, defining a "good" multidimensional schema may take several meetings with DW experts [5].

In such an uncertain scenario, decision makers may also express requirements that cannot actually be implemented on current OLAP tools or cannot be validated on data sources. For instance, data may be missing for some level or two source tables may not have an explicit relationship to build a well-defined hierarchy [5]. An early prototyping phase [3] and a method to validate the prototype on data sources [17] are very beneficial in this setting to confirm the feasibility of the implementation before the costs for error correction become too high.

Other key factors in reducing the time needed to converge to a proper multidimensional schema by reducing the number of errors are recognized to be a strong commitment of decision makers in the project and the adoption of an iterative methodology [19].

Finally, taking into account quality metrics in database and DW design has been investigated in the literature to some extent [52]; specifically, metrics related to DW usability [18], accessibility, correctness, etc. [10, 40] have been defined. While these quality metrics are normally used to evaluate a DW *a-posteriori*, we argue that they can also actively support the elicitation process, and particularly they can guide decision makers in expressing requirements that correspond to their analysis needs.

To sum up, on the decision makers' side, the requirements methodology should:

- Support the formalization of OLAP example queries;
- Support early and rapid prototyping of multidimensional schemata and their validation;
- Let decision makers be significantly involved in the project;
- Support design iterations to enable a more rapid convergence to correct multidimensional schemata;
- Take quality metrics into account to guide decision makers in properly expressing their requirements.

3.2 DW experts

Once the requirements have been elicited, they must be formalized and translated into a multidimensional schema. This is a complex process and is entirely based on the design skills of DW experts. An automatic translation of requirements into multidimensional schemata is needed [2, 7] to speed up this process and make it less prone to the errors introduced by DW experts.

Even the implementation of the multidimensional schema onto the OLAP tool and the underlying DBMS requires significant technical skills [3]. Thus, a tool for the automatic implementation of a cube from its multidimensional schema is also recommendable. This tool should support rapid prototyping, iterations and early validation.

Finally, an effort should be made to effectively support the communication between decision makers and DW experts during all steps of the methodology by defining a clear and flexible design workflow.

To sum up, on the DW experts' side, the requirements methodology should:

- Automate multidimensional modeling from requirements;
- Automate prototype implementation from multidimensional schemata;
- Provide clear guidance to the experts and allow flexibility in terms of number and type of iterations.

3.3 Irregular multidimensional schemata

Sometimes, in real-world projects, decision makers express requirements that correspond to cubes characterized by irregular structures [53] which go well beyond classical multidimensional schemata.

As mentioned in Sect. 2.1, hierarchies are normally characterized by a many-to-one relationship between the members of two subsequent levels. For example, one city belongs to exactly one department, and one department includes several cities (Fig. 4). These are called *strict hierarchies*.

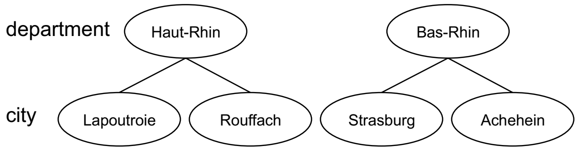


Fig. 4 A strict hierarchy, i.e., one only including many-to-one relationships

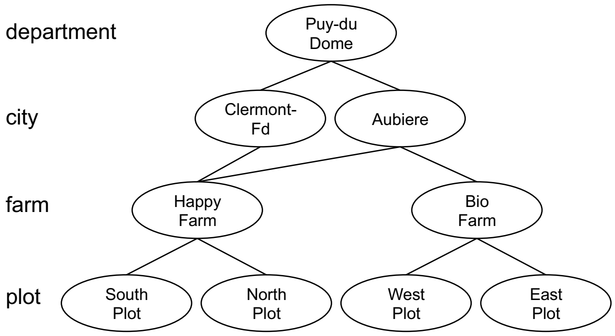


Fig. 5 A non-strict hierarchy, i.e., one also including many-to-many relationships

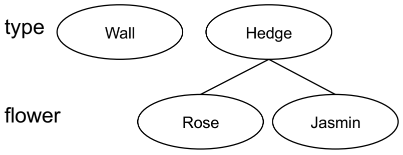


Fig. 6 A non-onto hierarchy, i.e., one whose finest level may have no members

However, in real-world settings also *non-strict hierarchies* characterized by many-to-many relationships may exist: for example, as shown in Fig. 5, a farm can be geographically located in two cities. Non-strict hierarchies are modeled in ICSOLAP using «AggRelationship» associations with 1..*-to-1..* multiplicity (see Fig. 2). They are known to possibly cause aggregation problems due to double counting [31], and some methods to transform them into strict hierarchies have been proposed [32].

A *non-onto hierarchy* is one where the finest level does not always present a member, so some events are directly related to a coarser level. For example, wall boundaries do not have flowers, so member "Wall" of level type in the boundary dimension does not have child members at the flower level (Fig. 6). As shown in Fig. 2, in ICSOLAP a non-onto hierarchy is represented using an «AggRelationship» association with 0..*-to-1 multiplicity, plus a direct «DimRelationship» between the coarser level and the fact; moreover, an OR constraint is added between the two «DimRelationship» associations of the same

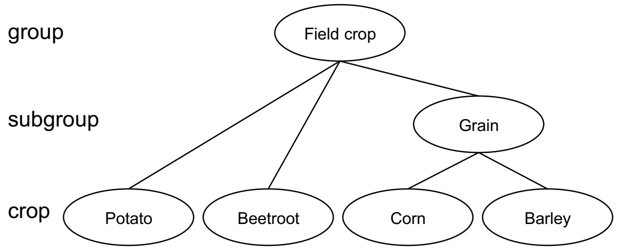


Fig. 7 A non-covering hierarchy, i.e., one where some members are missing in intermediate levels

dimension. Note that, at the logical level, non-onto hierarchies are managed by adding a dummy member in place of the missing child members.

Sometimes, members may be missing in other levels as well, giving rise to so-called *non-covering hierarchies*; for instance, a crop may belong to no subgroup (see Fig. 7). Dummy members are also used for implementing non-covering hierarchies. Non-covering hierarchies are represented in ICSOLAP using two «AggRelationship» associations with 1..*-to-0..1 multiplicity from the finer level, and adding an OR constraint (Fig. 2).

Finally, some particular relationships can exist between the fact and the dimensions. *Many-to-many fact-dimensions relationships* occur when some events are associated with multiple members of the same dimension; for example, a farmer can use two different treatments on the same day. In ICSOLAP, this is represented through a «DimRelationship» with 0..*-to-1..* multiplicity between the fact and the finest level (Fig. 2). An implementation solution here consist in creating a new dummy level whose members are lists of pesticides.

Another particular fact-dimension relationship is known as *multigranular fact*. A fact is multigranular when its events may be related not to a member of a dimension, but to a member of some coarser level. For example, when a farmer is not able to recognize a specific species, she relates the event to the group of species only. In ISCOLAP, this is represented by introducing an additional «DimRelationship» association between the coarser level and the fact, yielding an OR constraint (Fig. 2).

To wrap up, real-world OLAP applications need complex and often irregular multidimensional schemata, which can be easily represented using advanced models such as ICSOLAP but require specific implementation solutions. Since these solutions may introduce significant differences between the original schema and the logical one, decision makers can easily lose their interest in the project believing that it will not satisfy their needs.

4 The methodology

In this section, we present an overview of our methodology; then, in the subsequent subsections, we explain in detail each of its steps.

4.1 Overview

As sketched in Fig. 1, our methodology includes four iterative cycles as explained. Figure 8 shows the details of the different steps using a UML activity diagram.

0. *Tutorial* This preliminary step consists in presenting some existing OLAP applications to the decision makers and explains to them the main concepts of DW and OLAP. A web-based OLAP client is used to let decision makers “play” with these applications. This step normally takes 2–3 h. This tutorial is not meant to teach decision makers how to use an OLAP tool, but only to give them a glimpse of analysis possibilities inherent to the OLAP paradigm. Thus, in the end, decision makers do not have a thorough understanding of OLAP, which they will incrementally acquire during the next steps.
1. *Goal modeling* (**input**: interviews; **output**: goal model). This iterative step aims at creating a model that defines the analysis goals and subgoals of each decision maker. The first stage is requirements elicitation, which relies on a classical unstructured interview with decision makers, aimed at understanding their main analysis needs. The second stage is requirements specification, which uses the goal models introduced in [16]. The models obtained are finally validated by decision makers.
2. *Pivot table modeling* (**input**: goal model, semi-structured interviews; **output**: pivot tables model). This iterative step aims at refining the requirements specified in the goal models previously created. First, decision makers express detailed requirements for each goal/subgoal by drawing e-pivot tables. Basically, e-pivot tables enhance classical pivot tables by establishing a graphical convention to visualize data in irregular hierarchies. Then, DW experts use a semi-structured interview to ask some specific questions to decision makers about their e-pivot tables in order to understand their needs better, and to incite decision makers to reason about possible errors and changes to be made.
3. *Multidimensional modeling* (**input**: pivot tables model; **output**: DW schema based on the ICSOLAP profile). Each e-pivot table is automatically translated into a multidimensional schema. Then, as proposed in [16], all schemata associated with the same subgoal are merged. Finally, the overall multidimensional schema is vali-

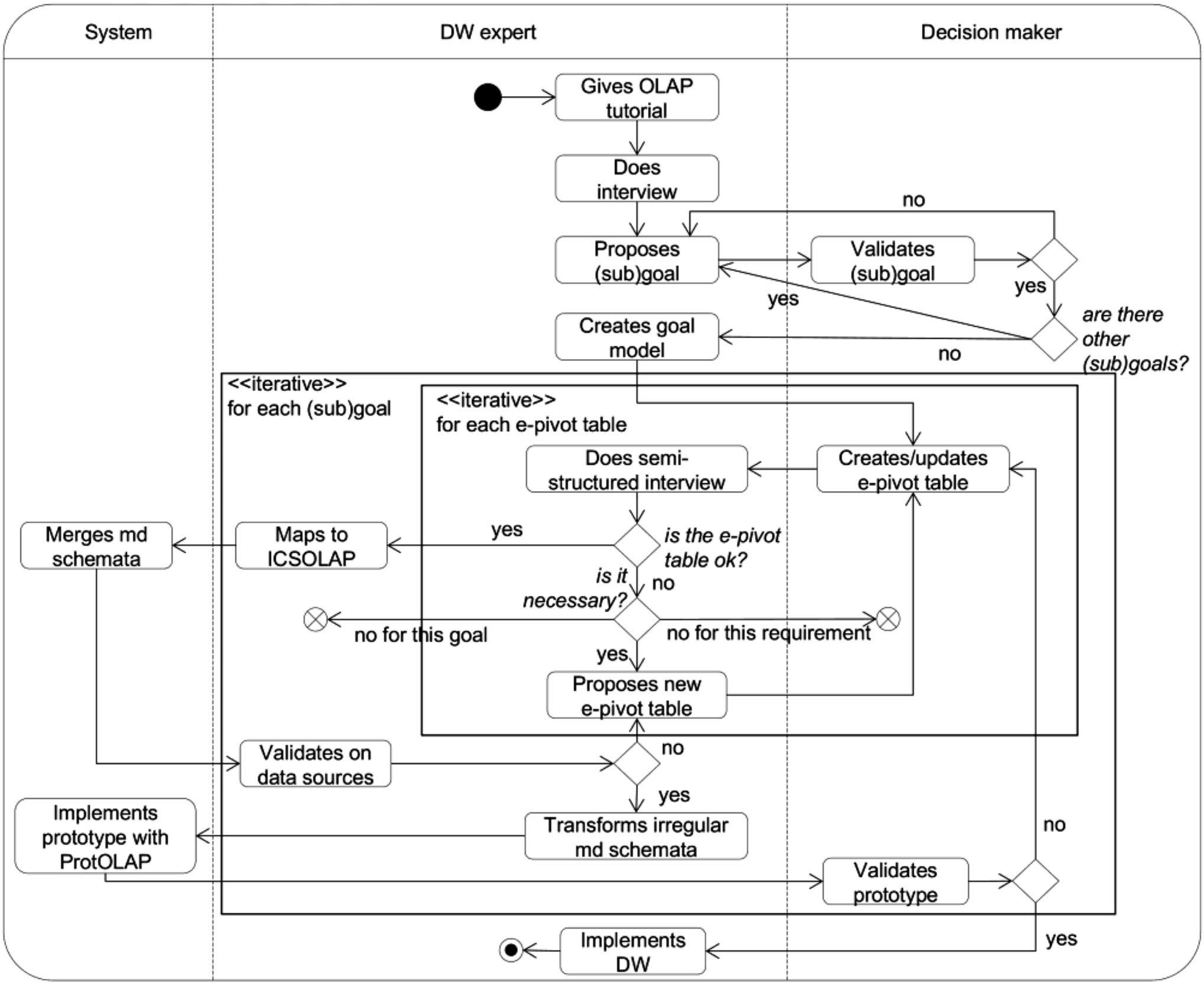


Fig. 8 UML activity diagram for our methodology

dated on the data sources; in practice, the DW experts verify that it can actually be fed with the source data.

4. *Implementation* (**input**: DW schema based on the ICSOLAP profile; **output**: SQL statements, Mondrian XML for DW prototype, and pivot tables issued from the DW prototype). The DW experts implement a prototype of the DW and show it to the decision makers. The e-pivot tables designed at step 2 are reproduced with the prototype and sent to the decision makers for validation. After the prototype has been validated, the DW implementation is completed by building the procedures that will load source data into the DW (so-called ETL, i.e., extraction, transformation and loading) and by optimizing performances [29].

4.2 Goal modeling

This step covers elicitation, specification and validation of requirements. These activities are carried out informally in some methodologies, for instance using interviews, questionnaires, workshops, etc. In other cases, they are carried out more formally, using for instance use case modeling or goal definition techniques as in [16]. Indeed, it is commonly agreed in the literature that goal definition plays an essential role in requirements-driven design methodologies to accurately capture users' requirements (see [23] for an extensive survey of the recent literature on goal-oriented requirements engineering). For this reason, to handle this step we adopt the software development methodology *Tropos* [8], which has been successfully adapted to DW design in *GRAnD* [16]. The main concepts of *GRAnD* are *actors*, which represent decision makers, and *rationale diagrams*, which represent the actor's goals, their dependencies with other actors and their AND/OR decompositions into subgoals.

Differently from what suggested in the GRAnD methodology, we skip organizational modeling and focus on *decisional modeling* and specifically on the *goal analysis* step, which considers the requirements of the DW from the perspective of the decision makers. First, all decision makers are identified and modeled as actors. Then, for each actor, her goals are analyzed and decomposed, to produce a set of rationale diagrams. Importantly, the following steps of decisional modeling according to GRAnD, namely *fact, dimension and measure analysis*, are not executed in our context because they require some conceptualization skills and a good familiarity with the multidimensional model. In practice, as discussed in Sects. 4.3 and 4.4, the gap between subgoals and multidimensional schemata will be filled in our methodology using e-pivot tables, rather than using fact/dimension/measure analysis as in GRAnD. The main reason for using e-pivot tables is actually to identify dimensions and measures in an example-driven way, without requiring a deep understanding of multidimensional modeling from decision makers. Specifically, dimensions and measures will be determined starting from the headers of each e-pivot table.

The goals and subgoals of each decision maker are iteratively defined by the DW expert and then validated by the decision maker herself, until a consensus is reached. Figure 9 shows, concerning our case study, the goal model for two decision makers. The model includes two main goals: "Biodiversity monitoring," from "Farmers and Ecologists" actors and "VGI monitoring," from "VGI Database managers" actors. The former is used to describe the ecological biodiversity phenomenon (for instance, using average abundance as an indicator). In particular, four subgoals have been defined by farmers and ecologists to analyze the spatio-temporal evolution of biodiversity: analyzing the impact of agricultural practices (such as plowing), analyzing the impact of plot management policies (e.g., using boundaries with flowers), analyzing the impact of territorial management (such as road building, schools, etc.) and finally analyzing the influence of climate changing on biodiversity (e.g., migration of species in urban areas). Using AND decomposition ensures that the "Biodiversity monitoring" goal can be achieved only when all subgoals are achieved. On the other hand, "VGI monitoring" has been specialized into two subgoals, namely "Protocol difficulties" and "Coverage." "Protocol difficulties" aim at monitoring whether the rules of the data acquisition protocol are correctly followed or not (e.g., has the data about the abundance of bees been collected monthly when the temperature was between 5 and 15 Celsius degrees?). The "Coverage" subgoal is further specialized into "Spatio-temporal coverage" and "Socio-economic coverage"; the former entails a temporal analysis of the spatial coverage in terms of data collection, while the latter concerns the profiling of volunteers (school students, young farmers, etc.). Here, OR decomposition is used to express that achieving a single subgoal can be sufficient to consider the goal as satisfied.

4.3 Pivot table modeling

This step entails a refinement of the refining requirements expressed by decision makers at the previous step; specifically, decision makers refine each goal/subgoal by drawing e-pivot tables. In fact, this step replaces fact, dimension and measure analyses of GRAnDin bridging the gap between (sub)goals and multidimensional concepts. As stated in Sect. 3.1, decision makers in real-world projects are often unskilled in DWs and OLAP. Therefore, it is very hard for them to express their analysis needs in terms of multidimensional concepts such as measures and dimensions, and even in terms of classical requirements engineering concepts such as goals, KPIs, etc.

To address this issue, some query-driven or example-driven methodologies have been devised, where queries are specified using either SQL statements [45], MDX expressions [39] or query trees [38]. Unfortunately, unskilled users are clearly unable to cope with formal languages such as SQL and MDX; conversely, as already shown in the literature [3], they can easily understand and validate the results of prototype implementations, which normally consist of pivot tables returned by OLAP tools. For example, Shimomura et al. [50] use pivot tables as prototypes to interact with users and elicit requirements. Prototypes have also been used as a tool for validating requirements [1, 28, 51]. For this reason, we propose to adopt (enhanced) pivot tables as a formalism for requirements elicitation and specification.

In the following, we give a formalization of e-pivot tables; differently from existing approaches [37, 44], our formalization supports irregular multidimensional schemata as described in Sect. 2. Besides, we model the order in which the different dimensions appear in the e-pivot table, because we argue that it is representative of the decision maker's interests.

Definition 1 *(E-Pivot table)* An e-pivot table has a schema and an instance. The *schema* consists of a sequence of level, $(l_1, \ldots, l_n)$, and one indicator, m. Each level l belongs to a dimension, $dim(l)$; the levels belonging to the same dimension are adjacent and ordered from the coarsest to the finest. The *instance* consists of a set of *pivot lines*; a pivot line is characterized as follows:

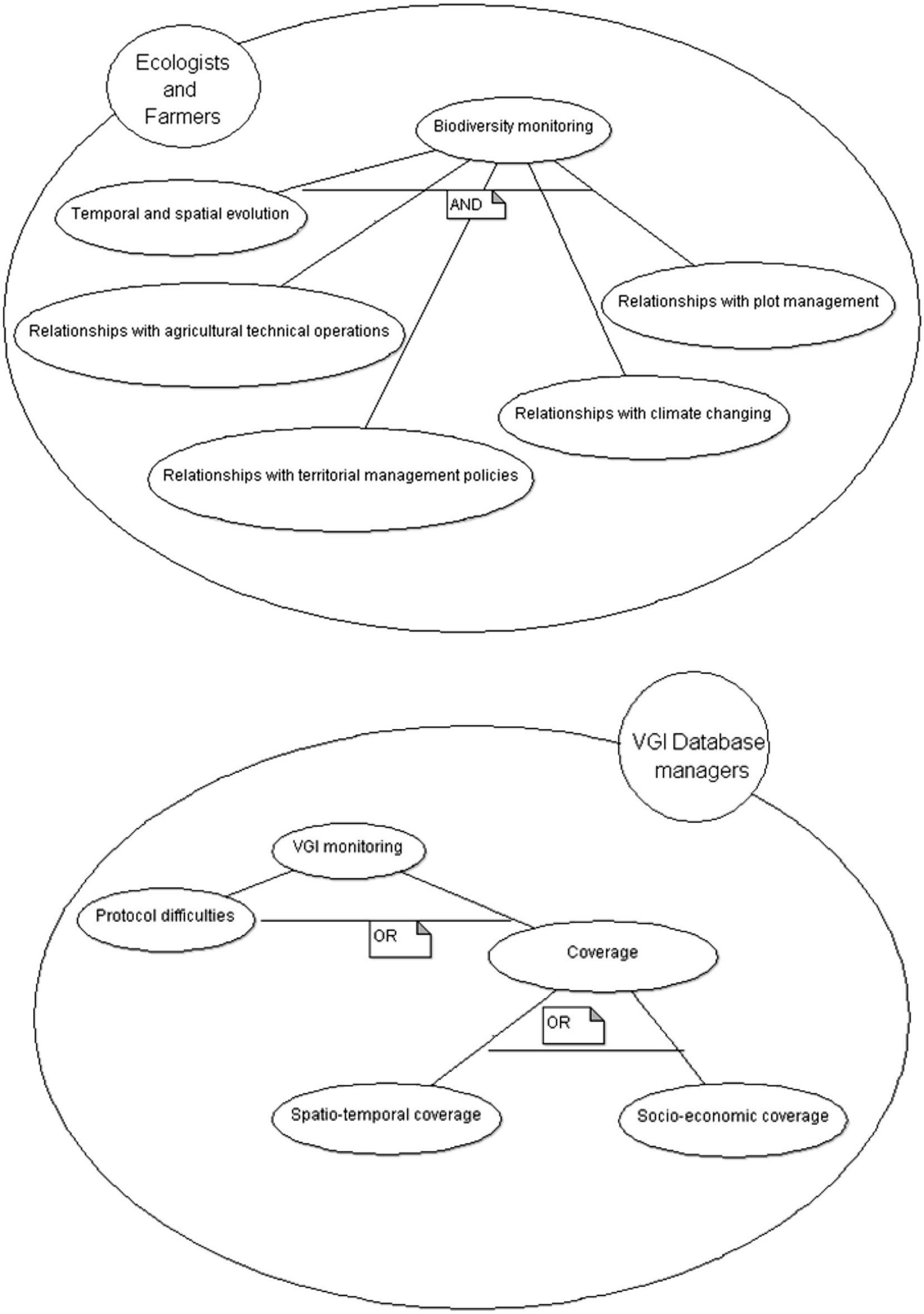
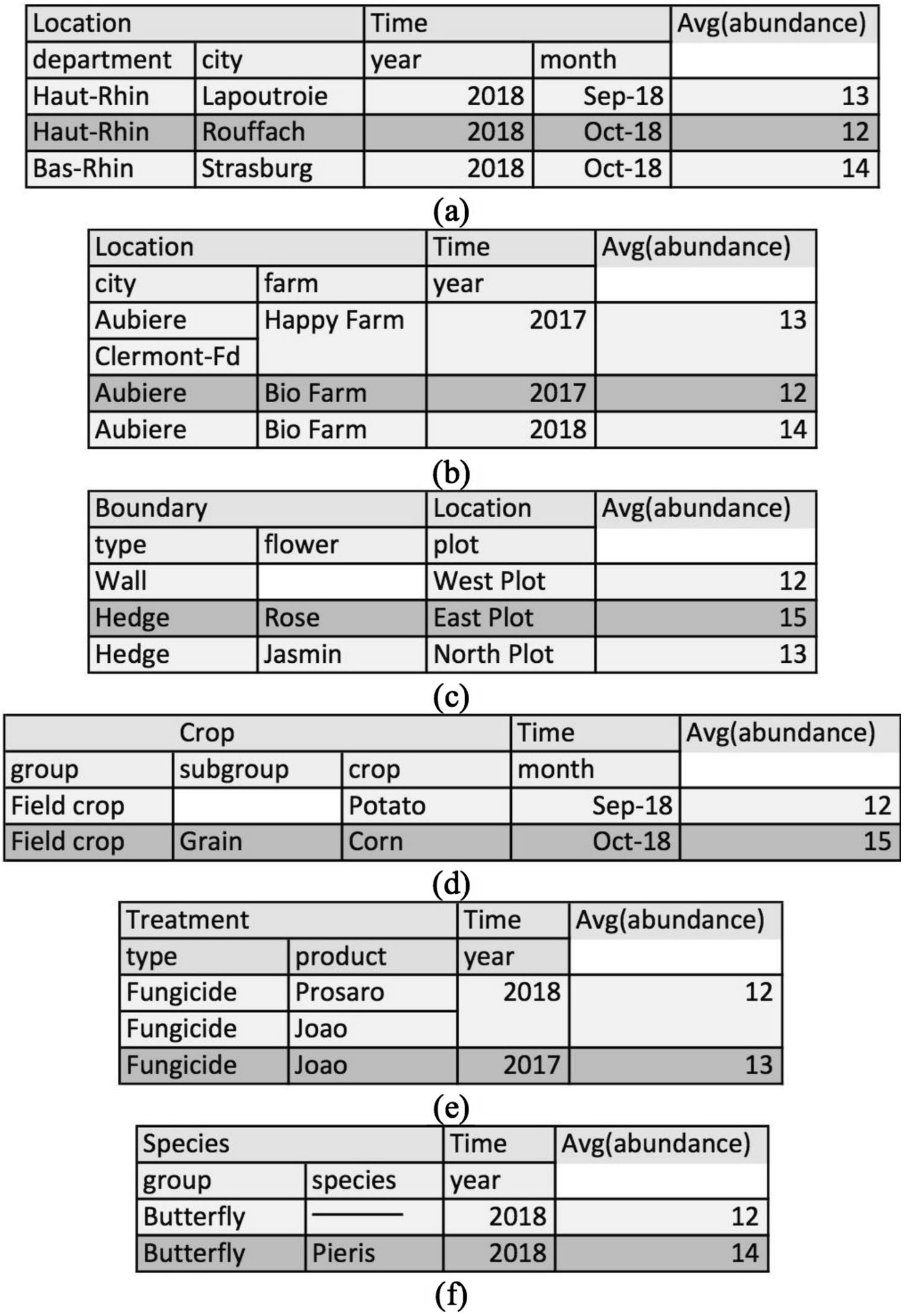

Ecologists and Farmers
Biodiversity monitoring
AND
Temporal and spatial evolution
Relationships with agricultural technical operations
Relationships with plot management
Relationships with climate changing
Relationships with territorial management policies
VGI Database managers
VGI monitoring
Protocol difficulties
OR
Coverage
OR
Spatio-temporal coverage
Socio-economic coverage

Fig. 9 Goal model for two stakeholders, showing their goals and subgoals

- It has exactly one value for the indicator *m*;
- For each level it has either a non-empty set of members, or "white" or "crossed";
- Value "white" is not admitted for the first level;
- If one level is "crossed," then all the following levels must be "crossed" too.

Fig. 10 E-pivot table representation for a strict hierarchy (**a**), a non-strict hierarchy (**b**), a non-onto hierarchy (**c**), a non-covering hierarchy (**d**), a many-to-many fact-dimension relationship (**e**) and a multi-granular fact (**f**)



Location		Time		Avg(abundance)
department	city	year	month	
Haut-Rhin	Lapoutroie	2018	Sep-18	13
Haut-Rhin	Rouffach	2018	Oct-18	12
Bas-Rhin	Strasburg	2018	Oct-18	14

(a)

Location		Time	Avg(abundance)
city	farm	year	
Aubiere Clermont-Fd	Happy Farm	2017	13
Aubiere	Bio Farm	2017	12
Aubiere	Bio Farm	2018	14

(b)

Boundary		Location	Avg(abundance)
type	flower	plot	
Wall		West Plot	12
Hedge	Rose	East Plot	15
Hedge	Jasmin	North Plot	13

(c)

Crop			Time	Avg(abundance)
group	subgroup	crop	month	
Field crop		Potato	Sep-18	12
Field crop	Grain	Corn	Oct-18	15

(d)

Treatment		Time	Avg(abundance)
type	product	year	
Fungicide Fungicide	Prosaro Joao	2018	12
Fungicide	Joao	2017	13

(e)

Species		Time	Avg(abundance)
group	species	year	
Butterfly	———	2018	12
Butterfly	Pieris	2018	14

(f)

An e-pivot table is graphically represented through a table as follows:

- The second row, called *level header*, orderly shows the names of the levels in the schema from left to right.
- The first row, called *dimension header*, shows the names of the dimensions to which each group of levels belongs; the rightmost column shows the name of the indicator.
- Each following row shows one pivot line of the instance; in case multiple members are present for some level, they are shown in separate subcells.

Remarkably, as shown in Fig. 10, our definition of e-pivot table supports the representation of all types of irregular multidimensional schemata listed in Sect. 3.3. In particular:

- A row showing multiple members for an intermediate level of a dimension represents a non-strict hierarchy (Fig. 10b);
- A row showing multiple members for the finest (rightmost) level of a dimension represents a many-to-many fact-dimension relationship (Fig. 10e);
- A white cell in the finest (rightmost) level of a dimension represents a missing value in a non-onto hierarchy (Fig. 10c);
- A white cell in an intermediate level of a dimension represents a missing value in a non-covering hierarchy (Fig. 10d);
- Crossed cells in the finest (rightmost) levels of a dimension represent a missing value in a multigranular fact (Fig. 10f).

Pivot tables normally include totals at different aggregation levels. We chose not to include them in e-pivot tables because, in our experience, they are hardly understood by non-skilled users; thus, in our approach, queries that compute totals are treated as separate e-pivot tables. For example, the query of Fig. 3 can be described by four e-pivot tables combining all the levels involved: (department, all species), (department, species), (city, all species) and (city, species).

Note that, in real-world DW projects, even in the presence of skilled users, specifying complex aggregations is a challenge, since (1) it may entail the writing of complex formulae, (2) different operators may be necessary when aggregating along the different hierarchies and even (3) different operators may be necessary when aggregating at the different levels of a single hierarchy. Thus, aggregations are normally specified and implemented by DW experts following either a textual description, or a simplified formula, or some example given by users, and they always have to be validated by users with real values during the testing phase because they are a common source of severe and hidden errors [18]. Interestingly, some methods to specify complex indicators via collaborative work have been devised recently [13]; however they are meant to be used by DW designers so they cannot be used by the unskilled decision makers targeted by our proposal. Thus, in the current version of our methodology, users can give DW experts a first hint about how indicators should be computed using their names. For instance, AVG (yearly maximum of total abundance by plot) means that the abundance measurements are summed up for each plot, then the yearly maximum is found, and finally the average is used when aggregating on all remaining dimensions. Then, the exact formula may be found through a few iterations between decision makers and DW experts and checked by the former during prototype validation in the implementation step.

As a side remark related to terminology, we observe that, in principle, choosing the right names for multidimensional concepts could be made easier by relying on a well-established domain ontology. However, to the best of our knowledge there is no single ontology for biodiversity in the agricultural context, so a complex integration between the partial ontologies available (such as AGROVOC[1]) would be required. Fortunately, in our setting decision makers are also data producers. Thus, they know very well the exact terminology to be adopted for measures, dimensions and levels. As a consequence, few relevant issues arose in mapping requirements onto the application domain model.

The role of DW experts during this phase is not only that of passively tracing the requirements expressed by decision makers in the form of e-pivot tables, but also that of actively guiding decision makers in producing good-quality requirements. In this direction, the specification of each e-pivot table is followed by a semi-structured interview whose questions are specifically aimed at addressing quality issues related to the multidimensional schema to be obtained. Similarly to [52], which provides some quality metrics (accuracy, consistency, completeness and timeliness) over databases and decisional queries, we introduce some quality features for DW schemata as summarized below:

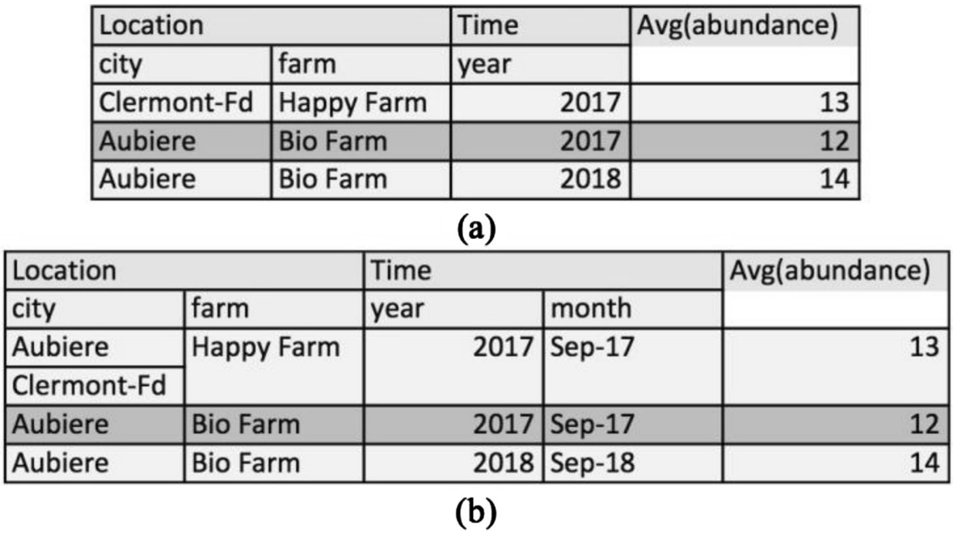
- *Completeness*: each hierarchy has all the necessary levels; the multidimensional schema has all the necessary dimensions and indicators.
- *Minimality*: there are no useless or redundant dimensions, levels and indicators
- *Correctness*: hierarchies are correctly structured, specifically with reference to irregular schemata
- *Accuracy*: the values of indicators are correctly computed.

As explained in Sect. 6.3, these quality features are related to the non-redundancy, depth of hierarchies and expressiveness metrics defined in [43] and to the completeness, minimality and aggregation correctness criteria introduced in [39].

In the following, we show the set of natural language questions derived from these quality issues, which are used as a base for the semi-structured interviews. Indeed, DW experts will complement questions with examples and suggestions during the interview.

Q1 “Are some columns missing?” (*completeness*)

[1] http://aims.fao.org/fr/agrovoc, https://www.sciencedirect.com/science/article/pii/S1574954107000362.



Location		Time	Avg(abundance)
city	farm	year	
Clermont-Fd	Happy Farm	2017	13
Aubiere	Bio Farm	2017	12
Aubiere	Bio Farm	2018	14

(a)

Location		Time		Avg(abundance)
city	farm	year	month	
Aubiere	Happy Farm	2017	Sep-17	13
Clermont-Fd				
Aubiere	Bio Farm	2017	Sep-17	12
Aubiere	Bio Farm	2018	Sep-18	14

(b)

Fig. 11 E-pivot table before (**a**) and after (**b**) validation triggered by semi-structured interview

IF YES Q1.1 “Can you suggest the name and the corresponding data for the new column?”
Q1.2 “Can you classify it in an existing group of columns?”
IF NO Q1.2.1 “What is the name of the new group?”
Q2 “Can some columns be deleted?”(*minimality*)
Q3 “Is the aggregation operator used for the numerical values correct?” (*accuracy*)
Q4 For each dimension, “could a finer level of detail be useful?” (*completeness*)
Q5 For each pair of columns of a group, “could the cell on the right be associated to multiple cells on the left?” (*correctness*, non-strict hierarchies)
Q6 For the rightmost column of each group, “could each numerical value be associated with multiplecells?” (*correctness*, many-to-many fact-dimension relationship)
Q7 For each column, “could some cells be empty?” (*correctness*, non-onto and non-covering hierarchies)
IF YES Q7.1 “What value could be shown without compromising readability?”
Q8 For each column, “could some cells be crossed?” (*correctness*, multigranular facts)

Note that there is no predefined order for these questions. They are asked to the decision makers depending on the specific element of the e-pivot table they are validating.

An example is proposed in Fig. 11. In refining subgoal “Temporal and spatial evolution” of Fig. 9, the decision maker drew the e-pivot table in Fig. 11a. After that, she went through the semi-structured interview. In response to questions Q1 and Q5, she improved the e-pivot table in terms of completeness and correctness by adding column month (a new level in dimension time) and a second city for happy farm (a non-strict hierarchy in location). The resulting e-pivot table is shown in Fig. 11b.

4.4 Multidimensional modeling

The goal of this phase is to translate each e-pivot table into a multidimensional schema and merge the resulting schemata, which may be related to the core phase of query-driven approaches. In [38], some a priori knowledge of the user requirements is assumed to be available, though no original techniques are proposed to build it, and the multidimensional schema is obtained by matching this knowledge against the query graphs provided by users. In our setting, this approach cannot be applied, since there is no a priori knowledge—e-pivot tables are the only source of requirements. In [45], users express queries in SQL over the source (relational) database; however, the algorithm that creates the multidimensional schema starting from these queries also uses information coded within the source database, so it cannot be reused in our setting because data sources are not known in advance. In [39], users express queries (in MDX) over a “general” cube which is known a priori (they assume a DW already exists) to derive the schemata of additional cubes. This cannot be done in our setting, as no previous cube schema exists.

Since previous approaches cannot be reused here, we provide our own method. Basically, the idea is to map the headers of each e-pivot table onto multidimensional concepts (namely hierarchies, dimensions, levels and indicators). Figure 12 shows the pseudo-code for translation; the resulting multidimensional schema is expressed using the ICSOLAP profile. Remarkably, choosing ICSOLAP gives the advantage to enable automated prototyping during the

```
Create «Fact» with the name of the rightmost column of the dimension header, $f$
Create «NumericalMeasure» with name $f$
Create «BaseIndicator» with {aggregatedAttribute = NumericalMeasure} and {Aggregator = the aggregation operator in the cell}
For each other column $d$ of the dimensions header
    Create «Dimension» with name $d$
    Create «Hierarchy» with name $d$
    For each column $l$ of the level header underlying $d$
        Create «Level» with name $l$
    Create «DimRelationship» between the rightmost column underlying $d$, $l_{finest}$, and «Fact» $f$
    If there is at least one row with multiple values in $l_{finest}$
        Give «DimRelationship» the cardinalities of a many-to-many fact-dimension relationship
    Else
        Give «DimRelationship» the cardinalities of a many-to-one fact-dimension relationship
    If there is at least one row with a white/crossed cell in $l_{finest}$
        Give «DimRelationship» the cardinalities of a non-onto hierarchy / multigranular fact
    Else
        Give «DimRelationship» the cardinalities of an onto hierarchy
    For each couple ($l$, $l'$) of adjacent columns of the level header underlying $d$
        Create «AggRelationship» between $l$ and $l'$
        If there is at least one row with multiple values in $l$
            Give «AggRelationship» the cardinalities of a non-strict hierarchy
        Else
            Give «AggRelationship» the cardinalities of a strict hierarchy
        If there is at least one row with a white cell in $l$
            Give «AggRelationship» the cardinalities of a non-covering hierarchy
        Else
            Give «AggRelationship» the cardinalities of a covering hierarchy
```

Fig. 12 Algorithm for translating an e-pivot table into an ICSOLAP multidimensional schema

implementation phase. We also observe that, while the order of dimensions is modeled in e-pivot tables, it is not part of the ICSOLAP specification, which is only focused on the multidimensional structures. An example of the result of the algorithm starting from the e-pivot table with a non-strict hierarchy of Fig. 11b is the ICSOLAP model of Fig. 13.

E-pivot tables encourage decision makers to think in terms of single queries rather than in terms of a global multidimensional schema. Thus, each multidimensional schema obtained by the above algorithm normally includes only a subset of the dimensions, levels and indicators that should be contained in the overall DW schema. For example, the time dimension of the e-pivot table in Fig. 13 contains only one level year, while the e-pivot table of Fig. 14 contains the month level. This means that the final DW schemata must be obtained by merging the multidimensional schemata obtained by translating all the e-pivot tables created by decision makers for each subgoal. In terms of DW design, this corresponds to the fusion of a set of multidimensional schemata, for which some approaches have been proposed in the literature. We adopt the one proposed in [47], which produces the schema shown in Fig. 15 starting from the ones in Figs. 14 and 15.

4.5 Implementation

Several approaches have been proposed for mapping requirements over data sources (e.g., [12]). In particular, [6] and [34] provide methods to validate conceptual multidimensional model over relational data sources, but they do not take into account irregular schemata as previously described. Indeed, the existing work assumes that the DW schema meets the multidimensional normal forms [35], which exclude irregular structures and assume data summarizability. Moreover, in the existing work data sources are typically represented as relational databases, since the mapping algorithms operate on primary and foreign keys. This is another important limitation, since data lakes, where different types of data and storage systems (relational, textual, NoSQL, etc.) are joint together, are increasingly present in companies.

Due to the inherent complexity of devising an automated approach for the validation of (irregular) multidimensional schemata over (heterogeneous) data sources, in this work we advocate for a manual mapping, to be performed by DW experts in collaboration with decision makers and databases administrators. When a multidimensional element cannot be found in data sources, it is removed by the DW schema, and the e-pivot table involved is sent back to the decision maker to be updated. This kind of situation was quite infrequent in

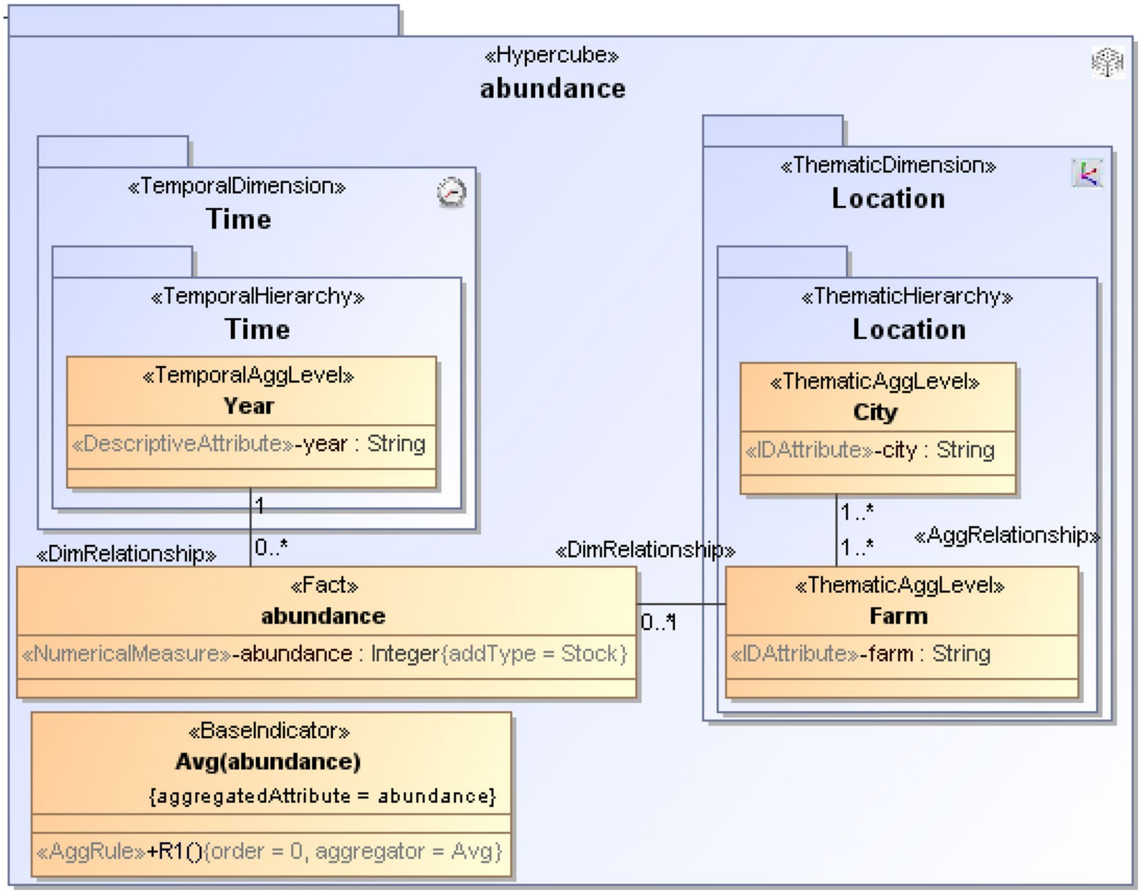


Fig. 13 ICSOLAP multidimensional schema for the e-pivot table of Fig. 11b

our context, as decision makers have good knowledge of the data sources [5].

Prototyping is one of the most used requirements elicitation and validation methods in the development of both DWs and generic software. In our methodology, once an ICSOLAP model has been obtained at the previous step, it is implemented by DW experts using the ProtOLAP tool [3]. In this way, decision makers can access their pivot tables on the OLAP client and use them to validate their requirements.

ProtOLAP takes in input a UML model defined with ICSOLAP through the MagicDraw CASE tool and automatically creates the SQL scripts for Postgres (table creation and data insertion) as well as the XML configuration files for the Mondrian OLAP server. Importantly, this step enables the quick creation of a prototype to be shown to decision makers on the one hand; on the other, it allows DW experts to check for the feasibility of implementing the DW schema on the OLAP server and the DBMS.

Before the ICSOLAP schema can be fed into ProtOLAP and prototyped, it may have to be transformed, because irregular multidimensional schemata are not natively supported by Mondrian. Some solutions have been proposed in the literature to transform irregular multidimensional schemata; these solutions have pros and cons, and the choice of the best one depends on the application domain and decision makers' needs. For example, according to [32], the following rules can be applied:

- Non-covering and non-onto hierarchies introduce dummy members in the levels that present missing data. In this way, all level cardinalities become 1-to-many, and all hierarchies become covering and onto.
- Non-strict hierarchies transform the parent level of the non-strict relationship into a new dimension, so that the many-to-many relationship is represented through the fact itself.
- Many-to-many fact-dimension relationship creates a dummy level whose members are sets of members, so that each event can be connected to a single member of this dummy level.
- Multigranular facts add a dummy member to the dimension level.

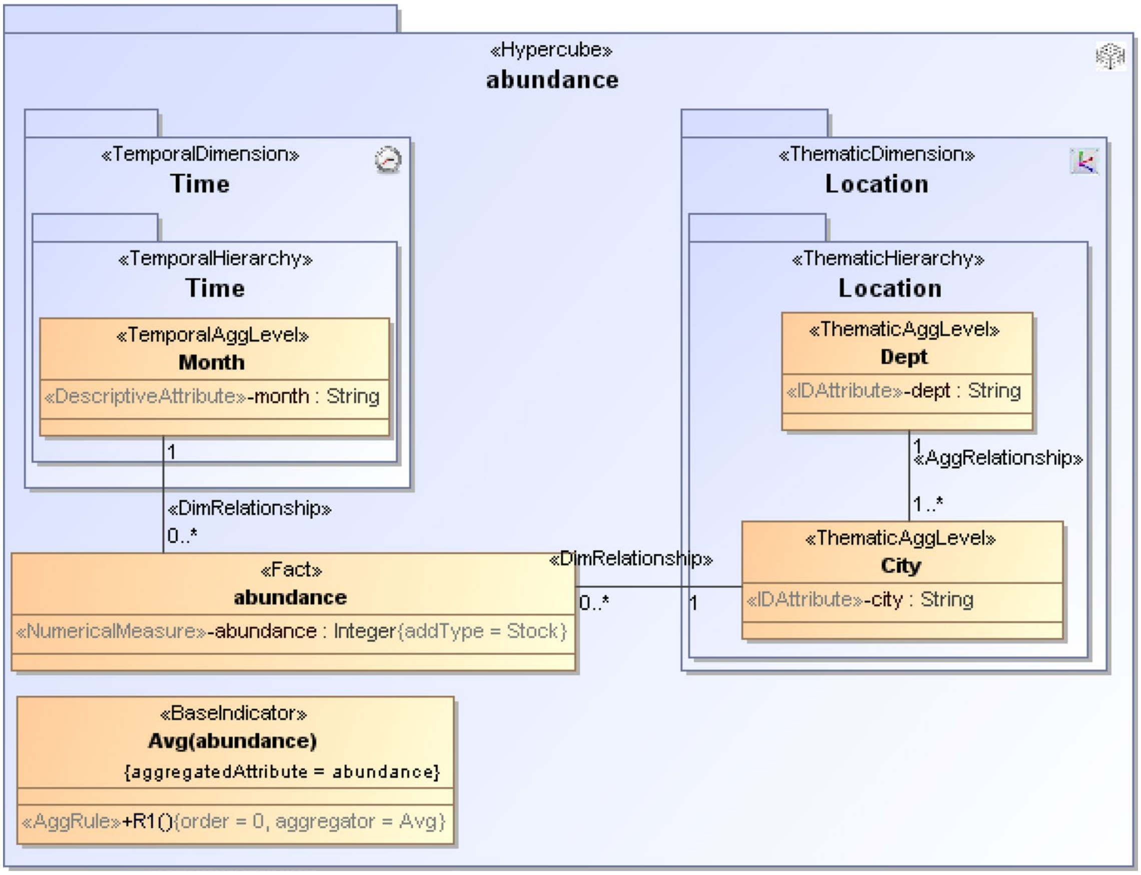


Fig. 14 Input of the fusion between multidimensional schemata

For example, for the non-strict hierarchy of the schema in Fig. 12, the spatial dimension is split into two dimensions, farm and city.

We close this section by remarking that this step primarily validates the requirements associated with a single multidimensional schema; however, when all prototypes have been built and validated, it is also used to check the completeness of requirements. Indeed, by playing with the prototypes, decision makers may find out that some e-pivot table they need cannot be obtained with the prototype, which means that some other requirement should be implemented. In this case, the workflow restarts from step 1, complementing the previously designed goal, e-pivot table and multidimensional schemata. As discussed in [5], different meetings may be necessary to finalize the DW prototype (in our case study, from 2 to 5). Note that, while the semi-structured interviews used to validate each single multidimensional schema require the participation of DW experts due to the poor skills of decision makers in OLAP, the validation of the completeness of the requirements covered by the DW prototype is up to the decision makers since it solely depends on their analysis needs. Interestingly, in our experience there is no need for DW experts to solicit the decision makers to find out more requirements; while playing with prototypes to create e-pivot tables, decision makers better understand the power of the OLAP paradigm, so they actively propose additional requirements until they consider the DW completely satisfactory.

5 Experiments

The *multidimensional modeling* and *implementation* steps of our methodology have been already evaluated in [5]. It was confirmed that the use of prototyping allows an iterative and rapid design since it speeds up the implementation time, and that decision makers find “playing” with the OLAP client very useful to validate their requirements. Therefore, in this section we focus on validating e-pivot tables as a tool for the elicitation phase, in particular on understanding to what extent they can be effectively used by non-skilled users to express their analysis needs. We have tested both the readability (the ability of users to understand a given e-pivot table) and the writability (the ability to create an e-pivot table) of e-pivot tables representing irregular multidimensional schemata.

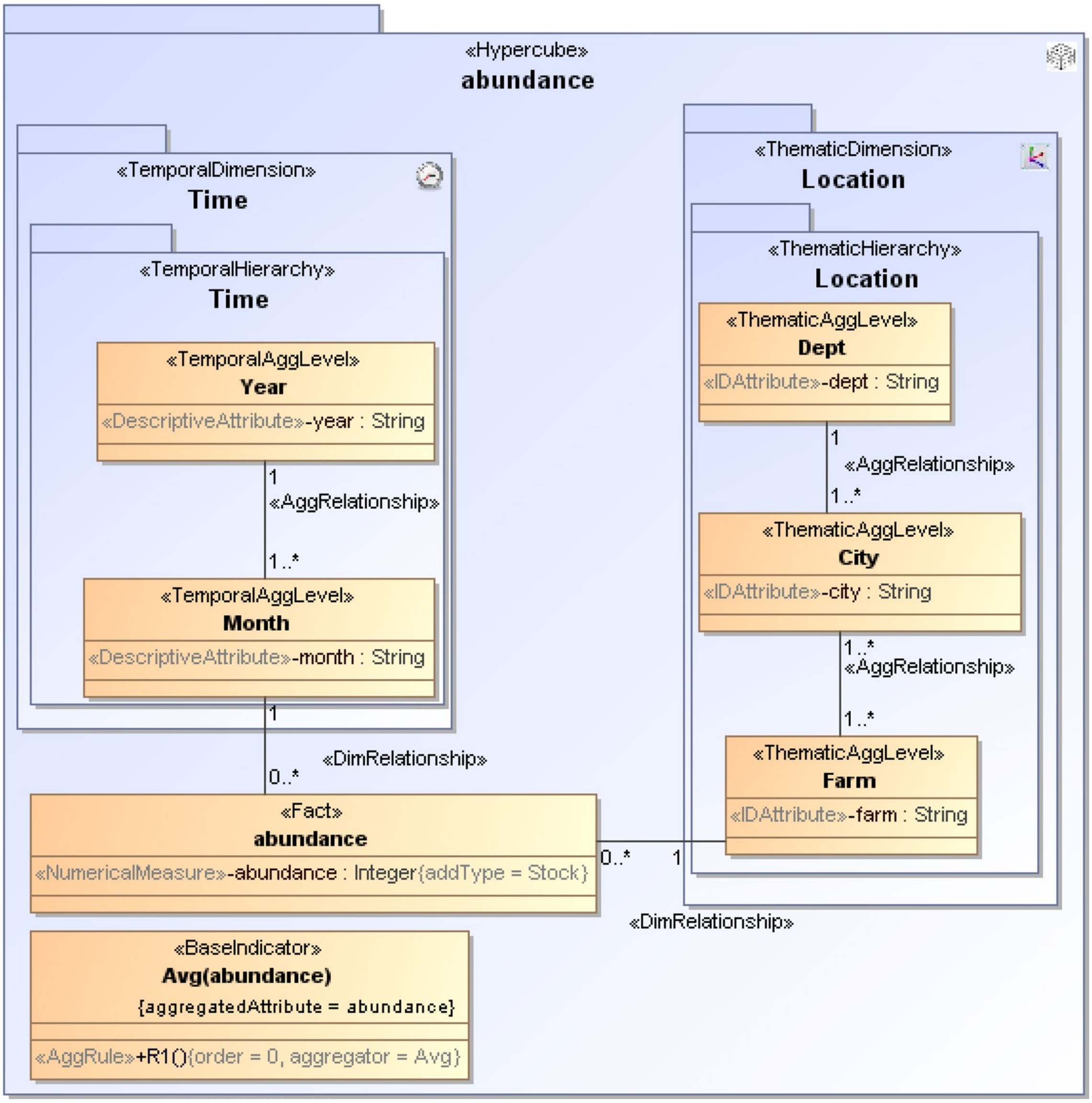


Fig. 15 Output of the fusion between multidimensional schemata

Eleven users have been recruited from the academia. They are students, professors and engineers in different domains: agronomy, robotic engineering and computer science. We have used a multidimensional application concerning the evaluation of students' votes. All users have been trained about the e-pivot table formalism with a 15 min session using the examples shown in the previous sections.

As to readability, we have shown to the users five e-pivot tables containing all the types of irregular structures mentioned in Sect. 3.3. Then, we have asked each user to describe in natural language the e-pivot tables. For each e-pivot table, we have checked whether the user has well understood each irregular structure. As a result, all users have correctly interpreted all e-pivot tables, which confirms our hypothesis that e-pivot tables embody a visual metaphor which is suitable for unskilled users.

As to writability, we have asked users to create an e-pivot table that allows answering this question: "What is the average vote per student and gender, per course and department, and per year?", considering that (1) students may avoid showing their gender for privacy (non-covering hierarchy), (2) a course can belong to multiple departments (non-strict hierarchy) and (3) a group of students could take an exam together, in which case a single mark is given to the whole group (many-to-many fact-dimension relationship). For each user, we have counted the number of errors and the type of multidimensional element involved; the percentage results are summarized in Fig. 16. Many-to-many fact-dimension relationships are clearly the irregular structures

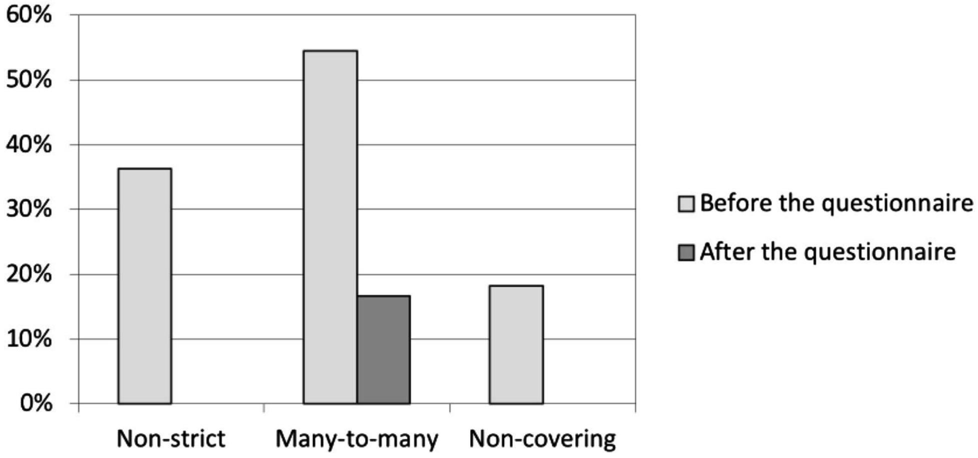


Fig. 16 Writability tests: percentage of errors per type

that most frequently produce errors (54% of cases). We have also measured how many errors were corrected after our questionnaire, since pivot table modeling is followed by a semi-structured interview in our approach. Remarkably, all errors related to non-strict and non-covering hierarchies were solved thanks to the questionnaire; only many-to-many fact-dimension relationship errors were not completely corrected by users, but then they were simply corrected by the DW experts.

We have also analyzed the errors based on the part of the e-pivot table where they occur. Two common errors appeared: the usage of the headers (46%) and the order of the columns representing the levels of a dimension (36%). However, all these errors were quickly solved by users themselves after the DW experts had shown them an example of well-formed e-pivot table. Therefore, we can conclude that these kinds of errors are due to the short training phase.

To sum up, our experiments show that e-pivot tables are easily understood by unskilled users, and the presence of irregular structures is not an obstacle to their adoption as a requirements elicitation tool for complex multidimensional schemata. Additionally, semi-structured interviews appear to be really effective in guiding users towards the resolution of problems and misunderstandings.

6 Related work

6.1 Data warehouse design

We can divide DW design methodologies into two main groups. One group considers the participation of the domain experts at the beginning of the process (during requirements and knowledge elicitation). The other group includes the experts during the following stages of the process. Winter et al. [55] propose an approach to obtain the schema of a DW beginning with the identification of targeted users, the sources of data and the requirements over the data. They propose an iterative and incremental approach, where users are involved at the beginning of the cycle. The process ends with the validation of the schema using the queries elicited at the beginning of the process. Elamin et al. [14] propose another approach to obtain the schema of a DW with three main steps: elicitation, normalization and schema construction. The experts participate in the first step. This proposal consists of a group of heuristics and algorithms in order to make the derivation. Mazon et al. [36] also propose a methodology to produce DW schemata using heuristics. The approach consists of well-defined steps, and the user is only involved in the first step. This proposal elicits requirements from the goals.

Salinesi and Gam [48] propose an iterative and incremental approach with three steps: elicitation, design and integration. Although the approach is similar to other proposals, the decision makers are highly involved in the activities. They participate in the first step to describe their business needs, but also in the design and integration steps to validate the models. The approach proposed does not use any heuristic, which is why the involvement of experts is so important. Bruckner et al. [9] propose a methodology with high involvement of experts and no rules to derive multidimensional schemata. Their approach includes several levels of abstraction: business level, user levels and finally, requirements levels. Kumar et al. [30] also consider different levels of abstraction, from early to late requirements. They also consider organizational and personal needs. Nevertheless, the approach is not iterative. Guo et al. [21] also deal with early requirements in a sequential approach. They begin with the elicitation of the organizational needs, to obtain the schema, to finally elicit requirements. It is interesting how they use requirements in order to validate the scheme, while in general, the schema is obtained from requirements. Paim and Castro [40] also deal with early requirements. Their approach does not produce the schema but only the requirements. However, they perform an intensive interaction with experts in order elicit, specify, verify, validate and improve the specification, until the quality and precision of the requirements are considered acceptable. Then, they finish the specification of requirements that can be fed to other approaches to obtain the schema.

In contrast to iterative methodologies that capture knowledge incrementally, there are waterfall methodologies that need a complete specification in order to produce the DW schema. These approaches generally rely on formal transformations and are composed of sequential steps. Nabli et al. [37] use requirements specification as a tool to derive the DW schema. The rules rely on substitution and merging operations. Giorgini et al. [16] also rely on formal transformations. They use a goal-oriented specification technique, and the experts are involved during most of the process. Finally, Kaldeich et al. [27] propose a similar approach, where they use user requirements and data models as well as goals.

DW design methodologies can also be grouped according to the techniques and the process they rely on. Several approaches rely on goal-driven modeling techniques [16, 21, 27, 30, 36]. In general, they rely on Tropos and i*. These techniques demand high involvement of experts during early stages of elicitation to determine the objectives of the organization and their participants to identify requirements that satisfy these goals [27, 48]. Some approaches rely on natural language [14, 21, 27]. These approaches elicit information (facts, requirements, data), and they analyze this information to design the DW model. Some other approaches use UML models [36]. Use cases are specifically used in order to elicit and validate requirements [9]. A similar tool, namely *scenarios*, is adopted by another proposal [40]. Finally, in [42] the driving force for requirements elicitation is *decisions*; to relate them with the required information, four techniques are proposed based on different critical success factors expressed by managers. Overall, interviews are the most common technique adopted [21, 26, 40]; other frequent techniques are questionnaires [26], focus groups [26], workshops [40], observation [26] and prototyping [40].

Generally, requirements analysis includes elicitation and specification. Nevertheless, some approaches consider requirements validation. Some rely on prototypes and expert users [26, 40], in some cases using questionnaires [22]. Some other approaches rely on verification, where the DW design is contrasted against early models of requirements [14, 21, 48, 55].

The following table summarizes the current work. Approaches are classified based on the elicitation technique, the process type, the user/expert involvement, the orientation (e.g., goal-driven), the level of process formalization and the method used for the validation of the requirements. Table 1 suggests that our approach is the only one using pivot tables as elicitation technique and based on an example-driven model. Finally, an important novelty of our proposal is that we use early prototyping for validation, which adds rapidity and flexibility to the methodology.

6.2 Data warehouses and unstructured data

Since a couple of decades ago, there has been an increasing interest in incorporating unstructured data into the decision-making process. The first attempts in this direction were essentially aimed at designing a DW in a data-driven fashion starting from unstructured (typically, XML) data rather than relational databases, e.g., [54]. These approaches are often classified as *schema-on-write*, as they force unstructured data to fit a structured and fixed schema at the time of loading them into the DW. A more recent trend is that of storing unstructured data in a so-called *data lake* rather than in a DW, leaving them in their raw format; then, data can be analyzed using a *schema-on-read* approach, i.e., adapting them to some (flexible and user-defined) schema only at the time of accessing them. The work done in this direction (e.g. [11, 15],) aims at satisfying the situational analysis need of data scientists rather than the stationary needs of decision makers, while preserving and exalting the variety and dynamics of schema less data. Overall, these schema-on-read approaches cannot be compared to ours, which relies on a DW and can be classified as schema-on-write. Note that a schema-on-read approach could not be adopted for the type of projects we consider in this paper (see Sect. 3), because it requires that users have some strong background in multidimensional modeling; besides, creating multidimensional schemata on-the-fly when accessing data makes it very hard to cope with irregular hierarchies.

6.3 Quality metrics

The problem of devising specific metrics to assess the quality of multidimensional schemata has been long debated in the literature. In a recent survey [20], it is observed that most approaches aim at introducing metrics for understandability and complexity, which in most cases come down to measuring the size of the schema by counting the number of its elements, both at the conceptual (e.g., number of dimensions and number of irregular hierarchies) and logical (e.g., number of tables and number of attributes) level [49].

Interestingly, in [43] a quality framework for multidimensional schemata is presented, based on three points of views: *specification* (the designer's one), *usage* (the decision maker's one) and *implementation* (the developer's one). With reference to the specification view, the paper introduces four criteria: *legibility*, *expressiveness*, *simplicity* and *correctness*. Legibility is related to non-redundancy (no repeated elements in the schema), factorization

Table 1 Comparison of our approach with the previous ones

Approach	Technique	Process	User involvement	Orientation	Process	Validation
Our approach	Interviews, e-pivot tables, prototyping	Iterative, incremental	High	Requirements-driven by examples	Formal	Early, through prototyping
[9]	Use cases	Iterative, incremental	High	No given information	Free	No given information
[14]	Natural language	Waterfall	Low	No given information	Formal, eight heuristics and some algorithms	Yes
[16]	Interviews	Waterfall	Medium, during the whole process	Goal-driven	Formal	Involvement of the stakeholders during the whole process to provide feedback
[21]	Interviews, natural language, interviews	Waterfall	Medium, one stage in the middle of the process relies on users	Goal-driven, data-driven, user-driven	Free	No given information
[26]	Interviews, focus groups, questionnaires, observation	Iterative, incremental	Low	No given information	Free	Through prototyping
[27]	Natural language	No given information	No given information	Demand-driven, data-driven, goal-driven	Formal	No given information
[30]	Interviews	Waterfall	No given information	Goal-driven	Free	No given information
[36]	i*, UML	Waterfall	Low	No given information	Formal	No given information
[37]	No given information	Waterfall	Low	No given information	Formal, rules to provide a unified schema (basically substitution and union operations)	No given information
[40]	Interviews, workshops, prototyping, scenarios	Iterative and incremental	High	No given information	No given information	Yes
[48]	Interviews	Iterative, incremental	High	No given information	Free	Yes
[55]	No given information	Iterative, incremental	Low	No given information	No given information	No given information

(reuse of hierarchies across different cubes, similarly to what done with the conformity factor in [18]) and zoom-in/zoom-out facilities (depth of hierarchies, similarly to what done with the roll-up factor in [18]). Expressiveness has to do with the number of measures, dimensions and levels in a fact. Simplicity deals with the number of facts, levels and arcs in the schema. Finally, correctness essentially counts the number of errors found on the schema elements. Of all these metrics, those related to a whole cube (e.g., number of dimensions) or even to a set of cubes (e.g., conformity factor) could not be used in our approach as a guideline to find the set of questions to be posed to decision makers during pivot table refinement. The reason is they require a *global* view of the multidimensional content—which decision makers do not have at this stage. Decision makers only have a *local* view related to the specific e-pivot table they are creating, so only local metrics can be taken into account, namely non-redundancy, depth of hierarchies and expressiveness. These metrics can be mapped, respectively, onto features minimality, correctness and completeness listed in Sect. 4.3.

Finally, the criteria used to assess the quality of a DW in [39] are *completeness*, *minimality*, *correct aggregations* and *minimal sparsity*. The first three of them can be easily mapped to those of Sect. 4.3, while minimal sparsity cannot be evaluated in our setting since it requires an estimate of the percentage of empty cells in the cube, which could hardly be obtained from our unskilled users.

7 Conclusion and future work

DWs and OLAP are first citizens of Business Intelligence systems. The more the warehoused data reflect the decision makers' analysis needs, the more the DW project will be successful. Therefore, several works have investigated DW design methodologies. Despite the importance of the elicitation phase in these methodologies, few works are specifically focused on how to collect, formalize and validate decision makers' needs. Motivated by this poor attention, and in particular by the lack of an approach to elicitation well-adapted to decision makers not skilled in OLAP and DWs; in this paper, we have proposed a requirements-driven DW design methodology based on the e-pivot table formalism. E-pivot tables allow decision makers to express and formalize their analysis needs by their own, in a by-example fashion. Then, they are also used to formally derive DW schemata in an interactive and iterative process.

Interestingly, with reference to the testing methodology proposed in [18], our methodology directly covers

- Three types of tests aimed at ensuring the quality of the multidimensional schema, namely
- The *workload test*, which verifies that the workload expressed by users during requirement analysis is actually supported by the multidimensional schema;
- The *hierarchy test*, which verifies that the functional dependencies represented by hierarchies in the multidimensional schema are actually valid on source data;
- The *usability test*, that verifies that the schema can be understood by users; and
- The *nomenclature test*, which checks that the names chosen for attributes, measures and domain values are appropriate, consistent and well interpretable by users;
- Two types of tests of the front-end, namely
- The *functional test*, of which a critical part consists in checking that aggregation are correctly defined and computed;
- The *usability test*, which assess the quality of the front-end in terms of *learnability* (how easy is it for users to accomplish basic tasks the first time they use the interface?), *efficiency* (once users have learned to use the interface, how quickly can they perform tasks?), *error recovery* (how many errors do users make, how severe are these errors, and how easily can they recover from the errors?), and *satisfaction* (how pleasant is it to use the interface?).

The experiments we conducted show that e-pivot tables are suitable for non-skilled decision makers during elicitation. In particular, they turned out to be both well-readable and well-writable, especially when coupled with semi-structured interviews. These experiments, associated to those previously conducted for ProtOLAP, confirm the advantages of our proposal.

However, using e-pivot tables indeed has some limitations:

- When complex aggregations or derived indicators are to be specified, the support given to decision makers by e-pivot tables is limited. While now the semantics of indicators is essentially carried by the indicator names, to more accurately cope with this issue we should give decision makers the possibility of writing a formula and have the system automatically check that the values they input are consistent with this formula. Though this solution is quite straightforward from the implementation point of view, finding a precise yet intuitive way to edit formulae (especially when different aggregation operators have to be applied for different hierarchies or different levels within a hierarchy) is indeed a challenge when unskilled decision makers are involved.
- The questions for semi-structured interviews listed in Sect. 4.2 are meant to be a base for stimulating a discussion between decision makers and DW experts about the quality of multidimensional schemata. Some issues deserve further investigation: (1) is the proposed set of questions complete with reference to the four quality factors we consider? (2) can an alternative set of more detailed and understandable questions be defined to reduce the additional work to be done by DW experts during semi-structured interviews? (3) which other quality metrics (e.g., credibility) could be taken as a reference?
- One issue with goal-based modeling is how to understand that goal refinement into subgoals is sufficient. No specific indications are given in [16] to this end, while in [33] it is argued that the refinement process ends when the finest-grain subgoals can be associated to actions that the system or the actors can perform to fulfill them. In our approach, each finest-grain subgoal can be refined, during pivot table modeling, into one or more e-pivot tables; thus, there is no criticality in ensuring that refinement ends only when subgoals corresponding to single e-pivot tables are found. On the other hand, finest-grain subgoals are used during multidimensional modeling to drive the fusion of the multidimensional schemata obtained from each e-pivot table. Thus, stopping goal refinement too early or too late may lead to cubes that are either very generic (i.e., they actually cover multiple facts) or very detailed (i.e., they could be further merged), respectively.

In our case study, in a few situations the cubes created were deemed too generic by decision makers, so we had to go back to the goal modeling step and repeat the following steps with finer subgoals. Clearly, this kind of iteration can be costly, so a more efficient solution to this issue should be devised.

Motivated by the feedback collected during the experiments, in our future work we plan to develop a web-based tool to support users when drawing e-pivot tables, since in the absence of a guide some errors may appear in the structure of the e-pivot table. We also plan to extend the proposed approach in order to support users in mapping requirements into a domain data model. To this end, we will provide rules to guide the analysis of the information obtained from the interviews (narrative descriptions) to identify keywords that can be considered as elements in the domain data model. This technique will rely on natural language processing, on the construction of glossaries and on the available ontologies.

References

1. Arnold D, Corriveau J, Shi W (2010) Modeling and validating requirements using executable contracts and scenarios. In: Proceedings eighth ACIS international conference on software engineering research, management and applications, pp 311–320
2. Benker T, Jürck C (2012) A case study on model-driven data warehouse development. Proc DaWaK 2012:54–64
3. Bimonte S, Edoh-Alove E, Nazih H, Kang M, Rizzi S (2013) ProtOLAP: rapid OLAP prototyping with on-demand data supply. In: Proceedings of the sixteenth international workshop on data warehousing and OLAP, ACM, pp 61–66
4. Bimonte S, Sakka A, Sautot L (2018) A new methodology for elicitation of datawarehouse requirements based on the pivot table formalism. In: Proceedings of 14eme edition de la conference EDA, pp 263–272
5. Bimonte S, Rizzi S, Sautot L, Fontaine B (2019) Volunteered multidimensional design to the test: The farmland bio-diversity VGI4Bio project's experiment. In: Proceedings of the 21st international workshop on design, optimization, languages and analytical processing of big data
6. Bonifati A, Cattaneo F, Ceri S, Fuggetta A, Paraboschi S (2001) Designing data marts for data warehouses. ACM Trans Softw Eng Methodol 10(4):452–483
7. Boulil K, Bimonte S, Pinet F (2015) Conceptual model for spatial data cubes: a UML profile and its automatic implementation. Comput Stand Interfaces 38:113–132
8. Bresciani P, Perini A, Giorgini P, Giunchiglia F, Mylopoulos J (2004) Tropos: an agent-oriented software development methodology. Auton Agent Multi Agent Syst 8(3):203–236
9. Bruckner R, List B, Schiefer J (2001) Developing requirements for data warehouse systems with use cases. In: Proceedings 7th Americas conference on information systems, pp 329–335
10. Chen L, Soliman K, Mao E, Frolick M (2000) Measuring user satisfaction with data warehouses: an exploratory study. Inf Manag 37(3):103–110
11. Chouder ML, Rizzi S, Chalal R (2019) EXODuS: exploratory OLAP over document stores. Inf Syst 79:44–57
12. Di Tria F, Lefons E, Tangorra F (2012) Hybrid methodology for data warehouse conceptual design by UML schemas. Inf Softw Technol 54(4):360–379
13. Diamantini C, Genga L, Potena D, Storti E (2014) Collaborative building of an ontology of key performance indicators. In: Proceedings conference on the move to meaningful internet systems, pp 148–165
14. Elamin E, Alshomrani S, Feki J (2017) SSReq: a method for designing star schemas from decisional requirements. In: Proceedings international conference on communication, control, computing and electronics engineering, pp 1–7
15. Gallinucci E, Golfarelli M, Rizzi S, Abelló A, Romero O (2018) Interactive multidimensional modeling of linked data for exploratory OLAP. Inf Syst 77:86–104
16. Giorgini P, Rizzi S, Garzetti M (2008) GRAnD: a goal-oriented approach to requirement analysis in data warehouses. Decis Support Syst 45(1):4–21
17. Golfarelli M, Maio D, Rizzi S (1998) The dimensional fact model: a conceptual model for data warehouses. Int J Coop Inf Syst 7(2–3):215–247
18. Golfarelli M, Rizzi S (2011) Data warehouse testing: a prototype-based methodology. Inf Softw Technol 53(11):1183–1198
19. Golfarelli M, Rizzi S, Turricchia E (2011b) Modern software engineering methodologies meet data warehouse design: 4WD. In: Proceedings of 13th international conference on data warehousing and knowledge discovery, Springer, pp 66–79
20. Gosain A, Heena (2015) Literature review of data model quality metrics of data warehouse. In: Proceedings international conference on intelligent computing, communication & convergence, pp 236–243
21. Guo Y, Tang S, Tong Y, Yang D (2006) Triple-driven data modeling methodology in data warehousing: a case study. In: Proceedings 9th international workshop on data warehousing and OLAP, pp 59–66
22. Hassine J, Amyot D (2016) A questionnaire-based survey methodology for systematically validating goal-oriented models. Requir Eng 21:285–308
23. Horkoff J et al (2019) Goal-oriented requirements engineering: an extended systematic mapping study. Requir Eng 24:133–160
24. Hwang H-G, Ku C-Y, Yen DC, Cheng C-C (2004) Critical factors influencing the adoption of data warehouse technology: a study of the banking industry in Taiwan. Decis Support Syst 37:1–21
25. Hwang MI, Xu H (2007) The effect of implementation factors on data warehousing success : an exploratory study. J Inf Inf Technol Org 2:1–16
26. Jukic N, Nicholas J (2010) A framework for collecting and defining requirements for data warehousing projects. CIT 18(4):377–384
27. Kaldeich C, e Sá JO (2004) Data warehouse methodology: a process driven approach. In: Proceedings 16th international conference on advanced information systems engineering, Springer, pp 536–549
28. Kamalrudin M, Grundy J (2011) Generating essential user interface prototypes to validate requirements. In: Proceedings 26th IEEE/ACM international conference on automated software engineering, pp 564–567
29. Kimball R, Ross M (2002) The data warehouse toolkit: the complete guide to dimensional modeling, 2nd edn. Wiley, Hoboken
30. Kumar M, Gosain A, Singh Y (2010) Stakeholders driven requirements engineering approach for data warehouse development. JIPS 6(3):385–402
31. Lechtenbörger L (2003) Vossen G (2003) Multidimensional normal forms for data warehouse design. Inf Syst 28(5):415–434
32. Malinowski M, Zimányi E (2008) Advanced data warehouse design—from conventional to spatial and temporal applications. Data-centric systems and applications. Springer, Berlin

33. Martinez A, Pastor O, Mylopoulos J, Giorgini P (2006) From early to late requirements: a goal-based approach. In: Proceedings workshop on agent-oriented information systems, pp 123–142
34. Mazón J, Trujillo J (2009) A hybrid model driven development framework for the multidimensional modeling of data warehouses. SIGMOD Record 38(2):12–17
35. Mazón J, Trujillo J, Lechtenboerger J (2007) Reconciling requirement-driven data warehouses with data sources via multidimensional normal forms. Data Knowl Eng 63(3):725–751
36. Mazón J, Trujillo J, Serrano MA, Piattini M (2005) Designing data warehouses: From business requirement analysis to multidimensional modeling. In: Proceedings of 1st international workshop on requirements engineering for business need and IT alignment, pp 44–53
37. Nabli A, Feki J, Gargouri F (2005) Automatic construction of multidimensional schema from OLAP requirements. In: Proceedings of international conference on computer systems and applications, p 28
38. Nair R, Campbell W, Srinivasan B (2007) A conceptual query-driven design framework for data warehouse. World Acad Sci Eng Technol 25:141–146
39. Niemi T, Nummenmaa J, Thanisch P (2001) Constructing OLAP cubes based on queries. In: Proceedings international workshop on data warehousing and OLAP, pp 9–15
40. Paim FRS, Castro J (2002) Enhancing data warehouse design with the NFR framework. In: Proceedings of workshop emengenharia de requisitos, pp 40–57
41. Pohl K (2010) Requirements engineering—fundamentals, principles, and techniques. Springer, Berlin
42. Prakash N, Prakash D (2019) A multifactor approach for elicitation of information requirements of data warehouses. Requir Eng 24:103–117
43. Prat N, Si-Said Cherfi S (2003) Multidimensional schemas quality assessment. In: Proceedings conference on advanced information systems engineering, pp 341–352
44. Ravat F, Teste O, Tournier R, Zurfluh G (2008) Algebraic and graphic languages for OLAP manipulations. IJDWM 4(1):17–46
45. Romero O, Abelló A (2006) Multidimensional design by examples. In: Proceedings international conference on data warehousing and knowledge discovery, pp 85–94
46. Romero O, Abelló A (2009) A survey of multidimensional modeling methodologies. IJDWM 5(2):1–23
47. Sakka A, Bimonte S, Sautot L, Camilleri G, Zaraté P, Besnard A (2018) A volunteer design methodology of data warehouses. In: Proceedings of 37th international conference on conceptual modeling, Springer, pp 286–300
48. Salinesi C, Gam I (2006) A requirement-driven approach for designing data warehouses. In: Proceedings of requirements engineering: foundations for software quality
49. Serrano M, Trujillo J, Calero C, Piattini M (2007) Metrics for data warehouse conceptual models understandability. Inf Softw Technol 49:851–870
50. Shimomura T, Ikeda K, Chen QL, Lang NS, Takahashi M (2007). Visual pivot-table components for web application development. In: Proceedings of the third conference on IASTED international conference: advances in computer science and technology (ACST'07). ACTA Press, Anaheim, CA, USA, pp 90–95
51. Suranto B (2015) Software prototypes: enhancing the quality of requirements engineering process. In: Proceedings international symposium on technology management and emerging technologies, pp 148–153
52. Vaisman AA (2006) Requirements elicitation for decision support systems: a data quality approach. In: Proceedings of eighth international conference on enterprise information systems: databases and information systems integration, pp 316–321
53. Vaisman AA, Zimanyi E (2014) Data warehouse systems—design and implementation. Data-centric systems and applications. Springer, Berlin
54. Vrdoljak B, Banek M, Rizzi S (2003) Designing web warehouses from XML schemas. In: Proceedings international conference on data warehousing and knowledge discovery, pp 89–98
55. Winter R, Strauch B (2003) A method for demand-driven information requirements analysis in data warehousing projects. In: Proceedings 36th Hawaii international conference on system science, p 231
56. Yeoh W, Koronios A (2010) Critical success factors for business intelligence systems. JCIS 50(3):23–32

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.