Towards a multi-model approach to support user-driven extensibility in data warehouses: agro-ecology case study

Fagnine Alassane Coulibaly, Sandro Bimonte, Stefano Rizzi, Sylvie Malembic-Maher, Frédéric Fabre

Towards a Multi-Model Approach to Support User-Driven Extensibility in Data Warehouses: Agro-ecology Case Study

Fagnine Alassane Coulibaly[1], Sandro Bimonte[1,*], Stefano Rizzi[2,*], Sylvie Malembic-Maher[3,*] and Frédéric Fabre[4]

[1]TSCF, INRAE Clermont-Ferrand, 9 Avenue Blaise Pascal, Aubière, 63178, France
[2]DISI, University of Bologna, Viale Risorgimento, 2, Bologna, 40136, Italy
[3]INRAE, University Bordeaux, BFP, Villenave d'Ornon, 33882, France
[4]INRAE, Bordeaux Sciences Agro, UMR SAVE, Villenave d'Ornon, 33882, France

Abstract

Information systems have evolved into complex data platforms supporting end-to-end data-intensive needs, aimed at coping with the different V's that characterize Big Data. In particular, multi-model databases (MMDBs) have been proposed to natively support storing and querying data in different (schemaless) models, so as to better handle Variety. In this work we envision a new data warehouse architecture in which an MMDB is used to enable on-the-fly user-driven extensions of multidimensional cubes with additional data, while ensuring support to variable and complex data and keeping the impact on ETL low. After proposing the architecture with the aid of a case study on the management of emerging plant disease, we discuss the main associated open issues.

Keywords

Data Warehouses, Multi-model databases, OLAP, Big Data

1. Introduction and motivation

The growing availability of data combined to advances in computational algorithms and statistical modelling is profoundly changing the practice of research on complex phenomena. Business Intelligence (BI) tools play a key role in this evolution by enabling the exploration of huge volumes of data. They benefit from a growing demand for these tools in new fields such as agro-ecology. The purpose of agro-ecology is to develop new farming practices that respect the environment while maintaining productivity and biodiversity [1]. Agro-ecology involves governmental, economic, social, and environmental data and actors. Traditionally, research in agroecology used the so-called "hypothesis-driven" process, which consists in eliciting all the data needed to challenge a testable hypothesis at design time. When some data are not available (for any reason), then they are excluded from the analytical process. Recently, with the advent of Big Data, "data-driven" analysis [2] has been emerging as an alternative that allows deriving knowledge from data that were not identified or available at design time. However, in the context of complex application domains, data-driven analysis is poorly feasible since the data collection process could become a too wide task. In the context of agro-ecology for instance, social, economic, agronomic, and meteorological data can be relevant in theory, but collecting them all in advance might be an overwhelming task. Solving this problem requires flexible BI tools that allow researchers to incorporate new data on-demand, whenever they need to test their hypotheses.

In the Big Data era, traditional database systems have evolved into complex data platforms supporting end-to-end data-intensive needs, such as storage, computation, and analysis of NoSQL data with heterogeneous structures. In particular, DBMSs that can handle different kinds of data, such as polyglot databases [3], have been introduced to better deal with the different *V* features that characterize Big Data, in particular Variety. In the same direction, *multi-model databases* (MMDBs) have been recently proposed to natively support storing and querying data in different models (graph-based, document-based, relational, etc.), which are often schemaless [4].

Data Warehouses (DW) also belong to this picture. Together with OLAP systems, they are widely recognized as main citizens of BI, as they enable interactive analyses of huge multidimensional cubes. While cubes are traditionally stored in relational databases, NoSQL databases are now used as well to this purpose. MMDBs have been also found to represent a suitable solution to enhance flexibility in DWs; in fact, some very recent works investigate the usage of MMDBs for storing multidimensional data [5, 6, 7]. We presented in [5] an extension of the clas-

DataPlat'23: 2nd International Workshop on Data Platform Design, Management, and Optimization, March 28, 2023, Ioannina, Greece
*Corresponding author.
fagnine-alassane.coulibaly@inrae.fr (F. A. Coulibaly); sandro.bimonte@inrae.fr (S. Bimonte); stefano.rizzi@unibo.it (S. Rizzi); sylvie.malembic-maher@inrae.fr (S. Malembic-Maher); frederic.fabre@inrae.fr (F. Fabre)
0000-0003-1727-6954 (S. Bimonte); 0000-0002-4617-217X (S. Rizzi)

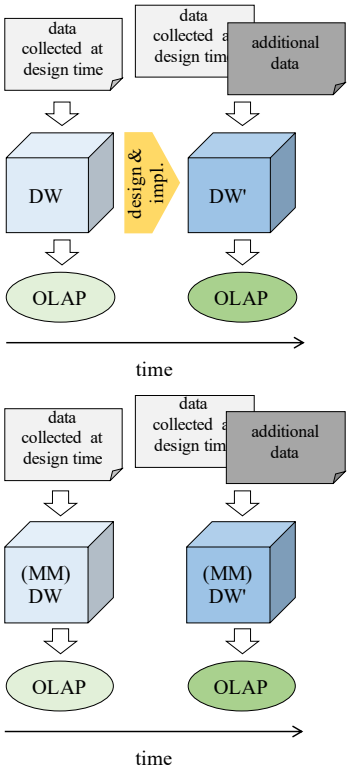

Figure 1: Two scenarios for extensibility: schema-on-write (top) and schema-on-read (bottom)

sical, relational star schema with factual and dimensional data stored via the document-based, graph-based, and key-value models, and we discussed the corresponding design guidelines in [7]. We claimed that *multi-model DWs* (MMDWs) simplify the Extraction, Transformation, and Loading (ETL) process, preserve the performances of OLAP queries, and encourage flexibility, extensibility, and evolvability. We proposed a UML profile for designing multidimensional models supporting variety in [6]; this profile supports type variability, complex objects, and extensibility for both dimensional and factual data. Specifically, extensibility refers to the possibility of adding new multidimensional elements to the MMDW that were unavailable or unknown at design time, so they can be used for future OLAP analysis. Such extensibility feature appears to be crucial when analyzing complex phenomena, as previously said for agro-ecology.

Figure 1 depicts two possible scenarios to provide extensibility in a DW. In the first one, following a classical *schema-on-write* approach where all source data are put into multidimensional form following a schema agreed at design time, the DW cannot be extended on-demand; thus, design and implementation must be redone in order to include additional data. This is impractical and time-consuming; in fact, a user-driven integration of new data in the DW should be easy and fast, requiring no intervention by designers and IT people. Thus, we envision a second scenario where a MMDW is used for storage and a schema on-read approach —which leaves data unchanged in their structure until they are accessed by the user [8]— is followed for OLAP queries. Here, decision-makers can launch OLAP queries over new multidimensional data on-the-fly, so no design and implementation iteration is required.

In this vision paper we investigate the schema-on-read scenario to extensible DWs by proposing an architecture to support it (Section 3) and discussing the main open issues associated (Section 4), using as a running example a real agro-ecological case study taken from the BEYOND project (Section 2). The paper is concluded by Section 5.

2. Case study

The BEYOND project (https://www6.inrae.fr/beyond/) aims at developing new indicators of plant disease risks in order to improve monitoring and prophylaxis strategies.

A specific goal of this project is the monitoring of *flavescence dorée*, an highly contagious quarantine disease threatening European vineyards [9]. Annual vineyard surveys inform the infectious status of plants. These historical data are gathered and analyzed using a DW in order to (i) understand the spatial and temporal dynamics of the disease, (ii) investigate the field and landscape factors that can (un)foster its propagation [10], (iii) better organize the observations tasks, and (iv) provide farmers with easily understandable indicators. The conceptual schema of the INFECTION multidimensional cube used to this end is shown in Figure 2 by means of the V-ICSOLAP profile [6]. It presents two measures, namely, the surface area surveyed (areaInHa) and the number of plants infected by *flavescence dorée* (numberInfectedWines), and four dimensions: a spatial Plot dimension, the temporal one, the winegrower one, and the team of professional organizations in charge of the detection of *flavescence dorée*.

The INFECTION cube has been designed taking into account both the requirements expressed by stakeholders and researchers and the available data. It presents two main issues. First of all, *additional data* are needed: the current dimensions and measures cannot be used to deeply understand the factors underlying disease spread. Thus, the multidimensional schema shows some *extensibility points*, i.e., parts of the schema where we expect that additional data will be available (the fact can be extended in terms of measures and dimensions, and new levels can be added to both the plot and the team dimensions; see the Extensibility properties in red in Figure 2). Secondly,

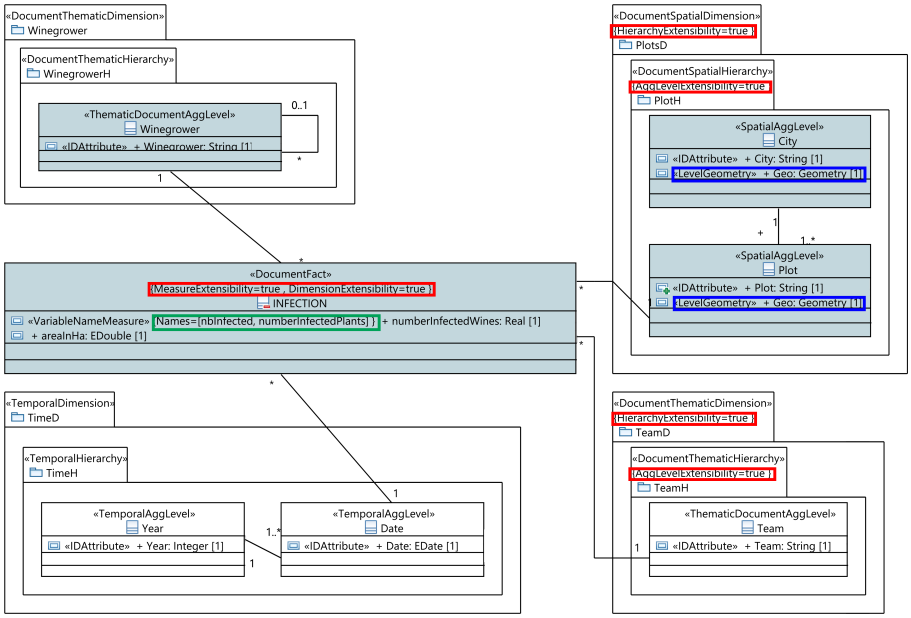

Figure 2: UML conceptual schema for *flavescence doreé* analysis based on the V-ICSOLAP profile [6]

the data involved are *too complex* to be seamlessly integrated into a classical relational DW. On the one hand, they come with *variability*; indeed, attributes names and types can change over time while maintaining the same meaning (e.g., measure numberInfectedWines can also be called numberInfectedPlants, in green), the same for data structures. On the other hand, these data follow *different models* (documents for spatial data, relational tables for contaminated plants campaigns, graphs for winegrowers, etc.), as in level Geo of the plot dimensions (in blue).

3. Envisioned architecture

In this section we describe the architecture we envision for extensible DW. As shown in Figure 3, it is composed of the following layers:

Data lake, where all data ingested are stored in their native formats (relational, document, graph, etc.). This layer is used to feed the next layer and can be defined as a data lake [11] since it will allow users to explore the source data as well as to extract/store the additional data to be loaded on-demand in the next layer.

Multi-model data warehouse, in charge of storing the warehoused data. An MMDW is used here because, as argued in [6], the extensibility points of the multidimensional schema can be easily implemented using the schemaless data structures allowed by MMDWs. As a result, when new data are fed to these points, no effort to adapt the DW schema is required for their inclusion in the decision-making process, and the effort for evolving the ETL is small. This layer is also in charge of hosting the additional data that the user selects from the source layer to discover new multidimensional elements and enhance the decision-making process.

OLAP, where users not only formulate multidimensional queries and visualize the results, but can also select additional data to be loaded and drive the process that discovers new multidimensional elements.

In our case study, the fact table in the MMDW layer is fed, via ETL, from a JSON collection stored in the data lake layer and including measurements of the number of infected wines. Assume that, starting from today, the JSON documents sent from some plots will also include some additional fields storing variables measured by sensors (e.g. automatic detection of the insect vector of *flavescence dorée* phytoplasma, Scaphoideus titanus). The fact has been defined as extensible, thus, the fact table comes with a JSON attribute that can store these new measures, with no effort to evolve the ETL process. Now, assume that a new table storing additional geographical data, e.g., the department each city belongs to, is loaded in the data lake. Should decision makers be interested in analyzing infection data by departments, they could (i) select this table to be loaded in the MMDW and (ii) use the many-to-one relationship between cities and departments to extend the plot hierarchy on the fly.

We close this section by remarking that our architecture cannot be classified as a *data lakehouse* [12], because

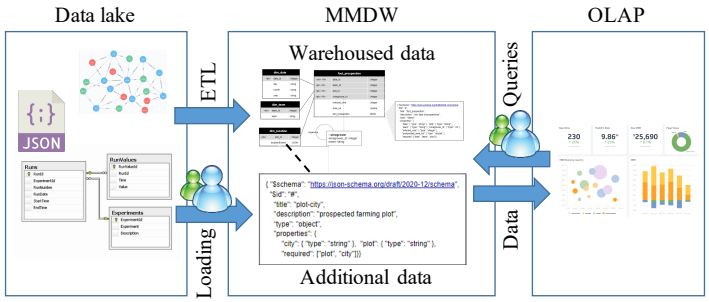

Figure 3: Envisioned architecture

it physically stores multidimensional data to be used for OLAP rather than letting OLAP queries by directly written on source data; indeed, this has been shown to entail a smaller effort when writing queries as well as better query performances [5].

4. Open issues

In this section we discuss the main research issues to be addressed to implement the approach proposed.

Cross-model data-driven multidimensional modeling. Data-driven methodologies for multidimensional design are usually based on the discovery of functional dependencies (FDs) in the source data to identify measures and dimension levels. The approaches proposed so far operate on source data fitting a single model. When source data are relational, the proposed algorithms detect exact FDs based on primary and foreign keys coded in the relational schema [13]. When source data are JSON documents, some methods have been devised to complement exact FDs coded in the document schema with approximate FDs [14] detected by parsing the data [15]. Other works investigate how to enable OLAP querying over graph databases [16]. Recently, some works have introduced formal models to give a unified representation of the schema of multi-model databases [17, 18], but they do not address the discovery of multidimensional schemata from these unified models. Overall, to enable on-the-fly extensions of cubes, there is a need for new algorithms capable of discovering exact/approximate FDs in multi-model databases by chasing cross-model references. These algorithms should be incremental, since extensions are created starting from the multidimensional schema of the existing cube. Clearly, consistently with what existing work proposes in schema-on-read approaches, they should also take into account the requirements of decision-makers.

Variability. Schemaless models inherently support variability in data types, names, and structures. However, most existing works on schema inference from schemaless data (e.g., [19, 20]) output distinct attributes in presence of variability, which would not make this variability transparent to users querying the DW. Type variability brings an additional problem, since measures are commonly identified with numerical attributes. Thus, to enable multidimensional elements to be correctly recognized when extending a cube, variability-aware approaches to properly infer schemata from schemaless data should be devised.

Complex multidimensional elements. Big Data sources can include complex data such as streams, trajectories, graphs, spatiotemporal data, etc. The possibility of defining measures as complex objects has been considered in [21], which also proposes a relational implementation for them, while in [6] a UML profile to model them at the conceptual level has been presented. However, how to recognize and design these complex measures —as well as complex dimensions and levels— starting from data sources is still an open question.

Multi-model design. Storing warehoused data in an MMDB gives rise to further questions: *Can different models be mixed to store warehoused data? Which factors impact the choice of the best model to be used for each piece of data? Which are the benefits of using an MMDW instead of a single-model implementation?* Some preliminary answers are given in [7], which provides some guidelines to design an MMDW that ensures a good trade-off between features such as querying performance, storage space, ETL complexity, and evolvability. However, a complete set of best practices for multi-model design has not been devised yet.

OLAP tools. The existing OLAP clients are able to connect to warehoused data stored in relational form, typically using star/snowflake schemata. A first issue here is related to creating clients that can transparently query DWs under different models while fully supporting the OLAP paradigm. Secondly, clients must be able to properly deal with variability (e.g., as suggested in [22]) and complex multidimensional elements. Another

challenge is how to ensure that the process of extending the cubes on-the-fly based on the user's requirements is smooth and effective, and at the same time efficient enough to be compatible with interactive analyses.

5. Conclusion

Several attempts have been made to improve data management in the Big Data era by moving from traditional database architectures to sophisticated data platforms. Among the architectures and technologies conceived to this end, we mention data lakes [11], lakehouses [12], polyglot databases [3], and MMDBs [4]. In particular, using MMDBs to store warehoused data has been found to ensure interesting features, such as low ETL costs and improved evolvability and flexibility [5]. However, no full support to the extensibility, variability, and complex data that characterize Big Data has been given yet [6]. To bridge this gap, in this work we have envisioned a new architecture where an MMDW is associated with additional data, loaded on-the-fly on the user's request and integrated with the existing cubes following a schema-on-read approach, so as to ensure extensibility. Using an MMDW also ensures that variability and complex data are seamlessly supported. The approach we propose leaves space for addressing several research questions, mainly related to detecting multidimensional elements from multi-model sources in presence of variability and complex data, and to creating OLAP tools that transparently supports all these features.

6. Acknowledgement

This work is supported by ANR-20-PCPA-0002.

References

[1] T. Dalgaard, N. Hutchings, J. Porter, Agroecology, scaling and interdisciplinarity, Agriculture, Ecosystems & Environment 100 (2003) 39–51.

[2] P. M. Hartmann, M. Zaki, N. Feldmann, A. Neely, Capturing value from big data: a taxonomy of data-driven business models used by start-up firms, Int. J. of Oper. & Prod. Manag. 36 (2016) 1382–1406.

[3] F. Kiehn, et al., Polyglot data management: State of the art & open challenges, Proc. VLDB Endow. 15 (2022) 3750–3753.

[4] J. Lu, I. Holubová, Multi-model databases: A new journey to handle the variety of data, ACM Comput. Surv. 52 (2019) 55:1–55:38.

[5] S. Bimonte, E. Gallinucci, P. Marcel, S. Rizzi, Data variety, come as you are in multi-model data warehouses, Inf. Syst. 104 (2022) 101734.

[6] S. Bimonte, H. Bazza, J. Laneurit, S. Rizzi, H. Badir, A UML profile for variety and variability awareness in multidimensional design, in: Proc. DOLAP, 2022, pp. 1–10.

[7] S. Bimonte, E. Gallinucci, P. Marcel, S. Rizzi, Logical design of multi-model data warehouses, Knowl. and Inf. Syst. 65 (2023) 1067–1103.

[8] Z. H. Liu, D. Gawlick, Management of flexible schema data in RDBMSs - opportunities and limitations for NoSQL, in: Proc. CIDR, 2015.

[9] EFSA, S. Tramontini, A. Delbianco, S. Vos, Pest survey card on flavescence dorée phytoplasma and its vector scaphoideus titanus, EFSA Supporting Publications 17 (2020) 1909E.

[10] H. K. Adrakey, et al., Field and landscape risk factors impacting flavescence dorée infection, Phytopathology 112 (2022).

[11] F. Nargesian, E. Zhu, R. J. Miller, K. Q. Pu, P. C. Arocena, Data lake management: Challenges and opportunities, Proc. VLDB Endow. 12 (2019) 1986–1989.

[12] M. Zaharia, A. Ghodsi, R. Xin, M. Armbrust, Lakehouse: A new generation of open platforms that unify data warehousing and advanced analytics, in: Proc. CIDR, 2021.

[13] M. Golfarelli, S. Rizzi, Methodological framework for data warehouse design, in: Proc. DOLAP, 1998, pp. 3–9.

[14] M. DiScala, D. J. Abadi, Automatic generation of normalized relational schemas from nested key-value data, in: Proc. SIGMOD, 2016, pp. 295–310.

[15] M. L. Chouder, S. Rizzi, R. Chalal, EXODuS: Exploratory OLAP over document stores, Inf. Syst. 79 (2019) 44–57.

[16] C. Chen, X. Yan, F. Zhu, J. Han, P. S. Yu, Graph OLAP: a multi-dimensional framework for graph data analysis, Knowl. and Inf. Sist. 21 (2009) 41–63.

[17] P. Koupil, I. Holubová, A unified representation and transformation of multi-model data using category theory, J. Big Data 9 (2022) 61.

[18] C. J. F. Candel, D. S. Ruiz, J. J. G. Molina, A unified metamodel for NoSQL and relational databases, Inf. Syst. 104 (2022) 101898.

[19] M. A. Baazizi, H. B. Lahmar, D. Colazzo, G. Ghelli, C. Sartiani, Schema inference for massive JSON datasets, in: Proc. EDBT, 2017, pp. 222–233.

[20] H. Lbath, A. Bonifati, R. Harmer, Schema inference for property graphs, in: Proc. EDBT, 2021, pp. 499–504.

[21] S. Bimonte, M. Miquel, When spatial analysis meets OLAP: multidimensional model and operators, Int. J. Data Warehous. Min. 6 (2010) 33–60.

[22] E. Gallinucci, M. Golfarelli, S. Rizzi, Approximate OLAP of document-oriented databases: A variety-aware approach, Inf. Syst. 85 (2019) 114–130.