Ecological Informatics 16 (2013) 41–52

Contents lists available at SciVerse ScienceDirect

Ecological Informatics

journal homepage: www.elsevier.com/locate/ecolinf

Guaranteeing the quality of multidimensional analysis in data warehouses of simulation results: Application to pesticide transfer data produced by the MACRO model

Kamal Boulil [a], François Pinet [a], Sandro Bimonte [a,*], Nadia Carluer [b], Claire Lauvernet [b], Bruno Cheviron [c], André Miralles [c], Jean-Pierre Chanet [a]

[a] Irstea, Clermont-Ferrand, France
[b] Irstea, Lyon, France
[c] Irstea, Montpellier, France

ARTICLE INFO

Article history:
Received 3 August 2012
Received in revised form 4 April 2013
Accepted 5 April 2013
Available online 23 April 2013

Keywords:
Simulation model result
Data warehouse
Software engineering
Unified Modeling Language

ABSTRACT

Currently, the vital impact of environmental pollution on economic, social and health dimensions has been recognized. The need for theoretical and implementation frameworks for the acquisition, modeling and analysis of environmental data as well as tools to conceive and validate scenarios is becoming increasingly important. For these reasons, different environmental simulation models have been developed. Researchers and stakeholders need efficient tools to store, display, compare and analyze data that are produced by simulation models. One common way to manage simulation results is to use text files; however, text files make it difficult to explore the data. Spreadsheet tools (e.g., OpenOffice, MS Excel) can help to display and analyze model results, but they are not suitable for very large volumes of information. Recently, some studies have shown the feasibility of using Data Warehouse (DW) and On-Line Analytical Processing (OLAP) technologies to store model results and to facilitate model visualization, analysis and comparisons. These technologies allow model users to easily produce graphical reports and charts. In this paper, we address the analysis of pesticide transfer simulation results by warehousing and OLAPing data, for which the data results from the MACRO simulation model. This model simulates hydrological transfers of pesticides at the plot scale. We demonstrate how the simulation results can be managed using DW technologies. We also demonstrate how the use of integrity constraints can improve OLAP analysis. These constraints are used to maintain the quality of the warehoused data as well as to maintain the aggregations and queries, which will lead to better analysis, conclusions and decisions.

1. Introduction

Environmental modeling is extensively used to study complex phenomena, such as urbanization, climate change and pollutant transfers (Hirabayashi et al., 2011; Li and Mao, 2011; Pogson et al., 2012; Trolle et al., 2011). These models allow researchers and stakeholders to represent, understand, explain and formulate/validate hypotheses about environmental phenomena to predict their evolution. These tasks usually involve several disciplines and several types of information (e.g., spatial, temporal, biological, meteorological, hydrological and economical information). The analysis of simulation/numerical model results is useful to compare or calibrate models with several sets of observed input data and to provide a global vision of model behavior.

Models can produce enormous volumes of result data (Fernández-Quiruelas et al., 2011). Moreover, models can simulate a stochastic process that requires several replications of each simulation run to obtain representative results. These replications increase the quantity of the result data, which makes their exploration and analysis difficult. As shown in Mahboubi et al. (2010), scientists or model users must extract aggregated information from large amounts of simulation results (e.g., regularities or synthetic indicators). Researchers and stakeholders also must perform comparative analyses of results that are produced from different sets of input data, different versions of the models or different sets of parameters.

For these purposes, efficient tools are needed to store, display, compare and analyze simulation results. One typical method for storing simulation results is to use text files; however, text-based storage makes it difficult to explore, select and visualize the data. Spreadsheet tools (for example, OpenOffice or MS Excel) can help to display and analyze the simulation results, but they are not suitable for very large volumes of information. Some recent studies have shown how Data Warehouse (DW) and On-Line Analytical Processing (OLAP) technologies can be used for the analysis of environmental simulation model results

[*] Corresponding author. Tel.: +33 671942187.
E-mail addresses: Kamal.Boulil@irstea.fr (K. Boulil), Francois.Pinet@irstea.fr (F. Pinet), Sandro.Bimonte@irstea.fr (S. Bimonte), Nadia.Carluer@irstea.fr (N. Carluer), Claire.Lauvernet@irstea.fr (C. Lauvernet), Bruno.Cheviron@irstea.fr (B. Cheviron), Andre.Miralles@irstea.fr (A. Miralles), Jean-Pierre.Chanet@irstea.fr (J.-P. Chanet).

1574-9541/$ – see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.ecoinf.2013.04.004

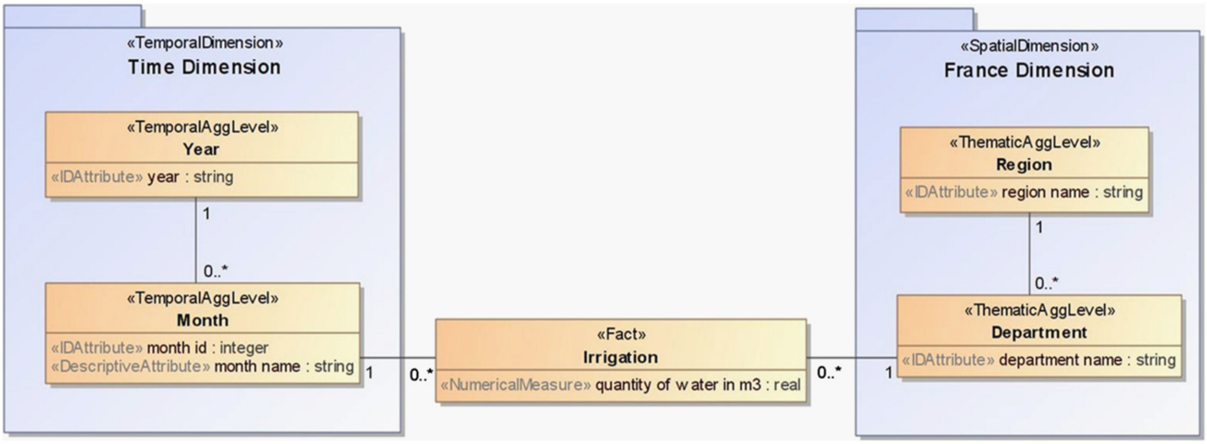

«TemporalDimension»
Time Dimension
«TemporalAggLevel»
Year
«IDAttribute» year : string
1
0..*
«TemporalAggLevel»
Month
«IDAttribute» month id : integer
«DescriptiveAttribute» month name : string
1
0..*
«Fact»
Irrigation
«NumericalMeasure» quantity of water in m3 : real
0..*
1
«SpatialDimension»
France Dimension
«ThematicAggLevel»
Region
«IDAttribute» region name : string
1
0..*
«ThematicAggLevel»
Department
«IDAttribute» department name : string

Fig. 1. Example of conceptual multidimensional schema.

(Mahboubi et al., 2010, 2011). DWs are dedicated to integrating and storing very large volumes of information (in the same database) to support multi-dimensional data analysis. OLAP tools allow DW data to be easily and rapidly explored and displayed using tables, different types of statistical diagrams and reports. DWs are usually relational databases (DBs); consequently, this technology provides efficient methods for user authentication, integrity constraint controls, data backup, data insertion and data selection (Basta and Zgola, 2011; Pokorný, 2006).

In the context of the project "Environmental Information Systems for pesticides" (2007–2010) (Miralles et al., 2011), financed by Irstea, and "Miriphyque", which was financed by the French Ministry of Ecology, we have used DW technologies for the storage and analysis of data that is produced by an environmental simulation model called MACRO. MACRO is a physically based model that simulates water and pesticide transfers using a dimensional approach for both microporous and macroporous media (Larsbo and Jarvis, 2003). We propose a DW schema (i.e., a multidimensional schema) for the analysis of the concentration, flux and discharge indicators of the MACRO model according to thematic and temporal dimensions. The schema is implemented in a traditional relational OLAP architecture using free software: PostgreSQL (PostgreSQL, 2012), JRubik (JRubik, 2012) and Mondrian (Pentaho, 2012).

In this paper, we demonstrate the limit of the quality of the analysis that is achieved with traditional DW and OLAP systems. We present the first experiment that validates the method introduced in Boulil et al. (2012) to specify integrity constraints (ICs) in DW and OLAP systems; we show how to apply this method on the DW that stores the MACRO model results. ICs are rules that are implemented and checked to ensure the consistency of the stored data, aggregations and OLAP queries. For example, these rules allow users to detect errors in stored simulation results. The errors can originate from the used parameters or from an inadequate implementation of the models. In our paper, we introduce examples of ICs that are modeled with the UML-based formalism proposed in Boulil et al. (submitted for publication). As illustrated in this paper, this method allows users to automatically generate the specific mechanisms that can be used to check whether the data, aggregations or queries comply with the specified ICs.

Table 1
The amount of water (in m^3) by year and department.

	Region 1			Region 2				…
	Dpt 1	Dpt 2	Dpt 3	Dpt 1	Dpt 2	Dpt 3	Dpt 4	…
Year 1	50,000	45,000	32,000	60,000	45,000	47,000	44,000	…
Year 2	50,000	47,000	30,000	58,000	45,000	47,000	43,000	…
Year 3	52,000	44,000	30,000	59,000	45,000	45,000	43,000	…
…	…	…	…	…	…	…	…	…

Table 2
The amount of water (in m^3) by year and region.

	Region 1	Region 2	…
Year 1	127,000	136,000	…
Year 2	127,000	135,000	…
Year 3	126,000	133,000	…
…	…	…	…

The main contributions presented in this paper are as follows: (i) a discussion on the advantages that DW and OLAP systems offer for the storage and analysis of the simulation results, which is addressed to environmental researchers; (ii) an OLAP system for the multidimensional analysis of pesticide transfer data generated by the MACRO model; and (iii) validation on a case study of the automated and standards-based approach proposed in Boulil et al. (2012) for the design and implementation of OLAP systems and their ICs.

This paper is structured in the following way. Section 2 introduces the main DW concepts. Section 3 presents the main concepts of our UML-based method (which is formalized as a UML profile). Section 4 discusses related work on DWs for simulation models. Section 5 presents the multidimensional model for the MACRO model and the associated integrity constraints and their implementation. Finally, Section 6 concludes the paper with a description of future work.

2. Data warehouse and quality: Main concepts

A DW is usually represented by a multidimensional schema (Pinet and Schneider, 2010). Fig. 1 shows an example of a conceptual multidimensional schema defined with the UML-based formalism proposed in Boulil et al. (2012). Data that will be visualized and analyzed by DW users are called facts. In this example, the facts are information about irrigation in agriculture. In the UML representation of Fig. 1, all of the facts are instances (i.e., objects) of the class marked "≪Fact≫". A fact can contain one or several attributes, which are numerical measures that are used for data analysis. In this example, the measure for analysis is the amount of water (in m^3) that is used for irrigation. The measures stored in the DW are the quantities of water per French administrative Department[1] per month.

In DWs, queries allow for the aggregation of measures (e.g., a sum or an average). In this example, the total amount of water that is used for irrigation can be calculated at different levels: by the year and/or by the French administrative Region. A hierarchy composed of one or several levels is called a dimension of analysis. Here, we have two dimensions: one temporal dimension (Month < Year) and one spatial dimension (Department < Region). In DWs, aggregations can be calculated for each

[1] In the geographical organization of France, an administrative Region is subdivided into several administrative Departments. France is composed of 27 Regions and 101 Departments.

Table 3
A comparison of spreadsheet tools and DWs.

Tools Criterion	Spreadsheets	DWs
Access time for large volumes of data	No data indexing method Limited optimization techniques All data are usually loaded into memory	Optimizations for data queries and insertions (provided by DB); efficient indexing methods Based on a client-server architecture; clients can load and display only a portion of the stored data
Security	Very basic method for user authentications is provided	Users' authentication and management of access rights Methods for data backups Integrity constraints can check and guarantee the consistency of the information Data encryption is usually possible
Internet/Intranet access	Additional implementation is needed to diffuse data to the Internet/Intranet	Based on an Internet/Intranet client-server architecture
Visualization	Tables with multiple entries Charts	Tables with multiple entries Charts Map production with spatial OLAP technology
Data exploration	Wizard and assistants are often provided to help create tables	OLAP allows users to easily choose measures and dimension levels, to perform quick swapping between aggregation functions and to select only a subset of the data OLAP server support for complex data queries (e.g., with the MDX language)

Table 4
Our UML profile for DWs.

Stereotype	Can be applied to	Semantics
≪Fact≫	Class	Fact class
≪NumericalMeasure≫	Attribute	Measure (having a numeric type) defined in fact classes
≪SpatialDimension≫	Package	Spatial dimension, e.g., a dimension that contains a hierarchy of geographical levels (different spatial scales)
≪TemporalDimension≫	Package	Temporal dimension that contains the different temporal scales
≪ThematicDimension≫	Package	Dimension that contains thematic (non-spatial and non-temporal) information
≪SpatialAggLevel≫, ≪TemporalAggLevel≫, ≪ThematicAggLevel≫	Class	The different levels of dimensions that contain spatial, temporal or thematic information
≪IDAttribute≫	Attribute	Attributes used to identify/distinguish instances of dimension levels (members)
≪DescriptiveAttribute≫	Attribute	Attributes that contain descriptive information

combination of levels. For example, if the sum aggregation function is chosen by the user, then the quantity of water can be calculated by any of the following: per Month; per Year; per Department; per Region; per Month and Department; per Month and Region; per Year and Department; or per Year and Region. This sum can also be calculated for all Regions and all years.

Relational DB management systems (e.g., PostgreSQL, Oracle, MySQL) are often used to store DW data. On-Line Analytical Processing (OLAP) tools (Berson and Smith, 1997) enable data aggregation and visualization of results in tables with multiple entries (i.e., pivot tables). To compose a pivot table with OLAP tools, users choose a measure, an aggregation function and different levels of dimensions. For example, suppose that a user creates a table with the measure “quantity of water in m^3”, the sum function and the levels “Department” and “Year”. Table 1 presents the resulting table (showing an aggregation by Year and Department). Each table cell contains the total amount of water used for each pair {department, year}. These values are calculated from the DW using the finest granularity found in the data, i.e., the amount per month and department. Next, suppose instead that a user chooses the levels “Region” and “Year”; Table 2 shows the resulting table. The addition of new levels and new dimensions to a DW will increase the number of possible tables. The OLAP systems also allow users to visualize results using graphic displays (e.g., graphs, pie-charts).

Here, we sum up the main components of the DW architecture (Berson and Smith, 1997). Typically, data extracted from different sources is cleaned and loaded in the DW using Extraction–Transformation–Loading (ETL) tools (e.g., Talend[2] or Pentaho[3] tools). At the implementation level, data can be stored in a relational DB using a specific structure (e.g., a star schema, a snowflake schema). Unlike traditional DB structures, DWs allow denormalized physical implementations; in other words, redundancies (repeated values) are tolerated in DWs, which speeds up data access time. Specific techniques such as materialized views, indexes and caching methods can also be implemented to improve the performance. OLAP servers are used to perform queries by exploiting a mapping between the multidimensional concepts of dimensions, facts, measures, aggregation functions and relational DB elements (e.g., tables and columns). Users communicate with these servers via OLAP clients. These tools offer a visual representation of query results by means of tabular (pivot tables) and graphic displays.

In Table 3, we propose a comparison of the DW/OLAP technologies versus spreadsheet tools, which constitute another possible method for data exploration. Table 3 shows that DW and spreadsheet tools present similar modes of visualization (e.g., information can be displayed in tables and charts), but spreadsheets are limited in terms of security, access time for large volumes of information and Intranet/Internet data diffusion.

DW technologies have been widely used in the retail sector, but many other potential applications exist. For example, integration of different data sources into a DW can also be performed for epidemiological studies (Bernier et al., 2009). However, DWs of environmental data are still quite rare, even though they could eventually generate a strong interest. Nilakanta et al. (2008), Pinet et al. (2010), and Schulze et al. (2007) provide a few examples of DWs for storing and visualizing environmental data. The use of DW technology for the storage of environmental simulation results is a new domain of application (Mahboubi et al., 2010).

Data quality in DWs is important because these systems are used in support of decisional processes. Poor data quality can lead to an inadequate analysis (Carpani and Ruggia, 2001; Ghozzi et al., 2003; Prat et al.,

[2] http://www.talend.com.
[3] http://www.pentaho.com.

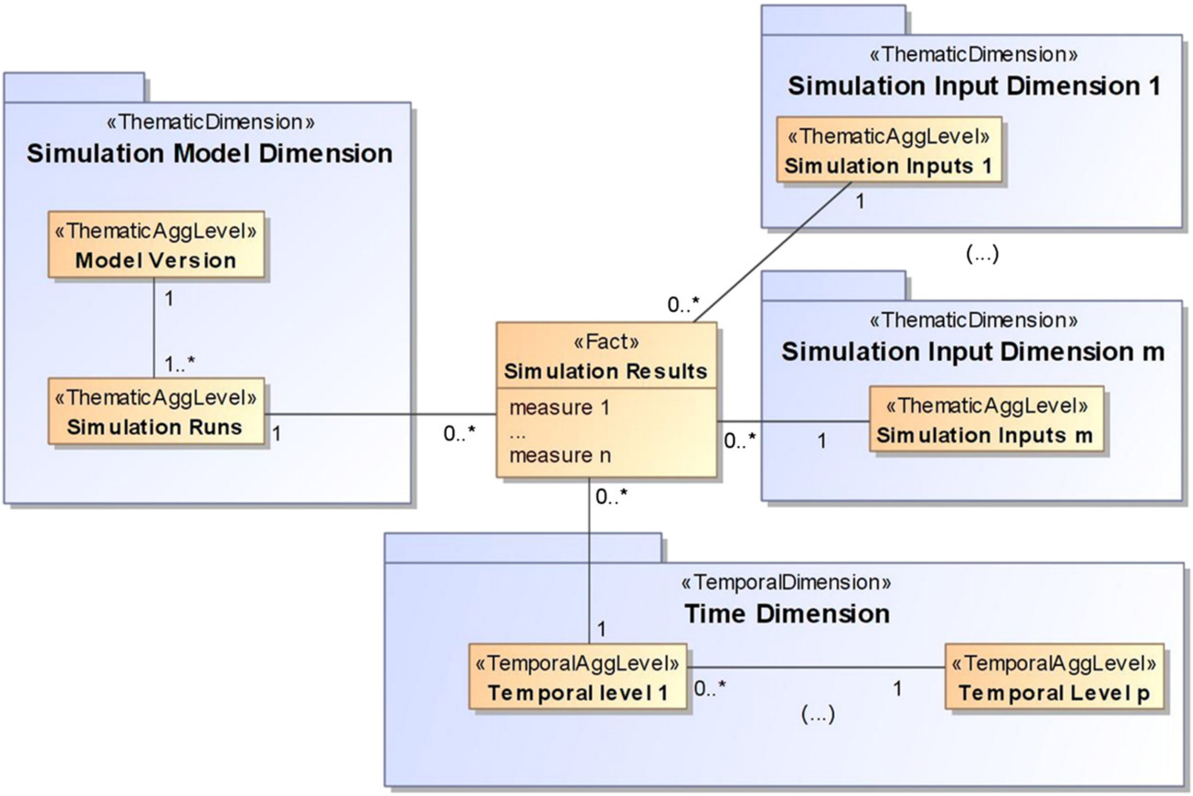


Fig. 2. A generic conceptual multidimensional schema for the storage of simulation results (inspired from the proposal of Mahboubi et al. (2010)).

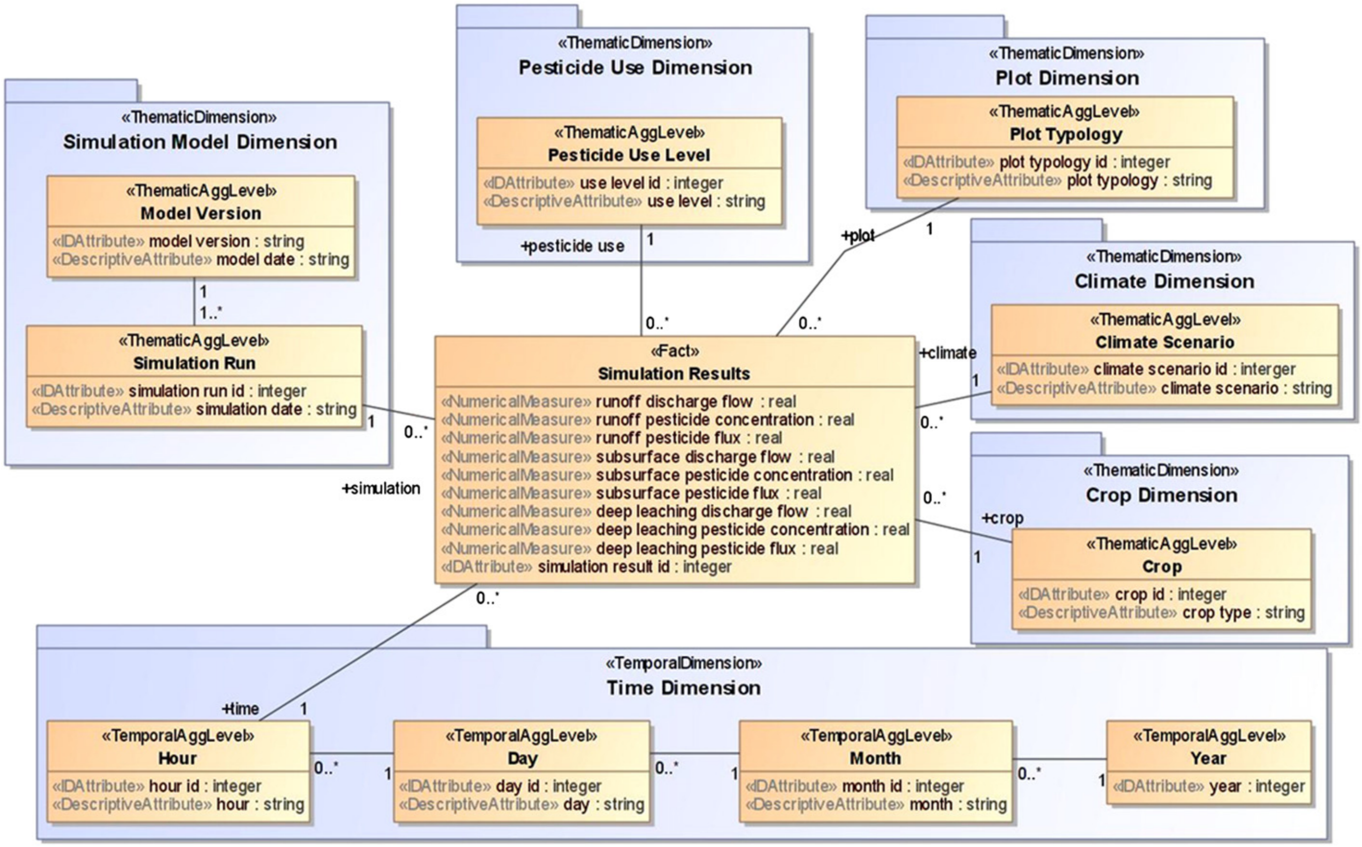

«ThematicDimension»
Simulation Model Dimension
«ThematicAggLevel»
Model Version
«IDAttribute» model version : string
«DescriptiveAttribute» model date : string
1
1..*
«ThematicAggLevel»
Simulation Run
«IDAttribute» simulation run id : integer
«DescriptiveAttribute» simulation date : string
1
0..*
+simulation
«ThematicDimension»
Pesticide Use Dimension
«ThematicAggLevel»
Pesticide Use Level
«IDAttribute» use level id : integer
«DescriptiveAttribute» use level : string
+pesticide use
1
0..*
«ThematicDimension»
Plot Dimension
«ThematicAggLevel»
Plot Typology
«IDAttribute» plot typology id : integer
«DescriptiveAttribute» plot typology : string
+plot
1
0..*
«Fact»
Simulation Results
«NumericalMeasure» runoff discharge flow : real
«NumericalMeasure» runoff pesticide concentration : real
«NumericalMeasure» runoff pesticide flux : real
«NumericalMeasure» subsurface discharge flow : real
«NumericalMeasure» subsurface pesticide concentration : real
«NumericalMeasure» subsurface pesticide flux : real
«NumericalMeasure» deep leaching discharge flow : real
«NumericalMeasure» deep leaching pesticide concentration : real
«NumericalMeasure» deep leaching pesticide flux : real
«IDAttribute» simulation result id : integer
«ThematicDimension»
Climate Dimension
+climate
1
0..*
«ThematicAggLevel»
Climate Scenario
«IDAttribute» climate scenario id : integer
«DescriptiveAttribute» climate scenario : string
«ThematicDimension»
Crop Dimension
0..*
+crop
1
«ThematicAggLevel»
Crop
«IDAttribute» crop id : integer
«DescriptiveAttribute» crop type : string
0..*
«TemporalDimension»
Time Dimension
+time
1
«TemporalAggLevel»
Hour
«IDAttribute» hour id : integer
«DescriptiveAttribute» hour : string
0..*
1
«TemporalAggLevel»
Day
«IDAttribute» day id : integer
«DescriptiveAttribute» day : string
0..*
1
«TemporalAggLevel»
Month
«IDAttribute» month id : integer
«DescriptiveAttribute» month : string
0..*
1
«TemporalAggLevel»
Year
«IDAttribute» year : integer

Fig. 3. Our conceptual multidimensional schema proposed for the storage of MACRO result data.

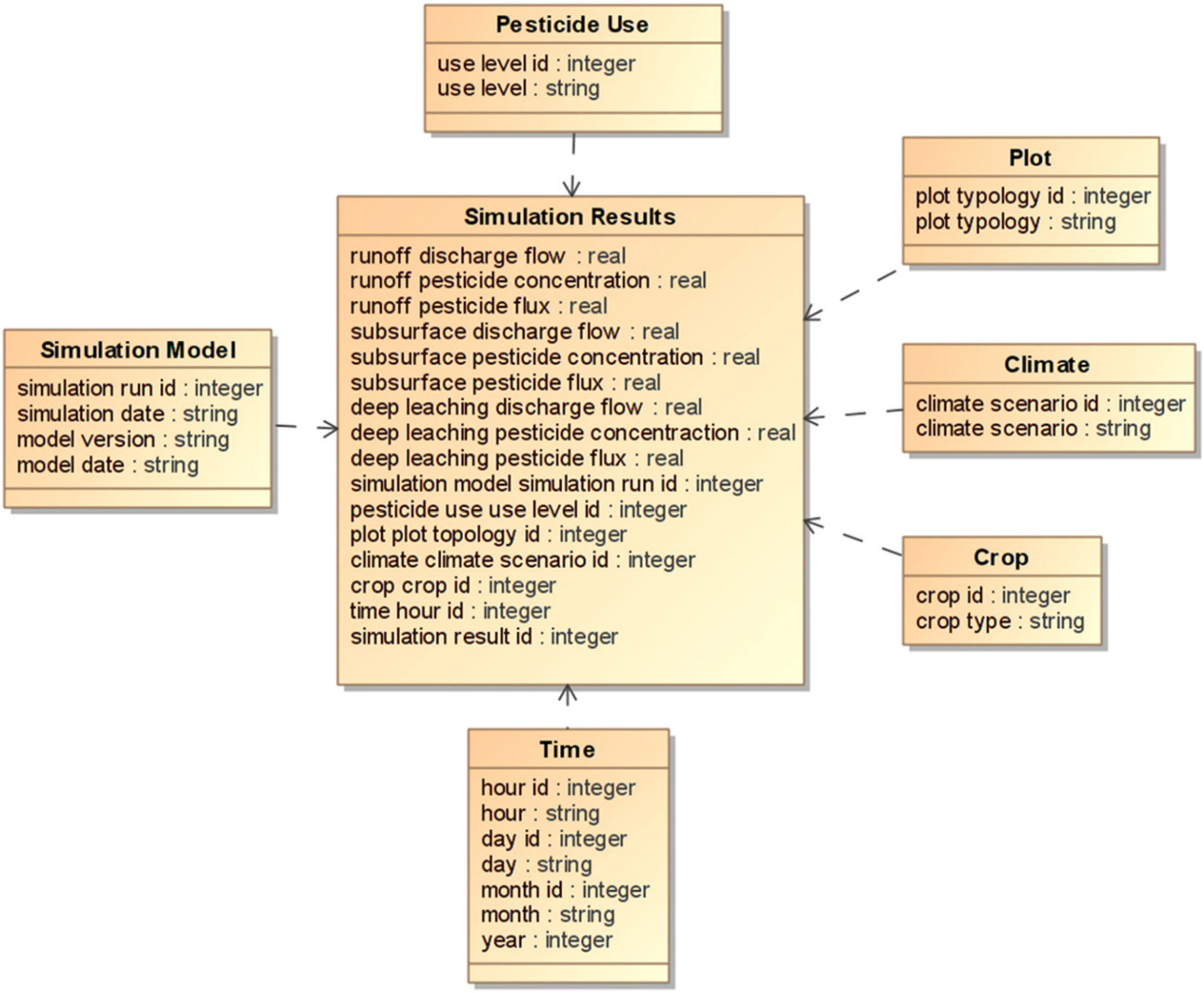

Pesticide Use
use level id : integer
use level : string
Plot
plot typology id : integer
plot typology : string
Simulation Results
runoff discharge flow : real
runoff pesticide concentration : real
runoff pesticide flux : real
subsurface discharge flow : real
subsurface pesticide concentration : real
subsurface pesticide flux : real
deep leaching discharge flow : real
deep leaching pesticide concentraction : real
deep leaching pesticide flux : real
simulation model simulation run id : integer
pesticide use use level id : integer
plot plot topology id : integer
climate climate scenario id : integer
crop crop id : integer
time hour id : integer
simulation result id : integer
Simulation Model
simulation run id : integer
simulation date : string
model version : string
model date : string
Climate
climate scenario id : integer
climate scenario : string
Crop
crop id : integer
crop type : string
Time
hour id : integer
hour : string
day id : integer
day : string
month id : integer
month : string
year : integer

Fig. 4. The DW implementation schema.

2010; Salehi, 2009). To maintain a correct quality level in DW systems, Boulil et al. (2012) propose a UML-based architecture that provides a conceptual formalism to represent the stored data and the associated integrity constraints. Code generation techniques are also introduced, to implement and check automatically these constraints in DWs. In comparison with other proposed techniques (Carpani and Ruggia, 2001; Ghozzi et al., 2003; Prat et al., 2010; Salehi, 2009), the method of Boulil et al. (2012) has a high expressive power; this method allows users to specify and implement three types of integrity constraints:

(a) Data constraints that ensure that warehoused data are consistent, e.g., geometries of cities must be topologically included in the geometries of their states;
(b) Aggregation constraints that ensure that aggregations of measures are correct and meaningful, e.g., the sum of the temperatures usually does not make sense (Lenz and Shoshani, 1997);
(c) Exploration constraints that alert users when OLAP queries can return inconsistent results, e.g., the query “sales per country after December 26, 1991” must return empty results for the USSR.

3. Overview of our UML profile for data warehouses

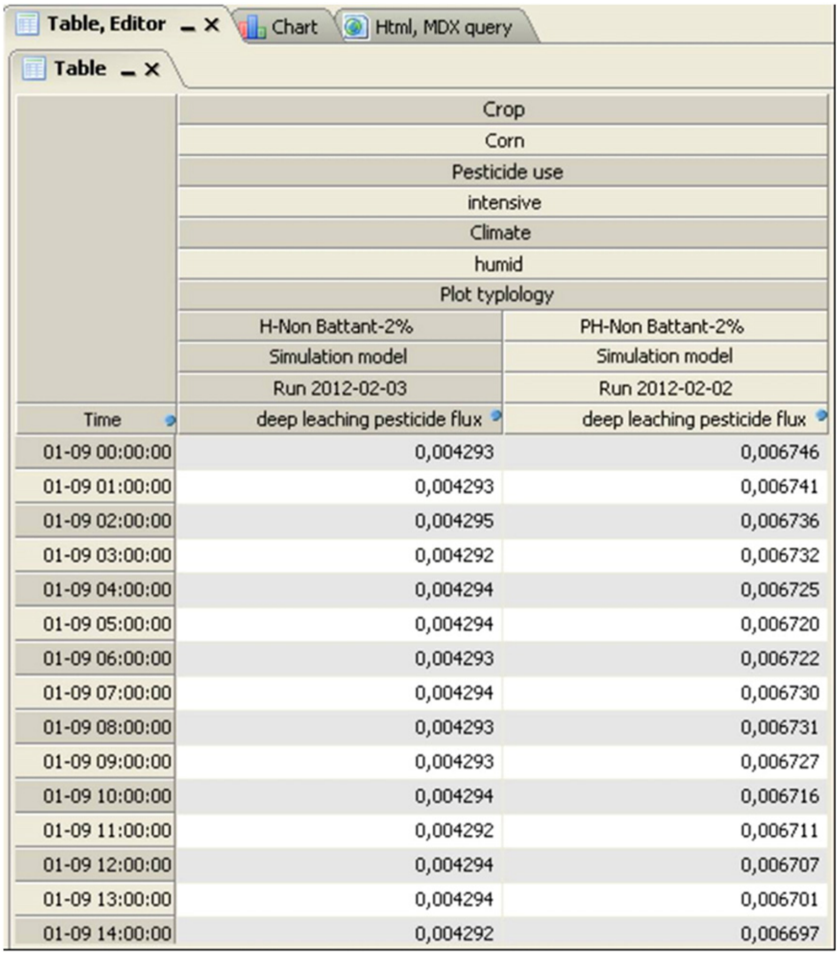
A UML profile provides a generic extension mechanism for customizing UML for specific applications (e.g., geographical information systems, web-services) (Pinet and Schneider, 2010). Profiles are mainly defined using stereotypes and tagged values that are applied to specific UML elements, such as classes, attributes and operations. A Profile is a collection of stereotypes and tagged values that modelers can reuse in different UML diagrams.

Stereotypes specialize UML elements and provide additional semantic information on UML diagrams. They are rendered as a name enclosed by ≪ ≫ in UML diagrams. For example, in Fig. 1:

- the stereotype ≪Fact≫ (which specializes the “class” UML element) is associated with the fact class “Irrigation”;
- the stereotype ≪NumericalMeasure≫ indicates that the attribute “quantity of water in m^3” is a measure that has a numeric type; this stereotype specializes the “property” UML element;
- identifier attributes and the other attributes can be distinguished with the stereotypes ≪IDAttribute≫ and ≪DescriptiveAttribute≫.

Table 4 shows the main stereotypes of our UML profile (Boulil et al., 2012). This profile is dedicated to the conceptual design of DW schemas (e.g., class diagrams).

Our UML profile for DWs can also be used to specify integrity constraints. A set of stereotypes and tagged values has been proposed to facilitate this specification (Boulil et al., 2012); tagged values allow modelers to associate a couple {variable = values} to UML diagram elements. Constraints specified in the Object Constraint Language (OCL) can also be used to express ICs. OCL is a standard and platform-independent language that is dedicated to the specification of constraints on UML diagrams (Pinet and Schneider, 2010). This language integrates notations that are close to a spoken language to express conditions. Examples will be provided in Section 5.2.



	Crop	
	Corn	
	Pesticide use	
	intensive	
	Climate	
	humid	
	Plot typlology	
	H-Non Battant-2%	PH-Non Battant-2%
	Simulation model	Simulation model
	Run 2012-02-03	Run 2012-02-02
Time	deep leaching pesticide flux	deep leaching pesticide flux
01-09 00:00:00	0,004293	0,006746
01-09 01:00:00	0,004293	0,006741
01-09 02:00:00	0,004295	0,006736
01-09 03:00:00	0,004292	0,006732
01-09 04:00:00	0,004294	0,006725
01-09 05:00:00	0,004294	0,006720
01-09 06:00:00	0,004293	0,006722
01-09 07:00:00	0,004294	0,006730
01-09 08:00:00	0,004293	0,006731
01-09 09:00:00	0,004293	0,006727
01-09 10:00:00	0,004294	0,006716
01-09 11:00:00	0,004292	0,006711
01-09 12:00:00	0,004294	0,006707
01-09 13:00:00	0,004294	0,006701
01-09 14:00:00	0,004292	0,006697

Fig. 5. Visualization of the data of two simulation runs with JRubik.

Our profile for DWs has been implemented with a UML-based tool called MagicDraw.[4] This implementation allows modelers to graphically design the DW conceptual schema using our UML profile and to check its validity (an absence of errors and contradictions in the schema).

4. Data warehouse of simulation result data: Related studies

The use of DW and OLAP systems for the analysis of simulation model results is quite new. To the best of our knowledge, only Mahboubi et al. (2010) and Vasilakis et al. (2004) address this topic. The preliminary work of Vasilakis et al. (2004) proposes a multidimensional model for analyzing the results of a simulation model for the patient flow through hospital departments. Mahboubi et al. (2010) presents a generic multidimensional schema that is dedicated to simulation results. This schema defines different concepts that can be found in a DW of simulation results.

The concepts that we can find in a DW of simulation data are summarized in the template schema of Fig. 2. The schema contains the following facts and main dimensions:

- The “Simulation Results” facts: These facts are composed of n measures that are calculated from simulation runs that can be visualized and analyzed according to different dimensions. In the schema, a simulation result is linked to its simulation runs, its set of inputs $(1,\ldots,m)$ and an instant (or a period) of time. These three types of dimensions are described below.
- The “Simulation Model” dimension: This dimension is important for the comparison of model runs or model versions. It can be organized into a hierarchy that is composed of two levels: (1) the simulation runs and (2) the model versions. Each simulation run is associated with a model version.
- The “Simulation Input” dimensions: These dimensions represent observed data (e.g., rain, land use) or variables/parameters of the models (e.g., parameters that describe the crop growth or the soil characteristics). Several “simulation input” dimensions can be present in the same multidimensional schema (i.e., in the same DW).
- The Time dimension: Environmental simulation runs usually model the evolution of a phenomenon and produce results at different time steps. This dimension indicates the periods/instants of time that are associated with the simulation results. Different levels $(1,\ldots,p)$ of this dimension can be applied (e.g., day, month, year).

Several UML notations are used in the diagram. ≪TemporalDimension≫ or ≪TemporalAggLevel≫ indicates that a dimension or a level is related to time (Boulil et al., submitted for publication). ≪ThematicDimension≫ and ≪ThematicAggLevel≫ are used to label other dimensions and levels.

In Mahboubi et al. (2011) and Mahboubi et al. (2013), an implementation for this type of data warehouse is proposed. Starting from the simulation model data structure, the tool proposed in these papers derives the possible multidimensional schemas and provides the possibility of implementing them in a relational OLAP architecture. A case study that concerns the analysis of result data that is produced by a demographic simulation model has been used to illustrate this proposal (Mahboubi et al., 2011), but our UML profile and our IC specification method have not been used in this application. These two points (applied to the MACRO model) constitute the originality of our new contribution, which is presented in our paper.

[4] http://www.nomagic.com.

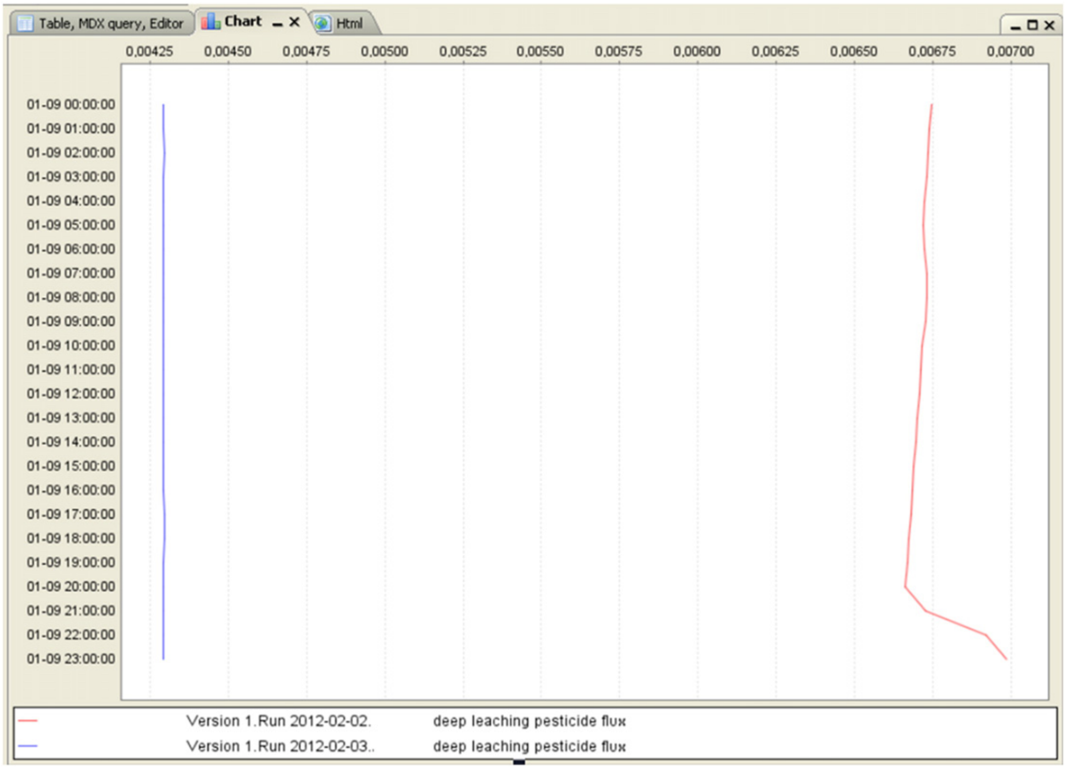


Fig. 6. An example chart produced by JRubik that compares deep leaching pesticide flux from two simulation runs.

5. A data warehouse for the pesticide transfer simulation model MACRO

In this section, we introduce the MACRO simulation model and we detail our DW for the storage of data that results from this environmental model (Section 5.1). We also present how to define ICs and implement them using the framework proposed by Boulil et al. (2012) (Section 5.2). In Section 5.3, we discuss our proposal, including its main contributions and limits.

MACRO is a physically based model that simulates water and pesticide transfers using a dimensional approach for both microporous and macroporous media (Larsbo and Jarvis, 2003). Water flow in microporous parts of soil is described by the classical Richards equation (Richards, 1931), which uses a kinematic wave to describe macroporous flow. Pesticide transfer is represented by an advection–diffusion equation, including sorption and degradation processes, in both microporous and macroporous parts of the soil. For each time step of a simulation, this model calculates the discharge and pesticide concentrations. These values are calculated for the following flow components: surface runoff; deep percolation under the soil profile; and tile drainage.

5.1. Data warehouse and OLAP

The schema in Fig. 3 shows the structure of the DW that is used to store the MACRO results. The facts are stored in the Simulation Results class (e.g., the data results produced from the simulation runs). The class is composed of different measures:

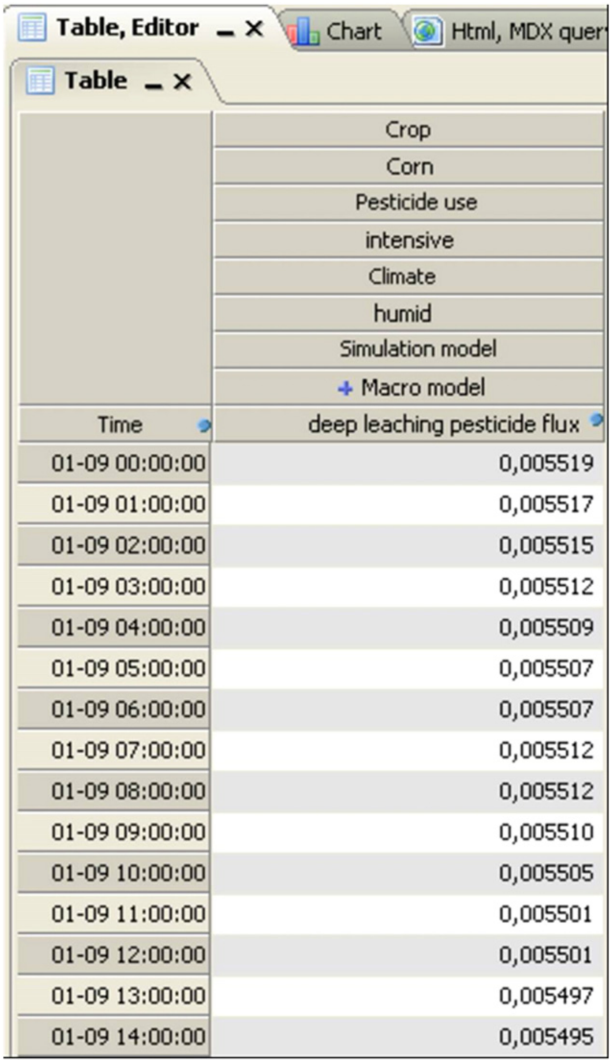
- Discharge flow $[L^3 \cdot T^{-1}]$;
- Concentration of considered pesticide in the flow $[M \cdot L^{-3}]$; and
- Flux of pesticide in the flow $[M \cdot T^{-1}]$,

  where M is the mass, L is the length and T is the time. Each of these three values can be associated with surface runoff, subsurface lateral fluxes or deep leaching.

  Our multidimensional schema includes four “Simulation Input” dimensions, which are linked to the following facts:

- Plot typology: The MACRO model needs a typology of plots. To avoid simulating each individual catchment field plot, each plot is linked to a “typical” plot, which depends on the soil type, the slope and whether the soil surface is slaking; thus, this approach dramatically reduces the number of simulations that are required.
- Climate scenario: Similarly, each climatic year is linked to a typical scenario based on the total amount of annual rainfall. The previous year is also considered because it influences the initial moisture content of the soil.
- Crop: This class indicates the considered crop.
- Level of pesticide use: Different strategies are considered (i.e., different levels of quantities of pesticide used).

The measures are calculated by the MACRO model for each hour, as indicated by the associations between the Simulation Results class (the fact class) and the Hour class (a dimension level), which is the smallest unit in the time dimension.



	Crop
	Corn
	Pesticide use
	intensive
	Climate
	humid
	Simulation model
	+ Macro model
Time	deep leaching pesticide flux
01-09 00:00:00	0,005519
01-09 01:00:00	0,005517
01-09 02:00:00	0,005515
01-09 03:00:00	0,005512
01-09 04:00:00	0,005509
01-09 05:00:00	0,005507
01-09 06:00:00	0,005507
01-09 07:00:00	0,005512
01-09 08:00:00	0,005512
01-09 09:00:00	0,005510
01-09 10:00:00	0,005505
01-09 11:00:00	0,005501
01-09 12:00:00	0,005501
01-09 13:00:00	0,005497
01-09 14:00:00	0,005495

Fig. 7. The average of the measures from two simulation runs as displayed within JRubik.

Our schema also includes a simulation model dimension. In our experiment, results produced by two MACRO simulations of the watershed of La Fontaine du Theil in Brittany (France) were stored in the DW. In total, 2 years (a total of 17,520 h) of data was simulated and stored. Here, only one model version was used.

We only used free software to implement our solution. Our DW has been implemented with PostgreSQL.[5] We chose two other popular tools to display and explore data: Mondrian[6] for the OLAP server and JRubik[7] as the OLAP client. Mondrian is an open-source OLAP server that builds OLAP logical structures (e.g., dimensions, measures) on top of any DB using a specific XML file. JRubik is a software package that provides a graphical presentation layer on top of Mondrian. This layer constitutes a set of user-friendly interfaces that trigger and display OLAP queries.

At the implementation level, we used a star schema (see Fig. 4). This implementation uses a fact table. Its attributes include measures and foreign keys that are linked to different dimensions. A fact table contains measure values at the most detailed levels. Measure values are stored by the hour, crop, climate scenario, plot typology, pesticide use level and simulation run. This schema also shows dimension tables. This type of table contains denormalized representations of dimension hierarchy levels; in other words, all of the levels of a dimension are stored in the same table (see the Time table). Denormalization is used very often in DWs. This method produces redundancies (repetitions) of values but vastly improves the data access time.

OLAP tools allow a user to compose a table by choosing the desired dimension levels in the multidimensional schema. Fig. 5 shows a part of the table that results from the selection of the measure “deep leaching pesticide flux” and the lowest dimension levels in JRubik. Each type of table entry corresponds to a dimension level that is presented in Fig. 3. The chart in Fig. 6 provides another example of data visualization produced with JRubik. This chart shows the evolution of the measure hour by hour for each simulation run. Many other types of graphics can be created.

Next, suppose that users want to compare two simulation runs. To accomplish this goal, they can select the aggregation function “average” and the level “Model Version” instead of “Simulation Run”. This selection will produce the average deep leaching pesticide flux between the two runs (see Fig. 7).

[5] http://www.postgresql.org.

[6] http://mondrian.pentaho.com.

[7] http://rubik.sourceforge.net.

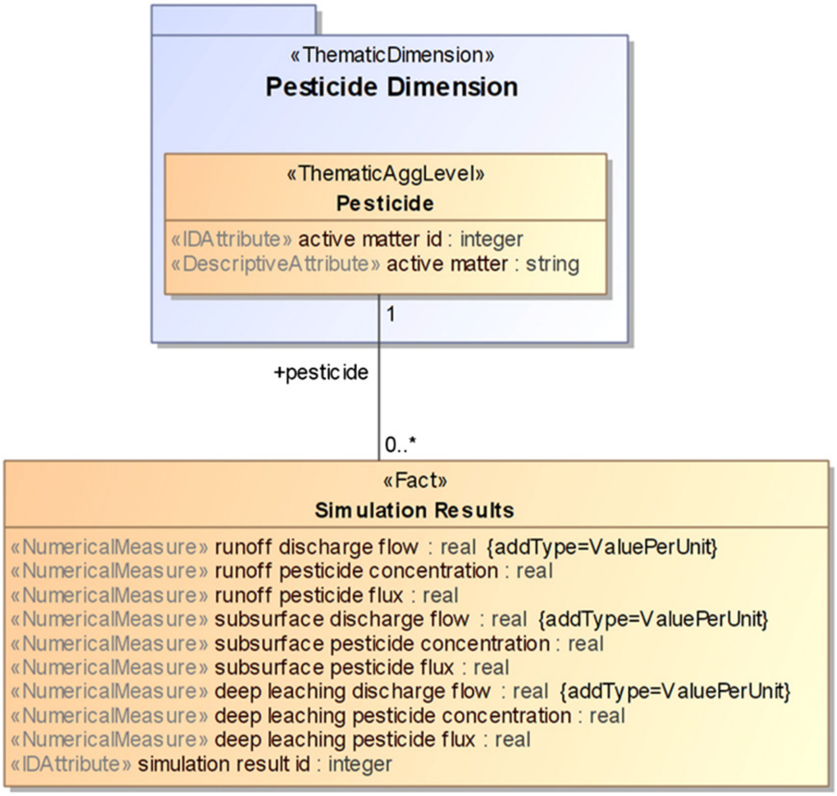

«ThematicDimension»
Pesticide Dimension
«ThematicAggLevel»
Pesticide
«IDAttribute» active matter id : integer
«DescriptiveAttribute» active matter : string
1
+pesticide
0..*
«Fact»
Simulation Results
«NumericalMeasure» runoff discharge flow : real {addType=ValuePerUnit}
«NumericalMeasure» runoff pesticide concentration : real
«NumericalMeasure» runoff pesticide flux : real
«NumericalMeasure» subsurface discharge flow : real {addType=ValuePerUnit}
«NumericalMeasure» subsurface pesticide concentration : real
«NumericalMeasure» subsurface pesticide flux : real
«NumericalMeasure» deep leaching discharge flow : real {addType=ValuePerUnit}
«NumericalMeasure» deep leaching pesticide concentration : real
«NumericalMeasure» deep leaching pesticide flux : real
«IDAttribute» simulation result id : integer

Fig. 8. The active matter dimension.

In the system that we implemented, we had only one type of pesticide. If we want to have a DW that stores simulation results that are produced for different categories of pesticides, we must add a pesticide active matter dimension (see Fig. 8). The other dimensions remain unchanged.

5.2. Integrity constraints

5.2.1. Formalization of integrity constraints

Here, we introduce some examples of ICs that can be expressed with our UML/OCL-based method proposed in Boulil et al. (submitted for publication). These ICs are defined on the conceptual multidimensional schema that was previously presented (see Figs. 3 and 8). As stated above (Section 2), three types of DW integrity constraints must be considered to ensure a good quality of the analysis: data IC, exploration IC and aggregation IC.

5.2.1.1. Data IC. Data ICs are specified using OCL (Boulil et al., 2012). For example, suppose that we use a transfer model with the pesticide active matter “sulcotrione”. The following constraint expresses that the pesticide active matter “sulcotrione” should be found in the DW.

$$\begin{array}{l}\texttt{context pesticide inv:}\\ \underbrace{\texttt{Pesticide.allInstances ( )}}_{(A)}\rightarrow\underbrace{\texttt{exists (active\_matter =' sulcotrione')}}_{(B)}\end{array}$$

The OCL function “allInstances ( )” in part (A) returns the instances of the Pesticide class. Part (B) checks whether there is at least one instance that satisfies the condition “active_matter =‘sulcotrione’”. Note that active_matter is an attribute of the Pesticide class.

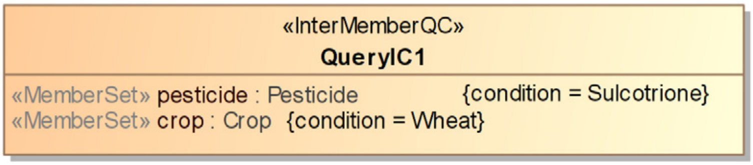
The following constraint states that the pesticide active matter “sulcotrione” cannot be used with the crop “wheat”:

```
(C1) context Simulation_Results inv:
(C2) not (
(C3)    self.pesticide.active_matter = ‘sulcotrione’ and
(C4)    self.crop.crop_type = ‘wheat’ ).
```

The context of an OCL constraint is a class. An OCL constraint must always be satisfied for each instance of this class. The term “self” denotes an instance of the class.

Line (C1) indicates that the context of the constraint is the Simulation Results class (Fig. 8). Line (C2) shows that the negation of the condition (active_matter = ‘sulcotrione’ and crop_type = ‘wheat’) must be satisfied. In line (C3), “self.pesticide” returns the instance of the Pesticide class that is associated with self; active_matter is an attribute of the Pesticide class. In line (C4), “self.crop” returns the instance of the Crop class that is associated with self; crop_type is an attribute of the Crop class (Fig. 3).

Data ICs can be checked during or after the data loading phase of the ETL process. This step allows for a confirmation that the simulation model provides good data or, in other words, that the model is correct from theoretical and implementation points of view.


«InterMemberQC»
QueryIC1
«MemberSet» pesticide : Pesticide {condition = Sulcotrione}
«MemberSet» crop : Crop {condition = Wheat}

```
context Pesticide inv Sulcotrione:

self.active_matter = 'sulcotrione'

context Crop inv Wheat: self.crop_type = 'wheat'
```

Fig. 9. An example of exploration IC.

5.2.1.2. Exploration IC. Alternatively, it is also possible to check constraints during DW explorations (i.e., during the data display with OLAP tools). Exploration IC represents an "invalid" combination of dimensional elements and can be expressed using our UML profile and OCL (Boulil et al., 2012). Fig. 9 shows the previous IC formalized as an exploration IC (i.e., a query constraint). The QueryIC1 class shows level instances that the user should not combine within the OLAP tool. Two OCL constraints are defined to select these level instances. These constraints allow users to have a good interpretation of query results according to the semantics of the data.

Here, we insist on the differences between data ICs and exploration ICs. Data ICs are checked at the database level. They are used by database managers to forbid the insertion of data that do not comply with the constraints or to select inconsistent data. In the latter case, the data can be deleted or "repaired"; the end users will never see these inconsistent data. In comparison, exploration ICs are used at the OLAP level (when the end-users visualize the data). Exploration ICs automatically alert end-users that the displayed data could contain inconsistencies.

5.2.1.3. Aggregation IC. Aggregation ICs can be specified with OCL as inherent constraints of our UML profile for DWs. For example, an aggregation IC has been defined in OCL inside our profile to indicate that the sum aggregation function cannot be used on "value per unit" measures (such as the runoff discharge flow measure in Fig. 8). This constraint is checked by MagicDraw at the conceptual design phase when the designer validates the DW schema.

5.2.2. Implementation of integrity constraints

Once ICs have been defined at the conceptual level using UML and OCL in MagicDraw and they have been validated by end-users, we can automatically implement them. We show in this section the different techniques that we propose to produce integrity-checking mechanisms.

We can automatically translate data ICs into SQL code using the OCL2SQL code generator (Demuth et al., 2001). The produced implementation can be used in the DW tier. For example, the type of SQL query generated from the second data IC presented in Section 5.2.1 will be the following:

```
 (Q1) select * from SIMULATION_RESULTS SELF where not (
 (Q2) NOT ((
 (Q3) (select ACTIVE_MATTER from PESTICIDE where ACTIVE_MATTER_ID in
 (Q4) (select PESTICIDE_ACTIVE_MATTER_ID from SIMULATION_RESULTS where SIMULATION_RESULT_ID =
 (Q5)                                                          SELF.SIMULATION_RESULT_ID))
 (Q6)                 = 'sulcotrione'
 (Q7)                 AND
 (Q8)  (select CROP_TYPE from CROP where CROP_ID in
 (Q9)  (select CROP_CROP_ID from SIMULATION_RESULTS where SIMULATION_RESULT_ID =
(Q10)                                             SELF. SIMULATION_RESULT_ID))
(Q11)                 = 'wheat'
(Q12)  ))).
```

This SQL query selects data (tuples) of the Simulation_Results table that do not satisfy the condition of the constraint (see line Q1). CROP_CROP_ID and PESTICIDE_ACTIVE_MATTER_ID are foreign keys in table Simulation_Results. If we compare this query with the associated OCL constraint:

- Line Q2 corresponds to line C2;
- Lines Q3–Q7 correspond to line C3;
- Lines Q8–Q12 correspond to line C4.

Table 5
Example of pivot table visualization with exploration IC checking.

	Wheat	Ray-grass
Sulcotrione	0.002000	0.008000
Ioxynil	0.002000	0.004000

OCL2SQL integrates this type of query in triggers and object views. This query has been generated with the enhanced version of OCL2SQL introduced in Pinet et al. (2007). More details on OCL2SQL can be found in Demuth et al. (2001).

Aggregation ICs are implemented as inherent constraints of our UML profile, which is implemented in MagicDraw (Boulil et al., 2012). They are defined using OCL and do not need any translation.

Exploration ICs are translated into MDX code by our automatic code generator UML2MDX (Boulil et al., 2012) and are implemented in the OLAP server tier. MDX (MultiDimensional eXpressions language) is a standard language for querying multidimensional and OLAP databases; MDX is analogous to SQL for relational databases. The MDX code that is generated for exploration ICs defines a visual policy to display differently cells in the JRubik pivot table. More precisely, the generated code changes the color of the pivot table cells to indicate to the users that certain cells do not satisfy exploration ICs. Table 5 provides a short example of this type of visualization. Suppose that a DW includes two active matters (sulcotrione and ioxynil) and two types of crop (wheat and ray-grass). In this table, the exploration IC in Fig. 9 is considered. The MDX code generated from this IC will display the pivot table cells in different colors:

- Invalid cells will be displayed with a red color — the cell combining “sulcotrione” with “wheat” is in red;
- Valid cells such as the cell combining “ioxynil” and “wheat” will be displayed with a green color.

It is important to show to users that a cell is invalid because it defines an impossible (unplayable) combination of members. In Table 5, the cell combining “sulcotrione” and “wheat” contains a non-null value; consequently, the exploration IC is not satisfied because this constraint has defined this combination as invalid. The red color of this cell will show to users that this value is an error.

5.3. Discussion

In this section, we presented a multidimensional model that was conceived for the analysis of pesticide transfer data and a set of integrity constraints to guarantee the quality of the analysis. This work validates the idea of using DWs (Mahboubi et al., 2010) and ICs for simulation results, which, in spite of its important possibilities, has been investigated very little. We have also shown that the implementation can be facilitated by using a UML-based framework that is dedicated to the design of multidimensional schemas and their ICs. The UML-based conceptual models allow designers, decision makers and application domain experts to exchange about analysis goals and data. It eases the design process of complex applications such as environmental applications, where several DW schema versions must usually be defined in an iterative way.

In our work, we have mixed data- and goal-oriented design approaches (Giorgini et al., 2008). The decision makers have defined their analysis requirements in terms of conceptual multidimensional schemas and ICs (using our UML profile). We have noticed that, in the specific case of DWs for simulation results, the choice of the stored data is completely correlated to the used simulation model. Finally, this project has also highlighted the need for a tool for the automatic integration of data results into databases. Cartographic display support can also be required for georeferenced data (Bimonte, 2010).

6. Conclusions and future research

The volumes of simulation data continue to increase and model users now need a technology that is specifically dedicated to storing and exploring the vast quantities of information that is produced by the models. As demonstrated in the experiment presented in this paper, DW/OLAP technologies provide an excellent means for managing simulation data.

Here, we summarize the methodology that we followed to implement our DW. First, we adapted the generic multidimensional schema proposed in Mahboubi et al. (2010) to the target model. We chose measures and input dimensions that were suitable for the MACRO environmental model. The temporal granularity of MACRO provides the lowest level of the temporal dimension (i.e., 1 h). We also added other temporal levels to allow users to aggregate measures per month or year. The DB schema was created with PostgreSQL. We chose a star schema (i.e., a denormalized implementation) to ensure the best data access performance. Then, the results data were loaded into the DB from text files that were produced by MACRO runs. Mondrian and JRubik allow users to explore DW data very easily by choosing the desired levels of analysis. Researchers can benefit from the user-friendly JRubik graphic user interface to change the dimension level, to traverse level hierarchies, to change the aggregate functions or to visualize a subset of the results data. Our architecture also allows users to export a selection of data into other formats. We also present the first experiment of the method introduced in Boulil et al. (2012) to specify integrity constraints in DWs.

The next steps of our work will be to enrich and improve the generic multidimensional schema by experimenting with other environmental models and by considering feedback from users. We also plan to study the possibilities offered by spatial DW/OLAP (Bédard et al., 2009; Bimonte, 2010; Malinowski and Zimanyi, 2008). These recent tools are dedicated to displaying information that is stored in DWs by means of maps and can be useful for visualizing georeferenced environmental models. The extraction, transformation and loading processes are also important aspects of the architecture and we also plan to propose a tool to facilitate these steps. To do so, we have started to extend SimExplorer (Chuffart et al., 2010), a software application for managing simulation runs, to help users create a DW and load model output.

Acknowledgments

This work has been funded by Irstea, the French “Ministère de l'Ecologie, du Développement durable et de l'Energie” (project Miriphyque) and the Auvergne Region.

References

Basta, A., Zgola, M., 2011. Database Security Delmar Cengage Learning.
Bédard, Y., Bernier, E., Larrivée, S., Nadeau, M., Proulx, M.-J., Rivest, S., 2009. Spatial OLAP. http://www.spatialbi.com/.
Bernier, E., Gosselin, P., Badard, T., Bédard, Y., 2009. Easier surveillance of climate-related health vulnerabilities through a Web-based spatial OLAP application. International Journal of Health Geographics 8. http://www.ij-healthgeographics.com/content/8/1/18.
Berson, A., Smith, S., 1997. Data Warehousing, Data Mining, and OLAP (Data Warehousing/Data Management). Computing Mcgraw-Hill.
Bimonte, S., 2010. A web-based tool for spatio-multidimensional analysis of geographic and complex data. International Journal of Agricultural and Environmental Information Systems 1, 42–67.
Boulil, K., Bimonte, S., Pinet, F., 2012. A UML & spatial OCL based approach for handling quality issues in SOLAP systems. 14th International Conference on Enterprise Information Systems.
Boulil, K., Bimonte, S., Pinet, F., 2013. A UML profile and OCL-based constraints for spatial data cubes. Information Systems (submitted for publication).
Carpani, F., Ruggia, R., 2001. An integrity constraints language for a conceptual multidimensional data model. SEKE 2001.
Chuffart, F., Dumoulin, N., Faure, T., Deffuant, G., 2010. SimExplorer: programming experimental design on models and managing quality of modelling process. International Journal of Agricultural and Environmental Information Systems 1, 55–68.

Demuth, B., Hußmann, H., Loecher, S., 2001. OCL as a specification language for business rules in database applications. Proceedings of the 4th International Conference on The Unified Modeling Language, Modeling Languages, Concepts, and Tools. Springer-Verlag, pp. 104–117.

Fernández-Quiruelas, V., Fernández, J., Cofiño, A.S., Fita, L., Gutiérrez, J.M., 2011. Benefits and requirements of grid computing for climate applications. An example with the community atmospheric model. Environmental Modelling & Software 26, 1057–1069.

Ghozzi, F., Ravat, F., Teste, O., Zurfluh, G., 2003. Constraints and Multidimensional Databases. ICEIS 2003.

Giorgini, P., Rizzi, S., Garzetti, M., 2008. GRAnD: a goal-oriented approach to requirement analysis in data warehouses. Decision Support Systems 45 (1), 4–21.

Hirabayashi, S., Kroll, C.N., Nowak, D.J., 2011. Component-based development and sensitivity analyses of an air pollutant dry deposition model. Environmental Modelling & Software 26, 804–816.

JRubik, 2012. http://rubik.sourceforge.net/jrubik/intro.html.

Larsbo, M., Jarvis, N.J., 2003. MACRO 5.0. A model of water flow and solute transport in macroporous soil. Technical Description, Studies in the Biogeophysical Environment.

Lenz, H.J., Shoshani, A., 1997. Summarizability in OLAP and Statistical Data Bases. IEEE 132–143.

Li, Z., Mao, X.-z., 2011. Global multiquadric collocation method for groundwater contaminant source identification. Environmental Modelling & Software 26, 1611–1621.

Mahboubi, H., Bimonte, S., Faure, T., Pinet, F., 2010. Data warehouse and OLAP for environmental simulation data. International Journal of Agricultural and Environmental Systems 1, 1–19.

Mahboubi, H., Bimonte, S., Deffuant, G., 2011. Analyzing demographic and economic simulation model results: a semi-automatic spatial OLAP approach. ICCSA (1), pp. 17–31.

Mahboubi, H., Bimonte, S., Deffuant, G., Chanet, J.P., Pinet, F., 2013. Semi-automatic design of spatial data cubes from structurally generic simulation model results. International Journal of Data Warehousing and Mining 9, 70–95.

Malinowski, E., Zimanyi, E., 2008. Advanced Data Warehouse Design: From Conventional to Spatial and Temporal Applications. Springer.

Miralles, A., Pinet, F., Carluer, N., Vernier, F., Bimonte, S., Lauvernet, C., Gouy, V., 2011. EIS Pesticide: An Information System for Data and Knowledge Capitalization and Analysis. PEER Euraqua, Montpellier, France.

Nilakanta, S., Scheibe, K., Rai, A., 2008. Dimensional issues in agricultural data warehouse designs. Computers and Electronics in Agriculture 60, 263–278.

No Magic, 2012. http://www.nomagic.com/products/magicdraw.html.

Pentaho, 2012. http://mondrian.pentaho.com/.

Pinet, F., Schneider, M., 2010. Precise design of environmental data warehouses. Operational Research 10, 349–369.

Pinet, F., Duboisset, M., Soulignac, V., 2007. Using UML and OCL to maintain the consistency of spatial data in environmental information systems. Environmental Modelling & Software 22, 1217–1220.

Pinet, F., Miralles, A., Bimonte, S., Vernier, F., Carluer, N., Gouy, V., Bernard, S., 2010. The use of UML to design agricultural data warehouses. International Conference on Agricultural Engineering. AGENG.

Pogson, M., Hastings, A., Smith, P., 2012. Sensitivity of crop model predictions to entire meteorological and soil input datasets highlights vulnerability to drought. Environmental Modelling & Software 29, 37–43.

Pokorný, J., 2006. Database architectures: current trends and their relationships to environmental data management. Environmental Modelling & Software 21, 1579–1586.

PostgreSQL, 2012. http://www.postgresql.org/.

Prat, N., Wattiau, I., Akoka, J., 2010. Representation of Aggregation Knowledge in OLAP Systems.

Richards, L.A., 1931. Capillary conduction of liquids through porous mediums. Journal of Applied Physics 1, 318–333.

Salehi, M., 2009. Developing a Model and a Language to Identify and Specify the Integrity Constraints in Spatial Datacubes. Université Laval.

Schulze, C., Spilke, J., Lehner, W., 2007. Data modeling for precision dairy farming within the competitive field of operational and analytical tasks. Computers and Electronics in Agriculture 59, 39–55.

Talend, 2012. Web site Talend.

Trolle, D., Hamilton, D.P., Pilditch, C.A., Duggan, I.C., Jeppesen, E., 2011. Predicting the effects of climate change on trophic status of three morphologically varying lakes: implications for lake restoration and management. Environmental Modelling & Software 26, 354–370.

Vasilakis, C., El-Darzi, Elia, Chountas, Panagiotis, 2004. A data warehouse environment for storing and analyzing simulation output data. Winter Simulation Conference, pp. 703–771.