DATA WAREHOUSE METADATA

Panos Vassiliadis

Department of Computer Science, University of Ioannina

Ioannina, Hellas

pvassil@cs.uoi.gr , http://www.cs.uoi.gr/~pvassil

SYNONYMS

None

DEFINITION

Data warehouse metadata are pieces of information stored in one or more special-purpose *metadata repositories* that include (a) information on the contents of the data warehouse, their location and their structure, (b) information on the processes that take place in the data warehouse back-stage, concerning the refreshment of the warehouse with clean, up-to-date, semantically and structurally reconciled data, (c) information on the implicit semantics of data (with respect to a common enterprise model), along with any other kind of data that aids the end-user exploit the information of the warehouse, (d) information on the infrastructure and physical characteristics of components and the sources of the data warehouse, and, (e) information including security, authentication, and usage statistics that aids the administrator tune the operation of the data warehouse as appropriate.

HISTORICAL BACKGROUND

Data warehouses are systems with significant complexity in their architecture and operation. Apart from the central data warehouse itself, which typically involves an elaborate hardware architecture, several sources of data, in different operational environments are involved, along with many clients that access the data warehouse in various ways. The infrastructure complexity is only one part of the problem; the largest part of the problem lies in the management of the data that are involved in the warehouse environment. Source data with different formats, structure, and hidden semantics are integrated in a central warehouse and then, these consolidated data are further propagated to different end-users, each with a completely different perception of the terminology and semantics behind the structure and

content of the data offered to them. Thus, the administrators, designers and application developers that cooperate towards bringing clean, up-to-date, consolidated and unambiguous data from the sources to the end-users need to have a clear understanding of the following issues (see more in the following section):

- the location of the data,
- the structure of each involved data source,
- the operations that take place towards the propagation, cleaning, transformation and consolidation of the data towards the central warehouse,
- any audit information concerning who has been using the warehouse and in what ways, so that its performance can be tuned,
- the way the structure (e.g., relational attributes) of each data repository is related to a common model that characterizes each module of information.

Data warehouse metadata repositories store large parts (if not all) of this kind of *data warehouse metadata* and provide a central point of reference for all the stakeholders that are involved in a data warehouse environment.

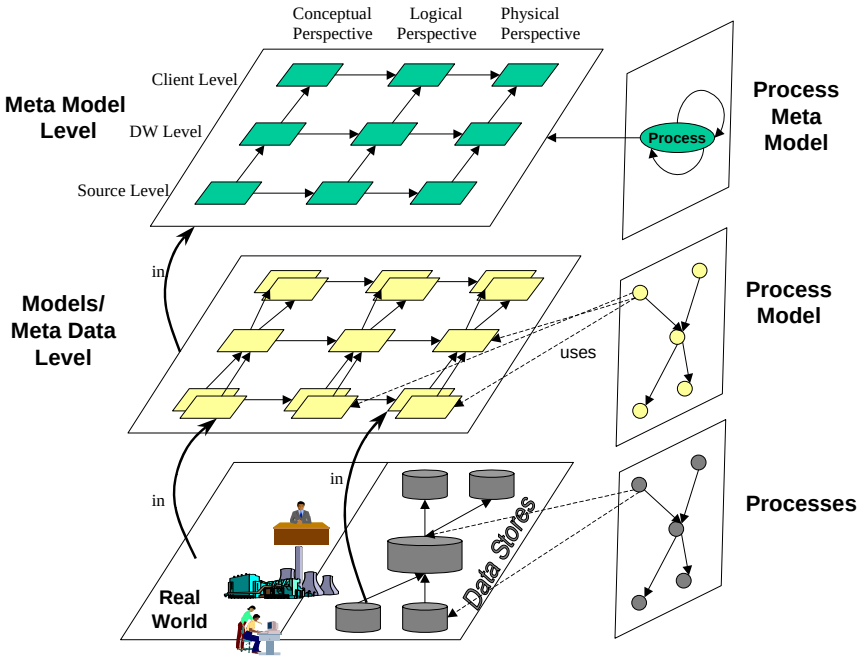

Figure 1. Role and structure of a data warehouse metadata repository [8]

As typically happened with all the area of data warehousing, ad-hoc solutions by industrial vendors and consultants were in place before the academic world provided a principled solution for the *problem of the structure and management of data warehouse metadata.* Early attempts of academic projects that related to wrapper-mediator schemes of information integration (Information Manifold, WHIPS, Squirrel, TSIMMIS -- see [9] for a detailed discussion of the related literature), did not treat metadata as first-class concepts in their deliberations; at the same time, early standardization efforts from the industrial world (e.g., the MDIS standard [12]) were also poor in their treatment of the problem.

The first focused attempt towards the problem of data warehouse metadata management was made in the context of the European Project "Foundations of Data Warehouse Quality (DWQ)" [7], [19]. In Fig. 1, the vertical links represent levels of abstraction: the data warehouse metadata repository, depicted in the middle layer, is an abstraction of the way the warehouse environment is structured in real life (depicted in the lowest layer of Fig. 1). At the same time, coming up with the appropriate formalism for expressing the contents of the repository (depicted in the upper layer of Fig. 1), provided an extra challenge that was tackled by [7] through the usage of the Telos language.

SCIENTIFIC FUNDAMENTALS

Structure of the data warehouse metadata repository. A principled approach towards organizing the structure of the data warehouse metadata repository was first offered by [7], [8]. The ideas of these papers were subsequently refined in [9] and formed the basis of the DWQ methodology for the management of data warehouse metadata. The specifics of the DWQ approach are fundamentally based on the separation of data and processes and their classification in a grid which is organized in three *perspectives*, specifically the conceptual, the logical and the physical one and three *location levels,* specifically, the source, warehouse and client levels (thus the 3x3 contents of the middle layer of Fig. 1 and the structure of Fig 2, too). The proposal was subsequently extended to incorporate a *program.vs.data* classification (Fig. 1) that discriminates static architectural elements of the warehouse environment (i.e., stored data) from process models (i.e., software modules).

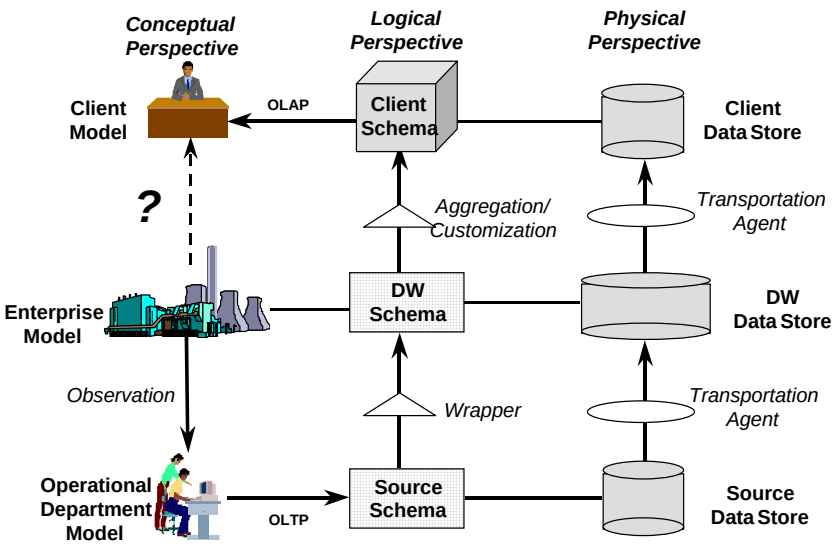

Figure 2. The DWQ proposal for the internal structure of the data warehouse metadata repository [7]

The *location* axis is straightforward and classifies elements as source, data warehouse and client elements. The data warehouse elements incorporate both the officially published data, contained in fact and dimension tables as well as any auxiliary data structures, concerning the Operational Data Store and the Data Staging Area. Similarly, any back-stage Extract-Transform-Clean (ETL) processes that populate the warehouse and the data marts with data are also classified according to the server in which they execute. The most interesting part of the DWQ method has to do with the management of the various *models* (a.k.a. *perspectives* in the DWQ terminology) of the system. Typically, in all DBMS's –and, thus, all deployed data warehouses- the system catalog includes both a *logical model* of the data structure (i.e., the database schema) as well as a *physical schema*, indicating the physical properties of the data (tablespaces, internal representation, indexing, statistics, etc) that are useful to the database administrator to perform his everyday maintenance and tuning tasks. The DWQ approach claimed that in a complicated and large environment like a data warehouse it is absolutely necessary to add a conceptual modeling perspective to the system that explains the role of each module of the system (be it a data or a software module). Clearly, due to the vast number of the involved information systems, each of them is accompanied by its own model, which is close enough to the perception of its users. Still, to master the complexity of all these submodels, it is possible to come up with a centralized, reference model of all the

collected information (a.k.a., *enterprise model*) – exploiting, thus, the centralized nature of data warehouses. The interesting part of the method is the idea of expressing every other submodel of the warehouse as a "view" over this enterprise model. Thus, once an interested user understands the enterprise model, he/she can ultimately understand the particularities of each submodel, independently of whether it concerns a source or client piece of data or software.

In [14], the authors discuss a coherent framework for the structuring of data warehouse metadata. The authors discriminate between back-stage *technical metadata*, concerning the structure and population of the warehouse and *semantic metadata*, concerning the front-end of the warehouse, which are used for querying purposes. Concerning the technical metadata, the proposed structure is based on (a) *entities*, comprising attributes as their structural components and (b) an early form of schema mappings, also called *mappings* in the paper's terminology, that try to capture the semantics of the back-stage ETL process by appropriately relating the involved data stores through aggregations, joins etc. Concerning the semantic metadata, the authors treat the enterprise model as a set of *business concepts*, related to the typical OLAP metadata concerning cubes, dimensions, dimension levels and hierarchies. The overall approach is a coherent, UML-based framework for data warehouse metadata, defined at a high-level of abstraction. Specialized approaches for specific parts (like definitions of OLAP models, or ETL workflows) can easily be employed in a complementary fashion to the framework of [14] (possibly through some kind of specialization) to add more detail to the metadata representation of the warehouse. It is also noteworthy to mention that the fundamental distinction between *technical* and *business metadata* has also deeply influenced the popular, industrially related literature [10].

Contents of the data warehouse metadata repository (data warehouse metadata in detail). The variety and complexity of metadata information in a data warehouse environment are so large that giving a detailed list of all metadata classes that can be recorded is mundane. The reader who is interested in a detailed list is referred to [11] for a broader discussion of all these possibilities, and to [10] for an in depth discussion with a particular emphasis on ETL aspects (with the note that the ETL process is indeed the main

provider of entries in the metadata repository concerning the technical parts of the warehouse). In the sequel, we classify our discussion in terms of data and processes.

DATA	Source	DW	Client
Conceptual	Source model	Enterprise model	Business concepts
Logical	Schemata and/or data formats	– Schemata and/or data formats – Surrogate Key, Slowly Changing Dimension information – Data cleaning standards/specs and business rules	– Schemata of any data marts – List of available pre-canned reports and their definitions – User documentation – User profiles – Security, authentication profiles
Physical	names of the involved data files or database installations physical properties like partitions, deployment/striping of data at disks, indexes, etc)		Map of available reports, spreadsheets, web pages

Figure 3. Metadata concerning the data of the warehouse

Data. Fig. 3 presents a summarized view of relevant metadata concerning the static parts of the warehouse architecture. The physical-perspective metadata are mostly related to (a) the location and naming of the information wherever data files are used and (b) DBMS catalog metadata wherever DBMS's are used. Observe the need for efficiently supporting the end-user in his navigation through the various reports, spreadsheets and web pages (i.e., answering the question "where can I find the information I am looking for?"). Observe also the need to support the questions "what information is available to me anyway?" which is supported at the logical perspective for the client level. The rest of the logical perspective is also straightforward and mostly concerns the schema of data; nevertheless business rules are also part of any schema and thus data cleaning requirements and the related business rules can also be recorded at this level. The conceptual perspective involves a clear recording of the involved concepts and their intra-level mappings (source-to-DW, client-to-DW). As expected, academic efforts adopt rigorous approaches at this level [9], whereas industrial literature suggests informal, but simpler methods (e.g., see the discussion on "Business metadata" at [10]).

It is important to stress the need of tracing the mappings between the different levels and perspectives in the warehouse. The physical-to-logical mapping is typically performed by the DBMS's and their administrative facilities; nevertheless, the logical-to-conceptual mapping is not. Two examples are appropriate in this place: (a) the developer who constructs (or worse,

maintains) a module that processes a source file of facts, has to translate cryptic code-and-value pairs (e.g., CDS_X1 = 145) to data that will be stored in the warehouse and (b) an end-user who should see data presented with names that relate to the concepts he is familiar with (e.g., see a description "Customer name" instead of the attribute name CSTR_NAME of a dimension table). In both cases, the logical-to-conceptual mappings are of extreme importance for the appropriate construction and maintenance of code and reports.

This is also the place to *stress the importance of naming conventions* in the schema of databases and the signatures of software modules: the huge numbers of involved attributes and software modules practically enforce the necessity of appropriately naming all data and software modules in order to facilitate the maintenance process (see [10] for detailed instructions).

PROCESSES	**Source**	**DW**	**Client**
Conceptual		Semantics of each activity of the workflow	
Logical	List of software modules related to the extraction task (and how)	– Structure of the ETL workflow – Scheduling for the execution of ETL workflows – Security settings	
Physical design	– Names & location of the involved scripts or software modules in the ETL process – Exception handling		
Physical Execution	Execution statistics	– Execution statistics – Audit & data lineage logs – Time statistics	Usage statistics

Figure 4. Metadata concerning the processes of the warehouse

Processes. When the discussion comes to the metadata that concern processes, things are not very complicated again, at the high level (Fig. 4). There is a set of ETL workflows that operate at the warehouse level, and populate the warehouse along with any pre-canned reports or data marts on a regular basis. The structure of the workflow, the semantics of the activities and the regular scheduling of the process form the conceptual and logical parts of the metadata. The physical locations and names of any module, along with the management of failures form the physical part of the metadata, concerning the design level of the software. Still, it is worth noting that the physical metadata can be enriched with information concerning the execution of the back-stage processes, the failures, the volumes of processed data, clean data, cleansed or impossible-to-clean data, the error codes returned by the DBMS

and the time that the different parts of the process took. This kind of metadata is of statistical importance for the tuning and maintenance of the warehouse back-stage by the administration team. At the same time, the audit information is of considerable value, since the data lineage is recorded as every step (i.e., transformation or cleaning) in the path that the data follow from the sources to their final destination can be traced.

Standards. The development of standards for data warehouse metadata has been one of the holy grails in the area of data warehousing. The standardization of data warehouse metadata allows the vendors of all kinds of warehouse-related tools to extract and retrieve metadata in a standard format. At the same time, metadata interchange among different sources and platforms –and even migration from one software configuration to another- is served by being able to export metadata from one configuration and loading it to another.

The first standardization effort came from the *MetaData Coalition (MDC)*, an industrial, non-profitable consortium. The standard was named *MetaData Interchange Specification (MDIS)* [12] and its structure was elementary, comprising descriptions for databases, records, dimensions and their hierarchies and relationships among them. Some years after MDIS, the *Open Information Model (OIM)* [13] followed. OIM was also developed in the context of the MetaData Coalition and significantly extends MDIS by capturing core metadata types found in the operational and data warehousing environment of enterprises. The MDC OIM uses UML both as a modeling language and as the basis for its core model. The OIM is divided in sub-models, or *packages*, which extend UML in order to address different areas of information management, including database schema elements, data transformations, OLAP schema elements and data types. Some years later, in 2001, the *Object Management Group (OMG)* initiated its own standard, named *Common Warehouse Metamodel (CWM)* [4]. CWM is built on top of other standard OMG notations (UML, MOF, XMI) also with the aim to facilitate the interchange of metadata between different tools and platforms. Currently, in 2007, CWM appears to be very popular, both due to its OMG origin and as it is quite close to the parts concerning data warehouse structure and operation. Much like OIM, CWM is built around packages, each covering a different part of the data warehouse lifecycle. Specifically, the packages defined by CWM cover metadata concerning (a) static parts of the warehouse architecture like relational, multidimensional and XML data sources, (b) back-

stage operations like data warehouse processes and operations, as well as data transformations and (c) front-end, user-oriented concepts like business concepts, OLAP hierarchies, data mining and information visualization tasks. A detailed comparison of earlier versions of OIM and CWM can be found in [18].

KEY APPLICATIONS

Data Warehouse Design. Typically, the data warehouse designers both populate the repository with data and benefit from the fact that the internal structure and architecture of the warehouse is documented in the metadata repository in a principled way. [16] implements a generic graphical modeling tool operating on top of a metadata repository management system that uses the IRDS standard. Similar results can be found in [3], [17].

Data Warehouse Maintenance. The same reasons with data warehouse design explain why the data warehouse administrators can effectively use the metadata repository for tuning the operation of the warehouse. In [15], there is a first proposal for the extension of the data warehouse metadata with operators characterizing the evolution of the warehouse's structure over time. A more formal approach on the problem is given by [5].

Data Warehouse Usage. Developers constructing or maintaining applications, as well as the end-users interactively exploring the contents of the warehouse can benefit from the documentation facilities that data warehouse metadata offer (refer to [10] for an example where metadata clarify semantic discrepancies for synonyms).

Data Warehouse Quality. The research on the annotation of data warehouse metadata with annotations concerning the quality of the collected data (a.k.a. *quality indicators*) is quite large; the interested reader is referred to [6], [9] for detailed discussions.

Model Management. Model management was built upon the results of having a principled structure of data warehouse metadata. The early attempts in the area [1], [2] were largely based on the idea of mapping source and client schemata to the data warehouse schema and tracing their attribute interdependencies.

Design of large Information Systems. The mental tools developed for the management of large, intra-organizational environments like data warehouses can possibly benefit other areas –even as a starting point. The most obvious candidate concerns any kind of open agoras of information systems (e.g., digital libraries) that clearly need a common agreement in the hidden semantics of exported information, before they can interchange data or services.

CROSS REFERENCES

METADATA, OIM, MDC, CWM, DATA WAREHOUSES, METADATA REPOSITORY, DATA WAREHOUSE LIFE-CYCLE AND DESIGN, MODEL MANAGEMENT, DATA QUALITY

RECOMMENDED READING*

[1] P. Bernstein, A. Levy, R. Pottinger, A Vision for Management of Complex Models, SIGMOD Record 29(4), Dec. 2000.

[2] P.A. Bernstein, E. Rahm. Data Warehouse Scenarios for Model Management. In Proc. 19th International Conference on Conceptual Modeling (ER 2000), pp. 1-15, Salt Lake City, Utah, USA, October 9-12, 2000.

[3] Liane Carneiro, Angelo Brayner: X-META: A methodology for data warehouse design with metadata management. 13-22 Design and Management of Data Warehouses 2002, Proceedings of the 4th Intl. Workshop DMDW'2002, Toronto, Canada, May 27, 2002.

[4] Common Warehouse Metamodel (CWM) Specification, version 1.1. OMG, March 2003. Available at http://www.omg.org/technology/documents/formal/cwm.htm

[5] M. Golfarelli, J. Lechtenbörger, S. Rizzi, G. Vossen. Schema versioning in data warehouses: Enabling cross-version querying via schema augmentation. Data Knowl. Eng. 59(2): 435-459 (2006).

[6] M.A. Jeusfeld, C. Quix, M. Jarke. Design and Analysis of Quality Information for Data Warehouses. In Proc. 17th International Conference on Conceptual Modeling (ER 1998), pp. 349-362, Singapore, November 16-19, 1998.

[7] M. Jarke, M.A. Jeusfeld, C. Quix, P. Vassiliadis. Architecture and quality in data warehouses. In Proc. 10th Conference on Advanced Information Systems Engineering (CAiSE '98), pp. 93-113, Pisa, Italy, June, 8-12, 1998. Lecture Notes in Computer Science, vol. 1413, Springer, 1998.

[8] M. Jarke, M.A. Jeusfeld, C. Quix, P. Vassiliadis. Architecture and quality in data warehouses. Information Systems, vol. 24, no. 3, pp. 229-253, May 1999. Elsevier Science Ltd. ISSN 0306-4379.

[9] M. Jarke, M. Lenzerini, Y. Vassiliou, P. Vassiliadis (eds.). Fundamentals of Data Warehouses. Springer-Verlag, Berlin Heidelberg 2003. 2nd Edition, ISBN 3-540-42089-4 (p. 207).

[10] R. Kimball, J. Caserta. The Data Warehouse ETL toolkit. Willey Publishing, Inc., 2004.

[11] R. Kimbal, L. Reeves, M. Ross, W. Thornthwaite. The Data Warehouse Lifecycle Toolkit: Expert Methods for Designing, Developing, and Deploying Data Warehouses. John Wiley & Sons, February 1998.

[12] Metadata Coalition: Proposal for version 1.0 metadata interchange specification. http://www.metadata.org/standards/toc.html, July 1996.

[13] MetaData Coalition. Open Information Model, version 1.0 (1999). Available at http://www.MDCinfo.com

[14] R. Müller, T. Stöhr, E. Rahm. An Integrative and Uniform Model for Metadata Management in Data Warehousing Environments. Proceedings of the Intl. Workshop on Design and Management of Data Warehouses, DMDW'99, Heidelberg, Germany, June 14-15, 1999.

[15] C. Quix. Repository Support for Data Warehouse Evolution. Proceedings of the Intl. Workshop on Design and Management of Data Warehouses, DMDW'99, Heidelberg, Germany, June 14-15, 1999.

[16] C. Sapia, M. Blaschka, G. Höfling. GraMMi: Using a Standard Repository Management System to Build a Generic Graphical Modeling Tool. 33rd Annual Hawaii International Conference on System Sciences (HICSS-33), Track 8: Software Technology, 4-7 January, 2000, Maui, Hawaii.

[17] Anca Vaduva, Jörg-Uwe Kietz, Regina Zücker. M4 - A Metamodel for Data Preprocessing. Fourth ACM International Workshop on Data Warehousing and OLAP (DOLAP 2001), November 9, 2001, Atlanta, Georgia, USA.

[18] Thomas Vetterli, Anca Vaduva, Martin Staudt. Metadata Standards for Data Warehousing: Open Information Model vs. Common Warehouse Metamodel. SIGMOD Record, 29(3), pp. 68-75, September 2000.

[19] Foundations of Data Warehouse Quality (DWQ) homepage. Available at http://www.dblab.ece.ntua.gr/~dwq/