Metadata Standards for Data Warehousing: Open Information Model vs. Common Warehouse Metamodel*

Thomas Vetterli† Anca Vaduva‡ Martin Staudt†

†Swiss Life
P.O. Box, CH-8022 Zurich
Switzerland
Thomas.Vetterli@swisslife.ch
Martin.Staudt@swisslife.ch

‡University of Zurich
Department of Information Technology
Winterthurerstr. 190
CH-8057 Zurich, Switzerland
vaduva@ifi.unizh.ch

Guest Column Introduction

This column typically addresses some aspects of the SQL standard, including the recently-published SQL:1999 standard and the still-emerging parts of the SQLJ standard. This issue, however, takes a slightly different approach. First, instead of writing the column ourselves, we are giving our space, as we will do occasionally, to a guest column. And, second, the subject matter is not directly related to the SQL standard (although there is certainly a relationship to the language).

One of the least well understood aspects of data management is the use of metadata. The increasing popularity of data warehouses raises the importance of comprehensive analysis of metadata far beyond its typical significance. Vetterli, Vaduva, and Staudt examine two standards for metadata in the context of their uses in the data warehousing context. We appreciate their willingness to share the results of their work with us and with you.

Andrew Eisenberg and Jim Melton

Abstract

Metadata has been identified as a key success factor in data warehouse projects. It captures all kinds of information necessary to analyse, design, build, use, and interpret the data warehouse contents. In order to spread the use of metadata, enable the interoperability between repositories, and tool integration within data warehousing architectures, a standard for metadata representation and exchange is needed. This paper considers two standards and compares them according to specific areas of interest within data warehousing. Despite their incontestable similarities, there are significant differences between the two standards which would make their unification difficult.

*This work was partly supported by Swiss Federal Office of Professional Education and Technology under grant KTI-3979.1 (SMART).

1 Introduction

Data warehousing aims at providing, managing and exploiting a set of integrated data (the data warehouse) for business decision support within an organization. Important business trends are discovered and better and faster decisions may be reached regarding multiple aspects of business like sales and customer service, marketing, and risk assessment.

In order to cope with the complexity of data structures and processes in data warehousing, a consistent management of metadata is required. Metadata (also called "data about data") is defined as any information that may be used to minimize the efforts for administration and to improve the exploitation of a data warehouse system. Typically, metadata is classified with regard to its use as *business metadata*, mainly needed by end-users, and *technical metadata*, produced and used by database administrators or by other software components of the data warehouse system. The category of business metadata contains end-user-specific documentation, dictionaries, thesauri, and domain-specific ontological knowledge, business concepts and terminology, details about predefined queries, and user reports. In

contrast, technical metadata includes schema definitions and configuration specifications, physical storage information, access rights, executable specifications like data transformation and plausibility rules, and runtime information like log files and performance results.

Consistent metadata management requires metadata to be captured and stored in a repository which is shared both by various user groups (e.g., end-users, system administrators, application developers, etc.) and software components. Users need the repository as a consistent documentation in order to effectively and efficiently achieve their particular tasks. On the other hand, software components may produce and consume (i.e., access, interpret and possibly execute at runtime) the information of the repository. Thus, the repository has to provide a structure fitting its utilisation purposes. First of all, this structure has to reflect the diversity of information required by users and tools (e.g., business and technical metadata, descriptive and transformational, conceptual and logical, etc). Then, it has to ensure the navigation requirements of users. That means, semantic links between metadata elements ought to be represented in the repository structure. At the conceptual level, the structure of the repository is described by a *metamodel*. Each repository should provide a documented and user-extendable metamodel (possibly stored itself in the repository) which easily allows the extension of the system for additional requirements (e.g., information types and sources).

The existence of a single repository for managing all kinds of metadata within an enterprise allows *centralized metadata management*. Metadata is uniformly and consistently managed, and accessed by all possible consumers. However, the use of different tools, software components and repositories with divergent data models, following diverse representation formats, hinders centralization. In this case, there are two alternatives to handle metadata management. A totally *decentralized* approach requires metadata to be mutually exchanged when necessary. *Federated metadata management* strikes a trade-off between the advantages of centralization and those of local control. However, in both cases a common metadata representation associated with a common metamodel is required: in the former case, a common interchange representation format has to be adopted at both sides and used when data has to be imported or exported between repositories and/or tools. In the latter case, a global conceptual view of all existing metadata has to be built as a means to integrate the architecture components sharing metadata in the system. Inherently, the easiest solution is to agree on a common metadata representation in a given tool environment or to use a predefined one-to-one interchange format between commercial products, assuming they provide any. However, the freedom of combining products from any manufacturer to create a customized system is only given if a standard exists and all products comply with it. In other words, the necessity of a metadata standard for representation and exchange is undisputable in order to ensure interoperability, integration and spreading of metadata use in data warehouse projects.

This paper compares the Open Information Model (OIM) [2] and the Common Warehouse Metamodel (CWM) specification [3], two accepted standards for metadata representation and exchange, proposed by Meta Data Coalition and OMG. To clarify, a standard for metadata representation requires the complete description of a metamodel with all its elements, their semantic content and interdependencies between elements. The standard is inherently independent of any specific implementation. A standard for exchange is based on a unique metamodel as well but it also contains the definition of ready-to-use interfaces that specify the metamodel in a wide-spread language like XML or CORBA IDL.

In the following, we discuss the OIM and CWM metamodels. Section 2 introduces the two specifications and present the sub-models from which they are built up. Section 3 compares the standards according to some criteria which play an important role for data warehousing. Section 4 summarizes the differences and a possible convergency in the future.

2 Overview

OIM and CWM are both industry standards developed by multi-vendor organizations with participation of industry leaders. They specify metamodels which could be seen as conceptual schemas for metadata incorporating application-specific aspects of data warehousing.

The purpose of OIM is to support tool interoperability across technologies and companies via a shared information model (another name for an enterprise-wide metamodel). OIM is designed to encompass all phases of information systems development, from analysis through deployment. OIM version 1.0 was adopted in July 1999, by the Meta Data Coalition as a standard. Version 1.1 is underway. The Meta Data Coalition aims at the definition, implementation and ongoing evolution of a metadata

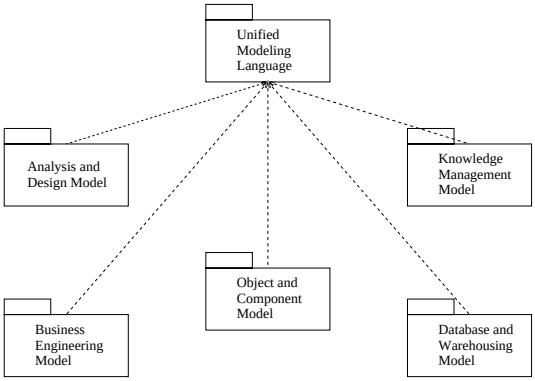

Figure 1: Open Information Model

interchange format standard and its support mechanisms. The coalition currently consists of more than 50 members, including Microsoft, Ardent Software, Brio Technologies, Evolutionary Technologies International (ETI), Informatica, Platinum, SAS Institute and Viasoft.

CWM is part of OMG's efforts to create a standardized object-oriented architectural framework for distributed applications in order to support reusability, portability and interoperability of object-oriented software components in heterogeneous environments. The proposed standard is designed to help companies integrate data warehouse, e-business and business intelligence systems quickly and easily by supplying a metadata representation and exchange format. The CWM specification was proposed to the OMG with a joint submission by IBM, Oracle, Unisys, Hyperion Solutions (Essbase Software), and others and was adopted as an OMG standard in June 2000.

2.1 Open Information Model (OIM)

OIM is a specialization of the abstract concepts of UML into domain specific sub-models that describe metadata. OIM is based on the industry standards UML, XML and SQL. Figure 1 shows the sub-models of OIM and their interdependencies using UML notation[1]:

- The *Analysis and Design Model* covers the domain of object-oriented modeling and design of software systems. The core of the model is the *UML package*, describing version 1.3 of the UML. Other packages are *UML extensions* (covering the presentational aspects of UML elements like fonts, coordinates etc., and other general-purpose additions to the UML package), *Common Data Types* (Date, Time etc.) and *Generic Elements* (describing a set of general-purpose classes that are relevant across diverse information models).

- The *Object and Component Model* comprises the *Component Description Model*, covering the different component development life-cycle deliverables. The model is divided into three distinct layers: specification, implementation (currently not defined), and executable. The *Component Description Model* covers the various aspects of the implementation of a component (defined as "a software package that offers services through interfaces"), but does not respect the specifics of any particular programming language.

- The *Business Engineering Model* (available in Version 1.1) provides all the necessary metadata types to capture goals, organization, and infrastructure of a business as well as the processes and rules that govern the business. It contains following packages: *Business Goals, Organizational Elements, Business Processes* and *Business Rules.*

- The *Knowledge Management Model* comprises the *Information Directory Model*, currently not defined, and the *Semantic Definitions.* It accommodates conceptual models of user information. Specifically, the package holds descriptions of semantic models and their relationships to the underlying database schema. These connections enable a user to query a database using English sentences. The *Information Directory Model* will provide metadata types to define a controlled vocabulary to classify business information.

- The *Database and Warehousing Model* consists of the following packages:

  1. *Database schema* elements describe information maintained in relational database systems. This package contains information about schema elements (tables, views, queries etc.) and database specific data types. The concepts of this package are modeled according to the ANSI/ISO SQL-92 standard, with selected proprietary extensions supported by popular relational database vendors.

[1] The reader should not be confused by the fact that OIM uses UML in three distinct roles: as modeling language to design and visualize OIM itself, as the main part of the Analysis and Design Model to express object-oriented models, and as the core-model of OIM from which sub-models inherit concepts.

2. OLAP schema elements describe multidimensional databases. The package allows the description of *cubes* (the basic component in multidimensional data analysis), *dimension hierarchies* (for roll-up and drill-down operations), and *aggregations* (precalculated roll-ups of data stored in a cube).
3. *Data transformations* elements cover relational-to-relational transformations. A *transformation* maps from a set of source objects into a set of target objects, both represented by a *transformable object set*. Transformations can be packaged into groups. There are three levels of grouping: The first uses a *transformation task*, which describes a set of transformations that must be executed together. The second level is a *transformation step*, executing a single transformation task. Steps are used to coordinate the flow of control between tasks. The third level is a *transformation package*, consisting of a set of transformation steps.
4. *Record-oriented database* elements describe information about data maintained in files or non-relational (legacy) databases. The package does not cover detailed logical-to-physical mapping or information about any of the concrete file systems or databases that may use record structures. These will be added later in subsequent packages.
5. *Report definitions* (available in Version 1.1) represent information necessary for data reporting tools and their relationships to the systems they report on.

2.2 Common Warehouse Metamodel (CWM)

The main purpose of CWM is to enable easy interchange of common warehouse metadata between warehouse tools and warehouse metadata repositories in distributed heterogeneous environments and thus considerable documents (and files) are provided with the CWM metamodel expressed in XML and interfaces generated as CORBA IDL. CWM is based on the following OMG standards: UML, MOF (Meta Object Facility), a metamodeling and metadata repository standard, and XMI, an XML-based standard for metadata exchange originating from OMG.

Figure 2 shows the dependencies between the submodels of CWM in UML notation:

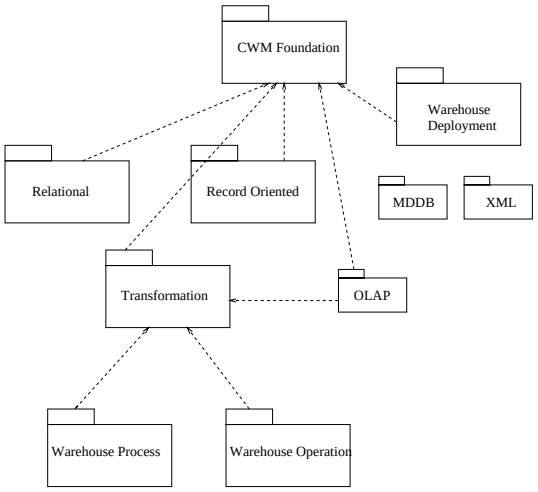

Figure 2: Common Warehouse Metamodel

- The *CWM Foundation* contains elements representing concepts and structures that are shared by other CWM packages. CWM Foundation covers the following conceptual areas:
  - *Business Information* defines so-called business-oriented information (like, e.g, responsible parties, how to contact them, general-purpose textual descriptions) about CWM model elements.
  - *Data Types* and *CWM Types* support the definition of data type metamodel constructs that modelers can use to create specific data types. Note that besides UML data types, CWM Foundation does not contain any specific data type definitions.
  - *Expressions* provide basic support for the definition of expression trees within the CWM.
  - *Keys & Indexes* are means for specifying instances and for identifying alternate sortings of instances. These concepts have been placed in the CWM Foundation because similar concepts are available in many types of data sources.
- The *Warehouse Deployment* package contains elements to record how the hard- and software in a data warehouse is used. The package tracks the installation of a software system (e.g. a DBMS, a software package with its components) as well as the location of individual computers.
- The *Relational* package describes data accessible through a relational interface. The package

follows the SQL:1999 standard.

- The *Record-Oriented* package covers the basic concept of a record and its structure. Traditional data records as well as data types of structured programming languages can be described.
- The *Multidimensional Database (MDDB)* package is a generic representation of a multidimensional database (i.e., MOLAP). In multidimensional databases, constructs like dimensions, hierarchies are represented by internal data structures and the OLAP operations are automatically provided as well. The package does not attempt to provide a complete representation of all aspects of commercially available multidimensional databases. Tool-specific extensions (Oracle Express, Essbase) are available as examples in the CWM Submission.
- The *XML* package contains types and associations that describe XML data resources. It is based on XML 1.0.
- The *Transformation* package covers transformations among different types of data sources and targets: object-oriented, relational, record-oriented, multidimensional, XML, and OLAP. Transformations can be grouped into logical units.
- The *OLAP* package defines a metamodel of essential OLAP constructs that are common across most OLAP applications and tools. This metamodel is mapped to deployment-capable structures (the *Relational* and *Multidimensional* packages) via the *Transformation* package.
- The *Warehouse Process* package documents the process flow used to execute the transformations.
- The *Warehouse Operation* package contains classes recording the day-to-day operation of the warehouse processes.

3 Comparison

This section provides a coarse comparison of the two competing specifications. OIM is designed to encompass all phases of information systems development and contains one package that is specifically intended for data warehousing, the *Database and Warehousing Model*. On the other hand, the entire CWM deals with metadata for data warehousing only and provides a framework for representing and exchanging metadata about data sources, data targets and warehouse processes that create and manage them. Therefore, OIM is much broader in scope than CWM.

Both standards use UML 1.3 as their foundation. However, there is a significant difference: CWM is compliant with the Meta Object Facility (MOF) standard, whereas OIM is not. Therefore, CWM has an inherent repository orientation provided by the MOF compliance (e.g. representation of the meta data as CORBA objects). OIM, on the other hand, is not a specification of a repository API or implementation; it focuses on the description of information, not on data access and management.

In the following, we investigate the coverage of certain areas of interest and the support provided by the two standards for various aspects of data warehousing, like the representation of data resources, data transformations and business aspects within the two metamodels.

3.1 Database Support

Both standards support the definition of relational data sources. The differences are:

- OIM is based on SQL-92, whereas CWM is based on SQL:1999.
- OIM defines the concept of keys and indexes within the (relational) *Database Schema*, whereas CWM defines them within the *Foundation* package, giving them a general applicability (e.g. for legacy data sources).

Support of legacy data sources (e.g. hierarchical, network, index-sequential models) is provided to some extent in both standards.

OIM provides a package to describe record-oriented data sources, covering the basic elements such as records, fields and relationships. There is no explicit support yet for common file systems and databases such as VSAM, IMS and so forth.

CWMs record-oriented package supports the definition of record-oriented structures, including both traditional data records as well as data types from structured programming languages (Cobol, C etc.). As an extension of the record-oriented model, support for IMS database definitions is provided as an example (not a normative part of the standard).

3.2 On-line Analytical Processing (OLAP) and Multidimensional Data

Since OLAP is the main analysis application for data warehousing, both OIM and CWM provide support to define OLAP metadata. Areas covered by both models include the definition of cubes, dimensions and hierarchies.

CWM provides two packages, namely the *OLAP package* which allows the definition of essential OLAP constructs that are common across most OLAP applications and tools. The *Multidimensional Package* covers the definition of a multidimensional database. The OLAP metamodel is mapped to deployment capable structures (the *Relational* or *Multidimensional* model) via the *Transformation* package. That means, CWM strictly separates the representation of data in a multidimensional format from the actual deployment, namely ROLAP or MOLAP.

In OIM, there is no separation of these two aspects. The *OLAP Schema* package combines the semantic aspects of OLAP with the deployment of these structures. Also, since the *OLAP* model is largely derived from the (relational) *Database schema* model, the deployment of an OLAP structure as a multidimensional database seems impossible – despite the fact that the *OLAP Schema* package provides the possibility to specify the OLAP model (hybrid, relational or multidimensional) as part of a cube specification.

3.3 Warehouse Deployment

Warehouse deployment deals with how the software in a data warehouse system is used, which software systems and data resources exist and how they interact. Warehouse deployment is handled differently in OIM and CWM. OIM generally provides logical and deployment subclasses (not further defined) of its semantic classes, whereas CWM defines a standalone *Warehouse Deployment* model which defines concepts of providers, data resources, and connections. Any logical data model within CWM (e.g. Relational, Multidimensional) may have a link to an appropriate metaclass of the CWM *Warehouse Deployment* package.

3.4 Data Transformations

Both CWM and OIM provide similar support to model data transformations, enabling the grouping of single transformations (i.e., mapping a data object, e.g. a table or column, to another data object) into transformation packages. There are, however, some weighty differences:

- The OIM *Transformation Model* currently supports relational-to-relational mappings, only. CWM, on the other hand, is not tied to a particular model.
- The specification of transformation expressions is stored in string format in OIM, whereas CWM supports both a textual string representation as well as a tree-based structure.

3.5 XML

Support for XML data resources is rapidly becoming very important. CWM provides an XML-package which supports the definition of XML data sources, e.g. XML schemas, element types etc. OIM does not support XML data sources.

Regarding the exchange of data using XML, CWM uses the already defined OMG standard called XMI (XML Metadata Interchange), whereas OIM defines its own XML encoding (which is not yet accepted as a standard by the Meta Data Coalition). There is again a difference in scope: The XMI standard is intended to exchange *any* MOF-based model, and consists of two major components:

1. the *XML DTD Production Rules* for producing XML Document Type Definitions (DTDs) for XMI encoded metadata, and
2. the *XML Document Production Rules* for encoding metadata in an XML compatible form.

Then, XMI is a pair of parallel mappings, one between MOF metamodels and XML DTDs, the other between MOF metadata and XML documents.

The OIM XML-encoding specification on the other side suggests a coarse DTD (specifying the necessary top element) for a complete transfer of metadata. A set of accompanying DTDs is also provided.

However, for both CWM and OIM, DTDs are not expressive enough to completely express the models semantics. Therefore, knowledge of the underlying model is necessary to correctly interpret a metadata exchange.

3.6 Business Metadata

Business metadata covers a wide range of topics: the definition of a business terminology, the link of this terminology to existing reports or data sources, the possibility to define key business figures (e.g. Balanced Scorecards), the non-technical description of

transformations and business rules, etc. At present, both standards do a relatively poor job in supporting business metadata. However, OIM version 1.1 will handle a considerable amount of business elements. The *Business Engineering* model will contain packages for the description of *Business Goals* and *Processes*. The *Knowledge Description* package within the *Knowledge Management* model will allow the definition of glossaries, thesauri and vocabularies. The *Database and Warehouse* model will be enhanced to allow the definition of reports (*Report Definition* package). No support for the definition of key business figures is planned.

CWM only supports the definition of business-oriented information about model elements: ContactInfo, EmailId, ResponsibleParty, Telephone, Location, Document and Description. Note that through the *Generic Elements* package, OIM also provides the possibility to model ContactInfo (i.e, a person), EmailID, Telephone and Location.

An important part of business metadata are the business rules. According to [1], a business rule is a statement that defines or constrains some aspect of the business and is intended to assert business structure or to control or influence the behavior of the business. In the ideal case, business rules should be defined on different levels of abstraction, including descriptions (in natural language) suitable for business users, as well as formal specifications for technical users. These specifications should also be linked to other areas of warehouse metadata, e.g. specification of transformation rules.

CWM does not support the specification of business rules per se. In OIM 1.1, support for business rules is provided by a part of the *Business Engineering* model. The suggested business rule model is mainly targeted towards business users. Furthermore, a *Grammar Model* is proposed, which will provide support for a specification of a business rule in a formal grammar.

3.7 Organizational Aspects

Currently, both standards do not cover organizational aspects of data warehousing, e.g. the definition of persons, roles, organization units etc. OIM 1.1 will introduce an *Organizational Model* as part of the *Business Engineering Model*.

3.8 Data Lineage

Data lineage is the ability to determine the source of the data and the process that "produced" a certain data item, e.g. a cell in a multidimensional cube.

The CWM *Transformation* package (see 3.4) covers transformations among all types of data sources and targets: object-oriented, relational, record-oriented, multidimensional, XML, and OLAP. Whenever data has to be converted, this can be modeled with a transformation. The *Warehouse Operation* package covers the actual executions of transformations within the warehouse, identifying when a transformation was executed and whether it was successfully completed. Therefore, a coarse data lineage is supported by CWM. However, it is not possible to track individual data items.

OIM provides a similar *TransformationPackage* class (see 3.4), limited to relational data only. An execution of a *TransformationPackage* produces an instance of *Package Execution* class which gets assigned an ExecutionId. This ID is stored in the instances of the target data. In this way, rudimentary data lineage support is possible in OIM as well.

3.9 Quality Issues

Quality issues in data warehousing comprise a wide spectrum. Roughly, three areas have to be considered: data definition and information architecture quality, data content quality and data presentation quality. Specifically, information about accuracy of data, data currentness, data completeness, believability etc. are of interest.

There is no support, however, for quality aspects in data warehousing, neither in OIM nor CWM.

3.10 Security Aspects

Both standards do not cover security aspects of data warehousing, e.g. the definition of access and execution rights.

3.11 Support for versions & configurations

In CWM, support for versions and configurations is given through MOF. The *MOF Model Package* provides a *copyModel()* operation, "deep-copying" a top-level package into a target namespace. This namespace must be an instance of *MOFRepository*. In UML, packages can have a version assigned with a *Tagged Value*, so both CWM and OIM provide a basic support for version management. Additionally, the *Generic Elements* package of OIM contains a class *NamedVersion*, enabling the ability to specify component version information.

3.12 Query Generation

The *Semantic Definitions* package of OIM offers the possibility to specify conceptual models of user information, comprising Entities, Relationships and Dictionary entries. The intention of the package is to support the definition of mappings between database schema and semantic constructs familiar to the users in order to enable end-users to interact with a database without learning a data manipulation language. CWM does not provide such a capability.

4 Summary

The tendency of local and proprietary metadata storage provided by specialized data warehouse tools causes a high demand for metadata integration and exchange. In this context, standardization is required. However, instead of a single one, there are two different competing streams of standardization with regard to metadata representation and exchange, namely MDC with Microsoft as leader promoting OIM on the one hand and its rivals Oracle, IBM etc. contributing to the OMG activities and adopting CWM as a standard on the other hand.

The differences and similarities between OIM and CWM can be summarized as follows:

1. They build on the same ground, namely UML and XML, raising the hope that a unification of the two models may someday be possible.

2. They differ in scope: while CWM is primarily restricted to data warehouse metadata, OIM has a broader purpose.

3. They do not provide mechanisms to categorize information according to different conceptual levels of metadata, as e.g. suggested by Zachman [4]. But both standards provide a package structure to separate different warehouse areas.

4. They provide support to represent and exchange core warehouse metadata with an emphasis on the technical metadata. For the time beeing, both standards fall short in providing mechanisms for handling business metadata. OIM version 1.1 (proposed but not yet adopted as a standard) will address business aspects in more detail.

5. In contrast to CWM, OIM is currently oriented and limited to the support of relational data (e.g., in transformations, OLAP).

6. While CWM provides a sophisticated mechanism to exchange metadata between tools or repositories relying on the XMI standard, the proposed solution for OIM is rather hand-knitted.

Despite the fact that the OMG and the MDC established a formal technical liaison (the MDC is now a Platform Member of the OMG, and the OMG is a member of the MDC) in order to "build consensus on metadata standards", significant efforts are required to unify OIM and CWM. It is likely that the two standards will co-exist for some time and metadata exchange between the two standards (via XML) will be commonplace. However, to support a complete integration of all warehouse metadata stored in repositories compliant with OIM or CWM, a unified and extended model will be absolutely necessary.

References

[1] GUIDE. *Business Rules Project*, Final Report, revision 1.2 edition, October 1997.

[2] Meta Data Coalition. *Open Information Model Version*, 1.0 edition, August 1999. http://www.MDCinfo.com.

[3] OMG. *Common Warehouse Metamodel (CWM) Specification*, OMG Document ad/99-09-01, Initial Submission edition, September 1999. http://www.omg.org.

[4] John A. Zachman. A framework for information systems architecture. *IBM Systems Journal*, 26(3), 1987.