UDC:681.324
Review paper

# METADATA INTERCHANGE IN SERVICE BASED ARCHITECTURE

**Alma Butković Tomac**
Nagravision Kudelski group, Cheseaux / Lausanne
*alma.butkovictomac@nagra.com*

**Dražen Tomac**
Cambridge Technology Partners, Genève
*drazen.tomac@ctp.com*

**Abstract:** *Metadata is important factor for understanding, controlling and planning in complex information systems both from technical and business perspective. However its usage is often limited due to opposing or non-existent standards. The article discusses current state of meta model standards and shows new possibilities for meta data interchange based on XML as de facto standard for data interchange and increasing acceptance of web services architecture.*

**Keywords:** *metadata, web services, XMI, CWM.*

## 1. INTRODUCTION

Term “meta” comes from old Greek work representing after, next, with; Latin word denoting something transcendental or beyond nature. Applying the prefix meta denotes more comprehensive or transcendent then the base itself. Meta data transcends the data in a way that it describes or references more than individual data instance: it provides domain, format, context of the data, data source, business related rule etc. Most common short definitions on metadata say: Meta data is data about data. This definition does not describe meta data in its whole scope. To quote [4]:” Meta data is all physical data (contained in software and other media) and knowledge (contained in employees and various media) from inside and outside an organization, including information about the physical data, technical and business processes, rules and constraints of the data, and structures of the data used by a corporation.”

## 2. METADATA SIGNIFICANCE

Although metadata significance is widely recognized and important in information systems, it gains on its importance for data warehouse projects, business intelligence and semantic web. It is almost impossible to effectively manage enterprise data without data and metadata standards. Technical meta data, mostly used by technical IT or company stuff are used in all project lifecycle phases, requirements gathering, analysis, design, development, it eases maintenance and understanding of the whole system, help planning future extensions and development. Business metadata support business users, establishing sort of link between business itself and the IT representation of it. Building a metadata repository is for sure a big effort, but its significance overpass initial costs and brings return of investments. Defining and documenting technical meta data: data struc tures, data sources, domain, its semantic, transformation rules, business rules, generally the context, meta data repository benefits, reducing the cost of future development, ensuring better understanding of the system, preventing the case where majority of significant data is stored in employees heads, thus also reducing possibility of development error. From business user point of view, qualitative metadata ensures users confidence in the system. Meta data exists in different forms and comes from many sources and each source possibly has its own meta model or a way to represent it. One of the common sources is data modelling tool (e.g. Oracle Designer) used in logical data and process modelling, physical and system design, different reporting tools. Other important metadata sources are company (business) documents and employees themselves – knowledge and experience they posses on business.

## 3. META DATA STANDARDS

Sharing meta data among various software products, teams and repositories is required to prevents errors, inconsistencies, especially on big projects where diverse teams are involved, geographically or somehow else dislocated. The amount of data enterprises are dealing with is significantly increasing, sourced from different systems, with possible redundancy or inconsistency issues it becomes quite impossible to handle, access or manage for any meaningful usage, e.g. for decision-making support. Rapidly grown different analysis and modelling tool with different models and implementations brings in more confusion into this field. Therefore, ability to effectively share meta model induces a need for standardization of meta model itself (meta model here represents physical data model used to store meta data). Recently, two major initiatives were significant bringing standard meta model: OMG (Object Management Group) and MDC (Meta Data Coalition), even more after their agreement to "merge" and to support single meta model standard in 2000. Unified Modelling Language (UML) and Meta Object Facility (MOF) specifications adoption in 1997 unifies modelling technology and repositories, which opens wider a door in metadata standardization efforts. Following, XML metadata interchange (XMI) specifications in 1999, which replaces numbers of different interchange formats, and finally Common Warehouse Metamodel (CWM) in 2001.

### *3.1. COMMON WAREHOUSE METAMODEL (CWM)*

"CWM provides a framework for representing metadata about data sources, data targets, transformation and analysis, and the processes and operations that can create and manage warehouse data and provide lineage information about its use." [1] CWM represents a complete specification of syntax and semantics, a language or a framework for metadata interchange purpose The Common Warehouse Metamodel (CWM) is a specification for modelling metadata for relational, non-relational, multidimensional systems. It is a result of Common Warehouse Metamodel Initiative (CWMI), issued by OMG whose major goals were to enable easy interchange of metadata between data warehous ing tools and metadata repositories in distributed heterogeneous environments. CWM addresses end-to-end management for databases, data marts, warehouses, and portals - managing and capturing all its data. CWM standard is extensible, allowing to meet specia l users needs. So-called transformation model in CWM is used to automatically transform possible propriety models into CWM and vice-versa. CWM provides a complete specification of syntax and semantics needed to export/import data warehousing and business intelligence metadata, a language or a framework for metadata interchange purpose. CWM model consists of sub/models, which represents metadata in following areas:

- Data resources (metal model for object-oriented, relational, multidimensional and CML sources)
- Data Analysis (metamodels for data mining, business naming, OLAP)

Warehouse Management (meta models for warehouse processes an its results). Major parts of CWM are metamodel specification, format for metadata interchange and API for metadata access. The core architecture is based on MOF, used to define CWM model, the design of the CWM model uses UML. CWM reuse Object Model, a subset from UML organized in Object Model package; Foundation package which includes business information package, data type package etc; Resource package to represent relation, record or multidimensional data resources, Analysis package and Management Package. To mention more CWM Relational metamodel which is used to specify relational metadata (as majority of operational DBs are rela tional) using classes of relational metadata and their associations like: schema, table, view, trigger, column set, SQL index, procedure, catalogue etc.

### *3.2. META OBJECT FACILITY (MOF)*

Information that is important to the enterprise exists in many forms: system and application programs, databases, process descriptions, technical manuals, regulation guides, and a host of others. To meet the need to represent and exchange this information among applications, Meta Object Facility (MOF) was developed. It provides a common way of expressing different types of modeling information and defines a hierarchy that enables to represent information at progressively higher levels of abstraction, as well as to define ways to express different types of models. MOF is the foundation technology for describing object models, which cover the wide range of object domains: analysis (UML), software (Java, C++), components (EJB, IDL, Corba Component Model), and databases (CWM). It is the OMG standard for metamodeling and metadata repositories. UML is the most widely known and implemented metamodel and has been formally defined using the MOF. In fact, MOF uses the UML notation for describing repository metamodels. MOF provides the capability to represent information at multiple levels of abstraction, or metalevels. Information at a higher metalevel provides a more abstract representation of the information at the metalevel below it. MOF represents core distributed metadata management architecture for OMG [2].

Table 1: MOF metadata architecture

| **Level** | **Information type** | **Example** |
|---|---|---|
| M3 | meta-metamodel | The “MOF Model” |
| M2 | metamodel, meta-metadata | UML metamodel, CWM metamodel |
| M1 | model, metadata | UML models, CWM metadata |
| M0 | object, data | Modelled systems, Warehouse data |

Although this may seem like a simple idea, it is actually the capability to represent information at higher levels that gives the MOF (and XMI) its expressive capability.

### *3.3 XML METADATA INTERCHANGE (XMI)*

XML Metadata Interchange (XMI) is a specification used to interchange metadata among tools, repositories, and applications in a vendor- and middleware-neutral way [3]. By expressing the objects using the W3C eXtensible Markup Language (XML), XMI allows the interchange of metadata among MOF-compliant applications. XMI is applicable to all levels of objects and metaobjects. Although it usually focuses on MOF metaobjects, general objects may be serialized and interchanged with XMI. XMI is closely related to modelling standards, enabling you to employ modelling effectively in your XML efforts. XMI has several advantages:

- XMI leverages XML technologies.
- XMI enables you to use modelling with XML.
- Software that supports XMI creates schemas from models.
- Software that supports XMI provides a higher level of abstraction than XML elements and attributes.
- XMI helps you produce XML documents that can be easily exchanged.
- XMI enables you to create simple documents and make more advanced ones as your application evolves.
- XMI enables you to tailor the XML representation of your objects and document your choices in your models.
- XMI enables you to work with data and metadata.

## 4. WEB SERVICES

Web services are self-contained, self-describing modular applications that can be published, located, and invoked from anywhere across the Web. They provide the mechanisms allowing business logic to be exposed and accessed across any network, thus enabling commercial exchanges and shared business operations. Web services represent a new model for software architecture by taking application design to a new level of openness and interoperability. They have very wide industry support and are part of every platform vendor’s offering. Web services combine the power of prevailing industry standards, such as SOAP, UDDI, WSDL, and ebXML. All standards that Web services are built on are XML-based. XML being a portable and extensible data format helps Web services achieve interoperability among disparate environments. While XML Web Services remain independent of each other, they can loosely link themselves into a collaborating group that performs a particular task. The features of XML Web Services are:

- They allow applications to share information through the Internet, regardless of the operating system or backend software that the applications are using.
- They are modular, self-contained and self-describing
- They are language independent and interoperable
- They are inherently open and standards based
- They are built on proven mature technology Besides XML, which is integrated in every part of Web Services, SOAP is the main protocol that enables business information exchange. It generally uses HTTP as the transport protocol but can also use other protocols such as Simple Mail Transfer Protocol (SMTP). SOAP messages are XML documents that carry messages, such as client requests, Web services response, and any error message that may occur while processing a

request. In order to use a Web service, a client should first be able to locate a Web service. A client locates Web services in a Web registry. Service providers who implement Web services publish their services in such registries for clients to discover. There are various registry standards such as UDDI, ebXML, and Oasis. The client uses the information to interact with the service. In order to publish, a service provider must describe the Web service. The XML-based language for describing a Web Service is called the Web Service Definition Language (WSDL). A Web Service is described in a WSDL document and published for discovery in a registry. The Web services architecture provides the following advantages:

- Reduced complexity
- Flexibility and scalability
- Extensibility
- Interoperability

## 5. META DATA INTERCHANGE

### *5.1 XMI AND MOF*

Since XMI is related to MOF, what can be expressed in a MOF-compliant model can be represented in XMI, at any of the levels in the MOF hierarchy. This provides a standard way to represent all of the different types of MOF-based models that you utilize in developing the architecture of your business's software. The XMI representation is expressed using XML - another industry standard. As a result of this standardization, information represented in XMI can be created and exchanged in the suite of tools you use to develop your applications. The process used in defining and implementing MOF- and XMI-based systems is straightforward. First you need to thoroughly understand the domain before you design a model or a metamodel. Use an UML-compliant tool (or you can do it manually or programmatically) to define your metamodel. If you need to exchange metadata using a stream- or file-based mechanism, use the MOF-to-XML mappings defined in the XMI spec to automatically generate an XML DTD or Schema that can be used to validate the metadata that is being exchanged between systems.

### *5.2 HOW DOES IT WORK TOGETHER?*

UML is used for modelling your objects, relationships, collaborations, MOF compliant tool or repository for implementation of this design, XMI to interchange in the MOF compliant -XMI automatically takes UML models and takes XML DTDs. The roles of MOF, UML, XMI and XML exchange are illustrated in a figure below.

### *5.3 THE BENEFITS OF CWM*

CWM allows to automatically and dynamically share database schemas, data warehouse plans, transformation rules and the business processes. As it is based on UML, significantly richer in semantics and relationships, an abstraction of all vocabulary that you could possibly need is ensured. The excellence of CWM usages of UML for capturing database and component definitions is that now it’s possible to easily exchange and publish document information.

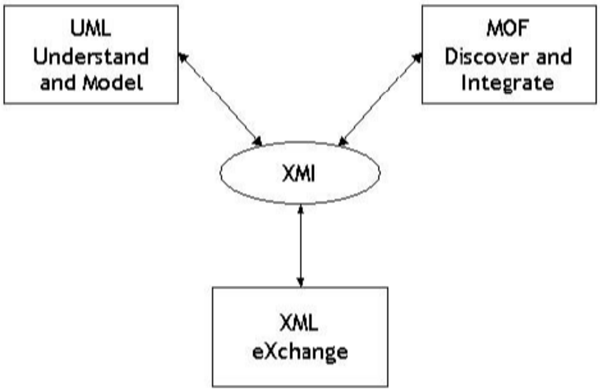

Figure 1: Roles in meta data interchange

### *5.4 WEB SERVICES IN META DATA INTERCHANGE*

CWM specifications describe the interchange of metadata, but do not define typical metadata interchange patterns, nor define protocols bas ed on request-response. As web services become another widely adopted paradigm, it is further discussed as a medium or architecture for metadata interchange.

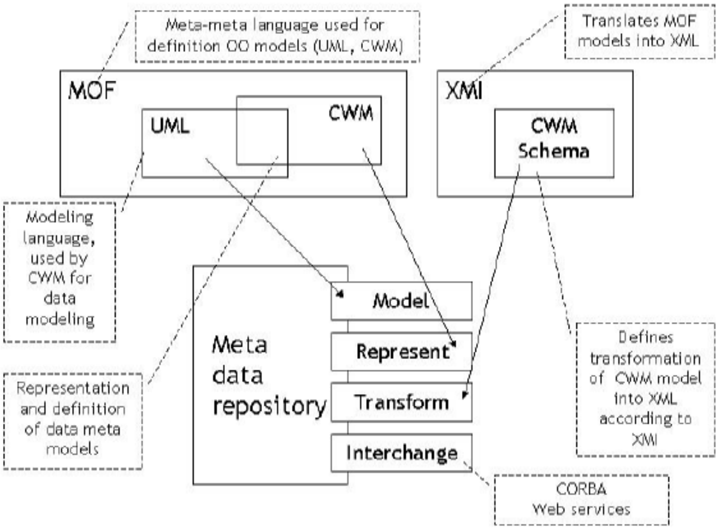
Meta-meta language used for definition OO models (UML, CWM)
Translates MOF models into XML
MOF
UML
CWM
XMI
CWM Schema
Modeling language, used by CWM for data modeling
Model
Meta data repository
Represent
Transform
Interchange
Representation and definition of data meta models
Defines transformation of CWM model into XML according to XMI
CORBA
Web services

Figure 2: XMI, meta models interchange and web services

## 6. CONCLUSION

Current conditions look very good for a “metadata world”. Standards for design, description and interchange are already here and new ones, targeting more specific domains are in process of development. They are all based on XML, data representation standard widely accepted both by users and big software vendors. The basis for efficient distributed metadata management is solid, and web service oriented architecture provides ideal solution for making it real. Infrastructure, protocols and applications already exist and being also XML based, it is perfectly compatible with all metadata standards. There are numerous ways of combining described standards and technologies into solutions. As the conclusion of this paper, here are some of ideas:

- Use Web services architecture as distribution media for metadata exchange, by integrating and automating XMI and Web Services.
- Create a layer providing functions for getting information about your models on various levels of MOF architecture.
- Implement security to control what level of information (model, data, metadata) are available to users.
- Provide metadata design pattern repository service, based on MOF and CWM for design, XMI for packaging and Web service for distribution.
- Enhance metadata pattern service to provide model comparison, quality checks and validation.

## REFERENCES

[1] Object Management Group (2000): *The Common Warehouse Metamodel (CWM) specification*, Object Management Group, www.omg.org

[2] Object Management Group (2000): *The Meta-object facility (MOF) specification*, Object Management Group, www.omg.org

[3] Object Management Group (2000): *The XML Metadata interchange (XMI) specification*, Object Management Group, www.omg.org

[4] Marco, D. (2000): *Building and Managing the Metadata Repository: A Full Lifecycle Guide*, John Wiley and Sons

[5] Barry, D.K. (2003): *Web Services and Service-Oriented Architecture: The Savvy Manager's Guide*, Morgan Kaufmann Publishers

**Received**: 17 December 2003
**Accepted**: 3 July 2004