Decentralized Metadata Development for Open B2B Electronic Business

Fred van Blommestein
Rijksuniversiteit Groningen
Tel: +31651353709
Fax:+31503632275
fred@flowcanto.com

Abstract:

This paper presents a mechanism to maintain metadata in a decentralized way, in order to flexibilize open electronic business. By defining instances as sets, data models can be manipulated in operational systems, together with those instances. Using the UN/Cefact Core Components Technical Specification as a framework, decentralized developed metadata becomes negotiable. The presented mechanism removes bottlenecks of the scalability of open B2B systems.

Keywords:

Metadata, B2B, Core Components, electronic business, XML, UML, EDI, UN/Cefact

1. Introduction

In Information Technology, but also in Ontology engineering, a strict distinction is made between metadata and data, or, in Object Oriented terms, between classes and instances. Metadata and data are defined in different domains. Data models contain Object Classes and cannot be changed by means of instance data. Designing data models is regarded as a strict top-down process and is conducted by developers, not by users.

Some attempts have been made to integrate metadata in run time processes in order to enable run time metadata manipulation. Meijler [1] proposes a system architecture where model data is stored in the run time system, with functions to selectively recompile parts of the information system after updating the model. Yet, model and instances, metadata and data, reside in different domains. Sciore et al [2] proposes to extend SQL compliant databases with metadata, using “Semantic Values”: metadata that is stored as normal data. Lee [3] builds forward on this concept, applying it to electronic business and more specifically to E-negotiation. The concept of Semantic Values bridges the meta-world to the instance world. Semantic Values however are mainly used to qualify simple attributes, like currency codes and measure units, and are not applied to complex associations or hierarchical relationships.

In ontology engineering too exists a strict separation between the ontology engineers and the users of those ontologies [4][5]. Because the latter have polluted the WWW with several billions of website without proper metadata labeling, ontology engineers have developed inference techniques to post-index such websites by analyzing the contents [6]. Yet, the construction of (upper) ontologies is not supposed to be an end-user (or even a webmaster) exercise.

Metadata management in e-business environments seems to be more complex than in media-, bibliographic or web-environments. Not only the number of different object types and their associations are significantly larger, think of products, conditions, roles, agreements, etc., each party in a business relationship may have a different view on those relationships. For a shipper, e.g., a consignment belongs (is a property of) an order, while for a carrier a (consolidated) consignment consists of goods items that may belong to various orders.

In business-to-business communication systems, good old Electronic Data Interchange directories are being designed by sector-oriented or international standardization bodies. XML dialects are mainly being developed by industry consortia or vendors of B2B software or services. Users are of course represented in standardization bodies and consortia, but the development is mainly a top-down process. Once agreement is reached in the consortium, or the standard has been approved in the standardization procedure, users must adopt the whole metadata standard, and nothing but the standard.

In real business processes though, metadata development is a dynamic, distributed process. Trade customs and procedures are evolving in small communities. Often local innovation triggers new logistic and business control processes. Such development is rather bottom-up than top down. Ultimately, commercial and logistical conditions, including metadata, must bilaterally be agreed before a contract is concluded and trade can begin. And even at the time of operational trading, the information to be exchanged may be renegotiated, based on personal preferences and local uncertainties.

Not only the document and message flow is part of the commercial and operation conditions, in case of computer-computer communication every code, business term definition and field length must be agreed. As information systems are locally being purchased, installed and customized, a centralized top-down process to determine the metadata on the interfaces between organizations participating in some commercial relationship will not prove to be feasible. The poor dissemination of EDI systems, despite the fact that EDI technology is mature for over ten years, supports this assessment.

In Business to business communication environments the development of metadata should be a democratic, or rather an organic process. Like most economic activities in a free world, metadata definition should be market driven. Effective information processing capabilities are assets to be used for economic competition. A top-down dictation of how information systems should be configured and what information should be exchanged is in contradiction with the increasing market value of information.

Ultimately metadata is negotiated in the course of normal business processes. This is even the case when the computers of the business partners are not directly being connected and information is transferred using paper documents. During contract negotiation often it is agreed what product information should be available, how delivery and payment conditions will be coded, what document types are needed to control the logistic operation, etc. etc. Each bilateral business relation is somewhat different.

Business process flows are negotiated as well. Business processes and information metadata are closely related. In fact business processes can be defined with the use of metadata. Metadata defines the representation of real world business objects. Business processes are defined by the state transitions and lifecycles of those objects. Discussion on the precise interrelation between metadata and process definition is however outside the scope of this paper. It is safe to assert though, that negotiable metadata are prerequisite for dynamic business process development.

In this research we make metadata negotiable, by treating metadata as normal data. Normal data may be manipulated by normal business users. If metadata can be manipulated by users as well, it can be negotiated in a normal business process. In this paper we stay at the conceptual level. We did not (yet) attempt to implement these ideas into working software. We are however convinced that the concepts presented can be supported in run time systems. But even without runtime support, the concepts can be used to decentralize and "democratize" development of standard ontologies and metadata (e.g. business vocabularies and XML schema).

2. B2B systems

We model a business to business system as two information bases, which are kept in sync by means of exchanging electronic messages. Both information systems are receiving information from real world events. An event can be a decision, made by an employee of one of the communication partners, or a signal from a sensor, like the scanning of some barcode. Events are always happening under responsibility of one of the communication partners (figure 1).

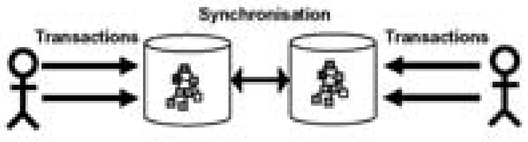

Figure 1. E-business architecture

In order for the databases to be synchronized, they must have a similar or mutually mappable structure. The common structure represents an agreed model of the world that is relevant for the business between the partners. Messages that are exchanged to synchronize the databases contain serialized information conform that common structure. To date such structures are developed and standardized by industry consortia and standardization bodies, leaving little room for localization.

We shall show that the partners, using the concepts of semantic values and semantic hierarchy, can specialize a common information structure to their specific information needs. We are not illustrating the negotiation process that leads to the agreed specialization in detail, as process definition and process control is outside the scope of this paper.

We assume that the mapping of the potential messages to the local databases and application systems has been defined in advance. We do not claim that the mechanism described here leads to self-configuring or self-adapting systems. The pre-defined mappings could be specified in semantic system profiles. The matching of such profiles could be supported by the mechanism described.

3. Mechanism

Metadata defines classes of business objects and their properties. Object classes represent sets of real life business objects or concepts. To avoid confusion with Object Oriented terminology, let us use the word "things" rather than "objects" to indicate the representation of such real life business objects or concepts. Things possess two types of properties: named associations with other things and simple valued attributes. In fact the latter are special cases of the former: a number or a text string can also be regarded as a thing.

So database models basically contain structures of types of things, consisting of the attribute types per thing type and the associations between the thing types. Attributes and associations each have a name and a cardinality (repetition factor) (see figure 2). The model defines the placeholders in the database for instance data: the actual values of the attributes and the actual associations between real things.

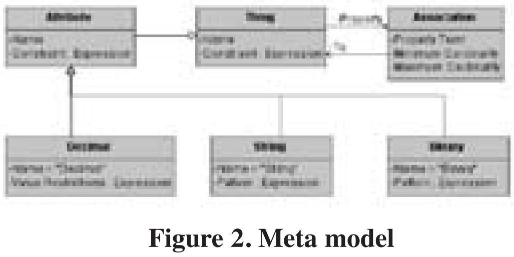

Figure 2. Meta model

However, instead of (only) representing instances in a database, as structures of named values, we could represent *sets* of instances as structures of value *domains*. We can extend the concept of a data type that is a placeholder for some simple number or text string, to represent the full set of possible values, constrained in some way. We then "store" the constraints that filter the allowed set of values from all possible values, rather than the value(s) themselves.

In other words, a data element would not contain a simple value, but would be the placeholder of the definition of a set of values. The set could be represented as a value pair (minimum – maximum), a regular expression or a complex constraint.

In fact, if we would be able to represent sets in the information system, we could also represent instances, as an instance can be regarded as a set with one member. So (on a conceptual level) the presented model includes the support for modeling traditional information systems.

From ontology engineering we have learned that a property can be defined as a labeled, directed association between two things [7]. As stated above, simple attributes are special cases of such associations. If we need to represent sets of things instead of things, we also need to represent sets of properties. Properties have cardinalities. The cardinality of a set of properties is the maximum sum of the cardinalities of the members of the set.

As "meta elements" like cardinalities and repetition factors, being semantic values, can also be regarded as data elements, associations between things can be defined and filtered this way as well.

So instead of (only) instantiating objects, we specialize things. Specialization means: defining a subset of some set that has already been defined. Instantiation is specialization, resulting in a subset with one member. The structure of the model limits the structures of data that may be processed. Note that that is exactly the function of a model: to represent the relevant world in order to predict its behavior. The prediction in this case is the limitation of the data structures that can be produced by (and interpreted from) the universe of discourse.

Specialization is narrowing the value domains of attributes, the set of properties and the set of property-target-sets, by imposing constraints. Specialization is different from (Object Oriented) inheritance. Inheritance is derivation by extension (all properties of the mother-object class are inherited and other may be added). Specialization is derivation by restriction (only a subset of properties is inherited, or properties are interpreted in a more restricted way).

Derivation by restriction is supported by ISO 15000-5, the UN/Cefact Core Components Technical Specification (CCTS) [8]. This specification is targeted to develop metadata for inter organizational business processes. The standard contains naming conventions (based on ISO 11179 [9]) that let the names reflect the specialization hierarchy. CCTS mainly is a profile on UML Class diagrams, and supports the storage of such Class Diagram artifacts in public metadata registries.

In figure 3 we have derived the CCTS Aggregate Business Information Entity from the more generic "Thing" in the meta model. The following example makes use of the CCTS naming conventions.

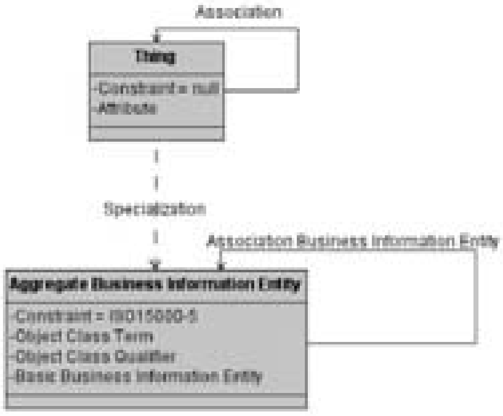

Figure 3. CCTS mapping.

4. Example

In this example we illustrate the mechanism of runtime metadata manipulation. Needless to say the example is highly simplified, not only with regard to business semantics, but also in the application of Core Components, UML and XML, that is not technically complete. In the example we make use of UML Class diagrams and XML messages, only for the sake of illustration. The mechanisms described are on a conceptual level and can be represented in other modeling and data manipulation languages as well.

Suppose two business partners wish to buy and sell vegetables. One of the partners (a wholesaler) has an information system that is dedicated to the sales of fresh vegetables. The other partner, a grocery store, has a general purpose system for food retail. In this paper we are not interested in the technical mapping of data between the two systems, but we show how on a conceptual level a data model can be specialized that supports the wholesale of vegetables. The model is incrementally refined in the course of a business process.

Initially, the data model is copied from a generic model, published by an industry consortium or standardization body. Such model is based on the Core Components specification, as shown in the diagram in figure 4.

The model then consists of the following Business Information Entities:

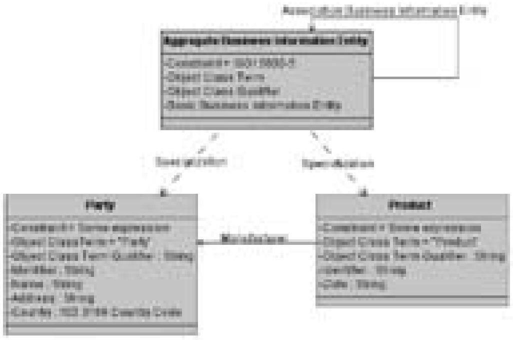
Association Business Information Entity
Aggregate Business Information Entity
-Constraint = ISO15000-5
-Object Class Term
-Object Class Qualifier
-Basic Business Information Entity
Specialization
Specialization
Party
-Constraint = Some expression
-Object Class Term = "Party"
-Object Class Term Qualifier : String
-Identifier : String
-Name : String
-Address : String
-Country : ISO 3166 Country Code
Manufacturer
Product
-Constraint = Some expression
-Object Class Term = "Product"
-Object Class Term Qualifier : String
-Identifier : String
-Date : String

Figure 4. Standard model

```
Product. Identifier. Text
Product. Date. Text
roduct. Manufacturer. Party
P
Party. Identifier. Text
Party. Name. Text
Party. Address. Text
Party. Country. ISO3166_ Code
```

In this example Identifiers, Date, Name, and Address are represented as strings for simplicity. Note that the CCTS defines a mechanism for rich datatyping that in reality would be used here. The Country Code is limited to a specific code list. In the example this could be defined in the Party Constraint. In reality, the Basic Business Information Entity Party. Country. ISO3166_ Code would be a complex datatype with its own constraints, e.g. an enumeration of the entries in the ISO 3166 list of 2 alpha country codes.

The standard model is not fitted for use in the vegetables business. The information systems of wholesaler and retailer need to exchange more vegetable-specific information in order to effectively control the trade operation. The vegetable meta-information is more specialized, and needs to be agreed between the partners before the actual trade can start. The agreement however is reached in the course of a "normal" automated business process, not in off-line negotiations between the IT personnel of the companies. The information systems are not being changed, the mapping between the information exchanged and the information stored in those systems may. Note that more specific (specialized) information can generally be stored in a system tailored to be more generic. Information that is extracted from a generic system needs to be checked and filtered before it can be sent to a specialized system, but this can be the function of intelligent middleware that is aware of the agreed information model.

Now suppose the Grocery Store needs a GTIN (barcode identification) instead of the more generic Product Identifier and a Production Date to control the ultimate sales date. The store would than add:

```
<Product>
  <Constraint>
    GS1_ Identifier.Format="9999999999999"
  </Constraint>
  <ObjectClassTermQualifier>
    "Food"&*
  </ObjectClassTermQualifier>
  <Identifier>
    <PropertyTermQualifier>GS1</PropertyTermQualifier>
    <DataType>
      <DataTypeQualifier>GTIN</DataTypeQualifier>
    </DataType>
  </Identifier>
  <Date>
    <PropertyTermQualifier>
      Production
    </PropertyTermQualifier>
  </Date>
  <Manufacturer>
    <PropertyTermQualifier>
      Food
    </PropertyTermQualifier>
  </Manufacturer>
</Product>
```

Most probably the store would add additional constraints to 'Food_ Product', to narrow representations of the various Business Information Entities.

The resulting Class diagram would be:

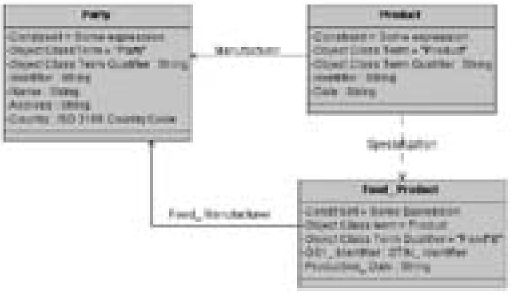
Party
Constraint = Some expression
Object Class Term = "Party"
Object Class Term Qualifier : String
Identifier : String
Name : String
Address : String
Country : ISO 3166 Country Code
Manufacturer
Product
Constraint = Some expression
Object Class Term = "Product"
Object Class Term Qualifier : String
Identifier : String
Date : String
Specialisation
Food_Manufacturer
Food_Product
Constraint = Some Expression
Object Class Term = Product
Object Class Term Qualifier = "Food"&"
GS1_Identifier : GTIN_Identifier
Production_Date : String

Figure 5. Food datamodel

consisting of the following BIE's:

```
 Food_ Product. GS1_ Identifier.

GTIN_Identifier
  Food_ Product. Production_ Date.
Text
  Food_ Product. Food_ Manufacturer.
Party
  Party. Identifier. Text


  Party. Name. Text
  Party. Address. Text
  Party. Country. ISO3166_ Code
```

Note again that in reality use would have been made of the rich CCTS datatyping.

Then the wholesaler needs to change generic "Party" into "NL_ Farmer_ Party", as he only sells products from Dutch farmers, which are identified using a specific identification scheme.

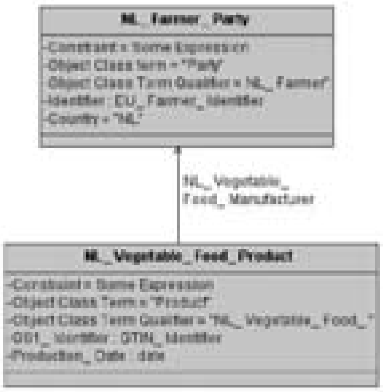
NL_Farmer_Party
-Constraint = Some Expression
-Object Class Term = "Party"
-Object Class Term Qualifier = NL_Farmer"
-Identifier : EU_Farmer_Identifier
-Country = "NL"
NL_Vegetable_
Food_Manufacturer
NL_Vegetable_Food_Product
-Constraint = Some Expression
-Object Class Term = "Product"
-Object Class Term Qualifier = "NL_Vegetable_Food_"
GS1_Identifier : GTIN_Identifier
-Production_Date : date

Figure 6. Vegetable data model

Note that the restrictions the wholesaler adds, do not necessarily influence the way the retailer stores and processes information. The data formats are subsets of the more generic Food data elements. The data definitions are more restrictive.

Now when the actual product catalog is filled with real vegetables (well, with their data representations), exactly the same mechanism takes place. Instead of being a placeholder for some GTIN identifier, the GS1_ Identifier content is limited to a specific product number. The Association with the set of Dutch farmers is per product narrowed down to just one farmer. Data is manipulated exactly the same way as metadata was.

5. Discussion

In this paper we have presented a mechanism to conceptually manipulate metadata as normal data. The application area targeted is the negotiation and standardization of metadata in electronic business.

Electronic Data Interchange systems have never seen the widespread use that was envisaged in the early 90's. A main reason was the inflexibility. Once an EDI system is installed, processes cannot be innovated. Also the bureaucracy of international standardization has proven to be a major bottleneck.

With the advent of XML messaging and inter organizational business process monitoring, the standardization paradigm (one central body issues the process definitions and metadata for an entire industry) has not changed. So most probably open XML messaging will, like EDI, come to a halt and not deliver the advantages foreseen.

By introducing a mechanism by which metadata can be adapted to the needs of local communities and even negotiated in bilateral relations, the bottleneck can be widened. Open XML messaging and open electronic business will become scalable.

We have illustrated the mechanism with the use of UN/Cefact CCTS. The CCTS offers a modular framework for metadata development.

The presented mechanism leads to the "intersection" of the information requirements and capabilities of the negotiating organizations and their information systems. When information systems are very heterogeneous, the intersection will be an empty set, preventing the organizations to do electronic business. Some commonality therefore must exist, e.g. a model provided by a sector organization or industry consortium. But then, without any commonality in data definitions organizations can not do business at all, with or without computers.

The major draw-back of this approach is that for many users and professionals it is counter-intuive to manipulate data and meta-data within the same domain. This may prevent adoption. The application area being open B2B intersectorial communication, wide adoption is needed for the mechanism to be successful. We envisage that a combination of the use of sector standards with limited flexibility will be the most successful introduction strategy.

6. Future work and conclusion

We have only sketched the mechanism, not elaborated it thoroughly. Our present research concentrates on validation, elaboration and formalization of the presented ideas. We also will try to implement the mechanism in working software.

Although here we applied the mechanism to (static) metadata only, we are extending it to process definitions.

We strongly believe that, by combining results from ontology engineering, metadata research and electronic business modeling, we can force a breakthrough in the application of inter organizational communication systems. Information content has long been neglected, by IT specialists (who state content is a user issue) and by users (who expect their IT specialists to solve content related problems).

The World Wide Web has placed 'content' on the research agenda for some time now. Electronic Business will prove to be an application area that directly turns the theories into practical (and economical) benefits.

References

1. T.D. Meijler. Incremental MDD Through Generative Causal Connectedness. Proceedings of UML 2004 - Modeling Languages and Applications Lisbon, Portugal. pages 254-258. Springer Verlag.
2. E. Sciore et al: Using Semantic Values to Facilitate Interoperability among Heterogeneous Information Systems. ACM Transactions on Database Systems, Vol. 19 No. 2 (1994).
3. M.R. Lee: Context Dependent Semantic Values for E-negotiation. Second international workshop on advanced Issues of E-Commerce and web-based information systems (2000) WECWIS 2000. ISBN: 0-7695-0610-0
4. D. Duric et al: MDA-based Ontology Infrastructure. Computer Science and Information Systems (1): (2004). ISSN:1820-0214
5. A. Osterwalder, Y. Pigneur. An e-Business Model Ontology for the Creation of New Management Software Tools and IS Requirement Engineering, Proc. of the CAiSE'2002 Doctoral Consortium, Toronto. 2002.
6. Y. Zou et al. F-OWL: an Inference Engine for the Semantic Web. Proceedings of the Third International Workshop (FAABS), April 2004, Greenbelt, MD, USA. Published by Springer-verlag.
7. N.F. Noy et al. Ontology Development 101: A Guide to Creating Your First Ontology. Stanford University, Stanford, CA, USA. dlm/papers/ontology-tutorial-noy-mcguinness.pdf. March 2001.
8. UN/CEFACT. Core Components Technical Specification – Part 8 of the ebXML Framework. Version 2.01. Adopted by ISO as ISO 15000-5. November 2003.
9. International Organization for Standardization. ISO 11179 1-6. Metadata registries / Specification and standardization of data elements. 1995-2005.