Supporting eBusiness with a dictionary designed in a vertical standardisation perspective

Piero De Sabbata  
ENEA, Bologna – Italy  
desabba@bologna.enea.it

Nicola Gessa  
University of Bologna  
gessa@cs.unibo.it

Guido Cucchiara  
Gruppo SOI, Italy  
guidocucchiara@libero.it

Thomas Imolesi  
FTI-Italy  
imolesi@libero.it

Fabio Vitali  
University of Bologna  
fabio@cs.unibo.it

Abstract

The authors, try to redefine the requirements and the logical architecture for a dictionary that supports a vertical collaboration framework in a standardisation perspective; to this aim the analysis of important vertical and horizontal standardisation initiatives (like EDIFACT and the most recent UBL) as well as the difficulties in setting B2B standards, is a key starting point.

The paper aims to put in evidence how the requirements deriving from an industrial sector, characterised by the large presence of SMEs, leaded to an advanced and flexible approach; it maintains relationships with the world of B2B standards but differs from the philosophy that animates B2B frameworks, influenced by the EDIFACT experience.

The dictionary and the related tools were experienced in the Moda-ML initiative that contributed to set the CEN/ISSS TexSpin and TexWeave specifications for the Textile Clothing industry.

1. Introduction

Business process integration among different and independent enterprises that co-operate along the supply chain is considered a strategic issue for the industry, to have a more flexible and responsive way to interoperate with their partners.

The basis of this integration is the definition of a common B2B (Business-to-business) framework; in a message-based approach, it consists of a set of templates of messages to be exchanged (content layer), transport and security protocols (transport layer) and collaboration models (business layer)[24].

The messages are usually defined by standardisation bodies after a well-defined standardisation process; EDIFACT [12], in the past, and XML, more recently, are the reference technologies to exchange electronic document, exploited by diverse standardisation processes which still present two main problems:

1. the business documents are complex to define, adopt and maintain, and standardisation life-cycle could be very complex. In [28] the need for standardisations life-cycle extensions is highlighted.
2. the standardisation processes related to B2B and ICT (Information and Communication Technology) in general appear to be in a problematic phase; since the end of the ‘90s the participation to the official standardisation bodies is decreasing [25] and one of the reason is the (low) speed of the standard definition correlated with technology life cycles [13]. Many private consortia have been set up to overcome the problem but the results so far are not significantly different.

These difficulties are clearly worsened by the wide complexity and heterogeneity of the business application scenarios. The multiplicity of e-business models, production processes and services surely enhances the business relationships through new commercial paradigms [27], but on the other hand it presents different issues and priorities to solve [26].

Due to these problems the top-down standardisation has proved to be very inefficient. This has led to the creation of closed proprietary islands, that are hampering the growth of the business. A more recent approach focuses the efforts on sectorial perspectives in order to limit the domain, improve the reactivity of the users and shorten the time to release [9]. Nevertheless this approach has resulted unsuccessful in some industrial sector. For example, in the T/C (Textile/Clothing), despite the
huge potentiality and the needs for such standards, the B2B document exchange is still considered a hampering factor [6]. In this sector the EDIFACT technologies never really caught on and still today new Internet-based systems are not spreading. In our opinion, this is due also to the large presence of SMEs, and to the absence of market leaders that rule the whole sector and can impose technological adoptions.

This scenario, common to other industrial sectors, requires the adoption of different approaches to the creation of common standards. In [30] it is recommended to rethink the current standard-setting processes and also to focus on different technical coordinates (i.e. support to an incremental approach, higher attention to simplicity and usability).

The content layer architecture of a collaborative framework should be strongly affected by these considerations; the vocabulary of business-terms upon which a document-based collaborative framework is set up represents the core of this layer.

In this paper we describe a new sector-based approach, tested in the Textile/Clothing sector, for defining document-based collaborative frameworks and facilitate the creation and the management of a sectorial standard. To this aim we present the structure of a sector specific business vocabulary, addressed to build a document-based interoperability framework; as mentioned, the target sector being characterized by a large presence of SMEs (and critical inter-company processes), solutions tailored for large companies cannot work. We also point out the advantages of a sectorial focus, the different philosophy with respect to general purpose (horizontal) initiatives and the technical consequences of implementing the XML document templates.

The paper is organised as follows. Section 2 contains an overall scenario in which business documents and dictionaries are used. Section 3 discusses the requirements and the structure of the vocabulary used to define the core components of business documents. In Section 4 the use case of the Moda-ML project is outlined. Section 5 contains conclusions based on the outcomes of the Moda-ML project together with some future directions of research.

2. The overall scenario

In the last ten years several approaches emerged to tackle the collaborative business process management: all these approaches leverage on XML as the reference language for data exchange and have common problems to face as compatibility, adaptability, security [23]. In [22] a historical overview of different approaches, architectures and initiatives for intra- and inter-enterprises business processes can be found.

Hasselbring and Weigand [18] focus on the semantic aspect of the interoperability problem: a really useful solution must define a common exchange language in which the business message can be expressed. In this paper XML is considered as the basic standard to exchange structured information. A number of standardization bodies have provided some guidelines in developing interoperability frameworks, together with various reference specifications for the development of such frameworks. These specifications cover many aspects, from languages available to define documents (XML), document templates (XML Schema [2]), business messages, process models (UML), and the means to exchange them (SOAP), to the communication protocols on the Internet. Leveraging on the emerging technologies, other initiatives propose architectures and guidelines to face the intra-enterprises collaboration problem.

[19,20,21] provide extensive surveys on B2B interaction frameworks and the different possible approaches in developing the content layer. Even if the number of surveys is significant, they all stress the lack of research initiatives concerning B2B frameworks that take care of the content layer and that develop efficient mechanisms to manage business document definition and maintenance. This effort is carried out mainly by standardization initiatives. Among the most relevant, we can mention the following:

- The CommerceNet eCO framework [31] is basically a document-based approach for B2B interaction. It does not target vertical industry domain, but provides an horizontal interoperability framework. Regarding the content layer, eCO provides a set of XML core business documents called xCBL – XML Common Business Library [16]. xCBL documents contain general information exchanged during business transactions. This information is common for many different application domains.
- ebXML[5] is a horizontal meta-framework to develop specific frameworks; it provides three types of components upon which to structure business documents: core components, domain components and business information objects.
RosettaNet [17] follows an approach similar to ebXML, providing two different dictionary types: the RosettaNet Business Dictionary specifies the properties of basic business activities and the RosettaNet Technical Dictionaries specify the properties of products. RosettaNet contains also a wide set of business process specifications together with the related business documents to exchange.

UBL [3] is a document-centric approach to electronic commerce, providing a narrow set of cross-domain business documents and a more wide set of reusable XML data elements. Our approach starts from the ebXML ideas to develop an original sectorial approach.

3. Document modeling and a dictionary for a SME-aware B2B framework

3.1 The impact of the requirements

Together with leading industries and trading association we have examined the requirements of the Textile/Clothing supply chain and found a few particularly relevant requirements for any document modeling

Readability

The T/C sector is characterized by the existence of a lot of small and medium sized companies with poor ICT skills (and with technology suppliers that are SMEs as well). This emphasizes the need for both document readability and specification understandability.

In the past, in particular regarding EDI systems, and in some early translations into XML, the emphasis was on the efficiency and normalization, to obtain compact representation of normalized data to be exchanged between hosts, no matter about naming rules and human readability. In the following an example of an early opaque XML document provides no hint as to its actual meaning without a complete documentation effort.

Example of poor readability, taken from the documentation of the XML/CEN/ISSS workshop

Nowadays, since the data storage is getting less and less costly, our focus is not on efficiency, nut rather on human readability, in order to facilitate setup and maintenance, and to improve flexibility of systems.

Ready to use: XML-based document models.

A second consequence of the small sizes and the low ICT skills of the actors is the need to build a ready-to-use framework. This means basically to define simple data models without ambiguity or, in other words, to limit the degrees of freedom in implementation. Such data models should be already implemented and represented in a specific technology (XML).

Some existing and past approaches (the BSR initiative for example [32], and also UN/TDED, the ONU dictionary for the international commerce) adopt a technology-neutral representation of the terms of the e-Business. The problem with these approaches is that too little semantics can be encoded in XML-based specifications and thus such specifications allow for different interpretations and consequently for different, often incompatible, implementations. All this hampers system interoperability. In our approach, we still adopt XML as the basic format for specification.

To avoid ambiguity the terms should be constrained by ‘facets’ to limit their features and values. An example is the definition of the precision or the allowable range of values for the numeric data via XML Schema; another example is the large use of enumerations to, a priori, define the set of allowable values; yet, the more recent UBL specifications [3], for example, are expressed using XML, but use only a part of the features available via XML Schema.

Our aim is to avoid the situation in which XML documents are accepted into an enterprise information system as ‘valid’, but contain data values that the internal systems will not be able to manage. On the contrary, we want to assure that the notion of compliancy with the specifications is as much as possible close to the notion of interoperability between enterprise information systems (about the relationship between compliancy and interoperability in ICT standards see [1]).

3.2 The modelling choices
Many process centered document models

The business processes of the T/C sector are quite different from those of other industrial sectors, because of timeliness, like some others ‘fashion’ sectors, and complexity of the production processes; so we believe that sector specific and highly specialised data models (XML Schemas) could result much clearer and more understandable by users and developers. The familiarity with the concepts and the terminology of their own processes is meant to reduce misunderstandings and errors and to create a lower cultural threshold for the adoption of the solution.

In this perspective we target to reduce the complexity of the supply chain with the design of many specialized models of documents rather than with different implementation of the same documents.

Moreover, each of these documents need to be, from the viewpoint of implementation, self-contained and independent. This approach is opposite to that of the most known EDI and XML/EDI implementations (EDIFACT, EANCOM, EAN.UCC, UBL). Two examples of the effects of this choice are the following:

- Differentiation by treated goods: XML documents within the same process but related to different kinds of goods were considered different (for example Fabric and Yarn purchase) because of the necessity to describe different properties of the goods and (for example) different packing instructions.
- No shared documents between different transactions: some transactions exchange similar information, but if the bulks of functionalities (and mandatory information) are different they cannot be supported by the same document model; for example, ‘purchase’ from a catalogue of products (purchase order, based on product identifiers) is different from the request to produce something whose specifications must be included in the order (manufacturing order). On the contrary, reusing common blocks of information inside different document models is strongly pursued.

It is worth noting that the risk of an exploding number of templates is avoided given the limited kinds of goods (“fabric”, “yarn”) to be exchanged; in fact the implementation to support the processes (related to purchase of fabrics and yarns, manufacturing orders and subcontracting for darning, dyeing and finishing) has led to the development of about 21 document templates.

Standard orientation

The proposed framework is close to standardisation activities and results such as:

1. the W3C Recommendations for using XML for EDI;
2. the ISO Naming Conventions (standard ISO IEC 11179 [29] that were adopted also within ebXML);
3. the reference to the recommended codifications (Terms of Payment, INCOTERMS, …); they were selected from the ONU dictionary for the international commerce (UN/TDED);
4. the ebXML architecture
5. the analysis of the business processes and of the related information in EDITEX (EDIFACT implementation for the textile clothing sector [11]).

The adoption of these references ensured a good affinity of the dictionary with the ebXML approach, although straightforward compliance cannot be demonstrated.

Maintenance and fast development

Finally, the large number of dictionary items to be managed, in multiple languages, recurring in many document templates (but combined in different processes), has led to the need to build dictionary tools that:

- support both the collaborative process description as well as the document structures and semantics;
- speed up the maintenance and design process, assuring fast and error-free release of the XML Schemas and related User Documentation;
- facilitate the reuse of semantic blocks (Aggregated Information Entities)

3.3 The dictionary structure

Content model

The data model of the dictionary is based on two sets of entities: the entities related to the representation of the processes and those related to the document templates and business information entities.
The dictionary is composed of the Business Information Entities (BIEs) (as defined in [29]). Within our model they are organised as follows:
a) we define two types of semantic units: the SIMPLE Business Information Entity (representing an "atomic" piece of data) and the COMPLEX Business Information Entity (representing a "molecular" data composed of diverse component units up to the full document template);

b) in order to keep things as simple and flexible as possible, for each semantic unit we create both an XML-type and an instance. In general a COMPLEX BIE may have children that are 'optional' (cardinality 0..N) or mandatory (cardinality 1…N);

c) in order to exploit the richness of the XML syntax, we have differentiated the Simple BIEs, which carry an effective content, into ELEMENTS (for data) and ATTRIBUTES (for meta-data);

d) the formal definition and control over the content of Simple BIE is obtained by applying "facets" (e.g.: maximum length, minimum inclusive, enumeration or code list, ..) to the "built-in simple types" defined in the W3C Recommendation (e.g.: string, integer, decimal, date, …).

It is worth to note that alternative constructions (different sets of children) are allowed when an enumeration type can be replaced by a free text describing the same concept. The use of free text, although not completely forbidden, is strongly discouraged, while the adoption of a set of predefined codes (enumerations) is preferred in order to reduce risks of ambiguities and errors.

The codes are either international (ISO, UN/CEFACT, ..etc..) or self-defined whenever international coding is not available.

**Table n. 1.**

The dictionary is composed of the Business Information Entities (BIEs) (as defined in [29]). Within our model they are organised as follows:

<table>
<thead>
<tr>
<th>object class</th>
<th>property term</th>
<th>Representation</th>
<th>moda-ml diction. entry name</th>
<th>moda-ml tag</th>
</tr>
</thead>
<tbody>
<tr>
<td>fabric</td>
<td>fastness acid perspiration</td>
<td>Measure</td>
<td>fastness to acid perspiration</td>
<td>&lt;CFperspirAcid&gt;</td>
</tr>
<tr>
<td>fabric fault</td>
<td>type</td>
<td>Code</td>
<td>fabric fault code</td>
<td>&lt;fabricFault&gt;</td>
</tr>
<tr>
<td>fabric fault</td>
<td>warpage end point</td>
<td>Measure</td>
<td>warpage ending point measure</td>
<td>&lt;warpEnd&gt;</td>
</tr>
<tr>
<td>fabric fault</td>
<td>type</td>
<td>Text</td>
<td>fabric fault text</td>
<td>&lt;fabricFaultText&gt;</td>
</tr>
<tr>
<td>fabric piece</td>
<td>weight</td>
<td>Measure</td>
<td>piece weight measure</td>
<td>&lt;pieceWeight&gt;</td>
</tr>
<tr>
<td>fabric piece</td>
<td>nuance</td>
<td>Identifier</td>
<td>fabric piece nuance identifier</td>
<td>&lt;mixMatch&gt;</td>
</tr>
</tbody>
</table>

depending on the goods category (for example the txtCOMINFO section, dedicated to commercial info as packaging instructions) and thus reusable in a smaller set of document types;

- depending on the pair "goods category, type of document" (for example toBody, related to order of textile) and thus specific of the document type.

In turn each of these complex BIEs can contain other complex BIEs, but only of equal or greater generality. So, as a thumb rule, generality grows top-down and reduces bottom-up.

The naming convention reflects this degree of semantic generality to clearly identify the generality of each BIE with respect to the document type and requires the adoption of meaningful tags in order to improve the readability.

Namespaces are used to clearly identify the different releases of the set of documents: each version corresponds to one namespace. For example xmlns:ml="http://www.moda-ml.net/moda-ml/repository/schema/v2004-1" clearly identifies that the XML Schema belongs to the 2004-1 version of Moda-ML; until today four versions have been released, two of them publicly available, and a new one is near to be released in March 2005.

**The dictionary entries**

The last foundation of our design is the independence of our Basic Information Entities (BIE) from the final Schema representation. In fact the unique repository of all the BIE is the on-line dictionary (that is public and freely accessible); from those elements, Schemas are computed dynamically to implement the different templates (that are publicly available as well).

This allows a separation between the dictionary (semantic aspects) on one hand, and the final XML Schemas (syntactic aspects) on the other.

In general the same entry can be used in different Schemas with different hierarchical positions, usage conditions (e.g., required, conditional, optional,…) or number of repetitions, providing that the “generality rule” above is maintained.
Each Dictionary Entry, be it a Simple BIE - Element, a Simple BIE – Attribute or a Complex BIE, is identified by
- a unique code (e.g. 35-755-04)
- a unique "tag" (e.g.: <specDate>)
- a unique name (being an ordered triple <object class, property, representation> as they are defined in [29])

Code and Name are interrelated, i.e. in the example above, 35 identifies the Object Class = product, 755 identifies the Property = specification and 04 identifies the Representation = date.

The adoption of the ebXML recommendation about the [29] specifications eases the maintenance and development of the dictionary: terms related with the same physical object or the same kind of property are easily selected from the dictionary (table 1).

It should be noticed that the Representations allowed in our Dictionary (as created from the original XML built-in Types) coincide with the "Approved Core Component Types" of ebXML [7].

Managing semantic diversity

The need to define very general documents types, typical of EDIFACT, leads to manage the “semantic diversity” through the combination of a general object (e.g.: Party) with specific qualifiers (e.g.: Party Qualifier assuming values such as “Buyer”, “Seller”, “Consignee”, etc.).

The result is the compression in the size of the data dictionary, but on the other hand this has often proved to be an obstacle to the clarity of the language. Furthermore, today, this approach bears some problem to fully exploit the XML Schema potentiality in data validation.

For instance, consider the fragment:

```
<measure application="physical dim" 
dim_type="length">10</measure>
```

This fragment is obviously particularly difficult and convoluted to read; but the main problem is that it is very difficult, when using XML Schema, to check separately the range of values of ‘length’ and of ‘weight’, since the XML-Schema syntax does not allow the definition of constraints of an element as a function of the value of one of its attributes.

On the contrary the fragment

```
<length>10</length>
<weight>1</weight>
```

is clearly comprehensible and allows to define different ranges of values using XML Schema. In this way we can find a lot of standard libraries capable to check more closely our documents and, therefore, we avoid the need of implementing ad hoc software from rules found in the implementation guides.

Specialised document rather than general

Analogous issues arise when different optional elements are in similar documents.

In the example (table 2), we might create two similar document types (textileOrder and clothingOrder) or one generic Order including all the optional elements. The textileOrder has an optional “selvedgeText” while the clothingOrder has a mandatory "size" element.

The example demonstrates that forcing few document structures to express too many content types leads to an increase of the complexity of the paper-based documentation and reduces the efficacy of automatic validation of the messages. The adoption of validation languages to express co- constraints (i.e. Schematron [34]) could solve the problem but they are not really as spread and accepted as XML Schema.

```
<textileOrder> 1) 
<header>1-1 
</header> 
<textileBody>1-1 
<articleCode/>1-1 
<selvedgeText/>0-1 
...</articleCode> </textileBody> </textileOrder>

<clothingOrder> 2) 
<header>1-1 
</header> 
<clothingBody>1-1 
<articleCode/>1-1 
<size/>1-1 
...</articleCode> </clothingBody> </clothingOrder>

<order> 3) 
@qualifier = "textile"|"clothing" 
<header> 1-1 
</header> 
<body> 1-1 
<item> 1-N 
<articleCode/>1-1 
<size/>0-1 
<selvedgeText/>0-1 
...</articleCode> </item> </body> </order>
```

Table 2. 1) Specialised textile order. “selvedgeText” is optional. 2) Specialised clothing order. “size” is clearly mandatory.
3) Generic order. It is not clear whether the indication of “size” is always optional or it depends upon the value taken by the order qualifier; “selvedge text” is not excluded for “clothing” order as well as “size” for “textile” order. Only the Guidelines can help.
Schema generators

The choices regarding the use of namespaces, the scope of the single elements, the role of type declarations versus instances can exploit the definition of different programming “styles” of XML Schema (such as “Venetian blind”, “Russian doll”, “Salami slices” [10], “Garden of Eden” [Maler02]).

In our approach the Schemas are automatically developed from the dictionary by the generator according with the “Venetian blind” model, that emphasizes compactness and readability. The choice to maintain all the elements of the Schema in one unique file, without spreading different elements in different files, characterises our Schemas with respect to other frameworks like UBL or EAN.

A potential negative effect is the repetition of many elements in different Schemas, with the risk of a difficult and expensive maintenance; but this risk is completely overcome in Moda-ML by the use of the automatic Schema generator based on the dictionary.

The positive effect, on the other side, is that everything related with a document model is completely contained in a single Schema, with higher comprehensibility and a reduced risk of error in the configuration of the applications. Only some very large tables related with enumeration types (for example codes representing the currencies) are left externally to the document schemas.

4. The use in the MODA-ML project

Our dictionary has been exploited within the Moda-ML (Middleware tOols and Documents to enhAncE the Textile/Clothing supply chain through xML - www.moda-ml.org) project[14]. This project originated in 2001 with the aim of developing an interoperability framework for the Textile/Clothing sector and still is running as the ‘Moda-ML initiative’.

The Moda-ML project collected various industrial and research organisations (ENEA, Politecnico di Milano, Domina, Gruppo SOI, IFTH) together with representative leading Italian Textile/Clothing manufacturers. It has been supported by the Fifth Framework programme of the European Commission within the IST (Information Society Technology) initiative and took part in the cluster of project about Agents and Middleware Technologies (EUTIST-AMI) (IST-2000-28221).

Differently from centralised architectures such as those found in ASPs (Application Service Providers), Moda-ML proposes a peer-to-peer architecture; this structure allows each firm to communicate directly with its partners using simple software modules and a transport interface, without passing from any other intermediary.

Together with the dictionary, the framework contains the necessary tools to provide a powerful way to exchange Moda-ML documents. To provide the necessary functionalities a communication system has been implemented. These tools are based on ebXML messaging service specifications and are collectively called the message switching system.

5. Feedback and conclusions

The first phase of the Moda-ML initiative ended in April 2003, with a good feedback from the pilot users and with the release of 14 document templates covering a portion of the T/C supply chain [8].

In the following period the framework received the support of about one hundred companies of the T/C industry and its results were included in the final specifications of the European standardisation initiative TexSpin, promoted by Euratex (European industry trade association of T/C industry) and by CEN/ISSS.

Using the Moda-ML dictionary a further set of document templates have been developed and published thanks to the support of TQR [33], a large consortium of T/C industries, and are going to be proposed to a new standardisation workshop (CEN/ISSS TexWeave).

A study is running to a posteriori evaluate the impact and the costs and benefits of the adoption of the Moda-ML framework.

The first evaluations, carried out by early Moda-ML adopters, show that it is close enough to the requirements of the industry as well as to the world of ICT standardisation. The search for documents readability and strong Schema typing appears to be a promising way for sectorial standardisation if joined with supporting tools based on a common dictionary.

The gained experiences show that the implementation of a vertical collaborative framework based on XML document templates is really different from those of horizontal frameworks, more close to the EDIFACT approach.

An open issue is the development of new logical and software infrastructures that more and more share information entities and their semantic and coding rather than static document templates [15].

6. References:


[3] B. Meadows, "Universal Business language (UBL) 1.0", cd-UBL-1.0-1.0, OASIS, September 15 2004; docs.oasis-open.org/ubl/cd-UBL-1.0/


[34]http://www.schematron.com/