BMEcat – an XML standard for electronic product data interchange

Claus Hümpel*, Volker Schmitz*

* Frankfurter Institut für Arbeitswirtschaft und Organisation Stuttgart. MT E-Business Systems, Nebenstraße 12, D-70169 Stuttgart, Germany, Tel.: +49 (711) 970 2449, Fax: -2461. E-mail: claus.humpel@iac.fhg.de. URL: http://www.e-business.iac.fhg.de

* Universität GH Essen, Beschaffung, Logistik und Informationmanagement, Universitätstr. 5, D-45117 Essen, Germany, Tel.: +49 (201) 183-4084, Fax: -4086. E-mail: volker.schmitz@uni-essen.de. URL: http://www.bli.uni-essen.de

Abstract. This paper proposes an XML encoding standard for product catalogs, the BMEcat format. So far electronic shop systems and procurement systems use different formats for catalog data. Consequently, suppliers have to support multiple data formats. A standardization with BMEcat simplifies the exchange of product data catalogs between suppliers and buying organizations. The BMEcat format is rich enough to encode hierarchical structures of article groups on the one hand and a unique classification of articles depending on their features on the other hand. Furthermore, different types of transactions (using different parts of a product catalog) can be defined such as the transfer of a whole catalog or a price update for instance.

Keywords: XML, business-to-business, product catalogs, standardization, e-procurement

1 Introduction

Currently, mid-sized and large companies more and more use e-procurement systems and electronic marketplaces to enhance their procurement processes. Here, the biggest challenge is content management, i.e. import and update of electronic product data in various formats. Thus standards for electronic product catalogs are needed.

In order to select an existing or to create a new standard for electronic product data a joint e-commerce working party was founded in late 1998 consisting of research institutes, large buying organizations, suppliers and the BME (Bundesverband Materialwirtschaft, Einkauf und Logistik e.V. = Federation of Materials Management, Purchasing and Logistics).

This group specified the following requirements for a catalog standard:

- Independence from software manufacturers or company specific standards
- Defined and controllable process of development
- High market penetration and acceptance by all partners, i.e. buying organizations, suppliers and software companies
- Applicable to various business domains
- Support for internationalization

We compared and evaluated existing standards like CBL 2.0 (Commerce One 1999), cXML 1.0 (Ariba 1999), OBI 1.1 (OBI Consortium 1998) with respect to the requirements above. As
most of the standards were specified by software companies the requirements of vendor independence and influence on the development process were not met. Additionally, most of the existing standards did not provide enough functionality for a supplier to differentiate his product portfolio from the one of a competitor, i.e. means to include keywords, multi-media attachments and catalog structures were not provided.

The working party concluded that none of the existing standards met the requirements and decided to develop a new standard, BMEcat, which was released in November 1999 (Hüspel/Schmitz/Remmer 1999).

We specified BMEcat in XML (eXensible Markup Language, W3C 1998). XML was chosen because it will probably become the most widely used data encoding standard. There are already software tools for processing XML-encoded data available such as editors, parsers and viewers. This simplifies the integration of an XML-based format like BMEcat into existing software systems. Furthermore, XML is flexible enough to encode the characteristics of product catalogs. This is not the case for HTML, for example.

BMEcat was created to meet the requirements of both suppliers and buying organizations. Suppliers need enough mechanisms to describe their products in an adequate way. For a buying organization a catalog must include all buying-relevant information plus all data that is needed for a seamless transaction support. The buying organization's interest is to provide its employees a rich and easy-to-use product catalog. BMEcat was designed for international use, i.e. support for different languages, currencies and regional requirements. Additionally, international standards were used for basic data formats like currencies, units, etc.

Of course these goals depend not only on the format of BMEcat itself but require furthermore good quality of the supplier's data and technical support by software systems (e.g. catalog generating tools, shop systems and e-procurement systems).

Despite the requirement to apply BMEcat to a wide range of business scenarios we chose a phased approach. First the emphasis was put on catalogs for low-value non-production goods (i.e. "C-articles"). In a next step it is planned to extend BMEcat with respect to production goods, services and highly-configurable goods, for instance personal computers. Further work will be done in the area of article features to incorporate logistical information.

2 General structure of a catalog document in the BMEcat format

One of the principal ideas behind the development of BMEcat was to obtain a standard that is flexible enough to encode complete catalogs on the one hand and transactions on catalogs such as updates, for example, on the other hand.

In the underlying model, a supplier compiles a catalog which conforms to the BMEcat standard in electronic form and transfers it to a buying organization. This catalog is referred to in the following as the catalog document.

The topmost element of a catalog document is BMECAT which contains a header and a transaction part (see Fig. 1). The header contains global data that is valid for all types of catalog data interchange, e.g. details about the supplier and contract information. The transaction part specifies which parts of the catalog are to be transferred (e.g. the complete catalog or just prices that have been updated). The transactions make use of data structures for product data (element ARTICLE) and structures of articles (elements ARTICLE_REFERENCE, CATALOG_GROUP_SYSTEM, FEATURE_GROUP). We will introduce these elements in the following
sections. Afterwards the different transactions will be explained.

```xml
<BMECAT version="1.01">
<HEADER>
...
</HEADER>
<T_NEW_CATALOG>
<ARTICLE>
...
</ARTICLE>
...
</T_NEW_CATALOG>
</BMECAT>
```

Figure 1: General structure of a BMEcat catalog document

3 Product and article data

BMEcat provides an element ARTICLE that is used to encode product data. The possible subelements of this element are rather straightforward. They allow to specify a detailed description (ARTICLEDETAILS), article features which belong to a separate feature system, prices, ordering information and multimedia attachments. Features provide a way to define article properties such as length, weight, etc. in a generic way, whereas the element ARTICLEDETAILS contains unstructured descriptions of the article and additional identifiers referring to various aspects of the article. The structure of the element ARTICLE is motivated by the need for modularization for different kinds of transactions and varies in the context of the chosen transaction type (see Fig. 2).

```xml
<T_NEW_CATALOG>
<ARTICLE>
<SUPPLIER_AID>54-Charlie-R</SUPPLIER_AID>
<ARTICLEDETAILS>
<DESCRIPTION_SHORT>Charlie casual shirt</DESCRIPTION_SHORT>
...
</ARTICLEDETAILS>
<ARTICLEFEATURES>
<FEATURESYSTEM>Menswear</FEATURESYSTEM>
<FEATUREGROUP_ID>123</FEATUREGROUP_ID>
<FEATURE>
<FNAME>Color</FNAME>
<FVALUE>Red</FVALUE>
</FEATURE>
</ARTICLEFEATURES>
...
</ARTICLE>
<T_NEW_CATALOG>
...
```

Figure 2: Structure of the element ARTICLE in the context of the transaction T_NEW_CATALOG

In addition to the structure provided by the DTD the definition of further user-defined elements is allowed at certain positions.
4 Structuring articles

BMEcat provides three different ways to structure articles, namely product structures, catalog group systems and feature group systems. These will be explained in the following sections.

4.1 Product structures

Product structures enable articles to refer to other articles. These references have a fixed meaning, i.e. they define a relationship between two articles.

![Diagram of product structures](image)

Figure 3: Example for a product structure

An article can refer to an associated spare part, for instance. BMEcat defines several types of relationships with a certain semantics plus a type other that can be used if none of the standard types is applicable. An article can have zero or more references to other articles. Each reference must be specified with the `ARTICLE_REFERENCE` tag in the `ARTICLE` section.

```xml
<ARTICLE>
  <SUPPLIER_AID>54-Charlie-R</SUPPLIER_AID>

  <ARTICLE_REFERENCE type="follow-up">
    <ART_ID_TO>54-Dennis-B</ART_ID_TO>
  </ARTICLE_REFERENCE>

  <ARTICLE_REFERENCE type="similar">
    <ART_ID_TO>57-Roger-Sc</ART_ID_TO>
    <CATALOG_ID>47425-4543-U</CATALOG_ID>
  </ARTICLE_REFERENCE>

  ...

</ARTICLE>
```

Figure 4: Product structures and the element `ARTICLE_REFERENCE`

4.2 Catalog group systems

Catalog group systems are used to define a hierarchical structure on articles in a catalog, so
that an end-user can navigate through the catalog and find an article more quickly. A catalog group system (element `CATALOG_GROUP_SYSTEM`) is a set of catalog groups (element `CATALOG_STRUCTURE`). These elements form a tree. A catalog group provides subelements to specify a detailed description, keywords, multimedia attachments and user-defined extensions for the group. Besides the hierarchical search following the structure of the catalog an end-user might use a keyword-based search to directly access certain groups.

The mapping between articles and catalog group is defined in a separate index (element `ARTICLE_TO_CATALOGGROUP_MAP`). Articles can only be mapped to catalog groups that represent leaves in the tree. In order to allow more flexible structures an article can be mapped to multiple groups.

The example below shows a catalog group system consisting of three levels (see Fig. 5) plus its representation in BMEcat (see Fig. 6). The boxes stand for the groups. The numbers inside the boxes are the `GROUP_ID`s of the groups. The lines represent the parent-child relationships.

```
...<CATALOG_GROUP_SYSTEM>
  <CATALOG_STRUCTURE type="root">  
    <GROUP_ID>1</GROUP_ID>
    <GROUP_NAME>Men'swear</GROUP_NAME>
    <PARENT_ID>0</PARENT_ID>
  </CATALOG_STRUCTURE>
  ...<CATALOG_STRUCTURE type="leaf">  
    <GROUP_ID>5</GROUP_ID>
    <GROUP_NAME>Shoes</GROUP_NAME>
    <PARENT_ID>2</PARENT_ID>
  </CATALOG_STRUCTURE>
</CATALOG_GROUP_SYSTEM>
...

<ARTICLE_TO_CATALOGGROUP_MAP>
  <ART_ID>54-Chariyo-R</ART_ID>
  <CATALOG_GROUP_ID>5</CATALOG_GROUP_ID>
</ARTICLE_TO_CATALOGGROUP_MAP>
...
```

**Figure 5:** Example of a catalog structure for menswear

**Figure 6:** The elements `CATALOG_GROUP_SYSTEM` and `ARTICLE_TO_CATALOGGROUP_MAP`
4.3 Feature group systems

Feature group systems allow to classify articles into classes in an unambiguous way. A feature group system is a set of feature groups. Each feature group provides a set of attributes together with their domains of possible values. If an article element is assigned to a feature group it has to provide values for all attributes of that feature group. This information can be used to define a search over the attributes of an article.

In contrast to catalog groups, where an article can be mapped to multiple groups, an article can only be assigned to exactly one feature group. This ensures that assignment conflicts are avoided, e.g. when two feature groups contain attributes that have the same names but different domains of values. The following examples show the definition of a feature group “trousers” (Fig. 7) and the assignment of an article to that feature group (Fig. 8).

```
<FEATURE_GROUP>
  <FEATURE_GROUP_ID>116</FEATURE_GROUP_ID>
  <FEATURE_GROUP_NAME>Trouser</FEATURE_GROUP_NAME>
  <FEATURE_TEMPLATE type="default">
    <PT_NAME>Waist</PT_NAME>
    <PT_UNIT>Inches</PT_UNIT>
    <PT_ORDER>10</PT_ORDER>
  </FEATURE_TEMPLATE>
</FEATURE_GROUP>
```

Figure 7: The element FEATURE_GROUP

```
<ARTICLE>
  <ARTICLE_FEATURES>
    <FEATURE_SYSTEM>Menswear</FEATURE_SYSTEM>
    <FEATURE_GROUP_ID>116</FEATURE_GROUP_ID>
    <FEATURE>
      <FNAME>Waist</FNAME>
      <FVALUE>32</FVALUE>
    </FEATURE>
  </ARTICLE_FEATURES>
</ARTICLE>
```

Figure 8: The element ARTICLE_FEATURES

Furthermore because of their unique classification of articles feature groups can be used to define markets, i.e. to partition a catalog into disjoint segments. These catalog segments can for example be subject of specific contracts, or they can serve as basis for statistical reports. Additionally, the classification can be the base for organizational workflow and accounting. For example all articles that belong to a group “radioactive substances” have to be approved in a special way.

4.4 Combination of catalog group systems and feature group systems

It is possible to combine the advantages of a hierarchical navigation through catalog group systems on the one hand and an unambiguous classification of articles with feature group systems on the other hand.
Therefore one has to define a catalog group system whose last layer of groups is identical to the groups of a feature group system. In a next step all articles have to be mapped to the catalog groups in an unambiguous way. With this approach BMEcat can serve as a transport medium for hierarchical classification standards like UNSPSC (UNSPSC 1999) or eCIS@ss (Institut der deutschen Wirtschaft 1999). The following example shows how to include an eCIS@ss classification schema into BMEcat (see Fig. 9).

Figure 9: Including an eCIS@ss-classification system into BMEcat

If a catalog group system or a feature group system (classification) is known to the receiver of the catalog, the two group systems have to be transferred again and the articles can be mapped immediately onto the groups. The mapping from articles to catalog groups is done using the element ARTICLE_TO_CATALOGGROUP_MAP. The feature group which classifies the article is declared within the element ARTICLE_FEATURES.

5 Transactions

Besides the header each catalog document contains a transaction part. Transactions specify which parts of a catalog are to be transferred in a catalog document. The current version of BMEcat distinguishes between three different kinds of transactions:

- transfer of a complete catalog (T_NEW_CATALOG)
- update of prices (T_UPDATE_PRICES)
- update of products (T_UPDATE_PRODUCTS).

Exactly one transaction has to be specified in each catalog document. The transaction is entered in the transaction part after the header.

A certain transaction type can be chosen to reduce the amount of data that is transferred when
exchanging a catalog document. For instance, a supplier could transfer a complete catalog once a year, using the T_NEW_CATALOG transaction or could perform price updates every three months, using the T_UPDATE_PRICES transaction. In the first case a whole catalog with article information, catalog groups etc. will be transferred whereby in the latter case only price information for certain articles will be transferred.

Depending on the transaction type a specific subset of the elements introduced in sections three and four is chosen. Technically this is achieved by using different DTDs which is explained in the next section.

```xml
<BMECAT version="1.01">
<HEADER>
</HEADER>
<T_UPDATE_PRICES>
<Article>
<Article_Price_Details>
...
</Article_Price_Details>
</Article>
...
</T_UPDATE_PRICES>
</BMECAT>
```

Figure 10: The transaction T_UPDATE_PRICES

6 Representing the specification with a Document Type Definition

The BMECat-specification is represented in four different document type definitions (DTDs), namely bmeCAT_base.dtd, bmeCAT_new_catalog.dtd, bmeCAT_update_prices.dtd and bmeCAT_update_products.dtd.

In the DTD bmeCAT_base.dtd all transaction-independent elements and entities are defined.

There are three other DTDs, each for one transaction. They include the DTD bmeCAT_base.dtd and specify the transaction-dependent elements. This modularization guarantees that the different transaction-dependent components are defined on a uniform basis. The DTDs are modeled in a way to ensure good maintainability.

6.1 The main DTD: bmeCAT_base.dtd

The DTD bmeCAT_base.dtd is partitioned into four logical parts. In the first part the basic data types are defined (e.g. STRING, NUMBER, BOOLEAN). These are defined as entities which are mapped on #PCDATA (see Fig. 1). They are used during the definition of the elements to describe which data type is expected in the XML-files. So they can not be used for a formal verification but can help to create the XML-files and to prepare the processing in the target-sytems. Once a standard W3C XML Schema Language has been approved, this language might be used to provide a more precise description of the BMECat specification.

```xml
<!-- definition of types -->
<!ENTITY & STRING "(#PCDATA)"/>
<!ENTITY & NUMBER "(#PCDATA)"/>
```
In the second part of the DTD several enumeration types are defined. For usability purposes they are also built by using their own entities. Enumeration types which are already specified by international standards, like for instance currency codes (UN/CEFACT ISO 4217-1995) are represented by strings. In future it may be necessary to be more precise and to list the codes in a separate DTD. Enumeration types defined in the specification are modeled by using the select structure from attribute definitions. If there is a context in the DTD, where a specific attribute value out of a list of possible attributes has to be used, an entity is defined which represents this combination visually. So the entity is mapped on the unspecified element, as shown below, and cannot be used for formal verification.

```xml
<!ENTITY & DATETIME_QUALIFIERS "(generation_date|
  agreement_start_date| agreement_end_date|
  valid_start_date| valid_end_date)"/>
```

*Figure 12: Definition of an enumeration type*

```xml
<!ENTITY & DATETIME_AGREEMENTSTARTDATE "DATETIME">
<!ENTITY & DATETIME_AGREEMENTENDDATE "DATETIME">
```

*Figure 13: Definition of two entities to emphasize an expected attribute value within an element

```xml
<!ELEMENT AGREEMENT_ID, (DATETIME_AGREEMENTSTARTDATE), >
( Datetime, AGREEMENTENDDATE ); >
```

*Figure 14: Use of these special entities*

The last entity in this section is used for building complex user-defined XML structures. It is mapped onto STRING, so that the manufacturer has to redefine the USERDEFINES entity with an integrated DTD in the head of the XML-file. This allows a user to create his own complex XML-structures.

```xml
<!DOCTYPE BMECAT SYSTEM "bmecat_new_catalog.dtd">
[
  !ENTITY & USERDEFINES 
  "(UX, UGE, ROOM?, UBX, UGE, VALUATION?, UBX, UGE, LEVEL?)" >
  !ENTITY & USERDEFINES 
  "(ux, xgs, xbc, (pxd, pxs, pxc, level))" >
  !ELEMENT UX, UGE, ROOM, (PCDATA)>
  !ELEMENT UX, UGE, VALUATION, (PCDATA)>
  !ELEMENT UX, UGE, LEVEL, (PCDATA)>
]
```

*Figure 14: Example for redefining the entity USERDEFINES within an XML-file*
In the third part of the DTD bmc_cat_base.dtd the atomic elements of the specification are defined and mapped to the different basic data types.

```xml
<ELEMENT DESCRIPTION_SHORT %STRING/>
```

Figure 15: Definition of an atomic element

The last section of the DTD is used to define non-atomic, transaction-independent elements. In this part of the DTD the selection attributes are specified as CDATA, which are filled with both a predefined list of values and a free user defined entry. This technique is applied to allow the manufacturer of the XML-file to use free attribute values in an easy way without defining new DTD definitions in the XML-file.

6.2 The transaction-dependent DTDs

Each transaction has its own element (T_NEW_CATALOG, T_UPDATE_PRICES and T_UPDATE_PRODUCTS) which is defined in the root element BMECAT within the transaction-dependent DTD. This element combines the main sections of the specification as defined in the DTD bmc_cat_base.dtd relevant for the respective transaction. Especially the subelements of T_NEW_CATALOG are ordered in a way to increase the performance during importing process of the XML-file. It does this by reducing the number of complete parsing cycles on the XML-file.

7 Conclusion and outlook

The XML standard BMEcat enables suppliers and buying organizations to exchange electronic catalogs in a well-defined and rich format that covers most of their requirements. Besides encoding product data BMEcat allows to define complex product structures such as catalog group systems or feature group systems that help the user of a catalog to find products more quickly. It has been shown how BMEcat could be used to encode a classification scheme such as eClass.

The use of BMEcat represents a significant step on the way to standardized business-to-business e-commerce. Firms that are capable of generating documents on the basis of BMEcat thus comply with one of the most important prerequisites for other areas of e-commerce, such as automatic order processing or electronic exchange of invoice data.

As soon as more software companies include the standard into their e-business products BMEcat has the potential to become one of the leading standards in the area of business-to-business e-commerce.

Acknowledgements

The authors would like to thank Thomas Renner who contributed as a co-author to the BMEcat specification and the members of the e-commerce working party.

This work was partly funded by the Fraunhofer Gesellschaft in the context of the Electronic Business Innovation Center (EBIZ) and the University of Essen, BLI.
Bibliography

Ariba Inc. (ed.): cXML/1.0, 1999, http://www.cxml.org, download at 1999-12-07

Commerce One (ed.): Common Business Library, Version 2.0, 1999


Humpel C.; Schmitz V.; Schroder S.: BMEcat Document Type Definitions, Version 1.01c, 1999.
http://www.bmeclat.org, download at 2000-02-08

Institut der deutschen Wirtschaft (ed.): ECL@ss 2.0 Specification, September 1999
http://ecllass.vision-net.de/download/ecllass2_0Edoc.zip, download at 1999-12-12

http://www.openbuy.org/obispecs, download at 1998-12-04

United nations standard products and services classification code organisation (ed.): UNSPSC, 1999.
http://www.unspsc.org/browse/

http://www.unece.org/trade/rec/cf/cfens.html/Notes/ISO4217-95

http://www.w3.org/TR/REC-xml, download as 1999-01-27HYPERLINK