# Business data structures for B2B commerce

Jeusfeld, M.A.

*Published in:*
Informationssysteme für E-Commerce (EMISA 2000)

*Publication date:*
2000

Link to publication

*Citation for published version (APA):*
Jeusfeld, M. A. (2000). Business data structures for B2B commerce. In H. Jasper, J. Küng, & G. Vossen (Eds.), *Informationssysteme für E-Commerce (EMISA 2000)* (pp. 111-130). Universitatsverlag Rudolf Taubner.

# Business data structures for B2B commerce[*]

Manfred A. Jeusfeld
Tilburg University, Infolab
Postbus 90153, 5000 LE Tilburg, The Netherlands
jeusfeld@kub.nl

**Abstract**

This paper elaborates on how product profiles and other business information can be integrated into a business data repository serving an electronic commerce broker. The repository is serving as a uniform access point for the user tools when referencing business data, esp. product and partner profiles. This paper specifies the principal data structures based on requirements derived from requirements analysis in the Dutch construction industry. The data structures subsume:

- product ontology structures,
- product profile structures,
- business partner profiles, and
- user profiles.

A new concept of business data spaces is developed which facilitates tool communication and can be used to interrelate business data sets from different information providers. Finally, this deliverable contains an argumentation about the use of repository software for solving the tasks of business data management andgives some estimates on the implications of investigating a second industry sector.

The main results of this paper are a) the new concept of business data spaces and b) the integration method for remote data sources. Another important result is the introduction of product ontologies into the business data structures. The ontologies address both the representation of multiple languages and of multiple perspectives. The latter enables the search engine to specialize on the terminology of a certain user class.

## 1 Introduction

Business-to-business (B2B) electronic commerce accounts for the lion share of electronic business transactions. Unlike in the business-to-consumer-area, the partners in B2B can be expected to be experts in their area and to be involved in business processes of their own enterprise, i.e. a company participates in B2B in order to support its business processes.

The Esprit project MEMO [MC00] has the goal to develop a a so-called broker, i.e. an Internet site where companies can register as member, display their business information (product profiles, company profiles etc.), search for suitable products and partners, and finally negotiate contracts with selected partners. One subtask is the management of the business information.

This paper is devoted to elaborate on the business data structures incorporated in the business data repository. To structure the paper, we elaborate on the following basic aspects.

### 1.1 What is business data?

Business data is information about a business entity or activity that is stored on some media. Business entities are among others companies, products, employees playing a certain role, contracts, money, rules & regulations. Data representation formats for business entities include text, database formats, and specialized file formats. Noteworthy, a data item becomes a *business data item* due to its involvement in some business process.

[*]This work has been partially supported by the Commission of the European Union under contract 26.895 (MEMO). Partners include ABN-AMRO Bank, CentER AR Tilburg, RWTH Aachen, IHK Aachen, EKD Spain, CheckMark Netherlands, Origin Spain, and Sarenet Spain.

### 1.2 Where does business data come from?

Business data are result or input of some business process inside an organization (company) or between organizations. A business process is an activity (of an organization) that contributes to its overall business goals. Not all business processes produce business data in computer readable form. For example, a phone call from a customer to a company asking for product details must not necessarily result in a computer-readable record about this activity. Nonetheless, essential business processes are typically supported by operational systems (application programs) and result in business data.

### 1.3 For what purposes can business data be used?

The main purpose of business data is its usage in the operational systems which in turn support business processes. For example, if a company start a business relation with a custumer, then it makes records about this company in order to correctly process interactions with the customer. A second purpose of business data is the analysis for decision making. Data warehouse systems are increasingly employed for this purpose. It should be noted that business data is usually confidential.

### 1.4 How should business data be represented?

The operational systems of a company determine the representation of business data. Relational tables are well-suited since they allow to make links between business data in a straightforward way. Analytical systems require business data from various operational systems in a uniform way. Data warehouses store them in a multi-dimensional way where operational facts (e.g. sales) are linked to business entities (like time, customer, product). In summary, the use of business data determines its representation. There is no single “best” representation for business data.

In MEMO, business data from multiple origins (multiple companies, multiple information providers) have to be maintained in order to support the main goal of users of the system: to establish business contacts. Hence, the business data in MEMO has an *inter-organizational* character. Producers and readers of MEMO business data are usually not from the same organization. A second difference to the *intra-organizational* business data is that MEMO business data is direclty incorporated in communication acts between human players of different organizations which use heterogeneous vocabularies.

The remainder of this paper is organized as follows: In section 2, requirements for the representation of business data are discussed and a generic model is proposed. Section 3 uses the generic model to derive data structures for product ontologies (as used in the search engine of MEMO). Finally, section 4 introduces then the notion of *business data spaces* which implements concepts of information ownership and are used to manage business data in MEMO.

## 2 Requirements for business data representation in MEMO

The MEMO broker aims to support business partners to find each other and to negotiate contracts. This circumscribes the *primary users* of the system. *Secondary users* are providing auxiliary business data, e.g. industry standards, law texts, financial data to mention a few. This paper focusses on the primary users since their requirements are essential whereas the requirements of secondary users are enhancements.

A primary user is a representative of a company which participates in the MEMO broker. There are two basic requirements of such a user:

**R1** A user wants to publish relevant business data from the own company. This includes the company profile and the product profiles. Linked to the company profiles are contact data which business partners of that company can use to communicate. Product profiles are those parts of the product catalog of the company which will be visible to other members of the MEMO broker. The act of publishing is part of the marketing activity of the company.

**R2** A user wants to access business data from other companies or sources. There may be different motivations for these accesses. The most obvious one is to search for potential business partners. Another motivation is comparison of own products with the products of competing companies.

In addition to the two basic requirements, there are some requirements about the control and about the quality of business data.

**R3** A company publishing its business data requires to have full control over the content. This includes the ability to update its business data and to remove (part of) its business data.

**R4** The business data accessed by a company must be trustworthy and traceable to its source.

**R5** If a company terminates its membership to the MEMO broker then it must be able to remove its own business data.

Requirements R3 and R5 are potentially harmful since there may be derived business data, e.g. results of searches, which refer to business data items that are removed. Thus, the business data repository has to consider the management of those *dangling references*. The next requirements concerns the indexing of business data:

**R6** Business data should be accessible via the business terms known by the user who accesses it.

This requirement has significant implications on the data structures of the repository. In general, different users have different terminologies for business entities. Hence, proper handling of these terminologies and their link to the actual business data is essential.

## 2.1 User requirements on data structures

The functional requirements [RH99] developed in the MEMO project contain some specifications about business data structures which have to be taken into account. Three blocks of business data structures are identified there:

**R7** User profiles have to be managed and utilized in the user identification function. User profiles include name, organization, address, telephone, and role of the user. The user profile may also contain logging data like the date of last access to the system.

The user profile fields described above have a book-keeping nature, comparable to company profiles. They are not subject to frequent change and basically serve as contact information. Logging data are more dynamic in nature and also more private. They can be used to adapt the user interface to dynamic changes in the behavior of users.

**R8** Company profiles should include name, identification code, juridicial status, address, telephone, fax, mobile phone number, email, Web page, registration authority, registration number, date of establishment of the company, legal type, type of organization (wholesaler, producer, agent, ...), main working geographic working area, description of activities, payment and delivery preferences, certificates, membership to organizations, environmental and health reports, financial data, and finally company officials.

This is an extensive list of company details. For product details, a shorter list of properties is proposed:

**R9** Products are described by their description, brand, form, color, shape, performance numbers, application and usage types, maintenance requirements, quality certificates, safety and health details, durability, technical drawings, etc.

## 2.2 Implications from the requirements

The requirements from [RH99] are rather specific and apparently derived from the construction industry. While the MEMO broker has to be able to fulfill these requirements, the data structures for company and product profiles should not be fixed.

The company profiles of [RH99] go far beyond the smaller list of the European Business Register (EBR). The common denominator of the two alternatives is that features (properties, attributes) are attached to a company. The more details about a company are accessible, the more precise can a search for companies be. On the other hand, features about a company must be kept up-to-date. Another problem of a rich feature list is that the likelihood that all companies provide data is rather low. The company profiles in the MEMO broker must cope with these conflicting goals. To solve this problem, requirement R8 has to be seen in conjunction with R1 and R3. Member companies should have full control over their data.

A company profile has to be seen in relationship to the event of becoming a member of the MEMO broker system. Membership application should be granted based on the publication of company details. A member company should be interested to provide correct details that allow potential partner companies to evaluate and contact them. A second source of company profiles are external company databases. Those databases have considerably less features than one can expect from a member company.

For product profiles, the situation becomes even more complex. Products can greatly differ in the collection of features that are appropriate to describe them. The common denominator of all product profiles is that a product is identifiable and that it is described by a collection of attributes.

Section 3 proposes a scheme which supports heterogeneous product profiles while still allowing for a uniform search interface. The solution lies in the introduction of ontologies, i.e. hierarchies of concepts (business terms) into which products are classified. To solve the problem of heterogeneous product profiles, the features of products are classifed into product attribute concepts. The use of ontologies also addresses requirement R6. The same idea can in principle be applied to company profiles. For reasons of simplicity, we will concentrate however on product profiles and then discuss its use for company profiles.

Section 4 addresses mainly requirements R1 and R3-R5. A structure, so-called business data spaces, is defined in which data ownership and access rules for the business data items are defined.

# 3 Product ontologies in the MEMO repository

Product ontologies were proposed by MEMO WP1 [Leun00] in order to create a topic-based search mechanism. Instead of dealing with product data structures, a user specifies semantic search terms (concepts) and the search engine returns those products which are classified into the search terms.

Communication with user groups [RH99] revealed that different user types use different concepts which may or may not overlap. Hence, two different users can find the same product by referring to different search terms. Consequently, multiple product ontologies, i.e. hierarchies of business terms have to be supported at the same time. The product ontology can be considered to be part of a user profile.

Subsequently, a generic ontology scheme is proposed. It is shown, that a given ontology (an instance of the scheme) can be extracted from the business data repository in a user-definiable output format. Finally, the ontology scheme is extended to represent product attributes.

## 3.1 Generic ontology scheme

A *perspective* is defined to be a set of ontology elements (concepts, lexicals, strings) which are interrelated in a semantic network. The same concept can have multiple denotations (lexicals in different languages). Attributes of concepts, e.g. the size of a door, are also considered as concepts.

In ConceptBase, this can be formalized in a generic ontology scheme with the following class definitions[1]:

```
ConceptNode with
  attribute
    conceptattribute: ConceptNode
end


ConceptNode Perspective with
  attribute
   contains: OntologyElement
end

ConceptNode OntologyElement end

ConceptNode Concept isA OntologyElement with
 conceptattribute
  denotation: Lexical;
  relationship: Concept;
  hasPart: Concept;
  hasNarrowerTerm: Concept
end
```

[1] These classes were developed in close cooperation with Kees Leune [Leun00]

```
Concept!denotation isA OntologyElement end
Concept!relationship isA OntologyElement end
Concept!hasPart isA Concept!relationship  end
Concept!hasNarrowerTerm isA Concept!relationship end


ConceptNode Lexical isA OntologyElement with
 conceptattribute
  label: String;
  language: Language
end

ConceptNode ConceptAttribute isA Concept with
 conceptattribute
    attributeOf: Concept
end

ConceptAttribute!attributeOf isA Concept!relationship end

ConceptNode Language end
ConceptNode String end

Language English end
Language Dutch end
Language German end
Language Spanish end
```

The first class, ConceptNode, is a meta class which is used to define the constructs of the ontology scheme. The second class, Perspective, is used to specify what elements belong to a given ontology. The class OntologyElement is just the super class of all possible elements of an ontology (enumerated in a perspective). The main type of ontology element is the Concept. A concept is denoted by a Lexical and may have relationships to other concepts. Two relationship types are highlighted: hasPart and hasNarrowerTerm. If a concept (e.g. a door) is related to another concept (e.g. a knob) by a hasPart link, then a real door can have a part which is a real knob. The second relationship type is hasNarrowerTerm. For example, a front door (as a concept) is narrower than a door (as a concept) since not all doors are front doors but all front doors are doors. More formally, the extension (or semantics) of a concept is defined as the set of objects. If two concepts are related by hasNarrowerTerm, then the extension of the narrower concept must be a subset of the extension of the less narrower term.

The collection of relationship types may be extended when necessary. If two concepts are known to be ralated by the relationship is neither hasPart nor hasNarrowerTerm, then the generic relationship type relationship can be used. Both hasNarrowerTerm and hasPart are defined as specializations of relationship.

A lexical is the denotation (a natural language expression) for a concept. A lexical is part of a language. We consider just the four languages English, Dutch, German, and Spanish. Of course, the set of languages supported can easliy be extended by just adding another line. The real work is to specify the lexicals, i.e. the instances of the class Lexical. This is being done for a small number of business terms in WP2.

The class ConceptAttribute is a special kind of concepts. It shares all properties of ordinary concepts and adds a new relationship attributeOf. This is the equivalent of the product or company features mentioned in section 2.

## 3.2 An example ontology

In this section, we present an example ontology which utilizes the ontology schema. Note that this example has no relevance to a real application domain and just serves for onderstanding the usage. All definitions can be entered into ConceptBase using its standard user interface.

The first three objects are concepts, i.e. instances of the class Concept. The first has LE0003 as denotation which happens to be an English lexical. The concept C1002 is defined as a part concept of C1001. The second concept, C1002, has two denotations (in English and in German). The third concept, `A0001`, is an attribute concept of `C1001` and has two denotations as well. Note that a concept may be defined without a denotation. This would establish an incompletely defined ontology. Also note that not all concepts may have denotations in all languages. It depends on the amount of effort that is put into the translation. Still, even an incompletely defined ontology is usable for finding business data.

Codes like `C1001` are artificial identifiers that are allocated as soon as a new ontology element is defined.

```
Concept C1001 with
  denotation
    english: LE0003
  hasPart
    part1: C1002
end

Concept C1002 with
  denotation
     english: LE0001;
     german: LG0001
end

ConceptAttribute A0001 with
  attributeOf c1: C1001
  denotation
    english: LE0002;
    german: LG0002
end
```

The following ontology elements are lexicals, the denotations of concepts.

```
Lexical LE0001 with
  label lab: "roof"
  language lang: English
end

Lexical LE0002 with
  label lab: "ground area"
  language lang: English
end

Lexical LE0003 with
  label lab: "house"
  language lang: English
end

Lexical LG0001 with
  label lab: "Dach"
  language lang: German
end
```

The example above shows how an ontology is represented using the ontology scheme. The definitions can be kept in a text file (the originals). One text file with all the definitions of an ontology constitutes a perspective, i.e. the viewpoint of a user on business terms.

To investigate a given ontology, one can load the corresponding original text file into the ConceptBase system [JGJ*95] and ask analytical queries. The following query returns all those concepts which have an English but not a Dutch denotation:

```
QueryClass ConceptinEnglishButNotDutch isA Concept with
```

```
  constraint
    c1: $ exists le/Lexical (~this denotation le) and
          (le language English) and
          (not exists lnl/Lexical (~this denotation lnl)
                 and (lnl language Dutch) ) $
end
```

Such queries are useful during the construction of a product ontology. Since this is beyond the scope of this paper, we do not further elaborate on this issue. It should however be noted that the ConceptBase system provides powerful query capabilities which are suitable for creating consistent ontologies.

### 3.3 Creating ontology reports

The interactive concept browser of the search engine [Leun00] requires a (product) ontology to let the user select the business terms she is interested in for doing a search. To do so, the ontology has to be available in a format that is processable by the search engine. For reasons of simplicity, a text-based exchange data format was agreed upon (see [Leun00] for exact definitions). An ontology loaded from the original can be transformed into this exchange format by utilizing the answer formatting facilities of ConceptBase. There are four ontology reports currently supported:

1. All ontology concepts, format: ”Concept, concept-id,”
2. All lexicals used in the ontology: ”Lexical, lexical-id,language,label”
3. All denotations: ”Denotation,concept-id,lexical-id”
4. All relationships: ”Relationship,relationship-class,concept-id,concept-id”

The reports are realized as a combination of a ConceptBase query plus an answer format definition conforming to the above specification. The following code shows how the third report is realized. The other reports are also supported and included in the first repository prototype.

```
QueryClass ontologyDenotations isA Concept!denotation with
  computed_attribute
    concept: Concept;
    lexical: Lexical
  constraint
    c1: $ From(~this,~concept) and To(~this,~lexical) $
end

AnswerFormat denotationFormat with
   forQuery fq: ontologyDenotations
   order o: ascending
   orderBy ob: "this"
   head hd: "<pre># Denotations of the ontology "
   pattern p: "Denotation,{this.concept},{this.lexical},"
   tail tl: "#end of file</pre> "
end
```

Definitions like the above query are stored in the ConceptBase repository and can be invoked by a HTTP POST operation. The exact host addresses depend on the computer on which the ConceptBase repository is installed. The example below shows the structure:

```
<form action="http://repserver.host.nl:8899" method="post">
<input name=conf type=hidden value="MEMODB">
<input name=action type=hidden value="ASKQUERY">
<input name=actionclass type=hidden value="ontologyDenotations">
<input type=submit value="Get denotations!">
</form>
```

The ConceptBase repository responds by the answer in the specified answer format:

```
# Denotations of the ontology
Denotation,sbk0,sbklex0,
Denotation,sbk0b,sbklex0b,
...
Denotation,top,toplex,
#end of file
```

The topic-based search engine [Leun00] reads the results of the ontology reports and uses them for concept browsing. The output format for ontology reports may also be in XML syntax (see XML structure ICBConcept in [Leun00]). This answer format has yet not been implemented in ConceptBase, but it can be easily done by specifying a different*pattern* in the previous example.

## 3.4 Product catalogs and profiles

A *product catalog* is a list of product descriptions, usually from a single supplier. The way how products are described depends on the product category (identifiable with the concepts of the product ontology). Some products are described by physical properties (size, heat resistance, weight, geometry) while others are described more by the way how the products can be used to solve a task.

This diversity creates a problem for product data representation in the MEMO repository. Existing business-to-business systems like [ECG] use a very general schema for product description:

```
  (EANcode, supplier code, description, product-group, last update)
```

The EANcode identifies a product using an international coding scheme. The EANcode does not characterize the product. It identifies the country, the supplier and the product of that supplier. The supplier code is an internal product identifier that the supplier uses independently of the EANcode. The description is plain text about the product. The product group is the name of the product category for which a given product is an example of. Finally, a date field is used to store the time of the last update of a product record.

Such a simple product data structure is not suitable for the purposes of MEMO:

- A product can only be grouped into a single product group; in MEMO, multiple product ontologies for different user types (contractor, agent, architect, wholesaler, etc.) are incorporated and thus require the ability to group the same product into multiple product groups.

- A purely textual description is only supporting keyword based search.

To achieve a more detailed product data structure without loosing the generality of the approach, a closer look into existing strategies to represent product data is required. We concentrate here on the primary industry sector selected for the MEMO user evaluation: the construction industry.

We consider three example for supplier-defined product descriptions. All descriptions are directed to potential buyers of the product.

The first example is from a supplier for facade panels [TRES]. The product called 'Trespa Meteon' is described by around 20 attributes which are mostly numerical. The attributes are sorted into categories: physical properties, optical properties, mechanical properties, thermal properties, chemical properties, and fire behavior. The physical properties are subdivided into specific gravitity (value example: +/-m 1400 $kg/m^3$ cf. ISO R1183-87), dimensional stability, water absorption, Vapour diffusion coefficient, and coefficient of thermal expansion. Optical properties has just one attribute: color stability hours (value example: 4-5 (3000 hrs; Xenon test) Grey scale cf. ISO 105 A02-87). Fire behavior has 4 attributes, namely the fire behavior norms fulfilled for four countries. Such a product description is suitable for buyers who need to evaluate whether the given product fulfill certain specifications. Typically, architects shall be interested in such descriptions.

The second example is from a supllier of roof windows [VELU]. A product description merely consists of a picture and about 5 lines of text describing the way of opening the window, the opening degrees, and locking machanism. Such a description is close to the simple product data structure proposed by EC-Gate. Apparently, end-consumers are targeted as readers of such a product catalog.

Finally, a vendor for roof panes [KDN] provides a rather detailed product catalog where attributes are grouped under categories. Twelve categories are provided ranging from product type, drawings, environmental features, to general product properties (see also [RH99]). The vendor uses the framework proposed by HCP-EDIBOUW (see below). Individual attribute values are mostly textual.

The construction sector is characterized by a relatively high level of organization, close cooperation between partner companies in project consortia, a high number of product suppliers, contractors, and other commercial partners like architects. The standardization in this industry is pursued by non-profit organizations. One of these organizations is HCP-EDIBOUW, a Dutch organization with the goal to enable electronic business in the construction industry. A relevant document from this organization is the “Branchemodel Elektronische Communicatie” [HCP98]. Among others, it defines a so-called product sheet specifying relevant features to describe a product. The features are grouped into the following categories. Each category has sub-categories which are indicated as well. For all categories, references to further descriptional documents are possible. The first two categories are mandatory. The others are optional.

- **product typing:** product description; brand; composition; conservation instructions and special treatment
- **product details:** details necessary for bookkeeping of the product (orders, accounting)
- **product form:** form; color; texture; dimensions (size); reference projects
- **performance:** mechanical; thermal; fire resistance; chemical; optical; electrical; magnetic; radiation; acoustic
- **utilization:** area of work; construction part; room; inside/outside use; function; restrictions on use
- **processing:** storage; transport; pre-processing; processing; montage; auxilary products; necessary equipment; returnment
- **maintenance:** cleansing and cleansing products; repair and repair products; maintenance and maintenance products; maintenance equipment
- **quality & warrants:** certificates and attests; test reports; guarantees
- **environment & health:** impact on humans; safety records; impact on environment; LCA, Pisa, duarability
- **drawing:** detailed pictures; technical drawings
- **counting:** part lists; receipts; computing models
- **product specification:** additional specifications of the product according to contract between supplier and customer (typically for buildings)
- **other aspects:** for all attributes that do no fall under the previous aspects

The values for the attributes are typically textual. Attributes from the product form category may involve numerival values. Performance numbers are augmentable by measurements norms, e.g. specified by an ISO code. The categories are suitable for the construction industry. Other business areas, e.g. the insurance business, would greatly differ. The principle of having a number of categories and for each category a number of attributes is however generic.

## 3.5 Product profiles in the MEMO repository

The extensive reference model for product sheets by HCP-EDIBOUW is a suitable object for creating product data structures. It omits however some aspects which are important within MEMO:

- Products are listed in product catalogs which are published by companies (usually the supplier).
- The same product can be described by more than one product sheet. Particulary, different markets using different languages and different standards require multiple product sheets.
- A product should be classified into product ontologies in order to support a topic-based search by users. The same product can be classified into multiple concepts of multiple ontologies.

The HCP-EDIBOUW framework [HCP98] also proposes a uniform product identification based on the EAN codes. This code is quite suitable for companies whose product catalog is limited. There are however companies, who can deliver a virtually unlimited number of different products. For example, a producer of paints can customize products based on client specifications. According [HCP98], such products are then identified by the EAN code of the generic product plus a number of parameters containing the client specifications.

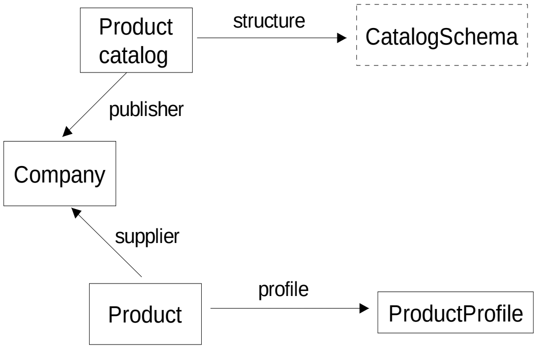

Figure 1: Product profiles and their relationships.

Figure 1 shows the relationship of product profiles to products, catalogs, and supplying companies. A product is uniquely identified, e.g. by its EAN code. Product in the sense of the MEMO repository subsumes customizable products, i.e. products whose properties can be specified by the customer. This is not further elaborated in this paper since it refers to the functionality offered by the negotiation component. A customized product is subject to communication between customer and supplier. Hence, it is not part of a product catalog.

The definitions of figure 1 are directly translated into book-keeping data structures for product profiling in the MEMO repository:

```
Class ProductCatalog with
  attribute
     structure: CatalogSchema;
     publisher: Company
end

Class Product with
  attribute
    supplier: Company;
    profile: ProductProfile
end

Class CatalogSchema isA ExternalObject end
```

A product catalog has a certain structure (normally a table structure). Products are supplied by companies and have a product profile. The product profile will utilize the catalog schema as explained later. The catalog schema is defined as sub-class of `ExternalObject` which is predefined in ConceptBase. It allows to specify a relational table structure together with the location of the database that exports the table.

The remaining issue to be solved is the mapping of external product catalogs to the above data structure and the product ontologies.

## 3.6 Importing product profiles

We assume that a product catalog exists externally from the MEMO repository as a database maintained by the publisher of the catalog. The catalog is exported by the owner as a relational table accessible via a standard Internet protocol, esp. JDBC. To be usable by the MEMO search engine, the entries in the external product catalog must be imported and classified into the product ontologies.

A product catalog schema is defined as follows. We assume that companies are aware of the proposed attributes for product descriptions in their industry sector, e.g. the model of HCP-EDIBOUW for the construction industry. For simplicity, all attributes in the exported product tables are textual (strings). Non-string values must be mapped to strings in case of numbers. Other non-textual values must be replaced by Web addresses (URL) of the document that contains the value, e.g. a picture or drawing of the product. Otherwise, the external product structure has no further limitations. A typical exported product profile looks as follows[2].

```
CatalogSchema CompXYProdSchema with
  field
   EANCode: String;
   description: String;
   area: String;
   warranty: String;
   pictureurl: String
end
```

Defining such a product catalog schema is the task of the member company which wants to publish its product profiles. With the above example, the definitions for the above profile extends to:

```
ProductCatalog CompXYCatalog1 with
  publisher company: CP452525
  structure schema: CompXYProdSchema
end

Company CP452525 end
```

The product profile has to be accompanied by classification instructions which specify the meaning of the attributes according to the product ontology, more specifically the classification of product profile fields into concept attributes. Technically, we just have to include a link from profile fields into concept attributes as well as a link from prodvuts to concepts:

```
Class Product with
  attribute
    classifiedAs: Concept
end

Attribute ExternalObject!field with
  attribute
    TOBECLASSIFIEDAS: ConceptAttribute
end

Class ConceptAttribute with
  rule
    car1: $ forall CA/ConceptAttribute f/Proposition!attribute
            (exists F/ExternalObject!field
             (F TOBECLASSIFIEDAS CA) and (f in F))
            ==> (f classifiedAs CA) $
end

Proposition!attribute with
  attribute
    classifiedAs: ConceptAttribute
end
```

The deductive rule `car1` inherits the concept classification of product field definitions to its instances. Due to this rule, the classification can be defined at the schema level. Instances, i.e. actual product fields like `area="120 m2"` are then automatically classified into the relevant concept attributes.

[2] Actual import of business data from external databases via ODBC is supported by ConceptBase

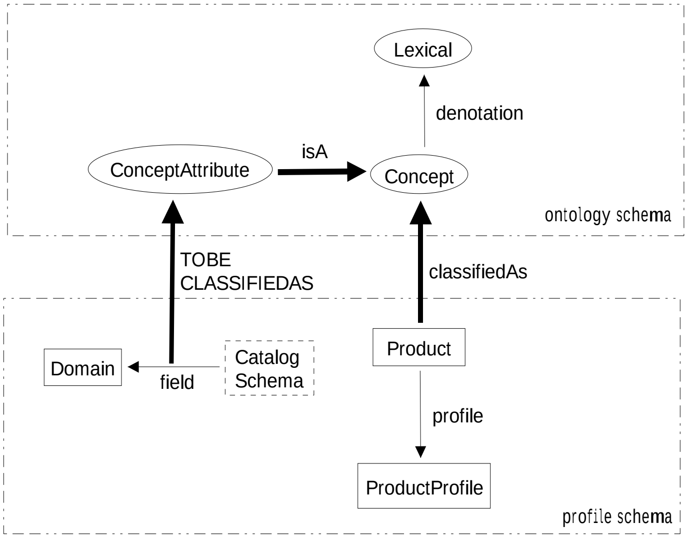

Figure 2: Product profiles classified into an ontology.

Figure 2 summarizes the above definitions and shows how product profiles are related to product ontologies. A product can be classified into multiple concepts of an ontology. It has product profiles (possibly more than one for different markets) which associate descriptive attributes (fields) to a product. The list of fields

### 3.7 Completing the example

This classification link is utilized as follows. We refer to the small example ontology at the beginning of this section:

```
Attribute CompXYProdSchema!area with
  TOBECLASSIFIEDAS ca: A0001
end
```

The product category of the area attribute may be defined as `product form` (cf. the HCP-EDIBOUW scheme). Such a super-category is part of the ontology:

```
ConceptAttribute A0002 with
  hasNarrowerTerm nt: A0001
  denotation english: LE0004
end

Lexical LE0004 with
  label lab: "product form"
  language lang: English
end
```

Now, assume that there our example company `CP452525` offers a product `"EAN 5-78029-123412"` which happens to be a house (concept `C1001` in the example ontology). The product has a product profile `CompXYProductProfile` which among others defines an attribute `area` which is known to be interpreted as concept attribute `A0001` (ground area) in the example. That attribute is known to be part of the category `A0002` (product form) in the ontology.

```
Product "EAN 5-78029-123412" with
  profile p: PP1234
  classifiedAs c: C1001
end

ProductProfile PP1234 in CompXYProdSchema with
```

```
    description d: "..."
    area a: "120 m2"
    ...
end
```

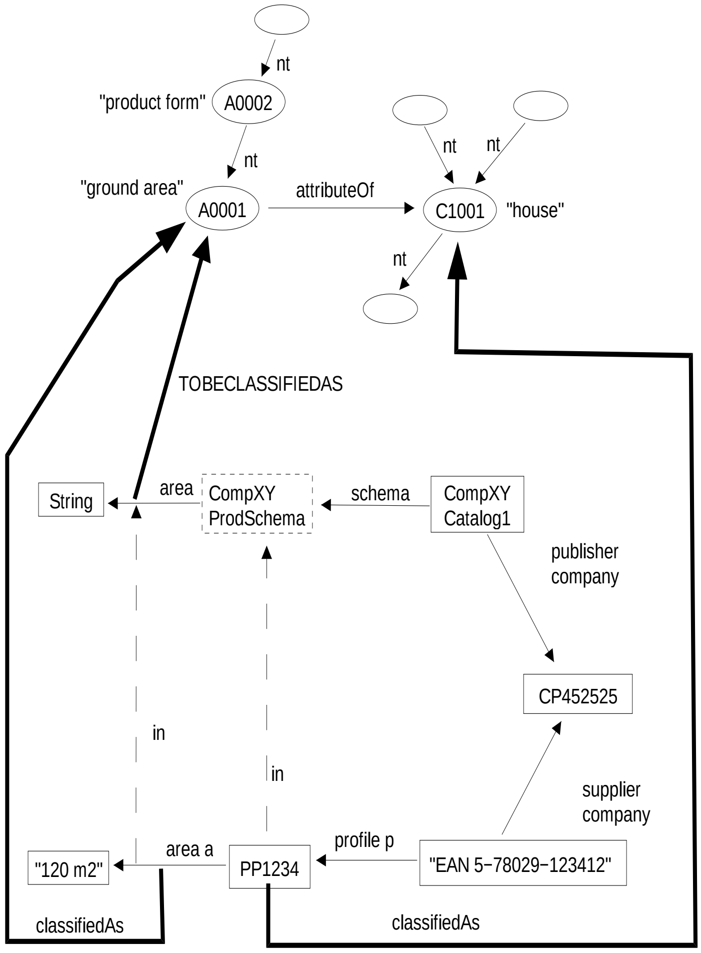

Figure 3: Example of a product classification.

Note that the `area` field of the catalog schema `CompXYProdSchema` is classified at schema definition time. The deductive rule `car1` makes sure that instance attributes (here: `area a="120 m2"`) are classified into the appropriate concept attribute according to the definition in `CompXYProdSchema`. The product `"EAN 5-78029-123412"` is classified at run time, i.e. when the product catalog is imported into the MEMO business data repository. To do so, a separate external database table has to be imported which has the structure

```
CatalogSchema ProductClassification with
   field
      prod_id: String;
      concept-id: String
end
```

Figure 3 summarizes the example. The top part is an excerpt of a product ontology. The empty ovals symbolize that the concepts are part of a network relating concepts and concept attributes. The lexical labels (here: in English) are attached to the concepts to improve the readability.

The central achievement of this chapter is the linkage of (multi-lingual & multi-perspective) product ontologies with product profiles whose descriptional attributes are user-definable in a so-called catalog schema. Standardization of product profiles is fully supported. For example, the product profile attributes proposed by HCP-EDIBOUW appear as corresponding definitions of concept attributes in the respective product ontology. A company that wants to publish its product catalog just has to define its schema (including the classification of schema fields into the concept attributes). Additionally, it has to provide a table which contains the classification of products into the concept hierarchy. As mentioned earlier, the same product may be classified into more than one concept of multiple ontologies. For example, manufacturer would classify a product into different concepts for wholesalers and for architects.

The proposed data structures for product profiles also supports multilinguality. To do so, a company would specify value identifiers for the descriptionary fields instead of actual values. The value identifiers can be treated like concepts in an ontology having different denotations (lexicals) for different languages.

# 4 Business data spaces

Business data spaces are a facility to support the management of business data. It can be argued that a MEMO broker for electronic commerce will only be successful if it provides innovative services (e.g. for negotiation) and access to high-quality and focussed business data. Therefore, we argue that a *market owner* as a user role has to be introduced whose responsibilty is to manage business data spaces for a well-defined market. Inside a market, users with different roles share an information space that is structured by so-called *business data spaces*. A business data space (BDS) is used to specify the context of business data, in particular visibility to members of the market.

Subsequently, data structures for users, user groups, and business data spaces are defined.

## 4.1 From users to business data spaces

The preceding section defined company profiles. We assume that only users from member companies (subsuming non-profit organizations) may become member of the MEMO broker system. A MEMO user has some profile (mainly contact details), a principal role, a MEMO user name and a password. The latter is used to check authorization and can be replaced by a more sophisticated method when required.

```
Class UserGroup with
  attribute
    member: MemoUser
end

Class MemberCompany isA Company,UserGroup
end

Class MemoUser with
  attribute
    password: String;
    viewpoint: Perspective;
    profile: UserProfile;
    role: UserRole
end

Class UserProfile
end

Class SimpleUserProfile isA UserProfile with
  attribute
    fullname: String;
    selfdescription: String;
    telephone: String;
    fax: String;
    email: String
end

Class BusinessDataSpace with
  attribute
    owner: MemoUser;
    memberGroup: UserGroup;
    purpose: String;
    state: State;
    creationtime: String;
    contains: Proposition;
    hasPartBDS: BusinessDataSpace
end

Class AccessRight
end

Attribute BusinessDataSpace!memberGroup with
  attribute
    withAccess: AccessRight
end
```

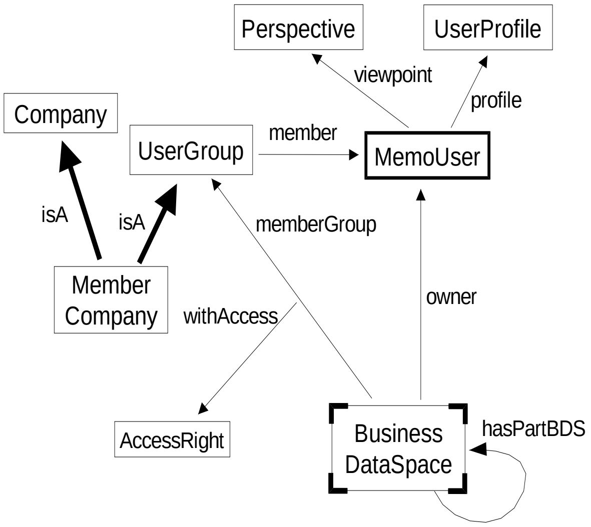

Figure 4: Business data space schema.

Figure 4 puts the concept of business data spaces into the context of the known concept of companies. Member companies are those companies which are member of the MEMO broker. They are considered as user groups since some of their employees are users of the MEMO broker. User have a user profile like companies have a company profile. Moreover, a user has a primary role (e.g. sales person) and a viewpoint being the ontology with the business terms she is dealing with (cf. section 3).

A business data space is owned by a user (possibly more than one in case of shared data spaces). The owner is usually the user who has created the BDS and assigned a purpose to it. The owner can specify user groups who participate as members of the BDS. The participation is subject to access right specification. The state of a BDS is a token like `searching`, `negotiating`, `finished`, `archived`. The set of states is extensible.

The above model follows the proven approach of the BSCW system [AM99]. A business data space corresponds to a BSCW workspace. MEMO users correspond to BSCW users. The part business data spaces correspond to folders inside a BSCW workspace. The new aspect the BDS concept is the close relationship to company and product profiles (via product ontologies). By assigning the viewpoint to a user record, the search context can be set individually for each user.

### 4.2 Using business data spaces

Business data spaces are proposed as the main organizing tool for managing information in the MEMO repository. Information providers like chambers of commerce will publish their company catalog via an appropriate BDS by specifying the user group which has access to it. Likewise, a company publishing product catalogs would do so via a BDS.

A second type of use is for session management. A MEMO user who wants to buy a certain product would start a new session, i.e. a BDS with the user as single owner and no further user group having access to it. The tools invoked by the user to fulfill the purpose of her BDS are supposed to store their results within the BDS via the `contains` feature.

The following scenario shows the usage for storing search results. The ConceptBase frames should be generated by the search engine.

```
* start new session

BusinessDataSpace session99881 with
  owner u: MU123
  purpose p: "Find window frames for construction project PX"
  state s: searching
  creationtime t: "2000-01-17; 10:37"
end

* store search results

SearchResults S66 with
  companies
     c1: CP12313;
     c2: CP34556
end

BusinessDataSpace session99881 with
  owner u: MU123
  contains sr: S66
end
```

The above object assume some prior defintions for search results and user records. The first object `MU123` defines the user and is the result of the user registration sub-system of the MEMO broker. Note that the viewpoint configures the search engines capabilities. The second frame is an example of an internal data structure that can be defined by the designers of the MEMO tools to pass results, e.g. from search engine to the negotiation component. Such data structure definitions are possible at any time with the MEMO repository. The tools using business data spaces must however conform to those data structure definitions.

```
MemoUser MU123 with
  username n: "Peter Smith"
  ...
  viewpoint vp: ArchitectPerspective
  role r Architect
end

Class SearchResults with
  attribute
   companies: Company
end
```

The following query shows how to extract business data from the repository to display the open sessions of a given user:

```
GenericQueryClass OpenSessions isA BusinessDataSpace with
```

```
  parameter
    u: MemoUser
  constraint
    c: $ (~this owner ~u) and
         not ((~this state finished) or (~this state archived)) $
end
```

When a user like `MU123` logs into the MEMO broker then her open sessions can be computed a single access to the MEMO repository: `OpenSessions[MU123/u]`. The user may then select for example `session99881` to continue her work. Results of tool invocations are then stored into the selected session. By this mechanism, the same user can have an arbitrary number of open sessions without mixing the context. Tools used to progress in the context of a session just need to refer to the session identifier like `session99881` in order to store results or pass results to other tools.

Further usage scenarios for business data spaces, esp. for publishing company and product catalogs remain to be elaborated but are not reported in this paper.

## 5 Conclusions

Deliverable D3.2 proposes data structures for product and company profiles. Product profiles are defined in combination with the product ontologies which are crucial for topic-based search.

Company profiles are defined around the standard of the European Business Register. It is shown how existing company profiles can be mapped into the EBR format by a view mechanism.

Business data spaces combine information about product (ontologies of users) and companies (membership of users) to form a data structure which allow to manage business data in the repository.

Finally, D3.2 elaborates on the reasons for using ConceptBase for the implementation of the repository system and estimates the effort for including a second industry sector for validation.

The implementation of the data structures still requires considerable effort. First, the import of product profiles requires to assign unique product identifiers (by attaching company codes to product codes). This is necessary since not all product catalogs use the EAN coding system. Second, a strategy to materialize the content of the repository in the file system (or a database management system capable of storing XML objects) has to be developed. The various product ontologies do not need to be loaded into a repository system when the MEMO broker system is running. Instead, its content has to be stored in a format readable by the search engine at a space accessible to the search engine. It is only accessed when a user with the appropriate viewpoint is working with the system. Finally, the concept of business data spaces has to be further elaborated and specialized for different usage scenarios.

## References

[AM99] W. Appelt, P. Mambrey. Experiences with the BSCW Shared Workspace System as the Backbone of a Virtual Learning Environment for Students. Proceedings of the World Conference on Educational Multimedia, Hypermedia and Telecommunications ED-MEDIA 99, Seattle, June 1999.

[ECG] EC-Gate Home page. `www.ecgate.nl`, January 2000.

[HCP98] HCP-Edibouw. Branchemodel Elektronische Communicatie. HCP-EDIBOUW, Driebergen, Netherlands, November 1998.

[IBN] IHK Aachen: Internet-Business-Network. `http://www.aachen.ihk.de/ihk_f_uk.htm`.

[JGJ*95] M. Jarke, R. Gallersdörfer, M. Jeusfeld, M. Staudt, S. Eherer. ConceptBase – a deductive object base for meta data management. *Journal of Intelligent Information Systems,* **4**(2), pp. 167–192, 1995.

[KDN] KDN-Teween home page. `www.kdn-teewn.nl`, January 2000.

[Leun00] C.J. Leune. Topic based search and term suggestion for trading information. MEMO Deliverable 1.2, MEMO internal workspace, January 2000.

[MC00] MEMO-Consortium. MEMO home page. Online http://www.abnamro.com/memo/, Feb 2000.

[PJWJ98] M.P. Papazoglou, M.A. Jeusfeld, H. Weigand, M. Jarke. Distributed, interoperable workflow support for Electronic Commerce. In *Proc. GI/IFIP Conf. Trends in Electronic Commerce (TREC'98)*, Hamburg, Germany, June 3-5, 1998.

[Plat] Computer Associates. Platinum Repository.
`http://www.cai.com/products/decisionbase/repository.htm`

[RH99] P.J. Raadsheer, J.H.M. Henselmans. Functional requirements. MEMO Deliverable D5.4, MEMO internal workspace, October 1999.

[TRES] Trespa home page. `www.trespa.nl`, January 2000.

[UREP] Unisys Universal Repository.
`http://www.unisys.com/marketplace/urep/`

[VELU] Velux home page. `www.velux.nl`, January 2000.