

An Extensible Business Communication Language

Hans Weigand & Wilhelm Hasselbring

Infolab, Dept. Information Management and Computer Science, Tilburg University,
PO Box 90153, NL-5000 LE Tilburg, Netherlands,
Email: {weigand|hasselbring}@kub.nl

Abstract. A main problem for electronic commerce, particularly for business-to-business applications, lies in the need for the involved information systems to *meaningfully* exchange information. Domain-specific standards may be used to define the semantics of common terms. However, in practice it is not easy to find those domain-specific standards that are detailed and stable enough to allow for real interoperability. Therefore, we propose an architecture that allows for incremental construction of a shared repository including a multilingual thesaurus, which is used in a business communication language. Communicating information systems then refer to the common thesaurus while exchanging messages. Our emphasis is on separating semantics (in the thesaurus) and syntax (in XML). Therefore, our extensibility is not only that of XML, but also the extensibility of the semantics that is modeled in the shared repository.

The business communication language XLBC is presented and how it can be used in electronic commerce applications. XLBC message patterns and conversation protocols are stored in the shared repository as well.

1 Introduction

In spite of all the surrounding hype, it is becoming increasingly clear that electronic commerce is taking off on a global scale, not only in the consumer market but also in business-to-business and business-to-administration application areas. However, there are also many barriers that still need to be taken. One barrier is the standardization of the message formats for business communication. Although business-to-business electronic commerce has a longer tradition of electronic data interchange in the form of EDIFACT, it is generally recognized that EDI is too costly and not flexible enough to cope with the dynamics of the new economy [KM97, KL96, MG98]. However, traditional EDI is often being re-examined to define the *meaning* of the transferred data (semantics), and XML is employed as the practical foundation in which to structure this information (syntax). XML is a markup language for creating self-descriptive data; in contrast to HTML, it separates style and content and is extensible in the sense that new tags can be used as long as they are defined in the DTD (document type definition). For electronic commerce, it is especially interesting that one format can be used both for electronic messages (to be processed by computers) and for human interfaces; an XML document itself is already, to some extent, readable for humans (what an EDI document is not), but especially when it is accompanied by a style document (XSL), it can be presented by means of a web browser in any desired layout. This feature not only allows to have one

single interface to application systems (for humans and for systems), but also enables hybrid set-ups in which humans and systems are involved in different stages of the process and the same format can be used throughout.

However, XML on itself will not do the job. The receiving party can recognize something as a valid XML document, and when it has the accompanying DTD, it can check whether it adheres to this DTD, but nothing is said yet about the meaning of the data elements. If every company were to develop its own DTDs, there would be no real interoperability. So, although XML is technically superior to EDI X.12, it does not solve the huge problem that EDIFACT has worked on for years, namely, how to define the contents of the messages. What elements should be there, how are they represented and what do they mean? If XML should be used in business-to-business electronic commerce, something equivalent to the EDIFACT standards must be in place. The standardization of messages can be at different levels: at the lexical level of character sets (data representation), the syntactical level of message structures, to a deeper semantic level of vocabulary and integrity constraints. If communicating parties want true communication, they must agree not only on the form but also on the meaning of the messages. It is not necessary and even undesirable to strive for explicit agreement on all semantics. All that is needed is that confusion is avoided and that messages can be processed automatically at both ends - which means that a mapping can be made to the local schema.

From an institutional point of view, standards are vehicles for facilitating coordination of economic activities [H⁺95]. Instead of repeated coordination between actors, a standard solves a number of dilemmas for actors in a situation where communication is required. A standard therefore diminishes the need for ad-hoc coordination. On the other hand, there is an increased need for concerted action when standards are created or changed. Normally, this concerted action is performed at the level of standardization committees. However, at present this often turns out to be infeasible, or only feasible to a very limited extent. In today's open and dynamic business environment, the partners have to take over part of the standardization process to themselves. This can involve two or more partners who intend to set up a business relationship on the spot, or an industrial platform/market owner who does this standardization for its members. For such a setting, a flexible system architecture that allows for dynamic evolution of the business communication language is required.

In this paper, we propose an architecture that allows for incremental construction of a shared repository, which is used in a business communication language. Our emphasis is on separating semantics (in the repository) and syntax (in XML). In this way, the architecture not only supports extensibility of message syntax (as offered by XML), but also the extensibility of the message semantics (by means of the shared repository).

2 Foundations for Formal Business Communication

Communication languages have been defined in several environments: KQML in the AI community, ACL in the FIPA agent consortium, and FLBC in the business communication field. In this section we give a short overview of FLBC that we take as our starting

point. We also indicate at what points we have deviated from FLBC while developing XLBC.

2.1 FLBC

A number of researchers have investigated the possibility of developing general-purpose formal languages for business communication (FLBC), notably Kimbrough, Moore, Covington and Lee. The impetus for this research has been a common assessment of the fact that existing EDI standards leave much to be desired in flexibility, in expressivity, in clarity, etc. Kimbrough & Moore mention two assumptions of the FLBC approach [KM97]. The first assumption states that a properly designed FLBC should permit business messaging to begin and to proceed without the business partners having to come to a separate and specific agreement concerning the content, structure, and proper interpretation of the messages to be exchanged. This assumption is very close to the approach called Open-EDI [KL96]. It does not require that every message be based entirely on public lexicons. Exchange of particular vocabularies should certainly be allowed, as should ‘linguistic bootstrapping’ (agreement to define new expressions in terms of existing expressions). The second assumption calls for a semantic foundation of the language in First-Order Logic.

FLBC is based on speech act theory that makes a distinction between the illocutionary force of a message and the propositional content [KM97, Moo99]. By explicating the illocutionary force, FLBC makes clear that messages are not just pieces of data, but (intend to) have some social effects, such as creating an obligation. Moreover, the propositional content is represented in such a way that it contains indeed a proposition, that is, a statement that can be logically true or not (in the case of an assertive message), or an action to be taken (in the case of a directive message). This is in contrast to traditional EDIFACT messages where all the necessary data elements are present (otherwise it would not work, as a matter of course), but not structured in the form of a proposition or action. As a result, the syntax definitions of traditional EDI are somewhat arbitrary and unpredictable. In the FLBC approach proposed by Kimbrough & Moore, the basic structure of FLBC messages is defined once for all. Of course, different message types (also called patterns) can be defined on this basis, such as for ORDER, INVOICE, etc. These message patterns differ in the actions that they refer to and the arguments that these actions take. However, they can always be parsed, and interpreted to some extent; for the full interpretation, the receiver should know the meaning of the terms and predicates. In contrast to earlier languages such as EDIFACT, formal semantics are considered important in FLBC in order to arrive at rigorous definitions and facilitate automatic processing. FLBC uses First-Order Logic as much as possible. XLBC is based on communication semantics described in [WVD95] called Illocutionary Deontic Logic, which is a modal logic. The following subsections summarize our main extensions and adaptations to FLBC.

2.2 Conversation objects

Business messages that are exchanged typically occur in conversation patterns. For example, an order is followed by an acknowledgment, and together they form a transac-

tion. It is not the order itself, but the order transaction that creates the obligation for the other party to deliver, as expressed in the legal definition of a purchase order as a “written authorization for a supplier to ship products at a specified price which becomes a legally binding contract once the supplier accepts it.” In [WvH99], a pattern language is described that distinguishes different levels of conversations. Here, we do not repeat the details of this framework, but just assume that conversation objects at different aggregation levels can be defined.

2.3 Roles

In line with linguistic theory, Kimbrough takes a predication/argument structure as the basic representation of events [Kim98]. This means that events are thought of as a special kind of entity of a certain type (e.g., delivery) and the arguments are the entities that play a role in the event (who delivers, what is delivered, etc). The role names are taken from a controlled set. In line with Functional Grammar [Dik89], we allow a predication to have restrictors in addition to the role arguments. Restrictors further identify the entity or event denoted by the predication. Using restrictors, it is possible to add any attribute to the predication, as long as it is semantically coherent. Predication restrictors are the semantic equivalent of adverbial expressions in natural language.

2.4 Predicates

FLBC does not say much about the structure of the lexicon in which predicates are defined. With XLBC, the lexicon – or thesaurus as we call it – is set up in a linguistically motivated way and with support for multilinguality. Predicates can be nominal, adjectival or verbal. Verbal predicates denote some action or activity. Predicates with their roles and possible selection restrictions are called predicate frames and are stored in the multilingual thesaurus. Some are very general (e.g. deliver, arrive), and some will be domain-specific. The thesaurus entries specify lexical information with the representation in one or more languages and semantic information.

3 The Shared Repository

XLBC is based on a separation of syntax (XML messages) and semantics. The semantic definitions are stored in a shared repository. This could be made public, but it could as well be restricted in use for one market only. The shared repository contains two parts: a multilingual thesaurus, and an XLBC component library. The former defines the elements of which messages are composed, and the latter defines the messages and higher aggregation structures. We stress that the shared repository distinguishes several component levels, and is not just a large set of possible DTDs. We strongly believe that such a structure is necessary in order to obtain real extensibility.

3.1 The Multilingual Thesaurus

The different elements that make up a message refer to certain real-world entities, such as the parties involved, the products/goods that are exchanged, etc. Usually, these items

will have different attributes such as location and price. Both the attributes themselves and the values of those attributes have a lexical representation, such as "company" or "product". These representations are basically words that denote some concepts. The thesaurus makes a distinction between words (or lexicals) and concepts, and supports a semantic network of concepts.

Figure 1 illustrates our general communication architecture, in which information systems communicate by means of a common business language (in our case XLBC messages that contain references to concepts in the multilingual thesaurus). The multilingual thesaurus for this language is managed by the vocabulary server, which allows for dynamic extension of the terms used in the business language. The communicating information systems are informed of changes to the common thesaurus by means of a notification service. The shared repository also manages the XLBC Document Type Definitions (see section 4).

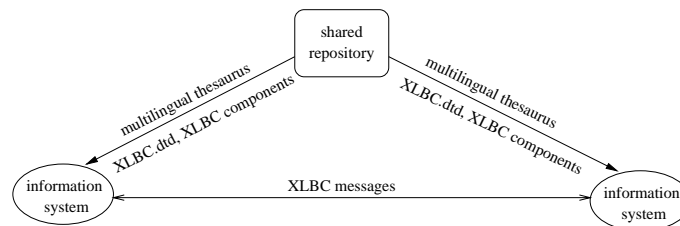


Fig. 1. The general communication architecture. The shared repository manages the common terms that are used in XLBC messages to be exchanged by the communicating information systems.

The thesaurus is built up around a semantic network of concepts. The concepts are defined through their relationships with words and other concepts. Words are the natural language representations of concepts. Multiple words can describe one concept (author and writer may be *synonyms*), and one word can be used to describe multiple concepts (company is a *homonym* that describes both an industrial organization or just a group of friends). We call the relation between a concept and a word a denotation. In addition, concepts are interrelated. For instance, the concept "author" may have as a generalizing parent (*hyperonym*) a concept that represents a person. This says something about the concept of author. A concept may have different types of relations with other concepts. As in typical object-oriented modeling techniques [Mey97], the parent relation (specialization/generalization), the part-of relation, and non-hierarchical relations between concepts (e.g., in the form of predicate frames) are used for defining concepts in the thesaurus. As can be seen in Figure 2, elements in the different message components that make up the conversation shall link to concepts which have been defined in the semantic network of the thesaurus.

Concepts are part of a semantic network but always organized in generalization hierarchies to facilitate top-down access. We have found it useful to distinguish three levels in this hierarchy. The *top level* contains categories such as Event, Physical Entity, Ge-

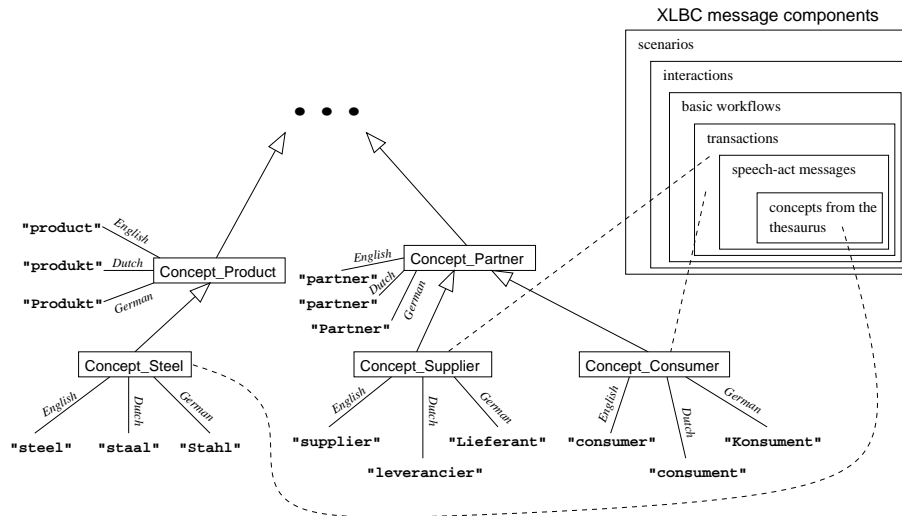


Fig. 2. Linking the XLBC components with the concepts in the semantic network of the multilingual thesaurus. Exemplary, translations of some concepts to English, Dutch and German are given.

ographical Entity, Measure Unit, Agent, and Time. It may also contain sub-categories, such as Transport Event, Transform Event, and Transfer Event. The middle level is called the *basic level* and it contains the concepts that are closest to human experience, such as Deliver (a Transport event), Pay (a Transfer event), Day, Month, City, etc. The *bottom level* is made up of sub-concepts, or specialized concepts, such as the many forms of delivery and payment. For retrieving information from the thesaurus, one usually starts at the basic level.

The thesaurus should also provide meaning definitions as far as these are relevant for the business application. Remember that we adhere to a minimalist approach as far as meaning definitions are concerned. We do not have one standard definition method because the relevance can differ and not all concepts are the same. Usually, real-world concepts (natural kinds) such as House, Horse, or Car, defy a formal definition but for the human interpreter, a verbal meaning definition (as in an ordinary dictionary) is usually sufficient, whereas the machine does not need to reason about them. For event types, it is often sufficient to specify the category (for example, Pay is a Transfer event) and the selection restrictions on the arguments (in this case, a Money theme). Sometimes, additional preconditions and postconditions are useful. For message types, the operational meaning typically depends heavily on the context. For example, an Invoice can be defined as a claim for payment in return for goods delivered or for a service provided. However, whether the Invoice causes an obligation to pay or presupposes it, is not so clear; in fact, different options can be chosen, leading to different trade scenarios. Trade scenarios are represented in the component library, and it is at this place where the operational semantics of message types such as Invoice are to be found (see

the next subsection). In the thesaurus, a verbal definition is sufficient then. Although real-world concepts usually do not need a complete definition, sometimes quality regulations are relevant. For example, a product may only be called ‘chocolate’ if it contains at least a certain percentage of cocoa. Although this is not yet possible in the current implementation of the thesaurus, we intend to allow the inclusion of such regulations, but not as part of the definition, but as what they are: rules on the way concept names are applied to instances. In this way, we not only separate syntax and semantics, but also semantics and pragmatics. For the representation of rules, we need an expressive knowledge representation such as is used in logic-based ontologies (cf. [D⁺98]).

3.2 The XLBC Component Library

XLBC is based on speech act theory. Furthermore, the XLBC messages have been grouped into different aggregation levels of conversations, as illustrated in Figure 2. At each composition level, various patterns can be defined. As explained briefly in section 2.2 and in detail in [WvH99], speech acts typically go in pairs. The request/accept transaction is an example of a pattern at transaction level. Transactions can be grouped into basic workflows, of which reciprocal interactions can be construed. It is possible to specify rules on, for instance, the sequence order in which elements of XLBC-patterns must occur.

The thesaurus and component library provide an architectural framework that is specifically aimed at extensibility. It is unrealistic to assume that any organization could arrive at an exhaustive repository of standard components and standard messages that any industry can adopt. This should not be the aim. Instead, what the repository should provide is a (well-structured) set of building blocks by means of which parties can develop new components with relative low effort. Given a shared repository with a certain content, users can develop new business processes top-down, by taking an existing business process and adapting it according to their needs, or bottom-up, by taking the basic terms of the Thesaurus - perhaps extended with newly defined ones - and compose them into messages etc. The advantage of the XLBC approach - and FLBC approaches in general - is that it supports compositional semantics. For example, if a new kind of business action has been defined - let’s say, ‘review’ instead of ‘deliver’ - then it can be combined without further effort into a request for reviewing, a commit to reviewing, or a report on the reviewing being finished. When the Thesaurus is owned and managed by a professional community instead of one party, the procedures for updating and extending the shared repository is equally important. In the conclusions, we will have a few suggestions on this point, but the issue as such is beyond the scope of this paper.

4 The Extensible Language for Business Communication XLBC

Below, we introduce the language XLBC (Extensible Language for Business Communication). XLBC combines the semantic orientation of FLBC with the extensible syntax of XML. The Extensible Markup Language (XML) is *subset* of SGML that is designed to make it easy to interchange structured documents over the Internet [McG98]. The main role of XML (as opposed to HTML) in interoperable systems is likely to be for

defining the structure of data to be exchanged between heterogeneous information systems. The syntactic structure of XML documents is specified by a Document Type Definition (DTD), which may be thought of as a schema of the document. Instances of XML documents can only be understood in relationship to their DTD. When we talk about XML documents we need to refer to explicit DTDs.

In XML, documents are made up of elements. Each element is *tagged* using a start-tag and usually an end-tag. For example a diagnosis element in a text might be tagged as follows: `<diagnosis>Diabetes mellitus</diagnosis>`. A start-tag takes the form `<name>` while the end-tag takes an identical form except that the opening angle bracket is followed by a slash character `</name>`. The XML standard is not concerned with the semantics of textual elements. The developers of XML tags have to choose intelligible names for the elements they identify and should document their proper use. The DTD sets out the structure using a simple syntax to specify the order and contents of each element. Various features can be combined to allow complex document structures to be defined. Elements can have *attributes* which provide additional information about each element. Attributes enable us to use one element in a variety of contexts without having to create many similar elements that resemble each other. Any number of attribute value pairs may be defined for any element.

The top-level DTD for XLBC messages is defined as follows:

```
<!ELEMENT MESSAGE (SENDER RECEIVER SPEECH-ACT+, CONTEXT) >
```

This definition defines a message as consisting of one or more speech acts and a context element. The sender and receiver are terms referring to a company or person.

```
<!ATTLIST MESSAGE
      ID          CDATA          #REQUIRED
      TYPE        CDATA          #THESAURUS
      DATE        CDATA          #REQUIRED>
```

Examples of message types are "QUOTE" and "PURCHASE ORDER". Since the message types are taken from a controlled vocabulary, we associate a check function with this message attribute. The check function is called when the XML document is processed and it checks in the multilingual thesaurus whether the value is indeed a valid message type.

4.1 Speech Act and Propositional Content

Speech act and propositional content is then defined as follows:

```
<!ELEMENT SPEECH-ACT (CONTENT)>
<!ATTLIST SPEECH-ACT
      TYPE        CDATA          #THESAURUS>

<!ELEMENT CONTENT (PREDICATION | COMPLEX-PRED)>

<!ELEMENT PREDICATION (ARG+, RESTRICTOR*)>
<!ATTLIST PREDICATION
      ID          CDATA          #REQUIRED
```



```

TYPE          CDATA          #THESAURUS
OPERATOR      ( 'NEG' | 'POS' ) #IMPLIED
ASPECT        ( 'INTEND' | 'START' | 'CONTINUE' |
               'STOP' | 'DONE' ) #IMPLIED>

```

```

<!ELEMENT ARG (TERM | PREDICATION | TIMEREF)>
<!ATTLIST ARG

```

```

        ROLE          CDATA          #THESAURUS>

```

These definitions contain the basic structure of the speech acts. A speech act is divided into an illocution and a propositional content. The speaker and addressee of the speech act are already determined by the sender/receiver of the message. The propositional content consists of a number of predications, where a simple predication takes the form of a predicate (usually a verb, such as ‘deliver’) followed by one or more arguments. An argument consists of its identifying role and a term. We also allow for sub-clauses, and therefore instead of a term, it is also possible to fill an argument recursively with a predication. The ID of a predication is the identifier of the action occurrence, as it is used in event semantics. The TYPE attribute defines the speech act type taken from the controlled vocabulary in the multilingual thesaurus (specified via the keyword #THESAURUS). The predicate operator can be positive or negative (default: positive). The aspect operator can be used to indicate a phase of the event (‘going to v’, ‘start v-ing’, ‘is v-ing’, ‘stop v-ing’, ‘has v-ed’). A complex predication allows for Boolean combinations of simple predications.

```

<!ELEMENT COMPLEX-PRED (PREDICATION|COMPLEX-PRED)+ >
<!ATTLIST CONNECTOR
CONNECTOR      (AND|OR)          #REQUIRED>

```

4.2 Context

The context of the message contains all kinds of pragmatic features, such as the session of which the message is a part, a link to a previous message, but also the thesaurus used or the preferred language setting. This definition is similar to the ones used in FLBC [Moo99] and FIPA [Fou].

```

<!ELEMENT CONTEXT ( #PCDATA ) >
<!ATTLIST CONTEXT
        SESSION      CDATA          #IMPLIED
        PREV-MSG     CDATA          #IMPLIED
        THESAURUS    CDATA          #REQUIRED
        LANGUAGE     CDATA          #THESAURUS>

```

4.3 Terms and References

A term is an expression by means of which the speaker refers to some entity [Dik89]. If the entity has a unique identifier, then the reference is simple, comparable to the reference by means of a personal name in natural language. However, this is not always the case. For example, when a customer orders 3 items of product X (identified by some EAN code), then the reference includes both a product type (identified by the EAN code) and a quantity. The situation becomes more complex when the product is

not sold in discrete items. In that case, some unit-of-measure is needed, for example, 200 kg. The entity type can be uniquely identified by an EAN code, but it can also be described by means of a general entity type and a list of restrictors. For example, the entity type can be 'Toyota Carina Model 1432' and restrictors can specify the color, the transmission system, etc.

```

<!ELEMENT TERM (NAME?, RESTRICTOR*)>
<!ATTLIST TERM
    TAG          CDATA          #REQUIRED
    TYPE         CDATA          #THESAURUS
    CODE         CDATA          #THESAURUS
    QUANTITY     CDATA          '1'
    UNIT         (ITEM|KG|CM|M|..) 'ITEM'>

<!ELEMENT NAME EMPTY>
<!ATTLIST NAME
    OID          CDATA          #REQUIRED
    CODE         CDATA          #THESAURUS>

<!ELEMENT RESTRICTOR (#PCDATA)>
<!ATTLIST RESTRICTOR
    ATTR-NAME    CDATA          #THESAURUS
    DOMAIN       CDATA          #THESAURUS
    UNIT         CDATA          #THESAURUS
    OPERATOR     (=|<|>)       #IMPLIED>

```

A term is defined as an optional name element followed by zero or many restrictors. In the minimal case, we only have the term attributes: tag, type and code. The most important one is the type; it specifies the concept type of the object referred to, for example, 'money' or 'brick'. The concept types are stored in the multilingual thesaurus (either in the general domain or in a specific domain). The domain from which the type is taken is indicated by means of the 'CODE' attribute. It is also possible that the concept type is defined by means of some external standard, such as the EAN product code. The tag attribute is used to identify the term within the context of the XML document and can be used elsewhere in the document (co-reference).

The quantity and measure-unit attributes take care of the quantitative aspect of the term. Note that the objects referred to can be both discrete (and countable), or non-discrete (mass-terms). If they are conceived as being non-discrete, a measure unit must be

Restrictors give further qualifications of the term reference. We have opted for a representation in terms of attribute/value pairs. Attribute names are color, weight, seize, price, etc. The possible values of the attribute are taken from some domain, such as the domain of colors (to be more precise, the ISO definition of color names, or some other formalization), the domain of money, etc. In some cases, the value is numeric, but note that this again implies the use of a measure unit (3.5 meter, 500 pound, etc). We also give (limited) opportunity to provide not just a value, but relationships with some values (e.g., 'higher than 3.5 meter').

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE FLBC-MESSAGE SYSTEM `xlbc.dtd`>
<MESSAGE MSG-ID='128576' MSG-TYPE='CONCEPT_DELIVERY-ORDER' DATE='19990812'>
<SENDER><TERM TAG='x1' TYPE='CONCEPT_COMPANY' CODE='OBI'>
      <NAME OID='5012345678900' CODE='EAN' />
      <RESTRICTOR ATTR-NAME='CONCEPT-CONTACT_PERSON'>V.Suos
    </RESTRICTOR></TERM>
</SENDER>
<RECEIVER><TERM TAG='x0' TYPE='CONCEPT_COMPANY' CODE='OBI'>
      <NAME OID='6012345678900' CODE='EAN' /> </TERM>
</RECEIVER>
<SPEECH-ACT ILLOCUTION='CONCEPT_REQUEST'>
  %<CONTENT>
    <PREDICATION ID='e1' PRED='CONCEPT_DELIVER'>
      <ARG ROLE='CONCEPT_AGENT'>
        <TERM TAG='x0' TYPE='CONCEPT_COMPANY' CODE='OBI' />
      </ARG>
      <ARG ROLE='CONCEPT_RECIPIENT'>
        <TERM TAG='x1' TYPE='CONCEPT_COMPANY' CODE='CONCEPT-OBI' />
      </ARG>
      <ARG ROLE='CONCEPT_THEME'>
        <TERM TAG='x2' TYPE='5012345678900' CODE='CONCEPT-EAN'
          QUANTITY='900' UNIT='ITEM' />
      </ARG>
    </PREDICATION>
  </CONTENT>
</SPEECH-ACT>
<CONTEXT LANGUAGE='ENGLISH' SESSION='ABC170' />
</MESSAGE>

```

Fig. 3. An example XLBC message for delivery orders.

4.4 Example

In this section, we present one simple, but complete example of an XLBC message that is equivalent to the FLBC message given earlier in section 2.1. The XLBC version of this message is displayed in Figure 3. given as follows:

For a better understanding of the merits of XLBC, we contrast the XLBC specification with a functionally equivalent encoding in a standard XML-EDI format, viz. SIMPL-EDI. In SIMPL-EDI this order message looks like:

```

<ORDER RefNo='0001'>
  <BGM>128576</BGM>
  <DTM1>19970812</DTM1>
  <RFF IDType='CT' FileID='652744' Line='112' />
  <NAD Of='BY' EAN='5012345678900' />
  <NAD Of='SU' EAN='6012345678900' />
  <LIN LineNo='1'>5012345678900</LIN>
  <QTY>900</QTY>

```

```
<DTM2>19970812</DTM2>  
</ORDER>
```

The XLBC message starts with a message header with attributes including message identifier, message type, date, sender, and receiver. As can be seen, the message date is also in the SIMPL-EDI message, although less explicitly, in the DTM1 field. The sender and receiver are represented by their names in simple terms.

The illocution of the speech act is a `CONCEPT_REQUEST` in XLBC since an order is a request to deliver. Note that neither the illocution, nor the action requested - the delivery - is explicit in the SIMPL-EDI message. The buyer and seller are identified by means of an EAN code. In the XLBC message, their roles are made explicit (agent/recipient), as well as their types (here `CONCEPT_COMPANY`). Although there are of course many ways to identify a party, we have chosen to use the same EAN coding in the XLBC message. The LIN segment in SIMPL-EDI contains the order lines. XLBC does not talk about order lines, but about objects that are the 'theme' of delivery. The objects are identified by means of an EAN code, as in SIMPL-EDI. Note that this is an identifier of a product type. What is ordered are 900 items of that product type.

Since the meaning of the RFF element in SIMPL-EDI message is not completely clear, we have not represented this information in the XLBC message. We have included a session number to the whole message. This session number can be used in later messages, for example, the `INVOICE`. The example shows that the XLBC message is more explicit, and therefore also longer than the SIMPL-EDI message. The concepts are defined in the thesaurus with the associated language translations. The thesaurus is built up around a semantic network of concepts.

5 MeMo

The language XLBC is currently used in the ESPRIT project MeMo. The main objectives of MeMo are:

1. To develop a prototype of an open market framework that mediates business-to-business communication and acts as a central marketplace where businesses can find relevant information and facilitate conducting transactions with suppliers
2. To develop a formal language for business communication that allows users to communicate meaningful business-tailored messages that can be processed by machines and by humans
3. to develop monitoring mechanisms (guaranteed by a Trusted Third Party) so that message exchanges between potential partners are validated and reconstructed for future references and trading activities

The MeMo project intends to improve the Electronic Commerce transactions between SMEs giving them the opportunity to establish trading relationships, regardless of their physical location or language.

The Negotiation Module of MeMo supports b2b negotiation and contract building. The MeMo negotiation module does not replace human informal communication but enables human agents to structure their communication. In this way, personal contact,

necessary for building up trust and long-term relationships, is not excluded, while at the same time the structuring of the messages allows automatic processing. The results of the negotiation can be forwarded automatically to the fulfilment phase, and do not have to be typed over. If there is already an EDI agreement between the parties, the fulfilment process can be generated completely without human intervention.

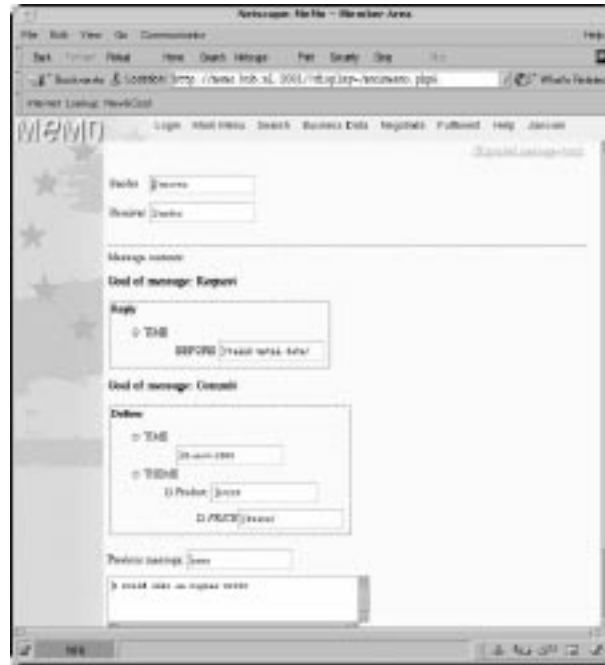


Fig. 4. Sample screen shot of MeMo prototype - an XLBC quotation message

Since language is often a big barrier for international trade, especially for small companies in Europe, the Negotiation Module also contains a multilingual thesaurus in which key terms of international trade are given in multiple languages, as well as necessary explanations.

The MeMo project started in 1999. Figure 4 presents a screen shot of the current prototype, showing a negotiation quotation message specified in XLBC. The message is simplified and contains the elementary fields only. The layout follows strictly the XLBC structure. It can be expected that different user communities demand their own layout adapted to their working practices. At the moment, this can be realized in MeMo only by tailoring the accompanying XSL-file which has to be done for each message type. Whether the layout specification can be done on a higher level than XSL is still under investigation.

6 Related Work

In the absence of a complete and comprehensive set of document formats, as EDIFACT intended to provide, several attempts are made to set up repositories of components that can be taken out and used by business partners. XML.ORG, for instance, aims at being an independent industry portal for the standardization of XML applications in electronic commerce, whereby it serves as a reference for XML DTDs. BizTalk.ORG is a competing industry initiative started by Microsoft. Commerce One's CBL defines a set of building blocks. These building blocks are then pulled together to make the actual documents describing the interactions between two organizations. OASIS is a non-profit international consortium dedicated to accelerating the adoption of product-independent formats based on public standards, notably XML. Oasis could serve as a host for an XML registry and repository, including XLBC. The sister organization ONTOLOGY.ORG has the objective to use ontologies to address the problems that impact the formation and sustainability of large electronic trading groups. In these initiatives, the goal is to put DTDs at the disposal for communication partners, but the dynamic, ad-hoc extension of common DTDs is usually not addressed.

[Lee98] suggests the use of a central repository in which formal trade procedures can be stored. Users can download these trade procedures — formally represented as Petri-Nets — adapt them if necessary, and then adopt them immediately for execution. The XLBC approach goes further by providing not only trade procedures (corresponding to XLBC components at the workflow level), but also term definitions and message types. [G⁺99] proposes a central repository of standard contracts that can be used by negotiating partners in the process of contract building. [Hue98] advocates a Trading Partner Agreement in which business partners describe a new business process. The definition can be exchanged by means of EDIFACT meta messages. However, this scenario makes not clear yet how the definitions are managed. Moreover, it requires that the message formats are adapted each time a semantic change is made, such as the addition of a new action. In our approach, the message format can be kept unchanged.

A somewhat older approach that bears similarity to XLBC is the Basic Semantic Repository (BSR) [ISO] and the Business System Interoperation (BSI) project at the University of Melbourne [ICA]. This project was not based on XML, but also aimed at standardization of business terms for EDI in the form of repositories. The BSR was set up with multilinguality support. The semantics that it did provide (at least in the first prototype of which information is available) was limited; the main contribution was a structured way of describing EDI data elements. The idea behind BSI was that outgoing messages would be translated automatically to the standardized form at the sender's site and translated back to the in-house file of the receiver at the receiver's site. The translation in both cases is performed by an BSI server. Although we basically agree with this general idea, there are still many problems to be solved before this actually works. One is that a standard should be available that is sufficiently expressive - such as aimed at by XLBC and the multilingual thesaurus.

FIPA has taken the initiative of defining an Agent Communication Language [Fou]. The language is also based on Speech Act Theory and its semantics is specified in BDI logic. The FIPA language provides almost no support for conversation objects. The propositional content can be defined for different domains using a nested attribute/value

scheme. The general message format is similar to XLBC; the main differences are that XLBC content is defined using predications and that the terminology is defined in a thesaurus (although FIPA also intends to set up directories where domain ontologies can be defined and published).

7 Conclusions and Future Work

The standardization process — defining a communication language and its semantics — is a process that is usually done by standardization committees, but if the users have to do it themselves, the question arises how it should be supported. We distinguish five aspects of this support:

Representation support: How to represent the syntax and semantics?

Accessibility support: How to store the definitions and make them available?

Methodological support: How to arrive at a definition of redefinition?

Process support: How to manage the standardization process?

Implementation support: How to implement the language in the context of existing legacy systems?

Drawing on FLBC, we introduced the XLBC language that defines the structure of messages. The meaning of the lexicals has to come from somewhere else. For this purpose, we have developed a multilingual thesaurus and a XLBC component library. By means of these techniques, the system is able to provide communication partners with representation and accessibility support, as mentioned above. Particularly, the specific (semantic) representation of business message components (in the shared repository) may incrementally evolve in our architecture. This would not be possible when relying only on prescribed XML DTD for message exchange. An important concern is the separation of the semantics in the repository and the concrete syntax in XML, as well as the separation of pragmatics (business rules on the application).

The other support aspects are not worked out in this paper, but we can make a few remarks. Process support is needed especially in the case that there are more than two stakeholders involved, for example, a business group or virtual community. In that case, the process should start by identifying all relevant stakeholders and ensure that everyone who wants to be involved has the possibility to do so. It is important that the process is legitimate so that the results are acceptable to all stakeholders. In [dM99], a method is described in which virtual professional communities can arrive at acceptable specifications. This method can be used also for a definition process.

Implementation support is especially important for the coupling of the standardized language with the legacy systems of the parties involved. Typically, the communication language is not identical to the language spoken by these legacy systems. A translation or mapping is needed to transform one representation into the other. This translation software is one of the major components of current EDI systems. [Has00] discusses the role of standards in the construction and mapping of global data models for cooperative information systems with different individual data models. The traditional bottom-up approach is to start with the data models to be integrated and then trying to define super-classes of which the original classes are specializations. The study shows that this can

lead to very complex integrated models. A top-down approach starts with an available domain model, as the multilingual thesaurus may provide, and maps this to the situation at hand in the legacy systems. In the case of a message standard, a top-down approach could be followed if generic concepts, such as order, invoice but also product, buyer, seller, or transport medium are available. The top-down approach and the bottom-up approach can be combined in a so-called *yo-yo* approach. On the technical level, wrappers that provide unified interfaces are an established technique for accessing legacy systems [RS97].

Acknowledgments

XLBC and the associated Shared Repository are currently being developed and implemented in the context of the ESPRIT project MEMO (Mediating and Monitoring Electronic Commerce) with project code 26.895. The partners are: ABN-AMRO, Tilburg University (CentER Applied Research), RWTH Aachen, IHK Aachen, EKD Spain, Sarnet Spain and Origin Spain. For more information, we refer to <http://www.abnamro.com/memo>.

References

- [D⁺98] S. Decker et al. ONTOBROKER: Ontology based access to distributed and semi-structured information. In R. Meersman, Z. Tari, and S. Stevens, editors, *Proc. DS-8: Database Semantics - Semantic Issues in Multimedia Systems*. Kluwer Academic Publishers, 1998.
- [Dik89] S.C. Dik. *The theory of Functional Grammar*. Mouton-De Gruyter, 1989.
- [dM99] A. de Moor. *Empowering Communities: A Method for the Legitimate User-Driven Specification of Network Information Systems*. PhD thesis, Tilburg University, 1999.
- [Fou] Foundation for Intelligent Physical Agents. *FIPA*. (available from <http://www.fipa.org/>).
- [G⁺99] M. Gisler et al. Requirements on secure electronic contracts. Technical Report MCM-institute-1999-01, Univ. St. Gallen, 1999.
- [H⁺95] C.-F. Helgesson et al. Standards as institutions. problems with creating all-european standards for terminal equipment. In J. Groenewegen, C. Pitelis, and S.-E. Sjostrand, editors, *On Economic Institutions*, pages 162–184. Edward Edgar Publ., 1995.
- [Has00] W. Hasselbring. The role of standards for interoperating information systems. In K. Jakobs, editor, *Information Technology Standards and Standardization: A Global Perspective*, pages 116–130. Idea Group Publishing, 2000.
- [Hue98] C. Huemer. The growing need for meta messages in electronic data interchange. In *Proc. 6th ECIS*, Aix-en-Provence, France, 1998.
- [ICA] ICARIS. *Business System Interoperation (BSI) – a Generic Solution*. (available from <http://www.cs.mu.oz.au/research/icaris/bsi.html>).
- [ISO] ISO/UN-ECE. *The Basic Semantic Repository*. (available from <http://www.cs.mu.oz.au/research/icaris/bsr.html>).
- [Kim98] S. Kimbrough. On ESTheta Theory and the Logic of X12 Date/Time Qualifiers. In *Proc. HICSS'98*, 1998.
- [KL96] S.O. Kimbrough and R.M. Lee. On formal aspects of electronic commerce: examples of research issues and challenges. In *Proceedings HICSS'96*. IEEE Computer Society Press, 1996.

- [KM97] S.O. Kimbrough and S.A. Moore. On automated message processing in electronic commerce and work support systems: speech act theory and expressive felicity. *ACM Transactions on Information Systems*, 15(4):321–367, 1997.
- [Lee98] R. Lee. Towards open electronic contracting. *Electronic Markets*, 8(3), 1998.
- [McG98] S. McGrath. *XML by example: building e-commerce applications*. Prentice Hall, Upper Saddle River, NJ, 1998.
- [Mey97] B. Meyer. *Object-Oriented Software Construction*. Prentice Hall, Englewood Cliffs, NJ, second edition, 1997.
- [MG98] B. Meltzer and R. Glushko. XML and electronic commerce enabling the network economy. *SIGMOD Record*, 27(4):21–24, 1998.
- [Moo99] S.A. Moore. A foundation for flexible automated electronic communication. Working paper, University of Michigan Business School, 1999. (available from <http://www-personal.umich.edu/~samoore/research/flbc/>).
- [RS97] M.T. Roth and P.M. Schwarz. Don't scrap it, wrap it! a wrapper architecture for legacy data sources. In *Proc. 23rd International Conference on Very Large Data Bases (VLDB'97), August 1997, Athens, Greece*, pages 266–275. Morgan Kaufmann, 1997.
- [WVD95] H. Weigand, E. Verharen, and F. Dignum. Integrated semantics for information and communication systems. In R. Meersman and L. Mark, editors, *Proc. IFIP WG 2.6 Conf on Data Application Semantics*, pages 500–525. Chapman & Hall, 1995.
- [WvH99] H. Weigand and W.-J. vd Heuvel. Meta-patterns for electronic commerce. *International Journal on Electronic Commerce*, 3(2):45–66, 1999.