# Meta Messages in Electronic Data Interchange (EDI)

Christian Huemer
Institute of Applied Computer Science and Information Systems
University of Vienna
Liebiggasse 4/3-4
1010 Vienna, Austria
phone: +43-1-4277-38443
fax: +43-1-4277-38449
ch@ifs.univie.ac.at

A Min Tjoa
Institute of Software Technology
Vienna University of Technology Resselgasse 3/188
1040 Vienna, Austria
phone: +43-1-58801-18800
tjoa@ifs.tuwien.ac.at

## Abstract

The economic advantages of electronic data interchange (EDI) are widely recognized. Nevertheless, the number of organisations and companies employing EDI is relatively small compared to the total number of businesses worldwide. The huge difference is caused by the fact that current EDI standards include a lot of complexity and their integration into existing applications is too expensive. This is due to the fact that current EDI standard messages are based on data models intended to capture all data that may appear in any business document of the corresponding business transaction. Business partners have to specify within a trading partner agreement a subset of the standard message which reflects their actual need before they are able to run an EDI partnership. Therefor, EDI should move to the meta level. The basic idea is that the concept of EDI is used for an agreement on the subset. Therefore, a meta message that is able to capture all the semantics in a trading partner agreement is needed. We demonstrate feasibility and advantages of a meta message approach by the example of the EDIFACT (Electronic Data Interchange For Administration, Commerce and Transport) standard.

## INTRODUCTION

Electronic Data Interchange (EDI) is the application-to-application exchange of electronic business-related data based on a format which is understood by both (all) trading partners using an electronic transmission medium in order to carry out a business transaction [1][2][4].

The problem of integrating EDI into businesses is a result of the fixed structure approach used in current EDI standards. Since all of the well-known EDI standards - ANSI X.12, EDIFACT, ODETTE etc. - use a similar approach, we further concentrate on only one example standard, namely EDIFACT. Each EDIFACT message is based on a data model of a single business transaction. It is created by volunteers from a specific business domain working together in standardization bodies. These volunteers describe their domain knowledge in a data model written down in EDIFACT syntax [8]. As a result an EDIFACT message is a schema intended to capture all data that may appear in any business document of the corresponding business transaction.

However, there does not exist an information system being able to process all the semantics included in EDIFACT messages. Therefore, there is no guarantee that a message created by a sending application in a standard conform format will be automatically processable by a receiving application. Business partners willing to exchange data electronically in a structured format have first to agree on the actual data they want to interchange. The format of these data is mainly determined by the semantics the involved information systems are able to process. Thus, business partners have to set up a trading partner agreement specifying a proprietary subset of the standard message and a set of rules to apply on the structure of the subset. Such a functional agreement is usually reached by conventional communication methods resulting in a serious drawback. The business partners have to (manually) implement these specifications within their translation software used to map EDIFACT messages to in-house data structures and vice versa.

We propose to exchange trading partner agreements by the means of EDI itself. This approach shifts the EDIFACT standardization to the meta level, because a meta message is needed to transmit the structure of the trading partner agreement. The electronic trading partner agreement might serve as the basis for the mapping between EDI message and in-house data structures. Consequently the receiver of a meta message needs not to care about the whole complexity included in the whole standard message, but can concentrate on the facts needed in the implementation. This simplifies the implementation of an EDI interface to the applications.

The remainder of the paper is structured as follows: In Section 2 we introduce the structure of EDIFACT and list the problems encountered by this standardization approach. The requirements on a trading partner agreement are given in Section 3. Section 4 presents EDIFACT meta messages, namely the directory definition message and the EDI implementation guideline definition message. Additionally, we propose further concepts to be incorporated into a meta message. Section 5 describes the software needed for the meta message approach and shows a possible scenario of exchanging trading partner agreements. We conclude with a short summary.

# EDIFACT

## Structure of EDIFACT

Besides the syntax (rules controlling the structure of a message) EDIFACT is based on the following key concepts: *Messages*, *Segments*, *Data Elements* and *Codes*. Standardized Codes are used to for representation of business terms. Data Elements are the smallest indivisible pieces of data. Furthermore, EDIFACT uses the concept of composite data elements, which are sequences of simple data elements that all together describe one logical unit. Segments are or groups of related data elements. A message is a sequence of segments and segment groups representing a specific business transaction

The EDIFACT syntax has the following main characteristics [1]:

- hierarchical structuring
- implicit data element identification
- special character data separation
- flexible length data structures
- mandatory or conditional status of data elements and segments

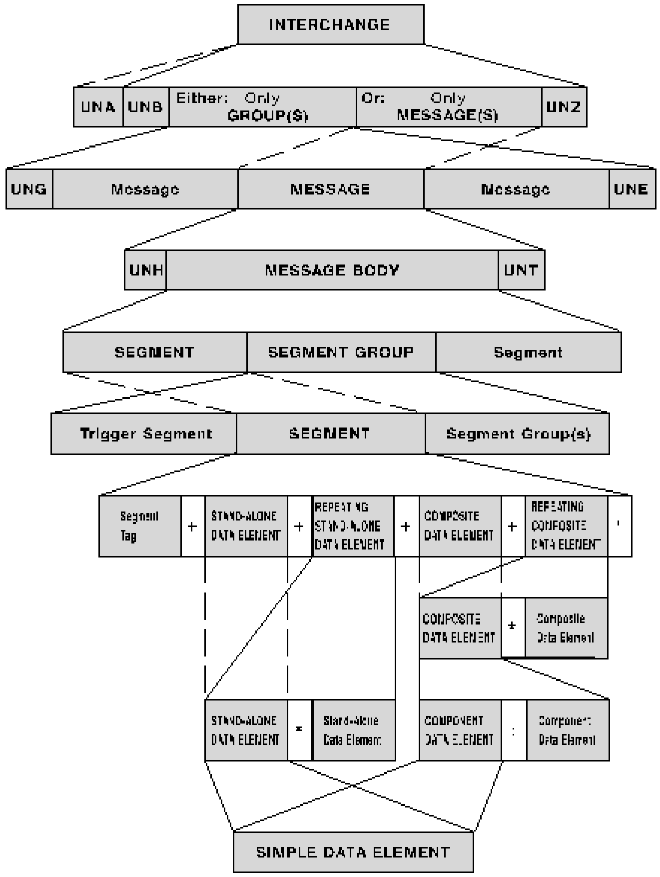


Figure 1. EDIFACT Structure

An interchange in EDIFACT format is structured as depicted in Figure 1 [11]. There may be more than one group or message within an interchange, each identified by its own header and terminated by its own trailer. A group is a conditional structure which is located between the interchange header and trailer and which comprises one or more messages.

A message comprises an ordered set of segments. Segments can be grouped. A segment group comprises an ordered set of segments. A segment is started and identified by a segment tag, which references a specific segment specification, and ends with the segment terminator ('). Furthermore, the message structure defines whether data segments and segment groups are mandatory or optional, and indicate how many times a particular segment or group can be repeated.

A segment comprises an ordered list of stand-alone data elements and/or composite data elements, which are separated by data element separators (+). The first data element is the segment tag, which references the segment definition. The segment definition indicates the data elements to be included in the segment, the sequence of the data elements and whether each data element is mandatory or optional.

A composite data element comprises an ordered list of two or more component data elements, which are separated by component data element separators (:). The composite data element definition specifies the component data elements to be included in the composite data element, the sequence of the component data elements and whether each component data element is mandatory or optional.

Both stand-alone and component data elements are simple data elements which contain a single data element value. Valid formats of data element values in terms of the number of characters and whether the characters are numeric or alphabetic are defined in the simple data element definition. Furthermore, the data element definition specifies the meaning of each data element. In addition to that some simple data elements are coded. Valid codes are defined in an associated code list definition

**Problem Areas in EDIFACT**

The Techniques and Methodology Working Group (TMWG) of the United Nations Centre for Facilitation of Procedures and Practices for Administration, Commerce and Transport (UN/CE-

FACT) have identified the problems resulting from the current standardization approach [10]. In addition to problems that are internal to the committees involved in the standardization TMWG has encountered the following problems that have an impact on the user of the EDIFACT standards:

Message development groups usually do not use a well-defined method to gather and structure the user requirements. In absence of a documentation on the user requirements, the only output of the standardization process is the EDIFACT message structure. Accordingly, the standards are not tied back to the entire business process. As a consequence, people not directly involved in development, may not understand the complexity includeded in the messages. Lack of comprehensive information on business contexts may lead in misunderstandings. One encounter difficulties in understanding the business needs and the solutions proposed.

The problem of missing semantics is handled by EDI branch organizations. They trim down the EDI standard messages to suite the requirements of business partners in a particular sector, in a particular part of the world. For the resulting subsets of EDI messages they specify the semantics in so-called implementation guidelines (MIGs) which govern the implementation of EDI in the specific sector of the specific local area. Since there are a number of these organizations, many different implementation guidelines for the same EDI message will coexist. The international and intersectorial EDIFACT standard is split into a number of national and branch specific subsets. Since, standard messages include optionality without explaining under which conditions options are to be used, the same information can be passed in different ways within a message. Therefore, different MIG specifications not only use different semantic features, but even worse they use different components of an EDI message to express the same semantics.

An implementation guideline also for a specific sector in a specific local area alone would be worthless if an information system of an involved business partner is not able to handle the induced semantics. Thus, the business partner have to adopt the standard message format or the message implementation guideline, respectively, to their actual business needs. The difficulty encountered by local users to understand the domains covered by the EDI messages may lead these users to redevelop new solutions from scratch. Therefore, they often interpret the message structure in their own way, which might be quite different to the standard's intention. As a consequence, trading partner agreements for different partnerships usually stay in conflict. This requires that an organization to implement different mapping tables between in-house format and EDI message format for different partnerships with in their translation software (which is usually based on the whole EDI standard message structure).It follows that business partners—although using the international and intersectorial EDIFACT standard—in fact, use a corresponding proprietary standard for each business relationship.

## TRADING PARTNER AGREEMENTS

To overcome the above mentioned problems long-term projects like Open-edi [7] or OO-edi [9] have been started. This projects present a top town approach (from the standard to the implementation) in order to incorporate the business requirements in the standardization process.

The definition of trading partner agreements by an EDI message is a bottom-up approach for short term solutions. Under the circumstances that a standard message does not include the necessary semantics, it is the goal to define the semantics at least for each partnership (bottom-up). This ensures on the one hand that business partner have a same understanding on the interchange data. On the other hand the EDI message structure is reduced to the subset required in the concrete transaction. Thus, business partners do not have to handle all the complexity included in the whole message structure. Instead they concentrate on the relevant items of the EDI message when specifying the mapping to the in-house data structures.

A trading partner agreement should specify a subset of an EDI message considering its implementation or application integration, respectively. Therefore, selecting those components of a standard structure that are considered in an implementation is not sufficient. The trading partner agreement must specify further constraints on the message structure, which have been investigated by the European Workshop for Open Systems (EWOS) [3].

The standard covers only two requirement designators to indicate that a component must be used or can be used in an interchange. A trading partner agreement should include much more constraints on the requirement designator. Mandatory components must be sent to ensure a correct interpretation of the interchange structure. Further components are required by the business context and have to be included in every interchange of the partnership. Some components are preferred by the receiver, but they are not absolutely required by the receiver. Vice versa, the sender needs to send some components which the receivers does not necessarily need to perform its business function. Furthermore, a requirement designator for signifying that a certain component is not used should be included. If a conditional component is not included in an interchange, a rule should specify whether a null-value or a default value should be applied.

Furthermore, if a component might be used multiple times, a mandatory requirement specification in the standard signifies just that the component must be used at least one time and can be used multiple times. In a trading partner agreement the specification of repetitions has to be bound to each requirement designator, to specify exactly how many times a component must be used, how many times it is preferred to be used, etc.

In the standard data element directory the representation of data elements is just defined by a format of numeric, alphabetic or alphanumeric and by a fixed or maximum length. For the purpose of application integration the trading partner agreement should provide means of expressing restrictions on data types, like those known in programming languages.

EDIFACT standard messages specify component usage just one step down the component hierarchy. This means that it is possible to cite which segments are used in a message. But it is not possible to designate the usage of data elements within a segment at a certain position in a message. Consequently, all segments of the same type are structured equally regardless of their position in a message. This might be sufficient for the standard specification, but is not adequate for the specification of trading partner agreements between business partners. The same problem applies for component data elements in composite data elements at a certain position in the message. Furthermore, the used code list and its assigned codes will also be static.

In a trading partner agreement a different usage of components at certain positions within the message is important. The status, value, repetition of a component in a certain position can be expressed as given. But this information can also be dependent on the status, value, repetition factor or data type of other components. For example, the usage of a segments depends whether the previous segment is used to describe the seller or the buyer (instances of that segment). Therefore, a meta message must ensure referencing mechanisms to other components.

# META MESSAGES

## Directory Definition Message (DIRDEF)

The directory definition message (DIRDEF) permits the transfer of the contents of a EDIFACT Directory set (Messages, Segments, Composite Data Elements, Elements and Code Sets) or part thereof [12]. An extract of the DIRDEF message structure is depicted in Figure 2. Note only the most meaningful segments are listed for reasons of place limitations.To describe the DIRDEF structure we add the term ‘meta’ to each DIRDEF component to distinguish it from a component which is defined in the instance of a DIRDEF message. Meta group 3 is used to define the message directory. Each instance of the meta group defines a message and within the meta subgroups 4 and 5 which segments and segment groups are used in this message. The following meta segment group 6 defines the segment directory. Each instance describes a segment and the data elements used therein. Next follows the meta group 7 defining the composite data element directory, which defines the sequence of component data elements within the composite data element. The meta group 8 is used to define the simple data elements of the data element directory. Finally, meta group 9 defines the code lists. Each entry assigns a code set to a data element. The values of each code list are defined within the meta subgroup 10.

| | | | |
|---|---|---|---|
| Segment Group 3 | | C | 9999 |
| MSG | Message type identification | M | 1 |
| Segment Group 4 | | C | 999 |
| SGU | Segment usage details | M | 1 |
| Segment Group 5 | | C | 1 |
| GRU | Segment group usage details | M | 1 |
| Segment Group 6 | | C | 9999 |
| SEG | Segment identification | M | 1 |
| ELU | Element usage details | C | 99 |
| Segment Group 7 | | C | 9999 |
| CMP | Composite data element identification | M | 1 |
| ELU | Element usage details | C | 99 |
| Segment Group 8 | | C | 9999 |
| ELM | Simple data elements details | M | 1 |
| Segment Group 9 | | C | 9999 |
| CDS | Code set identification | M | 1 |
| Segment Group 10 | | C | 9999 |
| CDV | Code value definition | M | 1 |

Figure 2. DIRDEF message

DIRDEF is best suited for its original purpose to transmit standard directory descriptions. It is also suited for exchanging subsets of the standard. This means, that it is possible to eliminate those messages, segments data elements, code sets and codes from a standard directory which are never used in an interchange between certain business partners. The structure of a message can be reduced to those component segments which are actually used when transmitting the corresponding message. The same applies to the composition of segments and composite data elements.

But DIRDEF cannot capture most of the requirements defined in the previous section. It is not possible to define position-specific component usage. Constraints on using components dependent on instances of other components cannot be specified. Furthermore, DIRDEF uses only mandatory and conditional  requirement designators. Only one combination of requirement designator and repetition factor can be indicated. The definition of data types is limited to that included in the standard.

**EDI implementation guideline definition message (IMPDEF)**

The EDI implementation guideline definition message (IMPDEF) permits the transfer of a document on guidelines for the implementation of one ore more EDI messages [3]. The basic structure of IMPDEF is derived from DIRDEF. Nevertheless, some changes in the structure have been made to overcome the shortcomings of DIRDEF to express implementation guidelines. In DIRDEF the meta groups specifying the message structure, the segment structure, the composite data element structure, the segment definition and the code list definition are at the same level. For this reason the component usage specification can be made only for the next level of the component hierarchy. To ensure position specific component definitions down the full EDIFACT hierarchy, the definition of a component will always be made in a meta subgroup of the component of the next higher level.

| | | | |
|---|---|---|---|
| Segment Group 3 | | C | 9999 |
| MSG | Message type identification | M | 1 |
| Segment Group 4 | | C | 999 |
| RFF | Reference | M | 1 |
| Segment Group 5 | | C | 999 |
| SGU | Segment usage details | M | 1 |
| Segment Group 6 | | C | 1 |
| GRU | Segment group usage details | M | 1 |
| Segment Group 7 | | C | 99 |
| REL | Relationship | M | 1 |
| Segment Group 8 | | C | 99 |
| RFF | Reference | M | 1 |
| Segment Group 9 | | C | 99 |
| ELU | Data element usage details | M | 1 |
| ELM | Simple data element details | C | 1 |
| Segment Group 10 | | C | 1 |
| DFL | Default value | M | 1 |
| Segment Group 11 | | C | 99999 |
| CDV | Code value definition | M | 1 |
| Segment Group 12 | | C | 99 |
| REL | Relationship | M | 1 |
| Segment Group 13 | | C | 99 |
| RFF | Reference | M | 1 |

Figure 3. IMPDEF message

Consequently, the outermost meta group 3 is that for message definition. Like in DIRDEF this meta groups includes meta subgroups 5 and 6 to identify which segments and segment groups are used in this message. But in contradiction to DIRDEF, meta group 9 which is used to describe the usage of data elements within a segment is a meta subgroup of meta group 5. Therefore, the usage of elements does not refer to the general segment definition, but to a segment in a specific position as defined by meta group 5. Furthermore, the meta segment ELM in meta group 9 is used to define any variation of a data element at a specific position compared to the general data element definition. In addition, meta group 10 provides a default value and meta group 11 a list of code values for each data element at a specific position in the message. Thus, meta groups 10 and 11 are subgroups of meta group 9 specifying the data element usage. By this concept IMPDEF fulfils the requirements of a trading partner agreement concerning different usage of components at certain positions within the message.

Furthermore, meta groups 7 and 12 are used to carry syntax rules that are specific to an implementation. Relationships between the various data elements in the a specific segment (meta group 7) and between the various simple data elements in a specific composite one (meta group 12) can be specified by this concept. Constraints to other components of an EDI message can be defined in the meta groups 4 (segment level), 8 (segment usage level) and 13 (composite usage level).

The IMPDEF message also provides a more reasonable choice for requirement designators. The data types of a data element can also be specified in more detail, but not comparable to that in programming languages. Therefore, IMPDEF is able to fulfil most of the requirements on a trading partner agreement.

### Possible Improvements for a Meta Message

Although IMPDEF is nearly perfect for the exchanging trading partner agreements a view adoptions can be proposed. IMPDEF works only under the precondition that the business partners are aware of the segment, composite data element, data element and code list directory. Therefore in a ‘meta’ environment a DIRDEF message must be sent to define these directories the IMPDEF message can work upon. This two step approach seems to be useful from the viewpoint that many business partners are already aware of the standard definitions and that the trading partner agreement always rely on a complete standard conform subset. Since the standardization process for a message might take several years, standardization often hinks behind the business requirements. To include business semantics which are not captured by the relevant message, this would require a non-standard conform adoption of the message. In an environment without meta messages standard conform subsets are an absolute requirement, because the standard message definitions (as incorporated in the translation software) provide the consistent link between the business partners. This can be undermined in a meta message approach, because the consistent link can be based on the electronic trading partner agreement. Accordingly, if a business partner generates a trading partner agreement which is not completely conform to the standard, he has to send two messages to its business partner. A DIRDEF message to specify the general component structure and an IMPDEF message to define the specifics of the trading partner agreement. The same will be true for the fact that the business partner is not aware of the standard definition. Therefore, both functions should be accomplished by an meta message. This could easily be realized by including the meta groups for segment, composite data element and data element directories specification of DIRDEF also into IMPDEF.

A further problem arises by the fact, that the IMPDEF message might include structures for specific component usage which are not defined in the general component usage. Therefore, we also prefer to include the general component definition in the meta message carrying the trading partner agreement. In this case explicit references between the specific and the general usage definition can be established. In this case the meta segments specifying the specific usage have to be redefined to support the referencing mechanism [5].

## IMPLEMENTING THE META MESSAGE APPROACH

### Software for the Meta Message Approach

The first important software tool is the *Subset Designer.* It must allow access at each level of the EDIFACT hierarchy (messages, segments, composite data elements and single data elements) and offers links between these levels. An edit function at each level has to ensure the development of subsets of standard directories and trading partner agreements. Although a stand-alone *Subset Designer* might be useful for the documentation of trading partner agreements, its full power will only be reached if the designed specifications can be accessed by the translation software. Two alternatives seem to be possible for this purpose. Firstly, a separate Subset Designer exports the trading partner agreement definition into a meta message format, which is imported into the translation software. At the moment commercially available translation software is usually not flexible enough to import directory specifications given in a meta message into the translation software database. But we expect them to be open at least for standardized meta messages in the foreseeable future. Secondly, the translation software itself is equipped with a Subset Designer module.

If the translation software is aware of the trading partner agreement, mapping specifications (mapping tables) can be established between in-house data structures and the self-designed trading partner agreement. Furthermore, the meta message created by the *Subset Designer* or by a comparable module of the translation software must be transmitted electronically to business partners. If a business partner uses EDI translation software able to accept the meta message as input format, he/she can create mapping specifications between his/her in-house data structure and the trading partner agreement designed by the initiating business partner. If the responding business partner is not willing to accept the trading partner agreement, he/she can edit the structure and send it back to the initiator. This process might proceed until consensus is reached.

Accordingly, EDIFACT messages based on an electronic trading partner agreement can be received via the communication interface, translated in compliance with the mapping tables into an interface file, which is finally imported with the converter software into the business application's database like any other EDIFACT message. Vice versa, data exported from the business application's database might be translated according to the mapping tables into an EDIFACT message based on an electronic trading partner agreement, and sent via the communication interface.

### Scenario For an Electronic Trading Partner Agreement

In this subsection we describe our proposed scenario for the definition and the following transmission based on a meta message for trading partner agreements [6]. The first step for the hub is to include the EDIFACT standard directories into the *Subset Designer*. Hopefully, this can be done by receiving a meta message from an EDIFACT reference database (1). Now the initiator specifies a subset on the basis of a standard version according to his requirements (2). The resulting subset specification will include only those components which will actually be exchanged in a business transaction with the responder. The subset definition can be exported from the *Subset Designer* into a meta message (3). In order to ensure that the responder has the same understanding on the subset definition the meta message must be sent to the responder. Thus, the resulting meta message is passed to the communication interface (4), which is responsible for the transmission to the responder (5). The responder receives the meta message via its communication interface (6). The responder imports the meta message into its Subset Designer (7) to verify the trading partner agreement. If the responder is not willing to accept the proposed trading partner agreement, he might adopt the proposed definition with its own Subset Designer. The refined trading partner agreement must be sent back to the originator (steps 8 - 12 correspond to steps 3 - 7 except of the direction of the information flow). This process (3 - 12) might last until consensus is reached, which is identified by sending back an identical trading partner agreement.

When there is an agreement on the subset, both business partners can import the meta message into their translation software in order to specify the mapping tables between their in-house data structures and the 'business' message(s) included in the trading partner agreement. Of course the import is only necessary, if the Subset Designer is not an integral part of the translation software. Afterwards the business partners are able to start exchanging EDI transactions on basis of the trading partner agreement. This exchange is similar to a conventional EDI transaction.

## SUMMARY

The meta message approach is designed to overcome major problems of current EDI standards. The basic idea is the reduction of standardization to only one type of standardized message, namely a meta message used to describe the other messages. The relevance of currently standardized business messages is devalued but not completely removed. Business messages will not longer serve as standards but more or less as recommendations. Consequently, business partners need not handle all the complexity included in current standard messages, but may concentrate on the actually needed structures.

The meta message approach is based on existing standards and is therefore fully compatible. An improvement of the current standardization approach instead of only focusing on new approaches is absolutely necessary, because many organizations already participating in EDI are not willing to move forward to a entirely new technique. Nevertheless the basic idea of the meta message approach might also be integrated into or merged with other initiatives in the EDI area. For example, pure XML/EDI [13] does not focus on the application integration problem. But the semantically complete meta data model could also be expressed in XML/EDI syntax, in order to serve for the same purpose as the EDIFACT meta message.

## REFERENCES


[1] Berge J.; *The EDIFACT Standards*; NCC Blackwell Limited; Oxford; 1991

[2] Emmelhainz M.A.; *Electronic Data Interchange: A Total Management Guide*; Van Nostrand Reinhold; New York; 1990

[3] EWOS; Assessment of IMPDEF (UN/EDIFACT Implementation Guidelines Definition message); EWOS/ETG 065; 1996

[4] Hill N.C., Ferguson D.M.; *Electronic Data Interchange: A Definition and Perspective*; EDI Forum: The Journal of Electric Data Interchange, Vol. 1, Issue 1; 1989; pp 5 - 12

[5] Huemer C.; Electronic Data Interchange (EDI): Standards, Shortcomings, Solutions; PhDThesis: University of Vienna;1997

[6] Huemer C.; The Growing Need for Meta Messages in Electronic Data Interchange; Proceedings of the 6th European Conference on Information Systems; Aix-en-Provence, France 1998

[7] ISO; The Open-edi Reference Model; IS 14662, ISO/IEC JTC/SC30; 1996

[8] Raman D.; *Cyber Assisted Business - EDI as the Backbone of Electronic Commerce*; EDI-TIE B.V.; Hoofddorp, Netherlands; 1996

[9] TMWG; Reference Guide - The Next Generation of UN/EDIFACT (Revision 12); CEFACT/TMWG/N010/Rev. 1; 1998

[10] TMWG; Review of the current EDI Standardisation process; CEFACT/TMWG/N021/Rev. 3; 1998

[11] United Nations, Economic and Social Council; *UN/EDIFACT Syntax Rules*; ISO 9735; International Organization for Standardization; Geneva; 1987

[12] United Nations, Economic and Social Council; *Directory Definition Message (TAG 52.315)*; Trade Data Interchange Protocols; Geneva; 1993

[13] XML/EDI Group; WWW-Page of the XML/EDI Group; http://www.xmledi.org