UN/CEFACT

United Nations Centre for Trade Facilitation and Electronic Business

UN/CEFACT
STANDARD BUSINESS DOCUMENT HEADER
Technical Specification

Version 1.3

2004-6-09

Table of Contents

1 STATUS OF THIS DOCUMENT

This Technical Specification is being developed in accordance with the UN/CEFACT/TRADE/22 Open Development Process for Technical Specifications. The Standard Business Document Header specification is a result of a work project of the UN/CEFACT Applied Technology Group (ATG). This specification will be supported by the two working groups within ATG, ATG1 (EDIFACT Syntax Structures) and ATG2 (XML Assembly Documents/Production Rules). The Standard Business Document Header (SBDH) [also known as Generic Header] Project Team has approved it for UN/CEFACT review.

This document contains information to guide in the interpretation or implementation of the specification.

This version: is Standard Business Document Header Technical Specification, Version 1.3 of 2004-06-09.

Previous versions: Standard Business Document Header Technical Specification, Draft Version 1.2 of 2004-03-10.

1.1 Disclaimer

The views and specification expressed in this document are those of the authors and are not necessarily those of their employers. The authors and their employers specifically disclaim responsibility for any problems arising from correct or incorrect implementation or use of this technical specification.

1.2 Contact Information

ATG Chair: Margaret Pemberton, Diskray Pty Ltd., diskray(at)w150.aone.net.au,
Project Leader: Melanie Kudela, Uniform Code Council, mkudela(at)uc-council.org
Document Editor: John Duker, Procter & Gamble, duker.jp(at)pg.com

2 INTRODUCTION

2.1 Summary

This specification defines the 'Standard Business Document Header' (SBDH) which will enable integration of documents between internal applications, enterprise applications, and business-to-business infrastructure by providing a consistent interface between applications. The standard header information will enable any application to determine the logical routing requirements and/or the logical processing requirements of a document based on information contained in the standard header. This can be accomplished with the use of key data elements including logical sender and recipient identifiers, a 'business document type', and other elements associated with a Standard Business Document (see Glossary) object.

Standard Business Documents (SBD) are used in supply chain, financial, and other processes to record and share data such as purchase orders, invoices, or item synchronizations. These business documents are typically created in one application and processed by one or more receiving applications, either within a single organization or an external organization (Trading Partner). A number of different proprietary approaches have been developed to route and process these documents.

The SBDH includes a set of standard elements necessary to determine the routing and processing of documents either as a header within, or linked with the document. The standard header can also optionally provide service and correlation information, at the business domain level, between trading partners. The standard header can provide the semantic information needed for the routing, processing and business domain context of documents, regardless of the data format of the document – XML or EDI or other format.

2.2 What is a Standard Business Document Header?

The SBDH contains information expressed in an XML format. The header provides information about the routing and processing of the Standard Business Document, whether the document is in an XML or EDI or other format. The SBDH is designed to be either an integral part of a Standard Business Document (e.g. either XML instance document or EDI interchange), or an object associated with the Standard Business Document itself.

2.3 How is it used in EDI and XML environments?

The UN/CEFACT Architecture supports both the EDI and XML communities. The Standard Business Document Header architecture will therefore support both EDI and XML e-business processes. Including a SBDH in (or with) each XML instance document, or with each EDI interchange reduces the effort needed to route and process documents and permits trading partner organizations to use different implementation approaches.

When implementing EDI, the provision of an additional standard header may not always be necessary, since EDI interchanges already contain functionality for some of the information in the SBDH. An example is the EDIFACT UNB interchange header, the UNH message header, and the 'function' part of the BGM. The SBDH specification will allow for this existing approach and provides an option to express additional functionality, such as service and correlation information.

Trust relationships among business applications and middleware applications providing services for those business applications are admittedly complex. For example, middleware communications software components may provide and enforce cryptographic properties such as data confidentiality and digital signatures, and are often implicitly delegated authorizing functions both for authentication (by signing or other means) or for access control (submission of business documents for further processing).

There are no additional security risks imposed by the use of a SBDH over those imposed by current middleware implicit delegation arrangements. The relationships between back end systems and middleware components are extremely diverse and heterogeneous. In such a situation, it is sufficient to allow the SBDH to work in two modes: no application level security and some application level security. In either case, the SBDH techniques can be made to work securely.

2.4 The Scope of the Standard Business Document Header

Many users, implementers and supporting industry standard bodies are in agreement on the need for a Standard Business Document Header. In their business-to-business activities, the SBDH will facilitate three business needs:

- The routing of business documents from one point to another. This refers not only to the transfer of information from an external originator to receiver, but also from one intermediate application to another. Information in the SBDH can help ensure that a document gets to the correct recipient.

- The simplified processing of documents. Processing refers to taking action on data, for example transforming it from one format into another. Information in the SBDH can reduce the effort required to determine the correct processing actions.
- Associating a data message with its originator is important from a business and legal perspective. It is especially important when using intermediaries for data transfer, as information from the transport protocol, may be lost after the initial transmission. Because information in the SBDH is retained, it can help ensure that a document's originator is correctly identified.

In addition to header functions provided by the SBDH for routing and/or processing of business documents, there is the need for a completely separate technical communications transport layer header which is defined by Business Collaboration Framework UN/CEFACT Modelling Methodology (BCF/UMM) as a message envelope. This technical communications layer header deals with communications protocols and physical addresses which are outside the scope of this technical specification. Transport specifications including EDIINT-AS2 and ebXML Message Service (ebMS) are among a number of possible transport options that address technical communications needs by defining a separate technical header. Transport layer headers are completely outside the scope and are a separate concern not addressed here (because they are in a different layer of the stack).

The SBDH is useful at the business application and middleware levels to provide for the routing and identifying of business documents. The information placed in the SBDH at the business payload level, will travel with the business information to many different workflows. In addition to the business payload information, it may be useful to the business application and middleware to know the original creator and intended receiver of the document. For the more complex creator and receiver business environments, there is a business need to use the SBDH for internal routing. The SBDH can enable this internal routing, eliminating the need to deeply parse and process an entire business document.

Within a legal context the terms 'Dispatch' and 'Reach' are commonly used to indicate when a data message leaves control of the originator and enters control of the recipient respectively. From a legal standpoint, these terms could replace the terms 'Send' and 'Receive' in some sections of this specification. These terms carry well defined semantics which are independent of any specific modelling methodology and technology. See UNCITRAL Model Law on Electronic Commerce < http://www.uncitral.org/english/texts/electcom/ml-ecomm.htm >.

2.4.1 What Makes the Standard Business Document Header Useful?

The main purpose of the Standard Business Document Header is to bridge the gap for standards, such as the UN/CEFACT EDI standard, that do not have the functionality of ebXML standards to perform a complete collaboration framework. It gives other technical frameworks and other standards an ability to simply use the payload in a collaborative exchange. These other standards and frameworks do not easily allow a user to accomplish this collaborative exchange without utilizing the attributes of the SBDH.

The BCF/UMM header of a business document provides information related to address, security and signatures as may be required by the associated Business Transaction' (please refer to BCF/UMM Business Transactions View (BTV)). Although according to BCF/UMM, some kind of document header is mandatory, the use of the Standard Business Document Header is not a replacement for the technical communications header nor is it mandatory. It is rather a useful business level header, which may be used optionally. As such we have identified three use case scenarios, which warrant the existence of the SBDH information as a separate header for business information. The three use cases are:

1. XML data in a non-ebXML environment; using middleware translation and transport
2. XML data in an ebXML environment, showing the Business Service Interface (BSI)
3. EDI payload in an ebXML environment with the SBDH (XML) header .

2.4.1.1 Legal Aspects of Electronic Data Exchange

A key use case for the SBDH is one where it may be used in a legal aspect to carry legal provisions and contract terms. UMM, ebXML and other collaboration frameworks provide only limited capabilities to associate the exchange of electronic information with legal provisions and contracts. A good example of this is an exchange of a "Price List" that may be accompanied by usage and confidentiality terms & conditions.

Associating messages with terms & conditions and legal documents is an important requirement and the SBDH may be useful in this role. The Unified Business Agreements and Contracts (UBAC) project is investigating the possibilities of adding an additional Business Scope in order to facilitate association between data messages and legal provisions. (See also section on Business Scopes in this document.) Likely candidates for this projected Agreement Scope are contract terms, signature reference and intent expression.

2.5 Business Opportunity and Benefits of the Standard Header

Although routing and processing instructions are not necessarily an integral part of a document, use of the Standard Business Document Header will allow organizations, with applications which are not yet fully process-centric, to take part in the process-centric approach and avoid wasted effort in developing customized routing and processing scenarios for each category of business data. Trading Partner organizations using different communication and integration approaches will find the SBDH a benefit since the business data payload will contain the information needed by the communication software to route and process this data in a standard way.

Operational decisions can be made by accessing the information in the SBDH and using that information to discover by which process context the business data should be driven. Routing and processing of Standard Business Documents (SBD) is facilitated regardless of whether all applications use a document driven, application programming interface (API), or agent approach. The use of logical parameters in the SBDH will minimize Trading Partner relationship management in both the Originating and Receiving organizations since the physical parameters can be derived from the values in the document.

2.6 Stakeholders and Audience

All organizations that manage infrastructure operations and business processes for various functional areas (e.g. ordering, invoicing, planning, or financial) which create, route and process Standard Business Documents can benefit from the use of the Standard Business Document Header.

2.7 Document Conventions

The keywords MUST, MUST NOT, REQUIRED, SHALL, SHALL NOT, SHOULD, SHOULD NOT, RECOMMENDED, MAY and OPTIONAL, when they appear in this document, are to be interpreted as described in [RFC2119] as quoted here:

- *MUST: This word, or the terms "REQUIRED" or "SHALL", means that the definition is an absolute requirement of the specification.*
- *MUST NOT: This phrase, or the phrase "SHALL NOT", means that the definition is an absolute prohibition of the specification.*
- *SHOULD: This word, or the adjective "RECOMMENDED", means that there may exist valid reasons in particular circumstances to ignore a particular item, but the full implications must be understood and carefully weighed before choosing a different course.*
- *SHOULD NOT: This phrase, or the phrase "NOT RECOMMENDED", means that there may exist valid reasons in particular circumstances when the particular behaviour is acceptable or even useful, but the full implications should be understood and the case carefully weighed before implementing any behaviour described with this label.*

- *MAY: This word, or the adjective "OPTIONAL", mean that an item is truly optional. One vendor may choose to include the item because a particular marketplace requires it or because the vendor feels that it enhances the product while another vendor may omit the same item. An implementation which does not include a particular option MUST be prepared to interoperate with another implementation which does include the option, though perhaps with reduced functionality. In the same vein an implementation which does include a particular option MUST be prepared to interoperate with another implementation which does not include the option (except, of course, for the feature the option provides).*

3 OBJECTIVES

3.1 Requirements

The objective of this specification is to define the attributes of a Standard Business Document Header. The SBDH will make it possible for originating and receiving applications to process Standard Business Documents in a way conformant to this specification. The objective of the SBDH specification is to facilitate the exchange of documents between applications in a standard way.

This specification will:

- Define SBDH semantics and associated values.
- Capture the details in a UN/CEFACT Modelling Methodology (UMM) logical information model for the SBDH.
- Assure the protocol independence of Message creation.
- Define standard, data driven processing and routing parameters in the SBDH.
- Define the role of the Business Information in the semantics and syntax transformation process.

The SBDH is a realization of the UMM meta model, with an example in XML syntax.

3.1.1 Constraints on the Standard Business Document Header

When using the Standard Business Document Header, the following constraints apply to the values provided in the header:

- Independence from proprietary routing rules.
- Location transparency in all except the ultimate partner facing functions
- Addressing transparency in all except the ultimate partner facing functions

- All proprietary semantics, syntax, and formats must be transformed into interoperable semantics and syntax.
- Protocol independence in all except the ultimate partner facing functions.

3.2 Principles of the Standard Business Document Header

The following table identifies the principles used to decide what kind of information is stored in the Standard Business Document Header, and what is not.

IN	OUT
1. Information known at the time of creation of the Standard Business Document (SBD) by the Business Data Creator Application (BDCA) or Translator/Parser. e.g., Standard Business Document (SBD) Type.	1. Information that can be known only at the time a message is sent. e.g., Transport Message Id.
2. Logical information that may be used to identify relevant physical information. e.g., partner name and role	2. Physical information useful for configuring the physical message transfer. e.g., channel information of partner such as protocol, port, etc. This physical information is to be extracted out of some profile, such as an OASIS CPP/A using the logical information provided.
3. Logical Information that may be used to route the document to specific external applications or services.	3. Physical Information identifying an external application such as its URL.
4. Logical Information that may be used to identify specific internal applications or services from where the document originated.	4. Physical Information identifying a specific internal application such as its IP address.

Table 1
In and Out Principles of the SBDH

3.3 Layered Processing Model

The layered processing model shows how the Standard Business Document Header may be populated, extracted and processed.

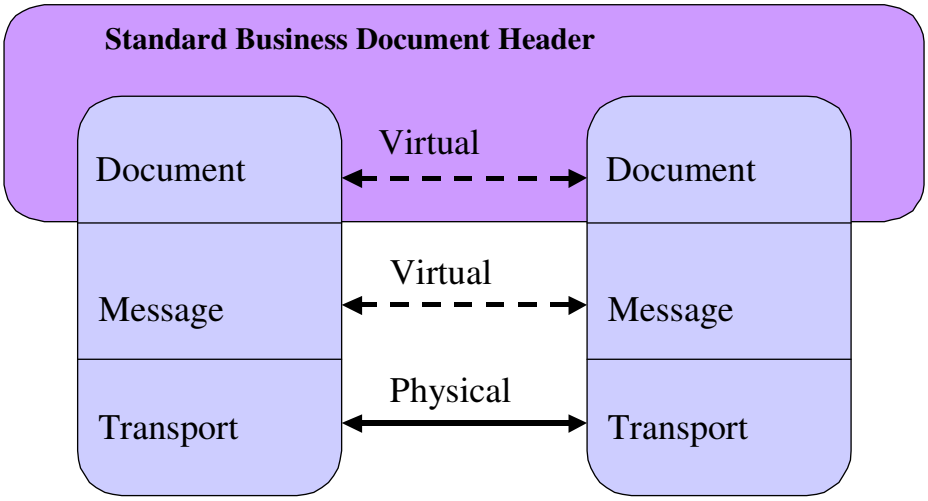


Figure 1

An interesting Standard Business Document Header element to consider is "Time Created" – each of the layers would have their own such element; for example, "Document CreationDateTime", "Message CreationDateTime", "Transport InitiationTime". The Document processor at the receiving end needs to worry or care about only the Document creation time, and not others. However, for auditing purposes, the other information may need to be logged, but such processing is outside the scope of SBDH.

3.4 Services

This section describes the use of the term "service" in the SBDH, Web Services, and UMM Business Collaboration Framework BTV and BSV terminology from UN/CEFACT. In the use of the SBDH, it is important to understand that the services defined by the service information object, are different from the services defined in ebXML and in web services. It is also important to understand that these terms are related and that the user must ensure that the services at each layer can map from one to the other.

EbXML Messaging Service (ebMS) and Web Services Description Language (WSDL) both use the term "service," but in slightly different ways. Here is a guide to navigating the terminological differences.

A WSDL file contains definitions and a wsdl:service is one element that can be defined. Within WSDL version 1.2, the decision has been made to have each service refer to only one wsdl:interface (formerly known as “portType”), and each wsdl:interface can aggregate one or more operations.

ebMS does not itself define “service,” and allows for bilaterally agreed upon values for both service and its action components. However, when ebMS is used with the UN/CEFACT Business Process Specification Schema (BPSS) and OASIS CPP/A, then the values for “service” and “action” derive from values in the BPSS instance. Basically, the service value indicates the entire package of Business Processes described in a BPSS instance document. Action values identify particular requests or responses within the Business Process.

So in both WSDL and ebMS, “service” is a kind of package of functionality, which can be defined by standards organizations or by members of a collaboration community. For ebMS, the package is of business processes, consisting of “actions”. For WSDL, the package is of elements, each called an “operation.” Operations bundle input, output, and fault definitions. Each input, output and fault at present gets associated with a “message” (and ultimately a schema defined type).

However, the ebMS action cannot be simply equated with an operation, because each business level action at present pertains to what is in WSDL either an input or an output. So, when an interface (formerly called “portType”) has both an input and an output operation, one interface name in WSDL can pertain to what will have two action names in ebMS, the action request and action response. Despite this one terminological asymmetry, ebMS actions and WSDL operations are very similar.

In the Standard Business Document Header, “service” is a kind of package of functionality, which is defined by standards organizations or by members of a collaboration community. It describes the business information in logical terms (it is similar to a requesting or responding business activity in BPSS or a group of operations in WSDL). However, it is not the same, because the SBDH provides a “syntax neutral” approach to facilitating the integration of the file systems of those users who need to preserve their current backend applications as they reformat their data into an XML format for transmitting it to their partners.

3.5 Routing

This section describes the use of the term routing at the technical messaging service level and at the Standard Business Document Header level, since the term is used differently in both of these aspects. At the business domain level,

which is the routing performed by the SBDH, routing describes the flow of a business document being transferred from one originating partner to another receiving partner.

At the lower level, the technical messaging service uses predefined transfer mechanisms such as HTTP to move the data across the Internet. At the network protocol level, individual packets are transferred from one router to another across the Internet network.

Because there are two kinds of routing - technical and business – it is useful to separate the headers into technical and business headers. The Standard Business Document Header handles business application level routing and specifying of business documents. The BCF/UMM which allows two business applications to have a virtual conversation, is another way of addressing this business need.

Standard Business Document Header routing does not refer to the lower levels of routing as they are transparent to the SBDH. However, the routing fields in the SBDH are capable of being mapped to the technical headers so that the document can be transmitted successfully to the partner. For instance, the routing information in the SBDH contains information for Sender and Receiver in a shared, well-known format, such as, a Global Location Number (GLN) or Dun & Bradstreet's Data Universal Numbering System (DUNS) number. This information can be mapped to different technical transport header fields. These technical headers use MIME in the case of AS2, or messaging service headers in the case of ebXML Message Service (ebMS).

3.6 Packaging

Since the Standard Business Document Header information is added to the business content that has been originally included in the business document, it is integral to the business document itself. It can be packaged as a part of the SBD, or for example as a separate MIME part.

There are varied reasons why the implementer would choose an integrated packaging approach or a non-integrated approach. The following arguments favour the integrated approach:

- If the SBDH is an integral part of the XML instance document, the document can be parsed at a high level and routing and processing decisions can easily be made.
- In older systems, if the SBDH is contained in a separate MIME body part, once the message is received by the Communications application, the linkage between the two MIME body parts can be lost and the routing/processing functionality becomes more complex.

494 The next arguments favour a non-integrated (e.g. a separate MIME parts)
495 approach:
496 • If the packaging is not integrated then the SBD can be easily encrypted
497 separately from the SBDH, and the information in the SBDH can be more
498 readily available to applications.
499 • Modern middleware can handle the linking between separate MIME parts.
500

501 ### 3.6.1 Access to the Standard Business Document Header Information
502 when the Payload is Encrypted

503
504 When using the integrated approach, once the message is inside one of the
505 partner's firewalls, the issue of application layer security and confidentiality may
506 arise under certain, special cases. This added concern over security and
507 confidentiality may be an issue on the entire Standard Business Document
508 Header and payload block or on some of the tags in the SBDH or payload.
509 Specifically identifiers or keys or financial information are examples that may
510 require additional security and confidentiality. The requirement may be that only
511 certain authorized individuals have the permission to view the contents.
512
513 For instance, a security requirement may be that the middleware environment
514 administrators should not have visibility to the payload, which could contain
515 sensitive trading partner data. In this requirement, only the receiving application
516 would be able to decrypt the data, potentially long after the data transport
517 process has ended. Some protocols may require the payload to be encrypted by
518 the sender, prior to transport, and to remain encrypted once received. If the
519 SBDH was received encrypted along with the payload, that would prevent further
520 routing from occurring. In these situations, requiring strict security and
521 confidentiality within the firewalls, there are two recommendations.
522
523 The first is to utilize selective encryption. Selective encryption is an XML
524 encryption option, which is available using the XML Encryption specification.
525
526 When using the older protocols, such as PKCS7, it will be more difficult to use
527 selective encryption. An alternative recommendation is that the SBDH is either
528 not encrypted or decrypted upon receipt. In the case where the payload needs to
529 be encrypted, there are two alternatives to handle this:
530 a) The first alternative is to send the SBDH and the attached,
531 encrypted payload in the manifest block. Both objects are contained
532 in one MIME part in one message.
533 b) The second alternative is to send the encrypted payload as a
534 separate MIME part. This option allows multiple recipients to read
535 the SBDH, while ensuring that only select recipients may read the
536 sensitive contents in the payload.
537

The manifest attachment is also the recommended way of sending a non-XML document or file. For example, an EDI document, with an SBDH should be sent as a manifest attachment. In this case, the non-XML payload can be encrypted and sent as the attachment, allowing the SBDH to be transported and received not encrypted or to be decrypted without impact to the rest of the payload.

4 MULTIPLE PARTNER ENVIRONMENT

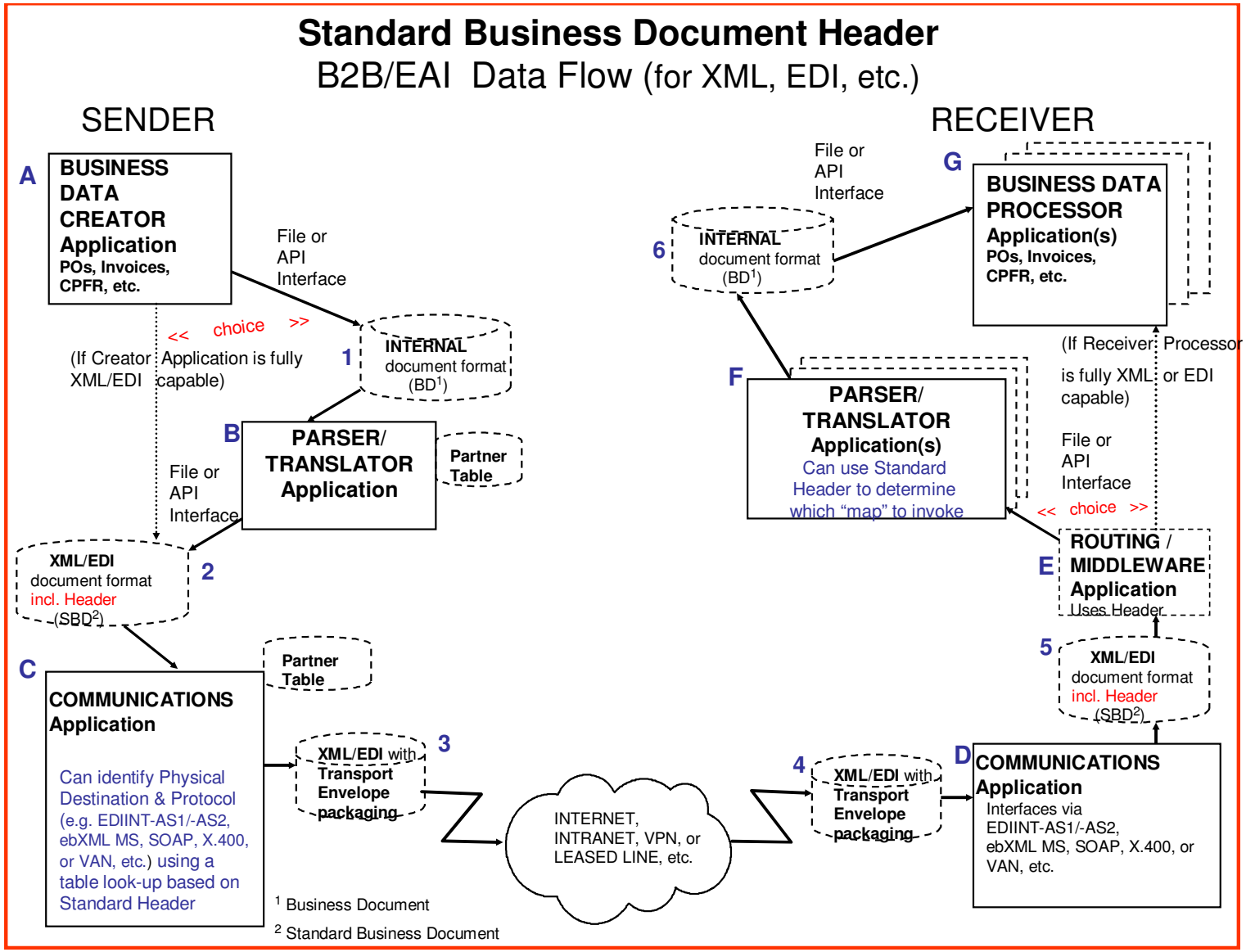
The Standard Business Document Header could be used in the scenarios where a SBD has to be sent to multiple partners or information related to a SBD needs to be collected from multiple partners. In that case the logical Receiver value could represent a ‘distribution list’, and the sending Communications application could send the SBD to multiple receivers.

The SBDH presupposes a point-to-point (sender-receiver) model. Effectively this infers that any hub-spoke or multi-party scenario will be broken down into collaborations between two partners. If it is extended to support an n-1 (hub-spokes) model, where n roles are interacting on a “business document” to do end-to-end processing, say order-to-cash, in a ‘multi-hop’ situation where the ‘middleman’ strictly performs a store and forward function without changing the SBD contents, the business document creating application should be insensitive to the presence of the middleman. If the SBD is altered by an intermediate role player, the logical Recipient should be that role player, not a subsequent recipient.

In a store and forward ‘multi-hop’ situation, legally relevant items such as the originator of a data message for example, may need to be retained with the identifying sender or receiver. The use of different types of technologies for example, the actions of an encryption service provider who unwraps and decrypts the message then re-encrypts it, may not preserve legally needed information that is needed when the payload arrives at the intended addressee. But by using the SBDH, the information is still preserved.

5 PROCESSING FLOW OVERVIEW

Figure 2



The figure and descriptions in this section are for illustrative purposes only, and are not normative. The various components depicted in Figure 2 are as follows:

Applications A – G:

Represent various applications in a data flow which move Business Documents (BD) from a Sender's back office application which creates data, to a Receiver's back office application which processes data.

Data Stores 1–6:

Represent various data storage locations indicating the format of data after it has been processed by one of the applications.

Application A:

Represents a '**Business Data Creator**' application (e.g. a legacy or ERP application) which creates business transactions for functional processes such as ordering, invoicing, planning, etc. either in:

a) Internal 'Business Document' (BD) format (shown in data store 1) e.g. a proprietary flat file which needs to be transformed into a SBD or,

b) If the creator application is fully XML or EDI capable, directly creates transactions in SBD format, including the standard header (shown in data store 2), and therefore bypasses Application B.

Data Store 1:

Represents one internally formatted BD which may contain one or more individual transactions of a single (or multiple closely related) business document type(s) such as purchase order, INVOIC/TAXCON, or shipment request, etc.

Application B:

Represents a '**Parser/Translator**' application that transforms a Business Document from its internal private format to an external Standard Business Document (SBD) format [shown in data store 2]. The SBD includes the Standard Business Document Header (SBDH). The SBDH provides logical information such as Sender, Receiver, Document Type, and optionally information such as business process identification.

Parser/Translator functions include optional parsing and transforming of Business Documents into standard semantics and syntax (i.e. a SBD). For example, a customer number is transformed into a Standard Partner Number, an internal stock keeping unit code is transformed into a Product Identification Number, and the structure is transformed from a proprietary flat file format into a standard format.

The transformation steps are optional. Not all Business Documents are created with proprietary semantics and syntax. Business Documents that are created in standard semantics or syntax will require fewer or no transformation steps.

Data Store 2:

Represents one externally formatted SBD, e.g. one XML instance document or one EDI interchange which includes the Standard Business Document Header.

Application C:

Represents a **Communications Application** that transmits the SBD from the Sender to the Receiver. The Communications Application can use logical information in the Standard Header to:

a) Determine the actual physical destination (i.e. where to route the SBD so that it gets to the Receiver, and

b) Determine the appropriate transport protocol, (e.g. ebXML MS, EDIINT-AS1/-AS2, SOAP, X.400, or a proprietary VAN protocol), managing the associated message creation, and protocol-specific envelope packaging.

Independence of transport protocol is provided by the syntax and protocol neutral Standard Business Document Header. Mapping of the SBDH logical values to the physical location and addressing parameters is handled by the Communications Application.

A Communication objective for the SBDH is to eliminate different proprietary approaches for determining transport protocol and destination. Providing a standard process will minimize the administration of Trading Partner relationships in the Communications Application by defining logical parameters in the SBDH.

Data Store 3:

Represents one transport message (as it is sent from Sender to Receiver) which contains the SBD plus the protocol specific envelope packaging.

Data Store 4:

Represents the same transport message (as it is accepted by the Receiver from the Sender.)

Application D:

Represents a Communications Application that receives the transport message, removes the protocol specific envelope packaging, and retrieves the SBD. The Communications Application can use information in the SBDH to determine further processing requirements.

Data Store 5:

Represents one externally formatted SBD, including the SBDH.

Application E:

Represents an optional routing and/or middleware application that uses the SBDH to determine which of several potential translator/parsers or back end applications to invoke or where to route the SBD. The application could also use the SBDH to determine Business Scope information such as Service Information and Correlation Information.

Application F:

Represents a ‘Parser/Translator’ application that transforms data from the external SBD format into a proprietary internal format. The ‘Parser/Translator’ can use

information in the SBDH to determine how to transform the SBD (i.e. which 'map' to invoke).

Data Store 6:

Represents one internally formatted 'Business Document' (BD) which may contain one or more individual transactions of a single (or multiple closely related) business document types(s) such as purchase order, INVOIC/TAXCON, or shipment request, etc.

Application G:

Represents a 'Business Data Processor' application (e.g. a legacy or ERP application) that receives data either in a Business Document, XML, or EDI format and processes business transactions.

6 Use Case Analysis

The Standard Business Document Header is compliant to and defined by using modelling elements of the UMM-Metamodel. The UMM is part of the Business Collaboration Framework (BCF). Figure 3, below, describes the scenario that the SBDH solution addresses. Basically, two partners engage in a UMM compliant business transaction that mandates the mutual exchange of one or more business messages. These messages, in turn, must be processed for relevant business data.

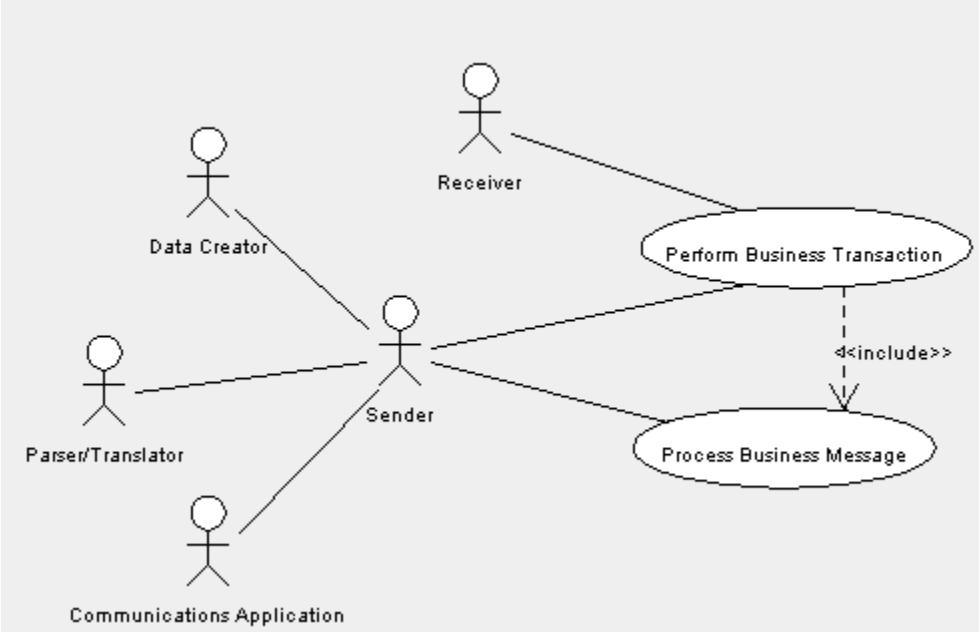


Figure 3

The use case diagram in Figure 3 illustrates the case where the Sender processes business messages, but note the receiver could follow the same

process being outlined. The remainder of this technical specification document will focus on the analysis of the Sender's domain (composed of three services: a Business Data Creator service, a Parser/Translator service and a transport or Communications Service); and then on the analysis of the Receiver's domain (composed of three services: a Communications Service, a Parser/Translator service and a Business Data Processor application).

6.1 Business Services

The specific services addressed by the UN/CEFACT ATG SBDH Data workflow are shown in Figure 4 below. To summarize, a Business Data Creator Service will create a Business Document, a Parser/Translator service will transform the Business Document into a SBD format, and a Communications Service will send the SBD to the Receiver.

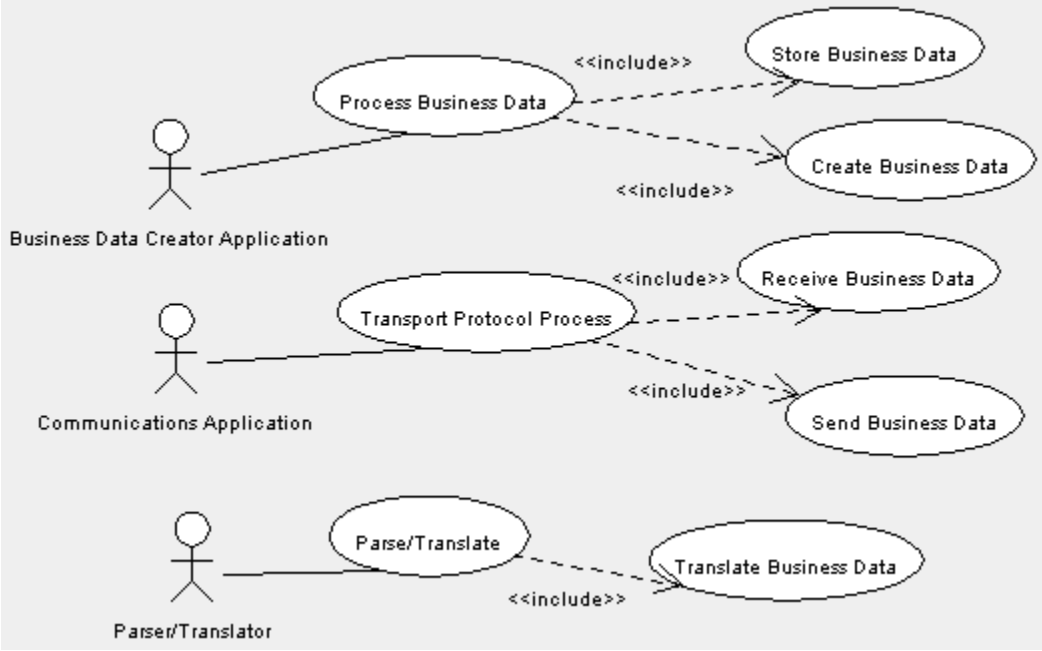


Figure 4

6.2 Description

Business Documents and their matching header data are created from data residing in the private space of the sender. Therefore, the BDs may be created using private semantics and syntax to describe and format the business data. The BDs can be used for purposes such as creating a purchase order, or an invoice, or some other purpose.

BDs can be created using:

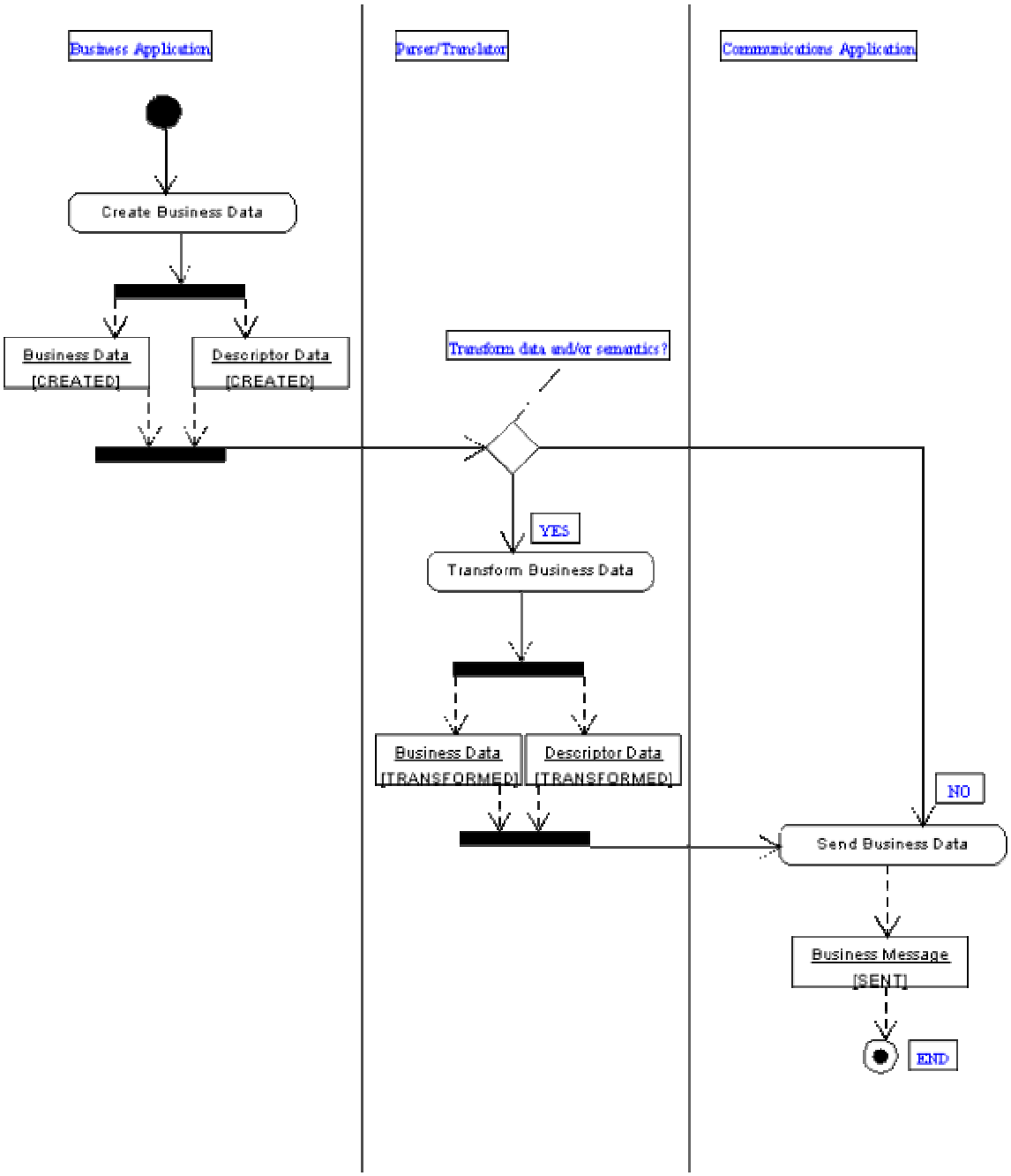
- legacy semantics
- legacy syntax
- standard semantics
- standard syntax, or
- some combination of the above.

The BD values will be derived from key semantics. The key semantic values must possess the intelligence required to:

- Ultimately derive the information for routing and processing the SBD.
- Map the BD logical values to the physical location and addressing parameters required by the Communications Services.
- Identify the appropriate Parser/Translator for this Business Document. Several parser/translators may exist depending upon the semantic and syntactical requirements of the BD. "Data-dependent routing" intelligence must be contained in the key values.

6.3 Workflow Analysis

There are two basic workflows for the ATG SBDH solution, each addressing a different, but complimentary, implicit UMM business function: originating and receiving business data. Figure 5, below, illustrates the prescribed ATG SBDH workflow for exchanging business data.



720

721 **Figure 5**

722 First, a Business Document and its matching header are created from
723 information residing in the private space of the sender (for example, one or more
724 internal business services). This data might be compliant (semantically and
725 syntactically) to some standard; otherwise it must undergo a data transformation
726 process. Note that the data and its corresponding header may initially contain
727 the information elements and semantics mandated by the ATG SBDH solution;
728 otherwise the data transformation service will ensure that such elements are

created. Finally, a communications service constructs a business message using the SBD with its SBDH. This message is sent to a peer through a predefined transport protocol.

The other workflow delineated by the ATG SBDH solution is shown in Figure 6 and illustrates the process of receiving a business message.

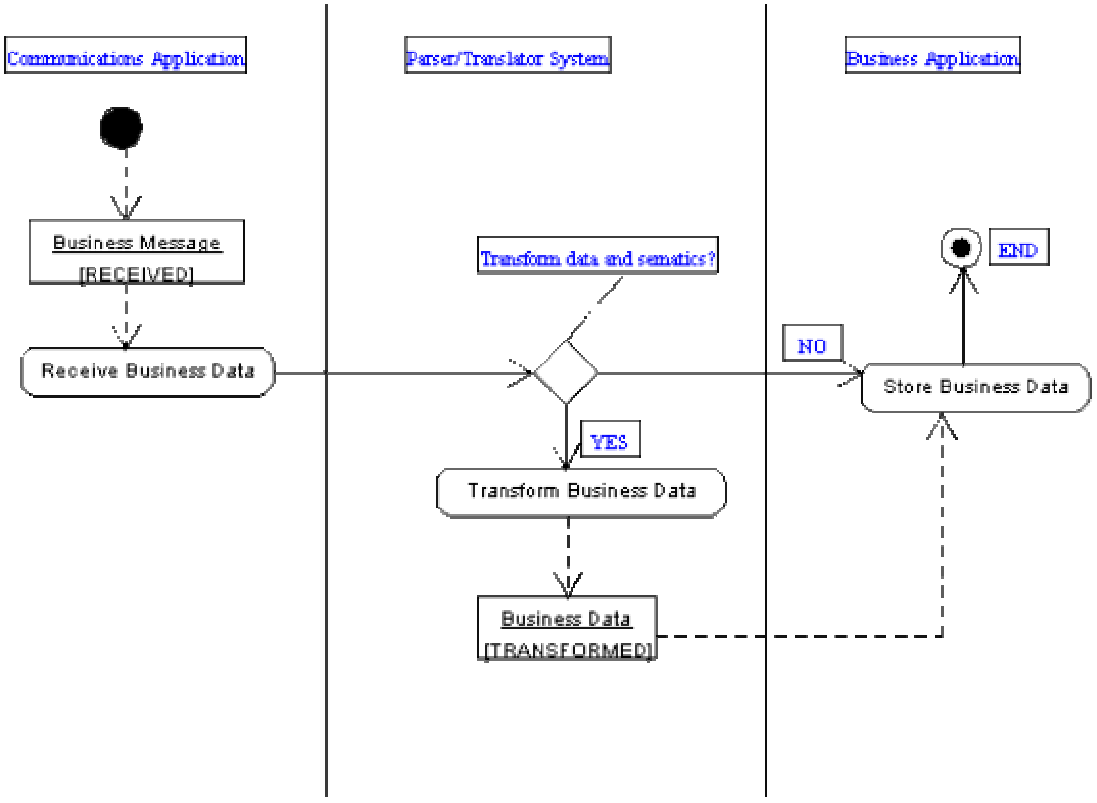


Figure 6

It is assumed that the message received by the Communications Service contains the key data elements and semantics mandated by the ATG SBDH solution. Key elements associated with information routing are then identified. The message may be sent to a parser/translator service or directly to a Business Data Processor service for processing and storage. If data transformation occurs, certain ATG SBDH elements will facilitate the process.

7 HIGH LEVEL SCENARIO

Assumption: In order to facilitate the exchange of business information in an electronic commerce environment, the specification addresses all the data flow in the message creation and processing:

1. the creation of the content
2. the transformation of the content into standard form
3. the packaging of the content into a message
4. the transfer of the message

5. the receipt of the message
6. the processing of the message
7. the storing of the message.

The high level scenario:

1. A BD is transformed and standardized into a SBD, e.g. standard EDI or XML with standard semantics. Logical SBDH elements are populated with standard semantic values.
2. The SBDH values are used to look up Message Envelope values to send the SBD using the appropriate transport protocol.
3. The SBD is received by receiver.
4. The SBD is transformed from standard EDI or XML and standard semantics to a proprietary BD format. Standard semantic values in the SBD are populated with Logical BD proprietary values.
5. A Response is sent by the receiver to indicate receipt of a SBD or a rejection indicates an exception has occurred with the sent SBD or the SBD has been rejected by the receiver. Response must indicate acceptance or rejection of the SBD.

8 PATTERNS

The UMM contains a series of message exchange patterns that rely on the concepts of Services and Agents, where a UMM Service exchanges a SBD, via messages, to another peer Service on behalf of an Agent.

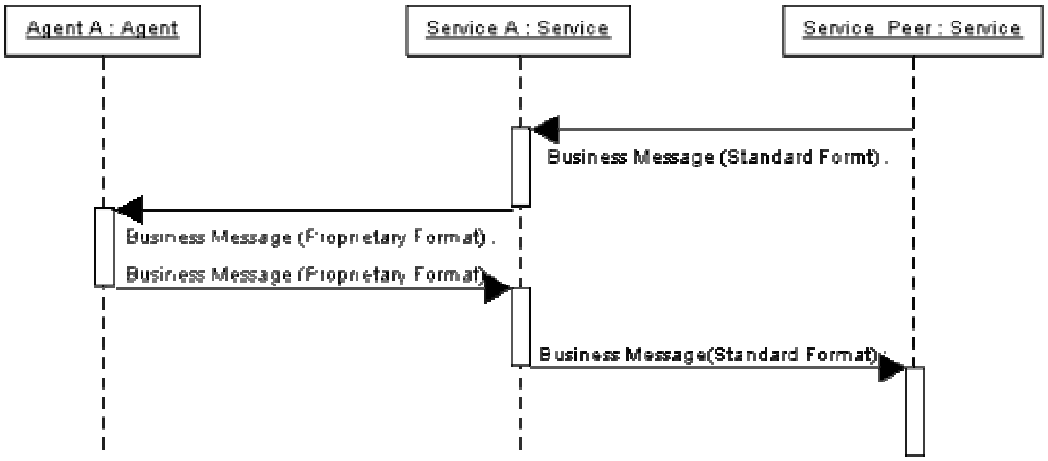


Figure 7

In the scenario depicted in Figure 7 the Services exchange business messages which comply with some standard. A secondary role of a Service can be to communicate the SBD contained within standard business messages to a corresponding Agent in some proprietary manner. In an e-Business enterprise an Agent could represent some legacy business application while a Service could be an interface to that legacy application that communicates to other enterprises in some standard fashion. The SBDH may be used to place a business document in the proper context for the UMM/Business Collaboration Framework service layer and transaction layer.

The ATG SBDH constructs a possible solution for a scenario that represents the UMM Service/Agent interaction patterns. It defines a generic workflow for the internal communication process between Service and Agent.

9 Business Scope

The business environment, circumstances, or scenario, in which trading partners conduct business is described by a set of domain context identifiers. This specification captures the information in the Business Scope block. The Business Scope specification being developed by the UN/CEFACT Techniques and Methodologies Group (TMG) Unified Business Agreements and Contracts (UBAC) team. Business process information is one of the characterizations of scope about messages exchanged in a business collaboration. However, there are other relevant characterizations of scopes and contexts as well. For example, it is relevant to know which business domain the collaboration of executing messages is associated with. Scope constraints clearly identify the business domain within which the transaction is executing, providing a basis for determining which rules are applicable to the transaction. The Business Scope* block in the SBDH provides the ability to associate a header and document with the proper business domain and thus constrain or extend its associated behaviour. (*See Appendix C for the theory behind the Business Scope.)

Scopes describe the environment within which transactions execute and allow a system to choose the correct environment. For example:

- Europe versus Asia,
- Direct-to-Consumer versus Replenishment, or
- Pre-Paid versus Credit.

Most systems, particularly legacy systems, have business domain rules coded into the application. By providing a Business Scope block in the SBDH, this information is forced up front so that all types of systems – no matter whether they are a Data Creator, a Parser/Translator or Communications Software – may select the rules correctly. The rules are selected depending on the scope received in the SBDH matched to the business domain selections within the implemented systems. When the system to be used to execute these transactions is being implemented, the implementer will write code against the

Business Scope and will have a very clear knowledge of which code needs to be triggered for execution of a specific domain rule.

The Business Scope in SBDH carries the information needed so that partners can identify and know which business rules to apply. There is a benefit to declare this information up front in the SBDH - partners can apply the rules even if the payload is encrypted. Knowing which of the domains the message is associated with allows business partners to make coordinated decisions for each context or business scope. For example, partners may agree that a transaction conducted with small businesses may require a credit card instead of a purchase order. The scope of that requirement constrains the business domain to be “small business”. Various scopes may select rules independently. For example, in addition to the “small business” scope, the partners may have an electronic collaboration mechanism in the form of an existing Trading Partner Agreement (TPA). The TPA identifies behaviour that is executed depending on the transaction exchange within the TPA domain. In the example, then, there are two scopes that are useful to identify the business domain of the collaboration:

- the small business domain and
- the domain of the pre-established TPA.

9.1 Technical Agreements and Business Agreements

Although partners may agree on technical agreements and pre-establish these agreements in a set-up step of the process, when it comes to business agreements, the partners’ behaviour during the collaboration runtime may vary depending on the business context being applied. This is the benefit of providing a Business Scope block in the SBDH. The required business behaviour for an exchange of messages is explicitly named in the Business Scope block. The business behaviour or relationship will vary in the instance of the transaction or collaboration. The same two partners, who submit replenishment purchase order collaborations, may exhibit similar technical behaviour but different business behaviour with each other when the purchase order is Direct-to-Consumer. The business behaviour is constrained by execution of a replenishment process or direct-to-consumer process. Which business process is executing determines the scope that is associated with the business behaviour. Being able to identify business behaviour with respect to active scope allows partners to clearly identify expected business behaviour in multiple scenarios

During an exchange of data messages, a number of specifications and legal provisions govern the exact interpretation and execution of ‘Dispatch’ and ‘Reach’. Specifications and agreements on business and technical levels often form a linked documentation set where various provisions are formulated in different resources. The SBDH and BusinessScope provide the capabilities to find the starting point for such dependent documents. However the current version of SBDH supports only identification of such resources (node) and not their relationships (edges). It was deemed that specification of relationships is an

area that needed further consideration and elaboration. In future versions of the SBDH relationships between scopes may be defined.

Pre-determined technical agreements describe technical protocols that partners will use when they conduct business electronically. In technical agreements, partners may decide upon using the OASIS CPP/A, a TPA, a RosettaNet PIP, or an AS2 connection. For example, a RosettaNet PIP and a CPP/A URI are used as two values in the filled out Business Scope block. This combination of PIP identification and CPA URI identifies the domain. This example is not exclusionary. The UN/CEFACT architecture describes a stack – a technical description at each layer of the stack. AS2 for example is at the bottom layer. Technical and business agreements can be declared going up the stack from AS2, following the UN/CEFACT architecture.

The CPP/A will have elements that govern both. It contains an SLA used by ebusiness software to monitor whether a response came back in time. The RosettaNet PIP provides a set of possible values, for example, for an order type, and the translation software would use that. The PIP will translate relationship attributes based upon “roles”. In a system, every user has rights based upon their role. Access management software has information on the role the user is playing in the current domain. This could be, for example, Read, Write, or No Access to data. The combination of values in the PIP and the CPP/A will provide information to all three services in the SBDH: the Data Creator, the Parser/Translator and the Communications Software.

It would be unrealistic to expect to renegotiate the technical agreement each time the business environment changes in some similar manner. The overhead of setting up numerous bindings and renegotiations to accommodate varied business perspectives would be prohibitive to the partners. Consider the case where a technical agreement is pre-arranged - in an existing TPA the business objective is to make deliveries from one partner to another partner's set of factories. In one particular exchange between the partners, the delivery must be made to one and only one specific factory. This specific business behaviour would be accommodated using the Business Scope and the existing TPA.

Behaviour is described by the business agreement, and then coded into the respective systems. By directly associating behaviour with scope, and then clearly identifying scope in the exchange, an agreed behaviour can be effectively triggered, monitored and enforced by the partners. They agree that when a particular value is detected in the business scope, the agreed upon business behaviour is exhibited. This behaviour is implemented in a variety of ways in the applications. The Business Scope class promotes this information up front in the partner facing part of the transaction. Most importantly, the Business Scope block makes the domain information available to both parties’ systems in the same way so that both of them can make use of the information. In this way, business considerations drive the transaction via the SBDH.

In EDI, a relevant example is the Order type field in the BEG line. The Order type is used to trigger different rules depending on whether the order type indicates Replenishment or Direct-to-Consumer, for example. In this case the Order type is constraining rules by inferring the transaction is within the scope of a process. This inference can become problematic because the Order type by itself does not fully define the process. There can in fact be several different processes required to make that Order type correct. Therefore, to know the right set of rules to use, additional information in the order is required. In this example, the order itself contains the information:

- The Order type plus
- Dates (and whether they are *n* weeks apart and)
- Whether the transaction is executing in one country, and so on.

In contrast, the Business scope is a clear and unambiguous holder to place that information, give it a name, and present it up front so that more applications than just the Business Creator applications can make use of it. In fact, all applications participating in the SBDH scenario – the Business Creator, Parser/Translators, and Communication Software Applications – can make use of the business scope information.

The Business Scope block as defined in the SBDH is general because the ability to identify domain associations changes over time. Rather than describing discrete values such a process, industry, etc. the SBDH Business Scope actually associates a message with its domain, execution environments and constraints. The association is made with multiple domain values such as:

- The process the message is executing within;
- The industry constraining processor;
- And the geopolitical policies.

For this reason, the Scope block within the Business Scope is repeatable.

9.2 Future Business Scopes

The Business Scope block is used to describe the complete business environment in which the SBDH and SBD will be processed. Standards bodies addressing business concerns will come up with enumerations of supply chain processes. The UN/CEFACT Technical Business Group (TBG) and Techniques and Methodologies Group (TMG) Unified Business Agreements and Contracts (UBAC) will be some of the entities that will define codes for the Business Scope. These will be used to fill out the SBDH Scope. The standards bodies will agree on how processes can differ. They will define the different business behaviours for each domain. The groups such as TBG will provide the content for the repeatable yet unique Scope within the Business Scope. The instance of Scope will be optional and used only if one or more such instances provide value to the partners within the current domain they are executing in.

Apart from the Business Scope defined in this version of the SBDH specification, there are other types of Scopes governing the exchange of words, messages, documents and business information in general. Agreements and contracts give legality to the information exchanges and form yet another type of Scope in another business environment. Standards bodies will identify the Scopes of the behaviour and their defined Scopes will impact implementation.

At the time of this specification, the defined extensions to Scopes are: Correlation and Service Information. In the future, additional scope extensions to the SBDH meta-model are probable. Business scopes such as "negotiation" may be added for example.

Another Scopes type and extension may be added to the Scope forming the concept of a Scope Profile. The Profile would contain various combinations of Scope Types and their extensions in an expression of a particular business domain within which an exchange of messages is occurring. This is described in the following figure.

Provisions are expressed in different resources that may logically and formally overlap each other. In order to achieve a clear and concise interpretation of the provisions, the dependencies between them must be exactly defined. Relationships such as superiority, replacement, modification are possible relationship types.

9.3 Scopes

The repeatable and general Scopes within the Business Scope blocks gives a structure and provides one mechanism to implement business scope knowledge in the code and allow the system to traverse all the relevant information. The Business Scope provides a method that supports a highly scripted discovery – agreements are easier to manage up front. That is the key function of the Business Scope block. When exchanging business information, documentation of only the lowest current level of scope is required. From this information all information exchanged can be deduced.

BusinessScope is a Scope reference mechanism and should not in general be used for Scope definitions. BusinessScope should be used to identify and reference the circumstances and scopes that govern a particular exchange of data messages. The referenced documents, resources, specifications etc. contain themselves complementary information relevant to the scope and information about relationships.

The BusinessScope is currently a list of governing Scopes. However such lists can handle flat structures as well as hierarchical structures (such UMM Business

Processes and ebXML Core Components), lattices and the more generic directed acyclic graph structures. This is because a Scope considers the Scope itself and not the Scope's relationship to other Scopes (i.e. reference to a node). Currently, Scopes are a linear list; however, there may be a relationship shown between the Scopes in the future. This will be accomplished by an extension to this version of the SBDH specification. The structure is described in the figure below.

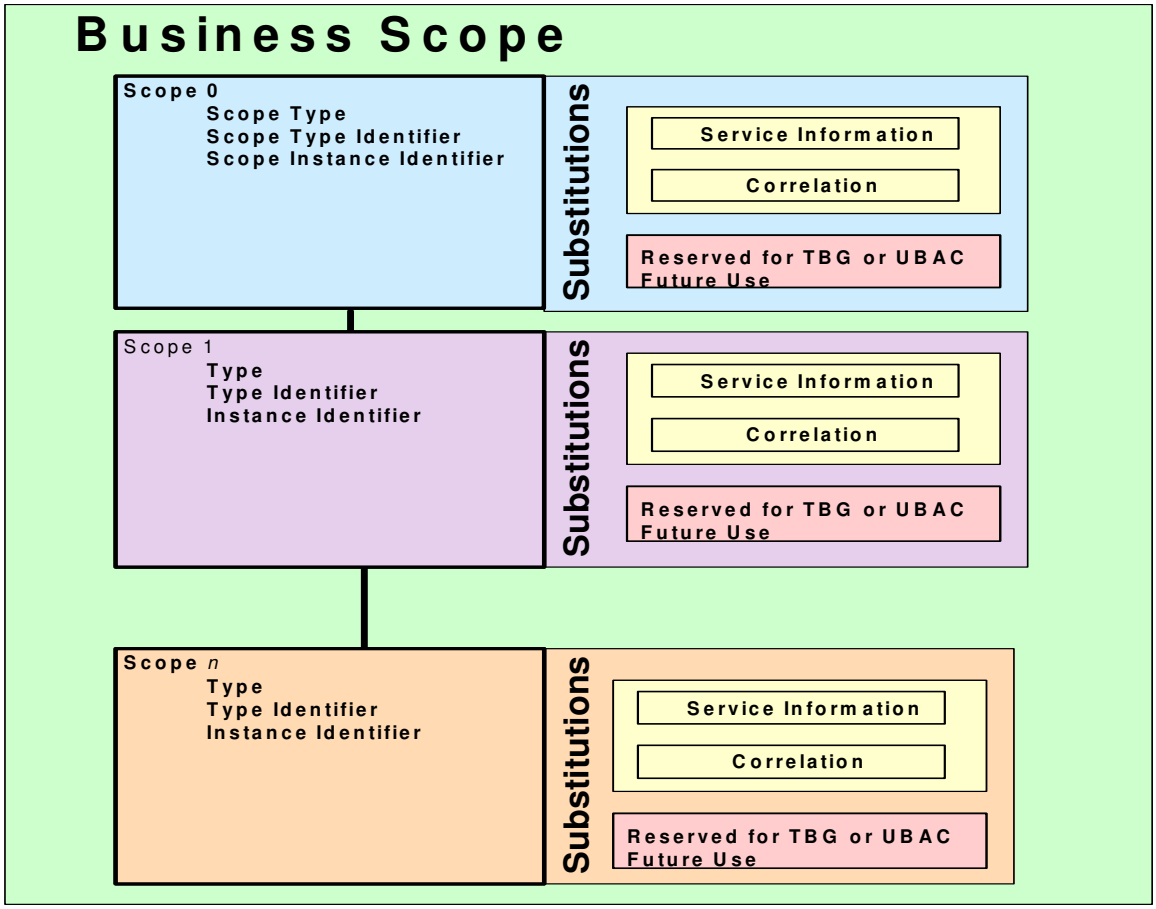


Figure 8

10. The figure below provides the UMM meta-model for the SBDH.

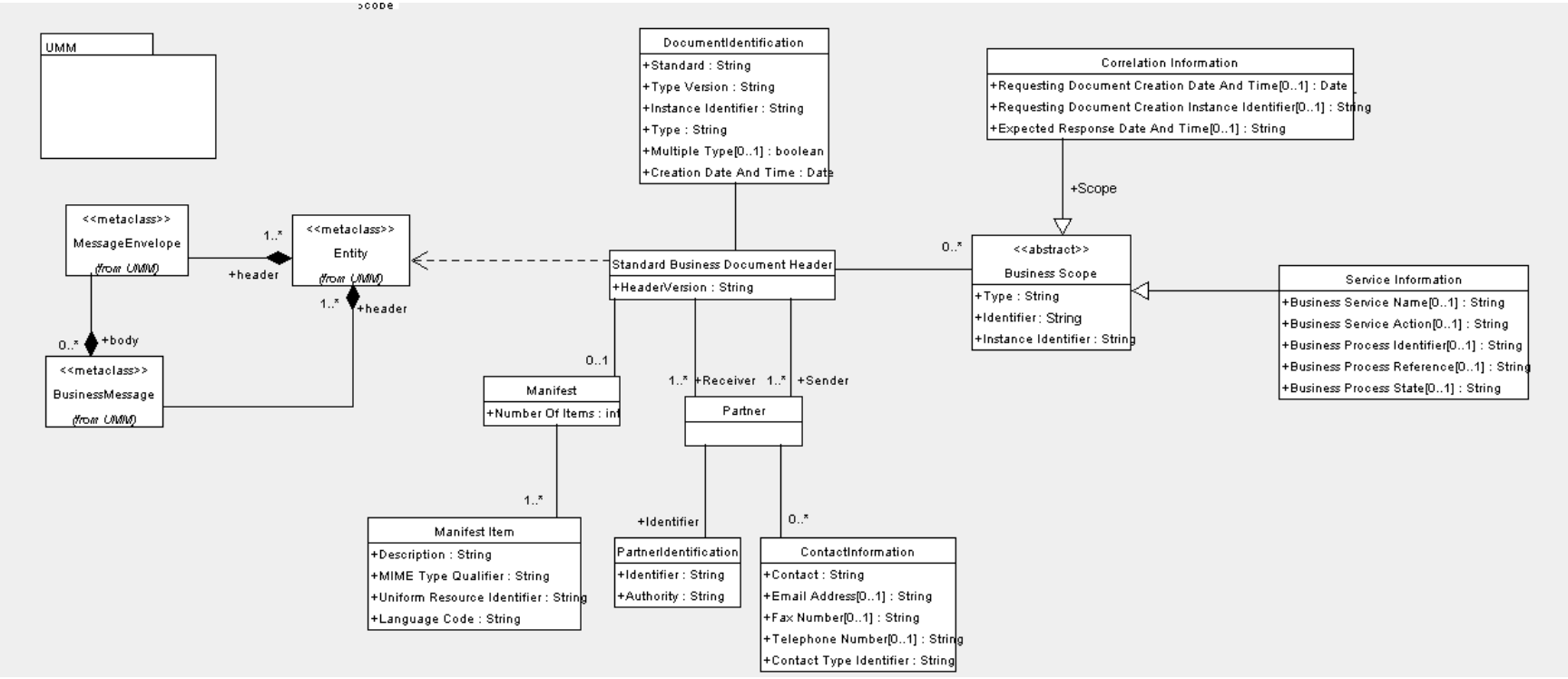


Figure 9

SBDH implementation of UMM Meta-Model

The next figure provides the UMM meta-class extension of the SBDH classes:

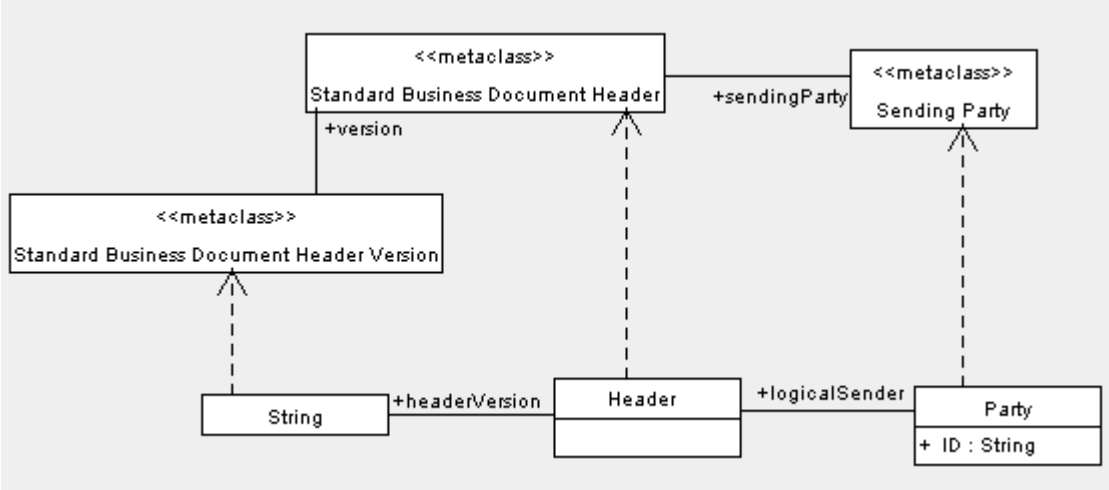


Figure 10

UMM Meta-Class Extension of the SBDH Classes

10 Standard Business Document Header Data Elements

The following Data Elements are components of the SBDH. The names here are the business terms, with (*proposed dictionary entry names)* in parenthesis. The proposed dictionary entry names are based on the Core Component Technical Specification Naming and Design Rules, version 2.01.
NOTE: The core components / business information entities may change after they have been processed through the UN/CEFACT harmonisation and approval process. In addition, the example schemas in Non-Normative Appendix A are for information only. These will be changed, and when published, will comply with the UN/CEFACT Naming and Design Rules and the UN/CEFACT UML to XML Transformation Rules, when available. The final version of this specification, after it has gone through the implementation verification process, will reflect these changes.

StandardBusinessDocument *(Standard Business Document Standard Business Document. Details*): The name of the XML tag required to wrap the SBDH and the SBD when the combined packaging into one instance file is used. This tag is only used under this packaging option, and in this case it becomes the root of the generated XML Instance Document. OPTIONAL, object.

StandardBusinessDocumentHeader *(Standard Business Document Standard Business Document Header. Details*): The name of the XML tag that contains the tags and contents of the SBDH. When the separate MIME part packaging

approach is used this tag becomes the root of the generated XML Instance Document. MANDATORY, object.

HeaderVersion (*Business Document Header. Version. Identifer*): Descriptor which contains version information for the SBDH (i.e. a number indicating the version of the SBDH). This Header Version information is not the same as the version information of the business document. REQUIRED, String. **NOTE**: The HeaderVersion value is currently "**1.0**". The HeaderVersion will be updated any time that the schema defining the HeaderVersion changes.

<**Sender Block**> (*Sender_ Party. Details*): Logical party representing the organization that has created the standard business document. This block is repeatable. If the Sender block is repeated then the first sender will be the primary sender and the second sender will be the secondary sender. The secondary sender will be used for internal routing purposes only to further identify the internal routing. The primary sender is REQUIRED, object. The secondary sender can repeat 1 to multiple times and is OPTIONAL, object.

1. **Identifier** (*Sender_ Party. Identification. Identifier*): Descriptor with information to identify this party; REQUIRED, String.
2. **Authority** (*Identification Scheme. Agency. Identifier*): Descriptor that qualifies the identifier used to identify the sending party; REQUIRED, String.
3. **ContactInformation** (*Sender_ Party. Contact. Contact*): Information about the contact for this document; Can repeat 0 to multiple times. OPTIONAL, object. Includes:
   a) **Contact** (*Contact. Name. Name*): contact for business, REQUIRED, String;
   b) **EmailAddress** (*Contact. EMail Address. Text*): email address of contact; OPTIONAL, String;
   c) **FaxNumber** (*Contact. Fax Number. Text*): of contact; OPTIONAL, String;
   d) **TelephoneNumber** (*Contact. Telephone Number. Text*): of contact; OPTIONAL, String;
   e) **ContactTypeIdentifier** (*Contact. Role Identification. Identifier*): role of the contact in this business process; OPTIONAL, String.

< **Receiver Block**> (*Receiver_ Party. Details*): Logical party representing the organization that receives the SBD. This block is repeatable. If the Receiver block is repeated than the first receiver will be the primary receiver and the second receiver will be the secondary receiver. The secondary receiver will be used for internal routing purposes only to further identify the internal routing. The primary sender is REQUIRED, object. The secondary sender can repeat 1 to multiple times and is OPTIONAL, object.

1. **Identifier** (*Receiver_ Party. Identification. Identifier*): Descriptor with information to identify this party; REQUIRED, String.
2. **Authority** (*Identification Scheme. Agency. Identifier*): Descriptor that qualifies the identifier used to identify the receiving party; REQUIRED, String. Includes:
3. **ContactInformation** (*Receiver_ Party. Contact. Contact*): Information about the contact for this document; OPTIONAL, object. Can repeat 0 to multiple times. Includes:
   a) **Contact** (*Contact. Name. Name*): contact for business, REQUIRED, String;
   b) **EmailAddress** (*Contact. EMail Address. Text*): email address of contact; OPTIONAL, String;
   c) **FaxNumber** (*Contact. Fax Number. Text*): of contact; OPTIONAL, String;
   d) **TelephoneNumber** (*Contact. Telephone Number. Text*): of contact; OPTIONAL, String;
   e) **ContactTypeIdentifier** (*Contact. Role Identification. Identifier*): role of the contact in this business process; OPTIONAL, String.

<**DocumentIdentification** block> (*Standard Business Document. Details*)
Characteristics containing identification about the document. REQUIRED, object.

1. **Standard** (*Standard Business Document. Standard Type. Code*): The originator of the type of the Business Data standard, e.g. SWIFT, OAG, EAN.UCC, EDIFACT, X12; references which Data Dictionary is being used. Used for the task of verifying that the grammar of a message is valid. Comment: This information may be provided in a URI if XML; probably not if EDI. REQUIRED, String.
2. **TypeVersion** (*Standard Business Document. Standard Type Version. Identifier*): Descriptor which contains versioning information or number of the standard that defines the document which is specified in the 'Type' data element, e.g. values could be '1.3' or 'D.96A', etc. . This is the version of the document itself and is different than the HeaderVersion. REQUIRED, string.
3. **InstanceIdentifier** (*Standard Business Document. Instance. Identifier*): Descriptor which contains reference information which uniquely identifies this instance of the SBD between the sender and the receiver. This identifier identifies this document as distinct from others. There is only one SBD instance per Standard Header. The Instance Identifier is usually automatically generated by the middleware. REQUIRED, string.
4. **Type** (*Standard Business Document. Type. Code*): A logical indicator representing the type of Business Data being sent or the named type of business data. This attribute identifies the type of document and not the instance of that document. The instance document or interchange can contain one or more business documents of a single document type or

closely related types. The industry standard body (as referenced in the 'Standard' element) is responsible for defining the Type value to be used in this field (e.g. 'order', 'catalogItemNotification', 'INVOIC', etc.). Comment: The type may be linked to the service. REQUIRED, string.

5. **MultipleType** (*Standard Business Document. Multiple Document Type. Indicator*): A flag to indicate that there is more than one type of Document in the instance. A "false" denotes that Type contains only one type of document; a "true" denotes that Type contains more than one type of document and that the name provided by the Standard authority identifies the multiple documents and not a single document. The instance document or interchange can contain one or more business documents of a single document type or multiple related document types. (E.g. Order, OrderSummary; or Invoice, TaxCon) Boolean, OPTIONAL.
6. **CreationDateAndTime** (*Standard Business Document. Creation. Date Time*): Descriptor which contains date and time of SBDH/document creation. In the SBDH the parser translator or service component assigns the SBD a Date and Time stamp. The creation date and time expressed here most likely will be different from the date and time stamped in the transport envelope. REQUIRED, dateTime.

<**Manifest** block> (*Manifest. Details*): Manifest that describes the related items or attachments (i.e., binary files), if any, being sent in this package. OPTIONAL, Object.

1. **NumberOfItems** (*Manifest. Item Count Number. Numeric*): The count of number of items associated with this package. Includes the base payload and any attachments. REQUIRED, Integer
2. **ManifestItem** (*Manifest. Item. Binary Object*): Provides information about the referenced item information; Repeatable if there is more than one item or attachments; REQUIRED, Object, Repeatable. Includes:
    a) **MimeTypeQualifierCode** (*Binary Object. Mime. Code*): Code describing whether the contents are XML or EDIFACT or X12, etc. syntax. Types are defined by IANA (see http://www.iana.org/assignments/media-types/) REQUIRED, String.
    b) **UniformResourceIdentifier** (*Binary Object. Uniform Resource. Identifier*): URI of the Manifest Item taken from its namespace; [For the useful guidance on how to reference external and internal message documents, the reader is referred to the RFC on Content Id URIs. This RFC 2392 (obsoletes 2111) can be found at the following location: http://www.faqs.org/rfcs/rfc2392.html]; REQUIRED, String.
    c) **Description** (*Binary Object. Description. Text*): Text Description of Item; OPTIONAL, String.
    d) **LanguageCode** (*Binary Object. Language. Identifier*): Language of Item in ISO 639; OPTIONAL, String.

<**BusinessScope** block> (*Business Scope. Details*): The business scope contains 1 to many [1..*] scopes. It is not mandatory to put all intermediary scopes in an SBDH. Only those scopes that the parties agree to are valid. The following examples are all valid: transaction; business process; collaboration. A Profile may be used to group well-formedness rules together. The business scope block consists of the Scope block. OPTIONAL, Object.

1. <**Scope** block> (*Business Scope. Scope*): Indicates the type of scope, the identifiers for the scope, other supporting information and the scope content itself. The importance of the Scope is that it allows the SBDH to operate under auspices of an agreement; that parties agree that they only include reference agreements (i.e. make a reference of SBDH and RosettaNet or OASIS CPP/A). Additional types of agreements are expected to be defined in the future. OPTIONAL, Object.
    a) **Type**: (*Business Scope. Scope Type. Code)*: Indicates the kind of scope; an attribute describing the Scope. Example entries include: UN/CEFACT Transaction, UMM:BusinessCollaboration, BusinessProcess, ebXML:BusinessService, BusinessServiceAction, BCF:AuthorizedRole, or Role Party. Could be used to indicate role reversal. MANDATORY, String.
    b) **InstanceIdentifier**: (*Business Scope. Scope Instance. Identifier*): A unique identifier that references the instance of the scope (e.g. process execution instance, document instance). For example, the Instance Identifier could be used to identify the specific instance of a Business Process. This identifier would be used to correlate all the way back to the business domain layer; it can be thought of as a session descriptor at the business domain application level. OPTIONAL, String.
    c) **Identifier**: (*Business Scope. Scope. Identifier)* An optional unique descriptor that identifies the "contract" or "agreement" that this instance relates to. It operates at the level of business domain, not at the transport or messaging level, by providing the information necessary and sufficient to configure the service at the other partner's end. Valid values for the Identifier may be in the form of a: URI, URN, ebXML CPAID, RosettaNet TPA, EDIFIEC or Partner Defined. Partners agree on how to describe the contract. A reference to the definition of legal compliance can be used as values in Identifier as well. It references the type of parent scope (e.g. process model, document specification). Several methods may be use to identify scopes: for example, Global identifiers (GUID) , relative identifiers (role name sequence number, local name). OPTIONAL, String.

The following objects are the first extensions of the Business Scope to be defined:

- the BusinessService block
- and the CorrelationInformation block.

In the future, the BusinessScope block will be extended with additional business scope and context extensions or substitutions, as these become defined by the business.

< **BusinessService** block> (*Business Service. Details*): Initiator's description of the service to be carried out on the SBD by receiver. The SBDH may be used to place a business document in the proper context for the UMM/Business Collaboration Framework (BCF) service layer and transaction layer. The SBDH does not model the BCF environment; it places the document within the context of a BCF environment which is modelled elsewhere in UN/CEFACT specifications. As such, a particular document will be in the context of one service transaction and one business transaction (i.e. in two different layers of the stack). OPTIONAL, Object.

1. **BusinessServiceName** (*Business Service. Name*): Initiator's description of service to be carried out on the SBD by receiver. Comment: A business service is a network component responding to business transaction requests initiated by other services. It has network identity as a business service. Business services monitor the execution of service collaborations. The service protocol implemented in the SBDH operates only in the document layer of the e-business network; it is not concerned with Transport or Message Layers. In the context of an ebXML business process model, a service is a set of related actions for an authorized role within a party. OPTIONAL, String.

2. **ServiceTransaction** (*Business Service. Service Transaction. Name*): BusinessServiceTransaction is a specific instruction to be executed by the 'BusinessServiceName' on the received Standard Business Document. The ServiceTransaction element identifies a process within a BusinessService that processes the SBD. BusinessServiceTransaction SHALL be unique within the Service in which it is defined. OPTIONAL, Object.
   (The following elements are an expression at a business level of what service an application wants and should not be construed as instructions to an infrastructure application.)
   a) **TypeOfServiceTransaction** *(BusinessService. ServiceTransaction. TypeOfServiceTransaction. Identifier)*: The value of the TypeOfServiceTransaction element is specified by UMM as: ‘Requesting Service Transaction’ or ‘Responding Service Transaction’. OPTIONAL, String.
   b) **IsNonRepudiationRequired** (*Business Service. Service Transaction. Is Non Repudiation Required. Indicator*): Non-repudiation of origin and content means that the originator must

digitally sign the business data and the recipient must store the business data (including the digital signature) in its original form for the duration mutually agreed to in a trading partner agreement. OPTIONAL, Boolean

c) **IsAuthenticationRequired** (*Business Service. Service Transaction. Is Authentication Required, Indicator*): If IsNonRepudiationRequired is true, this tag is superfluous. Otherwise, the tag indicates whether the identity of the sending role is verified. OPTIONAL, Boolean

d) **IsNonRepudiationOfReceiptRequired** (*Business Service. Service Transaction. Is Nonrepudiation Of Receipt Required. Indicator*): Indicates that both partners agree to mutually verify receipt of requested business data and that the receipt must be non-reputable. OPTIONAL, Boolean

e) **IsIntelligibleCheckRequired** (*Business Service. Service Transaction. Is Intelligible Check Required. Indicator*): Both partners agree that a responding partner role must check (e.g. via use of a document digest) that received data is not garbled (unreadable, unintelligible) and has integrity (i.e. has not been altered) before acknowledgment of proper receipt is returned to the requesting partner. OPTIONAL, Boolean

f) **IsApplicationErrorResponseRequested** (*Business Service. Service Transaction. Is Application Error Response Requested. Indicator*): Both partners agree that a responding partner's receiving business application must check for application level errors; and if any are detected, must respond with an Error Response Acknowledgment noting the errors detected. OPTIONAL, Boolean

g) **TimeToAcknowledgeReceipt** (*Business Service. Service Transaction. Time To Acknowledge Receipt*): Specifies the time period by which a Receipt Acknowledgment must be returned by the responding partner's receiving business application. The requesting and responding partners must jointly agree on the time period. It is measured from the time a business data request is sent by a requesting partner until the time verification of receipt is "properly received" by the requesting business partner. The Receipt Acknowledgment only indicates receipt of data by the business application; it does not indicate business acceptance of the contents of the message. If the TimeToAcknowledgeReceipt element is used, it indicates that a Receipt Acknowledgment is requested. OPTIONAL, TimeExpression

h) **TimeToAcknowledgeAcceptance** (*Business Service. Service Transaction.Time To Acknowledge Acceptance*): Specifies the time period that an Acceptance Acknowledgment (which indicates business acceptance of the contents of the document) must be returned by the responding role. It is measured from the time a

requesting partner sends business data until the time an acknowledgement of acceptance is "properly received" by the requesting partner. If the TimeToAcknowledgeAcceptance element is used, it indicates that an Acceptance Acknowledgment is requested. OPTIONAL, TimeExpression

i) **TimeToPerform** (*Business Service. Service Transaction.Time To Perform*): Specifies the time period by which this transaction must be completed (measured from the time the business data is "properly received"). OPTIONAL, TimeExpression

j) **Recurrence** (*Business Service. Service Transaction. Recurrence*): OPTIONAL, Unsigned Integer

<**CorrelationInformation** block> (Correlation. Details): A block of information used to correlate a requesting SBD to a responding SBD and to the contract in an executing choreography. A requesting document in the choreography could have: no response, a notification, or a response document. Therefore, the requesting and responding part of the choreography is not always one unit of activity. Using the correlation block, parties explicitly identify the document being responded to, rather than having only the content of the document to identify the requesting document. UN/CEFACT BPSS correlates information at the transaction level but not at the business domain level, which is the function of this block. This is valuable information for both parties' business data creator applications to correlate their document exchanges. The requesting document is often, but not necessarily, the very first document in the sequence. If the Requesting document is being sent, some of the information in this block is not necessary - the block attributes are OPTIONAL, Object. Includes:

1. **RequestingDocumentCreationDateTime** (*Correlation Requesting Document. Creation. Date Time*): Descriptor which contains date and time of the requesting SBDH and SBD, assigned to the requesting SBDH and SBD by the parser translator or service component. OPTIONAL, DateTime.
2. **RequestingDocumentInstanceIdentifier** (*Correlation Requesting Document. Identification. Identifier*): Identifier of requesting SBDH and SBD instance. OPTIONAL, String.
3. **ExpectedResponseDateTime** (*Correlation. Expected Response. Date Time*): Date and time when response is expected. This element could be populated in an initial message of a correlation sequence, and should be echoed back in a subsequent response. OPTIONAL, DateTime.

11 DETAILED USE CASE EXAMPLES

Note: These examples are subject to change by UN/CEFACT. Dictionary entry names for the core component / basic information entity names may change after they have been processed through the UN/CEFACT harmonisation / approval process.

11.1 Use case 1. XML data in a non-ebXML environment

Assumptions

- In this use case, the SBDH will be sent in a separate MIME Part from the rest of the payload. Therefore, the StandardBusinessDocument tag is not used in this example. The rest of the payload is not shown in this example.

- This use case requires the use of the optional Manifest object because there are two attachments to be sent.

- The middleware processing this use case does not require the information in the BusinessScope object; therefore, this information is not part of the payload.
- In this use case 2 sender blocks and 2 receiver blocks are shown. The first sender is the primary used for primary routing; the second sender is the secondary routing sender. There may be additional sender blocks and they would also be used for routing purposes. This same holds true for the receiver.

This use case shows the values that are known by the Business Data Creator in the first table. The second table shows the standard values after the original Data Creator values are transformed.

1) The Business Data Creator is the source of SBD creation and creates data in “Internal Business Document" format. The Business Data Creator application populates logical information only in the SBDH REQUIRED fields:

The following field values are populated by the Business Creator Application.

Table 2. Business Creator Application Business Terms and Values

Business Term			Example Value
Sender	Identifier		XYZ Retailer -12345
	Authority		XYZ Retailer
	ContactInformation	Contact	Corporate Headquarters
		EmailAddress	CorporateHeadquarters @XYZretailer.com
		FaxNumber	1-212-555-1212
		TelephoneNumber	1-212-555-2121
		ContactTypeIdentifier	Corporate Organization
Sender	Identifier		XYZ Retailer Purchasing Department
	Authority		XYZ Retailer
	ContactInformation	Contact	John Doe

		EmailAddress	John_Doe@purchasing.XYZretailer.com
		FaxNumber	1-212-555-1213
		TelephoneNumber	1-212-555-2122
		ContactTypeIdentifier	Buyer
Receiver	Identifier		WidgetsMarket
	Authority		Widgets
	ContactInformation	Contact	Mary Smith
		EmailAddress	Mary_Smith@widgets.com
		FaxNumber	1-312-555-1214
		TelephoneNumber	1-312-555-2125
		ContactTypeIdentifier	Seller
Receiver	Identifier		WidgetsSales-123
	Authority		Widgets
	ContactInformation	Contact	Jane Austin
		EmailAddress	Jane_Austin@widgets.com
		FaxNumber	1-312-555-1216
		TelephoneNumber	1-312-555-2127
		ContactTypeIdentifier	Assistant Seller
Document Identification	Standard		Trade Item Information Record
	TypeVersion		2.1.3.4
	Type		Trade Item Information Record
	CreationDateAndTime		Sept. 15, 2003 at 10:00:00
Manifest	NumberOfItems		2
	ManifestItem	MIMETypeQualifierCode	video/mpeg
		UniformResourceIdentifier	http://www.widgets.com//ProductImage
		Description	MPEG Video Image of Product
		LanguageCode	English

1401

1402 **Table 3. Parser/Translator Transformed Business Terms**

1403 **(see Sample 1 in Appendix B)**

Business Term			Transformed Example Value
StandardBusinessDocumentHeader			The root tag of the instance containing the SBDH information.
HeaderVersion			1.0
Sender	Identifier		6903148000007
	Authority		EAN.UCC
	ContactInformation	Contact	Corporate Headquarters *
		EmailAddress	CorporateHeadquarters@XY

			Zretailer.com *
		FaxNumber	1-212-555-1212 *
		TelephoneNumber	1-212-555-2121 *
		ContactTypeIdentifier	Corporate Headquarters *
Sender	Identifier		6903148000008
	Authority		EAN.UCC
	ContactInformation	Contact	John Doe *
		EmailAddress	John_Doe@purchasing.XYZretailer.com *
		FaxNumber	1-212-555-1213 *
		TelephoneNumber	1-212-555-2122 *
		ContactTypeIdentifier	Buyer *
Receiver	Identifier		2203148000007
	IdentifierAuthority	EAN.UCC	
	ContactInformation	Contact	Mary Smith *
		EmailAddress	Mary_Smith@widgets.com *
		FaxNumber	1-312-555-1214 *
		TelephoneNumber	1-312-555-2125 *
		ContactTypeIdentifier	Seller *
Receiver	Identifier		2203148000008
	IdentifierAuthority	EAN.UCC	
	ContactInformation	Contact	Jane Austin *
		EmailAddress	Jane_Austin@widgets.com *
		FaxNumber	1-312-555-1216 *
		TelephoneNumber	1-312-555-2127 *
		ContactTypeIdentifier	Assistant Seller *
Document Identification	Standard		http://www.uc-council.org/smp/schemas/simpl-eb
	TypeVersion		1.3
	Instance Identifier		100001
	Type		tradeItemDocument
	MultipleType		false
	CreationDateAndTime		2003-09-15T10:05:00Z
Manifest	NumberOfItems	2	
	ManifestItem	MIMETypeQualifierCode	video/mpeg
		UniformResourceIdentifier	http://www.widgets.com//ProductImage
		Description	MPEG Video Image of Product
		LanguageCode	EN

1404 * = No transformation changes between the Business Document and the Standard Business
1405 Document

1406

There are attachments to be sent along with the document; therefore the Business Data Creator populates the optional Manifest object.

2) The Business Data Creator collects the SBDH with the payload and attachments and passes all the data to the Parser Translator.

3) The Parser Translator receives the data and transforms the internal Business Document values into external SBDH semantic and format values and syntax and updates the Header with the new values. Logical information only is placed in the SBDH. The following field values are populated by the translator/parser to ensure that the values represent a well-known, shared standard. In this example, an XML syntax will be created.

Document Identification: In our example, this information is known from the URI of the namespace, so in this case it is redundant and does not need to be used. Still, we provide the URI as an example of the Standard.

4) The Parser Translator sends the data to the Communications Application.

5) The Communications Application receives the data and uses the SBDH to determine the physical destination of the document for external routing and the transport protocol used to move the data from the sender to the receiver. Typically, the Communications Application uses a table to lookup the destination and protocol.

Transport Headers are created by the Communications Application from the SBDH.

Schema examples for Sample 1 are located in Appendix A. The Instance Document is located in Appendix B.

11.2 Use case 2. XML data in an ebXML environment

Assumptions

- In this use case, the SBDH and the SBD will be packaged in one XML instance document. Therefore, the StandardBusinessDocument tag is used in this example. The rest of the payload is shown is a fragment of an Order document.

- This use case does not require the use of the optional Manifest object because there are no attachments to be sent.

- The middleware processing this use case requires that the information in the BusinessScope object is populated.
- In this use case there is only a primary routing for sender and a primary routing for receiver.

This use case shows only the standard values after the original Data Creator values are transformed in a single table. It does not include the Business Data Creator values.

This example contains a requesting and a responding example, useful in showing the use of the Business scopes.

The roles of the Business Data Creator, Parser/Translator and Communications Applications are the same as in the previous use case, even though the eventual SBD contents and packaging are somewhat different.

In this scenario, the Business Service to be carried out on the SBD is the Order-Sell service. Order-Sell service will invoke the Original-Order action. The Business Process that the Scopes are an instance of is the End-to-End-Order-to-Sell-Collaboration. The definitive reference to this Business Process is found at the location http://www.XYZretailer.com/ProcessReference/Order-Sell/version2. The current state of the executing Business Process from the sender's perspective is Pending. The receiver, having received communication of the Service Information from the sender's perspective, will act accordingly upon receipt of the SBD.

Once the Document Identification and Service Information are established, the parser/translator will use the Correlation object to establish explicit information about the requesting SBD (which contains the SBDH). Having the information explicitly stated allows both the sender and receiver to correlate the business domain information as the collaboration is in the process of execution. The date and time stamp of the Requesting SBD is: 2003-09-17T12:10:00Z as known from the Document Identification/ CreationDateAndTime. Therefore the requesting SBD will contain the same date and time stamp in the RequestingDocumentCreationDateTime. Since this information is redundant in this example, because it is the requesting example, the optional tag may be omitted. Likewise, the CorrelationInformation/ RequestingDocumentInstanceIdentifier is the same as the Document Identification/InstanceIdentifier in this requesting example. The response is expected by 2003-09-22T12:10:00Z (within 5 days from the 17th of September), and this is provided in the CorrelationInformation/ ExpectedResponseDateTime tag. The remainder of the values for Correlation object are shown in the table below.

1477

1478 **Table 4.**

1479 **Parser/Translator Transformed Business Terms for Requesting SBD**
1480 **(see Sample 2a in Appendix B)**
1481

Business Term				Transformed Example Value
StandardBusinessDocument				The root tag of the instance containing the SBDH and the SBD.
StandardBusinessDocumentHeader				The tag wrapping only the SBDH part.
HeaderVersion				1.0
Sender	Identifier			6903148000007
	Authority			EAN.UCC
	ContactInformation	Contact		John Doe *
		EmailAddress		John_Doe@purchasing.XYZretailer.com *
		FaxNumber		1-212-555-1213 *
		TelephoneNumber		1-212-555-2122 *
		ContactTypeIdentifier		Buyer *
Receiver	Identifier			2203148000007
	IdentifierAuthority			EAN.UCC
	ContactInformation	Contact		Mary Smith *
		EmailAddress		Mary_Smith@widgets.com *
		FaxNumber		1-312-555-1214 *
		TelephoneNumber		1-312-555-2125 *
		ContactTypeIdentifier		Seller *
Document Identification	Standard			http://www.uc-council.org/smp/schemas/simpl-eb/Order
	TypeVersion			1.3
	Instance Identifier			100002
	Type			order
	MultipleType			false
	CreationDateAndTime			2003-09-17T12:10:00Z
BusinessScope	Scope	Type		BusinessProcess
		InstanceIdentifier		End-to-End-Order-to-Sell-Collaboration
		Identifier		http://www.XYZretailer.com/ScopeIdentifier/Order-Sell/version2-123
		BusinessService		
		BusinessServiceName		Order-Sell

	ServiceTransaction	Original-Order
	TypeOfServiceTransaction	RequestingServiceTransaction
	IsNonRepudiationRequired	false
	IsAuthenticationRequired	true
	IsNonRepudiationOfReceiptRequired	true
	IsIntelligibleCheckRequired	true
	IsApplicationErrorResponseRequested	true
	TimeToAcknowledgeReceipt	P12H +
	TimeToAcknowledgeAcceptance	P2D +
	TimeToPerform	P5D +
	Recurrence	3
	CorrelationInformation	
	RequestingDocumentCreationDateTime	2003-09-17T12:10:00Z
	RequestingDocumentInstanceIdentifier	100002
	ExpectedResponseDateTime	2003-09-22T12:10:00Z
Order		This sample includes a fragment of an XML Order packaged as part of the Standard Business Document

* = No transformation changes between the Business Document and the Standard Business Document

+ See W3C Datatypes specification for the duration of time format.

In the Responding Document, Mary Smith is now the Sender and John Doe is now the Receiver. The type of document is an Order Response. The Document Identification/ InstanceIdentifier is 550001. The Document Identification/ CreationDateAndTime is May 9th, within the time allocated for a response. The Business Scope type is a Business Process with a new Instance Identifier. The Parent Scope is the same as the Scope for the Requesting Document.

The Correlation/ CreationDateAndTime, / InstanceIdentifier and / ExpectedResponseDateTime are not redundant in this responding example. The same information as found in the original requesting document is placed here. If there were several transactions in this collaboration, the original or first requesting document information would be placed here in all the SBDH instances. There could be several ongoing Request-Response collaborations between the two partners. This information “correlates” this response to the correct original request.

1504 **Table 5.**

1505 **Parser/Translator Transformed Business Terms for Responding SBD.**

1506 **(see Sample 2b in Appendix B)**

1507

Business Term			Transformed Example Value
StandardBusinessDocument			The root tag of the instance containing the SBDH and the SBD.
StandardBusinessDocumentHeader			The tag wrapping only the SBDH part.
HeaderVersion			1.0
Sender	Identifier		2203148000007
	Authority		EAN.UCC
	ContactInformation	Contact	Mary Smith *
		EmailAddress	Mary_Smith@widgets.com *
		FaxNumber	1-312-555-1214 *
		TelephoneNumber	1-312-555-2125 *
		ContactTypeIdentifier	Seller *
Receiver	Identifier		6903148000007
	IdentifierAuthority		EAN.UCC
	ContactInformation	Contact	John Doe *
		EmailAddress	John_Doe@purchasing.XYZretailer.com *
		FaxNumber	1-212-555-1213 *
		TelephoneNumber	1-212-555-2122 *
		ContactTypeIdentifier	John Doe *
Document Identification	Standard		http://www.uc-council.org/smp/schemas/simpl-eb/Order
	TypeVersion		1.3
	Instance Identifier		550001
	Type		Order
	MultipleType		false
	CreationDateAndTime		2003-09-17T12:10:00Z

BusinessScope	Scope	Type	BusinessProcess
		InstanceIdentifier	End-to-End-Order-to-Sell-Collaboration
		Identifier	http://www.XYZretailer.com/ScopeIdentifier/Order-Sell/version2-123
		BusinessService	
		BusinessServiceName	Order-Sell
		ServiceTransaction	Original-Order-Response
		TypeOfServiceTransaction	RespondingServiceTransaction
		IsNonRepudiationRequired	false
		IsAuthenticationRequired	true
		IsNonRepudiationOfReceiptRequired	true
		IsIntelligibleCheckRequired	true
		IsApplicationErrorResponseRequested	true
		TimeToAcknowledgeReceipt	P12H +
		TimeToAcknowledgeAcceptance	P2D +
		TimeToPerform	P5D +
		Recurrence	3
	Scope	Type	BusinessProcess
		InstanceIdentifier	XYZ
		Identifier	BP346
		ParentScope	
		Type	BusinessProcess
		InstanceIdentifier	ABC
		Identifier	BP345
	This is a placeholder for additional Business Scopes that will be defined by UN/CEFACT TBG, UN/CEFACT UBAC or other industry standards bodies.		
orderResponse			This sample includes a fragment of an XML Order Response packaged as part of the Standard Business Document

1508 * = No transformation changes between the Business Document and the Standard Business
1509 Document

1510 + See W3C Datatypes specification for the duration of time format.

Schemas for the SBDH and Sample order and order response are located in Appendix A. Instances for Sample 2 requesting and responding documents are located in Appendix B.

11.3 Use case 3. SBDH with EDI payload in an ebXML environment

Goal of this use case scenario

This scenario shows how the SBDH will work with an EDIFACT ORDERS message payload in a CEFACT Business Service to Business service and Business Process. In this Use Case, an EDI message is wrapped in the SBDH, in order to solve the problem of having no process information in EDI.

This scenario will show how to use the SBDH in an ebXML scenario and also how to help bring legacy systems forward by bringing collaborative knowledge in conjunction with the processes to non ebXML messages, such as EDI.

As an example, the following EDI messages form a process:

ORDERS Purchase Orders
INVOIC Invoices

In this scenario, those EDI messages could be handled as ebXML “business processes”.

The Business Data Creator is the source of message creation and creates data in “Internal Business Document" format. Because this is an ebXML environment, there is reason to use the Service Information. The Business Data Creator populates the ServiceInformation object. The Business Data Creator declares a Business Service using the EDI processes listed above. The Business Data Creator:

- establishes a context for the message;
- establishes a collaboration in which the established services are now capable of participation. The collaboration becomes associated with the set of information exchanges. The Business Data Creator and its partner on the other side will associate the set of EDI messages with the collaboration - the processes and instances of messages exchanged within the process.

The Business Data Creator sends all the data in “Internal Business Document" format to the Middleware.

The Middleware Parser Translator function receives the data and transforms the internal Business Document values into external SBDH format values. Only logical information is placed in the SBDH.

The Parser Translator sends the data to the Communications Application.

The Communications Application receives the data and uses the SBDH to determine the physical destination of the document (external routing) and the transport protocol used to move the data from the sender to the receiver. Typically, the Communications Application uses a table to lookup the destination and protocol.

Transport envelope values are created by the Communications Application from information in the SBDH.

An example of exchanging BP state information for a group of EDI transaction sets forming an “Order-Sell” process follows.

Below are the SBDH fields and their data values.

Assumptions:

- This use case will pass all the payload information as one instance document. The StandardBusinessDocument tag is used as the root.

This use case example shows only the requesting document.

Table 6. Parser/Translator Transformed Business Terms
(see Sample 3 in Appendix B)

Business Term			**Transformed Example Value in its XML Representation**
StandardBusinessDocument			Tag used to include the entire contents of the SBDH and the EDI Order.
StandardBusinessDocumentHeader			Tag used to wrap the contents of the SBDH
HeaderVersion			1.0
Sender	Identifier		6903148000007
	Authority		14
	ContactInformation	Contact	John Doe
		EmailAddress	John_Doe@XYZretailer.com

		FaxNumber	1-212-555-1213
		TelephoneNumber	1-212-555-2122
		ContactTypeIdentifier	Buyer
Receiver	Identifier		2203148000007
	Authority		14
	ContactInformation	Contact	Mary Smith
		EmailAddress	Mary_Smith@widgets.com
		FaxNumber	1-312-555-1214
		TelephoneNumber	1-312-555-2125
		ContactTypeIdentifier	Seller
DocumentIdentification	Standard		EDIFACT
	TypeVersion		D.96A
	InstanceIdentifier		100002
	Type		Order
	MultipleType		false
	CreationDateAndTime		2003-05-02T00:31:52Z
BusinessScope	Scope	Type	BusinessProcess
		InstanceIdentifier	Order-Sell/version2-251
		Identifier	EDI Order-Sell
		BusinessService	
		BusinessServiceName	Order-Sell
		ServiceTransaction	Original-Order
		TypeOfServiceTransaction	RequestingServiceTransaction
		IsNonRepudiationRequired	false
		IsAuthenticationRequired	true
		IsNonRepudiationOfReceiptRequired	true
		IsIntelligibleCheckRequired	true
		IsApplicationErrorResponseRequested	true
		TimeToAcknowledgeReceipt	+P12H
		TimeToAcknowledgeAcceptance	+P2D
		TimeToPerform	+P5D
		Recurrence	3
	CorrelationInformation	RequestingDocumentCreationDateTime	2003-05-02T00:31:52Z
		RequestingDocumentInstanceIdentifier	100002
		ExpectedResponseDateTime	2003-05-10T00:31:52Z

Order			This sample includes an EDI Order converted to an XML String packaged as part of the Standard Business Document

+ See W3C Datatypes specification for the duration of time format.

An XML instance document with an embedded EDI interchange matching Use Case 3 can be found in Appendix B Sample 3. The EDI data could have also been sent as an attachment in the Manifest.

11.4 Use of SBDH in Acknowledgement and Exception Situations

Use of the SBDH in acknowledgements and exception situations depends on the use case. If reliable messaging is used (e.g. AS2 or ebMS), then the transport acknowledgement signal would be part of the transport protocol. In that case, the SBDH would not be used.

However, if a business application generates an acknowledgment or exception message, then the inclusion of the SBDH would be useful. This deployment scenario would make the responding message just like any other business message with a SBDH included.

12 GLOSSARY

Applied Technology Group (ATG)	The purpose of the Applied Technologies Group (ATG) is to be responsible for the creation and maintenance of the trade, business and administration document structures that are based on a specific technology or standard. The function of the ATG is the design, assembly and production of syntax specific solutions based on identified business and/or technical requirements from the empowered groups of UN/CEFACT.
BCF	UN/CEFACT Business Collaboration Framework.
Business Document (BD)	A document used by a back office application, typically expressed in a proprietary format. The BD is typically transformed into a SBD via a middleware application such as a parser or a translator.
BPSS	Business Process Specification Schema. A UN/CEFACT requirements document.
Business Data Creator	The legacy, ERP or other application that creates business transactions for functional processes, such as ordering, invoicing, etc.
Business Service Interface (BSI)	The business layer interface described in ebXML.
Collaboration-Protocol Profile / Agreement (CPP/A)	An explicit TPA format defined by OASIS.
Communications Application	The application that transmits the SBD from the Sender to the Receiver.
DUNS	The identifier within the Dun & Bradstreet Universal Numbering System for partner identification.

ebMS	The electronic business Messaging Service specified by ebXML. Also known as ebXML-MS
EDI	Electronic Data Interchange
EDIFACT	Electronic Data Interchange for Administration, Commerce and Transport
GLN	The EAN Global Location Number for partner identification.
Parser/Translator	The application that transfers BDs from internal private format to an external SBD format including the SBDH.
Standard Business Document (SBD)	A document expressed in a format approved by a standards organization such as UN/CEFACT, EAN.UCC, Rosettanet, etc. Documents are typically shared between external trading partners in a SBD format.
Standard Business Document Header (SBDH)	The business level header in a standard format as described in this document. The SBDH is designed to be either an integral part of a Standard Business Document, or an object associated with the Standard Business Document.
Trading Partner Agreement (TPA)	An agreement between trading partners describing how they will exchange business information.
UN/CEFACT	United Nations Centre for Trade Facilitation and Electronic Business
UMM	UN/CEFACT Modelling Methodology
WSDL	W3C Web Services Definition Language.
XML	eXtensible Markup Language

1590

1591 Copyright Statement

Appendix A SBDH Schemas

NOTE: The example schemas in Appendix A are Non-Normative and are for information only. These will be changed, and when published, will comply with the UN/CEFACT Naming and Design Rules and the UN/CEFACT UML to XML Transformation Rules, when available.

A.1 BasicTypes.xsd

```
<?xml version="1.0"?>
<xs:schema
targetNamespace="http://www.unece.org/cefact/namespaces/StandardBusiness
DocumentHeader"
xmlns="http://www.unece.org/cefact/namespaces/StandardBusinessDocumentH
eader" xmlns:xs="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified" attributeFormDefault="unqualified">
    <xs:simpleType name="MimeTypeQualifier">
        <xs:annotation>
            <xs:documentation>The MIME type as defined by IANA. Please refer to
           http://www.iana.org/assignments/media-types/ for a list of types.
     </xs:documentation>
        </xs:annotation>
        <xs:restriction base="xs:string"/>
    </xs:simpleType>
    <xs:simpleType name="Language">
        <xs:annotation>
            <xs:documentation>ISO 639-2; 1998 representation of Language name.
Refer to http://www.loc.gov/standards/iso639-2/iso639jac.html to get the latest
version of the standard.
     </xs:documentation>
        </xs:annotation>
        <xs:restriction base="xs:string"/>
    </xs:simpleType>
</xs:schema>
```

A.2 BusinessScope.xsd

```
<?xml version="1.0"?>
<xs:schema
targetNamespace="http://www.unece.org/cefact/namespaces/StandardBusiness
DocumentHeader" xmlns:xs="http://www.w3.org/2001/XMLSchema"
```

```
1649  xmlns="http://www.unece.org/cefact/namespaces/StandardBusinessDocumentH
1650  eader" elementFormDefault="qualified" attributeFormDefault="unqualified">
1651     <xs:complexType name="BusinessScope">
1652        <xs:sequence>
1653           <xs:element name="Scope" type="Scope" minOccurs="0"
1654  maxOccurs="unbounded"/>
1655        </xs:sequence>
1656     </xs:complexType>
1657     <xs:complexType name="Scope">
1658        <xs:sequence>
1659           <xs:group ref="ScopeAttributes"/>
1660           <xs:element ref="ScopeInformation" minOccurs="0"
1661  maxOccurs="unbounded"/>
1662        </xs:sequence>
1663     </xs:complexType>
1664     <xs:group name="ScopeAttributes">
1665        <xs:sequence>
1666           <xs:element name="Type" type="xs:string"/>
1667           <xs:element name="InstanceIdentifier" type="xs:string"/>
1668           <xs:element name="Identifier" type="xs:string" minOccurs="0"/>
1669        </xs:sequence>
1670     </xs:group>
1671     <xs:element name="ScopeInformation" type="xs:anyType" abstract="true"/>
1672     <xs:element name="CorrelationInformation" type="CorrelationInformation"
1673  substitutionGroup="ScopeInformation"/>
1674     <xs:complexType name="CorrelationInformation">
1675        <xs:sequence>
1676           <xs:element name="RequestingDocumentCreationDateTime"
1677  type="xs:dateTime" minOccurs="0"/>
1678           <xs:element name="RequestingDocumentInstanceIdentifier"
1679  type="xs:string" minOccurs="0"/>
1680           <xs:element name="ExpectedResponseDateTime" type="xs:dateTime"
1681  minOccurs="0"/>
1682        </xs:sequence>
1683     </xs:complexType>
1684     <xs:element name="BusinessService" type="BusinessService"
1685  substitutionGroup="ScopeInformation"/>
1686     <xs:complexType name="BusinessService">
1687        <xs:sequence>
1688           <xs:element name="BusinessServiceName" type="xs:string"
1689  minOccurs="0"/>
1690           <xs:element name="ServiceTransaction" type="ServiceTransaction"
1691  minOccurs="0"/>
1692        </xs:sequence>
1693     </xs:complexType>
1694     <xs:complexType name="ServiceTransaction">
```

```
        <xs:attribute name="TypeOfServiceTransaction"
type="TypeOfServiceTransaction" use="optional"/>
        <xs:attribute name="IsNonRepudiationRequired" type="xs:string"/>
        <xs:attribute name="IsAuthenticationRequired" type="xs:string"/>
        <xs:attribute name="IsNonRepudiationOfReceiptRequired"
type="xs:string"/>
        <xs:attribute name="IsIntelligibleCheckRequired" type="xs:string"/>
        <xs:attribute name="IsApplicationErrorResponseRequested"
type="xs:string"/>
        <xs:attribute name="TimeToAcknowledgeReceipt" type="xs:string"/>
        <xs:attribute name="TimeToAcknowledgeAcceptance" type="xs:string"/>
        <xs:attribute name="TimeToPerform" type="xs:string"/>
        <xs:attribute name="Recurrence" type="xs:string"/>
    </xs:complexType>
    <xs:simpleType name="TypeOfServiceTransaction">
        <xs:restriction base="xs:string">
            <xs:enumeration value="RequestingServiceTransaction"/>
            <xs:enumeration value="RespondingServiceTransaction"/>
        </xs:restriction>
    </xs:simpleType>
</xs:schema>
```

A.3 DocumentIdentification.xsd

```
<?xml version="1.0"?>
<xs:schema
targetNamespace="http://www.unece.org/cefact/namespaces/StandardBusiness
DocumentHeader"
xmlns="http://www.unece.org/cefact/namespaces/StandardBusinessDocumentH
eader" xmlns:xs="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified" attributeFormDefault="unqualified">
    <xs:complexType name="DocumentIdentification">
        <xs:sequence>
            <xs:element name="Standard" type="xs:string"/>
            <xs:element name="TypeVersion" type="xs:string"/>
            <xs:element name="InstanceIdentifier" type="xs:string"/>
            <xs:element name="Type" type="xs:string"/>
            <xs:element name="MultipleType" type="xs:boolean" minOccurs="0"/>
            <xs:element name="CreationDateAndTime" type="xs:dateTime"/>
        </xs:sequence>
    </xs:complexType>
</xs:schema>
```

A.4 Manifest.xsd

```
<?xml version="1.0"?>
<xs:schema
targetNamespace="http://www.unece.org/cefact/namespaces/StandardBusiness
DocumentHeader"
xmlns="http://www.unece.org/cefact/namespaces/StandardBusinessDocumentH
eader" xmlns:xs="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified" attributeFormDefault="unqualified">
   <xs:include schemaLocation="BasicTypes.xsd"/>
   <xs:complexType name="Manifest">
      <xs:sequence>
         <xs:element name="NumberOfItems" type="xs:integer"/>
         <xs:element name="ManifestItem" type="ManifestItem"
maxOccurs="unbounded"/>
      </xs:sequence>
   </xs:complexType>
   <xs:complexType name="ManifestItem">
      <xs:sequence>
         <xs:element name="MimeTypeQualifierCode"
type="MimeTypeQualifier"/>
         <xs:element name="UniformResourceIdentifier" type="xs:anyURI"/>
         <xs:element name="Description" type="xs:string" minOccurs="0"/>
         <xs:element name="LanguageCode" type="Language" minOccurs="0"/>
      </xs:sequence>
   </xs:complexType>
</xs:schema>
```

A.5 StandardBusinessDocumentHeader.xsd

```
<?xml version="1.0"?>
<xs:schema
targetNamespace="http://www.unece.org/cefact/namespaces/StandardBusiness
DocumentHeader"
xmlns="http://www.unece.org/cefact/namespaces/StandardBusinessDocumentH
eader" xmlns:xs="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified" attributeFormDefault="unqualified">
  <xs:include schemaLocation="DocumentIdentification.xsd"/>
  <xs:include schemaLocation="Partner.xsd"/>
  <xs:include schemaLocation="Manifest.xsd"/>
  <xs:include schemaLocation="BusinessScope.xsd"/>
  <xs:complexType name="StandardBusinessDocumentHeader">
    <xs:sequence>
```

```
      <xs:element name="HeaderVersion" type="xs:string"/>
      <xs:element name="Sender" type="Partner" maxOccurs="unbounded"/>
      <xs:element name="Receiver" type="Partner" maxOccurs="unbounded"/>
      <xs:element name="DocumentIdentification" type="DocumentIdentification"/>
      <xs:element name="Manifest" type="Manifest" minOccurs="0"/>
      <xs:element name="BusinessScope" type="BusinessScope"
minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
  <xs:element name="StandardBusinessDocumentHeader"
type="StandardBusinessDocumentHeader"/>
  <xs:element name="StandardBusinessDocument"
type="StandardBusinessDocument"/>
  <xs:complexType name="StandardBusinessDocument">
    <xs:sequence>
      <xs:element ref="StandardBusinessDocumentHeader" minOccurs="0"/>
      <xs:any namespace="##other" processContents="lax"/>
    </xs:sequence>
  </xs:complexType>
</xs:schema>
```

A.6 Partner.xsd

```
<?xml version="1.0"?>
<xs:schema
targetNamespace="http://www.unece.org/cefact/namespaces/StandardBusiness
DocumentHeader"
xmlns="http://www.unece.org/cefact/namespaces/StandardBusinessDocumentH
eader" xmlns:xs="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified" attributeFormDefault="unqualified">
    <xs:complexType name="Partner">
        <xs:sequence>
            <xs:element name="Identifier" type="PartnerIdentification"/>
            <xs:element name="ContactInformation" type="ContactInformation"
minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="PartnerIdentification">
        <xs:simpleContent>
            <xs:extension base="xs:string">
                <xs:attribute name="Authority" type="xs:string"/>
            </xs:extension>
        </xs:simpleContent>
    </xs:complexType>
```

```
    <xs:complexType name="ContactInformation">
        <xs:sequence>
            <xs:element name="Contact" type="xs:string"/>
            <xs:element name="EmailAddress" type="xs:string" minOccurs="0"/>
            <xs:element name="FaxNumber" type="xs:string" minOccurs="0"/>
            <xs:element name="TelephoneNumber" type="xs:string"
minOccurs="0"/>
            <xs:element name="ContactTypeIdentifier" type="xs:string"
minOccurs="0"/>
        </xs:sequence>
    </xs:complexType>
</xs:schema>
```

A.7 Schemas for Use with Samples

A.7.1 Simulated Order.xsd for Use with Sample 2

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema targetNamespace="http://www.ean-ucc.org/schemas/1.3/eanucc"
xmlns="http://www.ean-ucc.org/schemas/1.3/eanucc"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
elementFormDefault="unqualified" attributeFormDefault="unqualified">
    <xsd:element name="order" type="OrderType"/>
    <xsd:complexType name="OrderType">
        <xsd:sequence>
            <xsd:element name="orderIdentification" type="xsd:string"/>
            <!-- rest of content model would go here -->
        </xsd:sequence>
    </xsd:complexType>
</xsd:schema>
```

A.7.2 Simulated OrderResponse.xsd for Use with Sample 2

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema targetNamespace="http://www.ean-ucc.org/schemas/1.3/eanucc"
xmlns="http://www.ean-ucc.org/schemas/1.3/eanucc"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
elementFormDefault="unqualified" attributeFormDefault="unqualified">
    <xsd:element name="orderResponse" type="OrderResponseType"/>
    <xsd:complexType name="OrderResponseType">
        <xsd:sequence>
            <xsd:element name="orderResponseIdentification" type="xsd:string"/>
```

```
            <!-- rest of content model would go here -->
        </xsd:sequence>
    </xsd:complexType>
</xsd:schema>
```

A.7.3 Simulated OrderProxy.xsd for Use with Sample 2

```
<?xml version="1.0"?>
<xs:schema targetNamespace="http://www.ean-ucc.org/schemas/1.3/eanucc"
xmlns:unece="http://www.unece.org/cefact/namespaces/StandardBusinessDocu
mentHeader" xmlns="http://www.ean-ucc.org/schemas/1.3/eanucc"
xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
attributeFormDefault="unqualified">
    <xs:import
namespace="http://www.unece.org/cefact/namespaces/StandardBusinessDocum
entHeader" schemaLocation="StandardBusinessDocumentHeader.xsd"/>
    <xs:include schemaLocation="Order.xsd"/>
</xs:schema>
```

A.7.4 Simulated OrderResponseProxy.xsd for Use with Sample 2

```
<?xml version="1.0"?>
<xs:schema targetNamespace="http://www.ean-ucc.org/schemas/1.3/eanucc"
xmlns:unece="http://www.unece.org/cefact/namespaces/StandardBusinessDocu
mentHeader" xmlns="http://www.ean-ucc.org/schemas/1.3/eanucc"
xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
attributeFormDefault="unqualified">
    <xs:import
namespace="http://www.unece.org/cefact/namespaces/StandardBusinessDocum
entHeader" schemaLocation="StandardBusinessDocumentHeader.xsd"/>
    <xs:include schemaLocation="OrderResponse.xsd"/>
</xs:schema>
```

A.7.5 Simulated EDIOrder.xsd for Use with Sample 3

```
<?xml version="1.0"?>
<xs:schema targetNamespace="http://www.edi-order.org/schemas"
xmlns:unece="http://www.unece.org/cefact/namespaces/StandardBusinessDocu
mentHeader" xmlns="http://www.edi-order.org/schemas"
```

```
xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
attributeFormDefault="unqualified">
    <xs:import
namespace="http://www.unece.org/cefact/namespaces/StandardBusinessDocum
entHeader" schemaLocation="StandardBusinessDocumentHeader.xsd"/>
    <xs:include schemaLocation="EDIOrder.xsd"/>
</xs:schema>
```

A.7.6 Simulated EDIOrderProxy.xsd for Use with Sample 3

```
<?xml version="1.0"?>
<xs:schema targetNamespace="http://www.edi-order.org/schemas"
xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns="http://www.edi-
order.org/schemas"
xmlns:unece="http://www.unece.org/cefact/namespaces/StandardBusinessDocu
mentHeader" elementFormDefault="qualified"
attributeFormDefault="unqualified">
    <xs:import
namespace="http://www.unece.org/cefact/namespaces/StandardBusinessDocum
entHeader" schemaLocation="StandardBusinessDocumentHeader.xsd"/>
    <xs:include schemaLocation="EDIOrder.xsd"/>
</xs:schema>
```

Appendix B Sample XML Instance Files

NOTE: The sample XML instance files in Appendix B are Non-Normative and are for information only. These will be changed, and when published, will comply with the UN/CEFACT Naming and Design Rules and the UN/CEFACT UML to XML Transformation Rules, when available.

B.1 Sample 1

(see Table 3 in the Use Case examples)

```
<?xml version="1.0" encoding="UTF-8"?>
<sh:StandardBusinessDocumentHeader
xmlns:sh="http://www.unece.org/cefact/namespaces/StandardBusinessDocumen
tHeader" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.unece.org/cefact/namespaces/StandardBusines
sDocumentHeader StandardBusinessDocumentHeader.xsd">
   <sh:HeaderVersion>1.0</sh:HeaderVersion>
   <sh:Sender>
      <sh:Identifier Authority="EAN.UCC">6903148000007</sh:Identifier>
      <sh:ContactInformation>
         <sh:Contact>Corporate Headquarters</sh:Contact>

   <sh:EmailAddress>Corporate_Headquarters@XYZretailer.com</sh:EmailAddr
ess>
         <sh:FaxNumber>+1-212-555-1212</sh:FaxNumber>
         <sh:TelephoneNumber>+1-212-555-2121</sh:TelephoneNumber>
         <sh:ContactTypeIdentifier>Corporate
Organization</sh:ContactTypeIdentifier>
      </sh:ContactInformation>
   </sh:Sender>
   <sh:Sender>
      <sh:Identifier Authority="EAN.UCC">6903148000008</sh:Identifier>
      <sh:ContactInformation>
         <sh:Contact>John Doe</sh:Contact>

   <sh:EmailAddress>John_Doe@purchasing.XYZretailer.com</sh:EmailAddres
s>
         <sh:FaxNumber>+1-212-555-1213</sh:FaxNumber>
         <sh:TelephoneNumber>+1-212-555-2122</sh:TelephoneNumber>
         <sh:ContactTypeIdentifier>Buyer</sh:ContactTypeIdentifier>
      </sh:ContactInformation>
   </sh:Sender>
```

```
1977        <sh:Receiver>
1978            <sh:Identifier Authority="Widgets">2203148000007</sh:Identifier>
1979            <sh:ContactInformation>
1980                <sh:Contact>Mary Smith</sh:Contact>
1981                <sh:EmailAddress>Mary_Smith@widgets.com</sh:EmailAddress>
1982                <sh:FaxNumber>+1-312-555-1214</sh:FaxNumber>
1983                <sh:TelephoneNumber>+1-312-555-2125</sh:TelephoneNumber>
1984                <sh:ContactTypeIdentifier>Seller</sh:ContactTypeIdentifier>
1985            </sh:ContactInformation>
1986        </sh:Receiver>
1987        <sh:Receiver>
1988            <sh:Identifier Authority="Widgets">2203148000008</sh:Identifier>
1989            <sh:ContactInformation>
1990                <sh:Contact>Jane Austin</sh:Contact>
1991                <sh:EmailAddress>Jane _Austin@widgets.com</sh:EmailAddress>
1992                <sh:FaxNumber>+1-312-555-1216</sh:FaxNumber>
1993                <sh:TelephoneNumber>+1-312-555-2127</sh:TelephoneNumber>
1994                <sh:ContactTypeIdentifier>Assistant Seller</sh:ContactTypeIdentifier>
1995            </sh:ContactInformation>
1996        </sh:Receiver>
1997        <sh:DocumentIdentification>
1998            <sh:Standard>http://www.uc-council.org/smp/schemas/simpl-
1999    eb/Order</sh:Standard>
2000            <sh:TypeVersion>1.3</sh:TypeVersion>
2001            <sh:InstanceIdentifier>100001</sh:InstanceIdentifier>
2002            <sh:Type> tradeItemDocument </sh:Type>
2003            <sh:MultipleType>false</sh:MultipleType>
2004            <sh:CreationDateAndTime>2003-09-
2005    15T10:05:00Z</sh:CreationDateAndTime>
2006        </sh:DocumentIdentification>
2007        <sh:Manifest>
2008            <sh:NumberOfItems>2</sh:NumberOfItems>
2009            <sh:ManifestItem>
2010
2011        <sh:MimeTypeQualifierCode>application/xml</sh:MimeTypeQualifierCode>
2012                <sh:UniformResourceIdentifier> http://www.widgets.com//ProductImage
2013    </sh:UniformResourceIdentifier>
2014                <sh:Description>MPEG Video Image of Product</sh:Description>
2015                <sh:LanguageCode>EN</sh:LanguageCode>
2016            </sh:ManifestItem>
2017        </sh:Manifest>
2018    </sh:StandardBusinessDocumentHeader>
2019
2020
2021
2022
```

B.2 Sample 2

B.2.1 Sample 2a Requesting Document

(see Table 4 in the Use Case examples)

```
<?xml version="1.0" encoding="UTF-8"?>
<sh:StandardBusinessDocument
xmlns:sh="http://www.unece.org/cefact/namespaces/StandardBusinessDocumen
tHeader" xmlns:eanucc="http://www.ean-ucc.org/schemas/1.3/eanucc"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.unece.org/cefact/namespaces/StandardBusines
sDocumentHeader OrderProxy.xsd">
   <sh:StandardBusinessDocumentHeader>
      <sh:HeaderVersion>1.0</sh:HeaderVersion>
      <sh:Sender>
         <sh:Identifier Authority="EAN.UCC">6903148000007</sh:Identifier>
         <sh:ContactInformation>
            <sh:Contact>John Doe</sh:Contact>

   <sh:EmailAddress>John_Doe@purchasing.XYZretailer.com</sh:EmailAddres
s>
               <sh:FaxNumber>+1-212-555-1213</sh:FaxNumber>
               <sh:TelephoneNumber>+1-212-555-2122</sh:TelephoneNumber>
               <sh:ContactTypeIdentifier>Buyer</sh:ContactTypeIdentifier>
         </sh:ContactInformation>
      </sh:Sender>
      <sh:Receiver>
         <sh:Identifier Authority="EAN.UCC">2203148000007</sh:Identifier>
         <sh:ContactInformation>
            <sh:Contact>Mary Smith</sh:Contact>
            <sh:EmailAddress>Mary_Smith@widgets.com</sh:EmailAddress>
            <sh:FaxNumber>+1-312-555-1214</sh:FaxNumber>
            <sh:TelephoneNumber>+1-312-555-2125</sh:TelephoneNumber>
            <sh:ContactTypeIdentifier>Seller</sh:ContactTypeIdentifier>
         </sh:ContactInformation>
      </sh:Receiver>
      <sh:DocumentIdentification>
         <sh:Standard>http://www.uc-council.org/smp/schemas/simpl-
eb/Order</sh:Standard>
         <sh:TypeVersion>1.3</sh:TypeVersion>
         <sh:InstanceIdentifier>100002</sh:InstanceIdentifier>
         <sh:Type>order</sh:Type>
         <sh:MultipleType>false</sh:MultipleType>
         <sh:CreationDateAndTime>2003-09-
17T12:10:00Z</sh:CreationDateAndTime>
```

```
2068          </sh:DocumentIdentification>
2069          <sh:BusinessScope>
2070             <sh:Scope>
2071                <sh:Type>BusinessProcess</sh:Type>
2072                <sh:InstanceIdentifier>Order-Sell/version2-
2073    123</sh:InstanceIdentifier>
2074                <sh:Identifier>Contract Order-Sell</sh:Identifier>
2075                <sh:BusinessService>
2076                   <sh:BusinessServiceName>Order-
2077    Sell</sh:BusinessServiceName>
2078                   <sh:ServiceTransaction
2079    TypeOfServiceTransaction="RequestingServiceTransaction"
2080    IsAuthenticationRequired="true" IsNonRepudiationRequired="true"
2081    IsNonRepudiationOfReceiptRequired="true" IsIntelligibleCheckRequired="true"
2082    IsApplicationErrorResponseRequested="true"
2083    TimeToAcknowledgeReceipt="P12H" TimeToAcknowledgeAcceptance="P2D"
2084    TimeToPerform="P5D" Recurrence="3"/>
2085                </sh:BusinessService>
2086                <sh:CorrelationInformation>
2087                   <sh:RequestingDocumentCreationDateTime>2003-09-
2088    17T12:10:00Z</sh:RequestingDocumentCreationDateTime>
2089
2090       <sh:RequestingDocumentInstanceIdentifier>100002</sh:RequestingDocumen
2091    tInstanceIdentifier>
2092                   <sh:ExpectedResponseDateTime>2003-09-
2093    22T12:10:00Z</sh:ExpectedResponseDateTime>
2094                </sh:CorrelationInformation>
2095             </sh:Scope>
2096             <sh:Scope>
2097                <sh:Type>BusinessProcess</sh:Type>
2098                <sh:InstanceIdentifier>XYZ</sh:InstanceIdentifier>
2099                <sh:Identifier>BP346</sh:Identifier>
2100                <!--<sh:AsYetUndefined>...</sh:AsYetUndefined>-->
2101             </sh:Scope>
2102          </sh:BusinessScope>
2103       </sh:StandardBusinessDocumentHeader>
2104       <eanucc:order>
2105          <orderIdentification>5412345000013</orderIdentification>
2106          <!-- rest of order document goes here -->
2107       </eanucc:order>
2108    </sh:StandardBusinessDocument>
2109
2110
```

2111 B.2.2 Sample 2b Responding Document

2112 (see Table 5 in the Use Case examples)

2113

```
<?xml version="1.0" encoding="UTF-8"?>
<sh:StandardBusinessDocument
xmlns:sh="http://www.unece.org/cefact/namespaces/StandardBusinessDocumen
tHeader" xmlns:eanucc="http://www.ean-ucc.org/schemas/1.3/eanucc"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.unece.org/cefact/namespaces/StandardBusines
sDocumentHeader OrderResponseProxy.xsd">
   <sh:StandardBusinessDocumentHeader>
      <sh:HeaderVersion>1.0</sh:HeaderVersion>
      <sh:Sender>
         <sh:Identifier Authority="EAN.UCC">2203148000007</sh:Identifier>
         <sh:ContactInformation>
            <sh:Contact>Mary Smith</sh:Contact>
            <sh:EmailAddress>Mary_Smith@widgets.com</sh:EmailAddress>
            <sh:FaxNumber>+1-312-555-1214</sh:FaxNumber>
            <sh:TelephoneNumber>+1-312-555-2125</sh:TelephoneNumber>
            <sh:ContactTypeIdentifier>Seller</sh:ContactTypeIdentifier>
         </sh:ContactInformation>
      </sh:Sender>
      <sh:Receiver>
         <sh:Identifier Authority="EAN.UCC">6903148000007</sh:Identifier>
         <sh:ContactInformation>
            <sh:Contact>John Doe</sh:Contact>

   <sh:EmailAddress>John_Doe@purchasing.XYZretailer.com</sh:EmailAddres
s>
            <sh:FaxNumber>+1-212-555-1213</sh:FaxNumber>
            <sh:TelephoneNumber>+1-212-555-2122</sh:TelephoneNumber>
            <sh:ContactTypeIdentifier>Buyer</sh:ContactTypeIdentifier>
         </sh:ContactInformation>
      </sh:Receiver>
      <sh:DocumentIdentification>
         <sh:Standard>http://www.uc-council.org/smp/schemas/simpl-
eb/OrderResponse</sh:Standard>
         <sh:TypeVersion>1.3</sh:TypeVersion>
         <sh:InstanceIdentifier>550001</sh:InstanceIdentifier>
         <sh:Type>OrderResponse</sh:Type>
         <sh:MultipleType>false</sh:MultipleType>
         <sh:CreationDateAndTime>2003-05-
09T00:31:52Z</sh:CreationDateAndTime>
      </sh:DocumentIdentification>
      <sh:BusinessScope>
         <sh:Scope>
            <sh:Type>BusinessProcess</sh:Type>
            <sh:InstanceIdentifier>Order-Sell/version2-
130</sh:InstanceIdentifier>
```

```
            <sh:Identifier>Contract Order-Sell</sh:Identifier>
            <sh:BusinessService>
               <sh:BusinessServiceName>Order-
Sell</sh:BusinessServiceName>
               <sh:ServiceTransaction
TypeOfServiceTransaction="RequestingServiceTransaction"
IsAuthenticationRequired="true" IsNonRepudiationRequired="true"
IsNonRepudiationOfReceiptRequired="true" IsIntelligibleCheckRequired="true"
IsApplicationErrorResponseRequested="true"
TimeToAcknowledgeReceipt="P12H" TimeToAcknowledgeAcceptance="P2D"
TimeToPerform="P5D" Recurrence="3"/>
            </sh:BusinessService>
            <sh:CorrelationInformation>
               <sh:RequestingDocumentCreationDateTime>2003-05-
02T00:31:52Z</sh:RequestingDocumentCreationDateTime>

   <sh:RequestingDocumentInstanceIdentifier>100002</sh:RequestingDocumen
tInstanceIdentifier>
               <sh:ExpectedResponseDateTime>2003-05-
10T00:31:52Z</sh:ExpectedResponseDateTime>
            </sh:CorrelationInformation>
         </sh:Scope>
         <sh:Scope>
            <sh:Type>BusinessProcess</sh:Type>
            <sh:InstanceIdentifier>XYZ</sh:InstanceIdentifier>
            <sh:Identifier>BP346</sh:Identifier>
            <!--<sh:AsYetUndefined>...</sh:AsYetUndefined>-->
         </sh:Scope>
      </sh:BusinessScope>
   </sh:StandardBusinessDocumentHeader>
   <eanucc:orderResponse>

   <orderResponseIdentification>5412345000013</orderResponseIdentification>
      <!-- rest of order document goes here -->
   </eanucc:orderResponse>
</sh:StandardBusinessDocument>
```

B.3 Sample 3

(see Table 6 in the Use Case examples)

```
<?xml version="1.0" encoding="UTF-8"?>
<sh:StandardBusinessDocument
xmlns:sh="http://www.unece.org/cefact/namespaces/StandardBusinessDocumen
```

```
tHeader" xmlns:ediorder="http://www.edi-order.org/schemas"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.unece.org/cefact/namespaces/StandardBusines
sDocumentHeader EDIOrderProxy.xsd">
    <sh:StandardBusinessDocumentHeader>
        <sh:HeaderVersion>1.0</sh:HeaderVersion>
        <sh:Sender>
            <sh:Identifier Authority="EAN.UCC">2203148000007</sh:Identifier>
            <sh:ContactInformation>
                <sh:Contact>Mary Smith</sh:Contact>
                <sh:EmailAddress>Mary_Smith@widgets.com</sh:EmailAddress>
                <sh:FaxNumber>+1-312-555-1214</sh:FaxNumber>
                <sh:TelephoneNumber>+1-312-555-2125</sh:TelephoneNumber>
                <sh:ContactTypeIdentifier>Seller</sh:ContactTypeIdentifier>
            </sh:ContactInformation>
        </sh:Sender>
        <sh:Receiver>
            <sh:Identifier Authority="EAN.UCC">6903148000007</sh:Identifier>
            <sh:ContactInformation>
                <sh:Contact>John Doe</sh:Contact>

    <sh:EmailAddress>John_Doe@purchasing.XYZretailer.com</sh:EmailAddres
s>
                <sh:FaxNumber>+1-212-555-1213</sh:FaxNumber>
                <sh:TelephoneNumber>+1-212-555-2122</sh:TelephoneNumber>
                <sh:ContactTypeIdentifier>Buyer</sh:ContactTypeIdentifier>
            </sh:ContactInformation>
        </sh:Receiver>
        <sh:DocumentIdentification>
            <sh:Standard>http://www.uc-council.org/smp/schemas/simpl-
eb/OrderResponse</sh:Standard>
            <sh:TypeVersion>D:96A</sh:TypeVersion>
            <sh:InstanceIdentifier>100003</sh:InstanceIdentifier>
            <sh:Type>ORDERS</sh:Type>
            <sh:MultipleType>false</sh:MultipleType>
            <sh:CreationDateAndTime>2003-05-
09T00:31:52Z</sh:CreationDateAndTime>
        </sh:DocumentIdentification>
        <sh:BusinessScope>
            <sh:Scope>
                <sh:Type>BusinessProcess</sh:Type>
                <sh:InstanceIdentifier>Order-Sell/version2-
251</sh:InstanceIdentifier>
                <sh:Identifier>EDI-Order-Sell</sh:Identifier>
                <sh:BusinessService>
```

```
                <sh:BusinessServiceName>Order-
Sell</sh:BusinessServiceName>
                <sh:ServiceTransaction
TypeOfServiceTransaction="RequestingServiceTransaction"
IsAuthenticationRequired="true" IsNonRepudiationRequired="true"
IsNonRepudiationOfReceiptRequired="true" IsIntelligibleCheckRequired="true"
IsApplicationErrorResponseRequested="true"
TimeToAcknowledgeReceipt="P12H" TimeToAcknowledgeAcceptance="P2D"
TimeToPerform="P5D" Recurrence="3"/>
            </sh:BusinessService>
            <sh:CorrelationInformation>
                <sh:RequestingDocumentCreationDateTime>2003-05-
02T00:31:52Z</sh:RequestingDocumentCreationDateTime>

   <sh:RequestingDocumentInstanceIdentifier>100002</sh:RequestingDocumen
tInstanceIdentifier>
                <sh:ExpectedResponseDateTime>2003-05-
10T00:31:52Z</sh:ExpectedResponseDateTime>
            </sh:CorrelationInformation>
         </sh:Scope>
      </sh:BusinessScope>
   </sh:StandardBusinessDocumentHeader>
   <ediorder:Order>

UNB+UNOA:3+6907777000001:14+6903148000007:14+030608:2206+811'
      UNH+1+ORDERS:D:96A:UN'
    …
   UNT+37+5'
   UNZ+5+811'
  </ediorder:Order>
</sh:StandardBusinessDocument>
```

Appendix C Theory Behind the SBDH Business Scope

Information about a Business Process specifies the scope or context of a single message exchange, however there are other types of *governing scopes* and contexts. Examples are TPA, economic contracts, technical agreements, and transaction specification. They are all governing message exchanges and are relevant to processing, parsing, translation, and routing etc. The following generalized header meta model provides for other types of business scopes and contexts which have business relevance to the sender and receiver.

From a philosophical and theoretical point of view, scope and context are commonly occurring. In order to interpret and process a message it is important to know in which business scope or context a business dialog is being conducted.

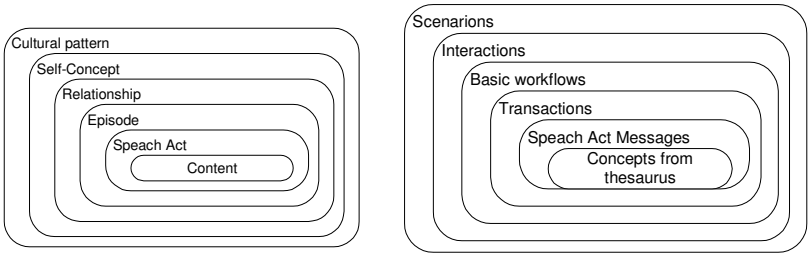


Figure C.1.
From theory of Coordinated Management Meaning

The business scopes and contexts often form natural hierarchies such as depicted in the diagram below. Often an exchange of words or business information, in the world of e-business, is conducted within several contexts:

- Within supply chains there may be business processes;
- within a process there may be several dialogs or collaborations;
- within a collaboration there may be sub collaborations;
- within a collaboration there may be multiple transactions;
- within a transaction there may be messages and signals being transmitted;
- within a message exchange there may be resending, reliability signals etc.

Apart from behavioural and state scopes there are other types of governing scopes and contexts in which an exchange of words, messages, documents or business information may be conducted. Agreements and contracts provide legal governance of information exchanges in order to satisfy the goals of business relationships.
An example: In order to fulfil a commitment to deliver goods, a business dialog or collaboration must be defined and agreed upon. Since the parties already have been engaged in electronic collaborations over an existing communication channel. they may decide to reuse an existing Trading Partner Agreement, its general provisions and technical details. Furthermore a generic business level agreement may specify that all deliveries of a certain kind must or should be made to a specific factory.

It is unrealistic to prescribe that all governing details must be accessible from a single specification document, including all business and technical properties. This vision involves unnecessary bindings between the business perspective and technical details. If a delivery location is changed it should not cause a TPA to be renegotiated and agreed and vice versa.

Therefore a general and federated model based on dependencies is preferable.

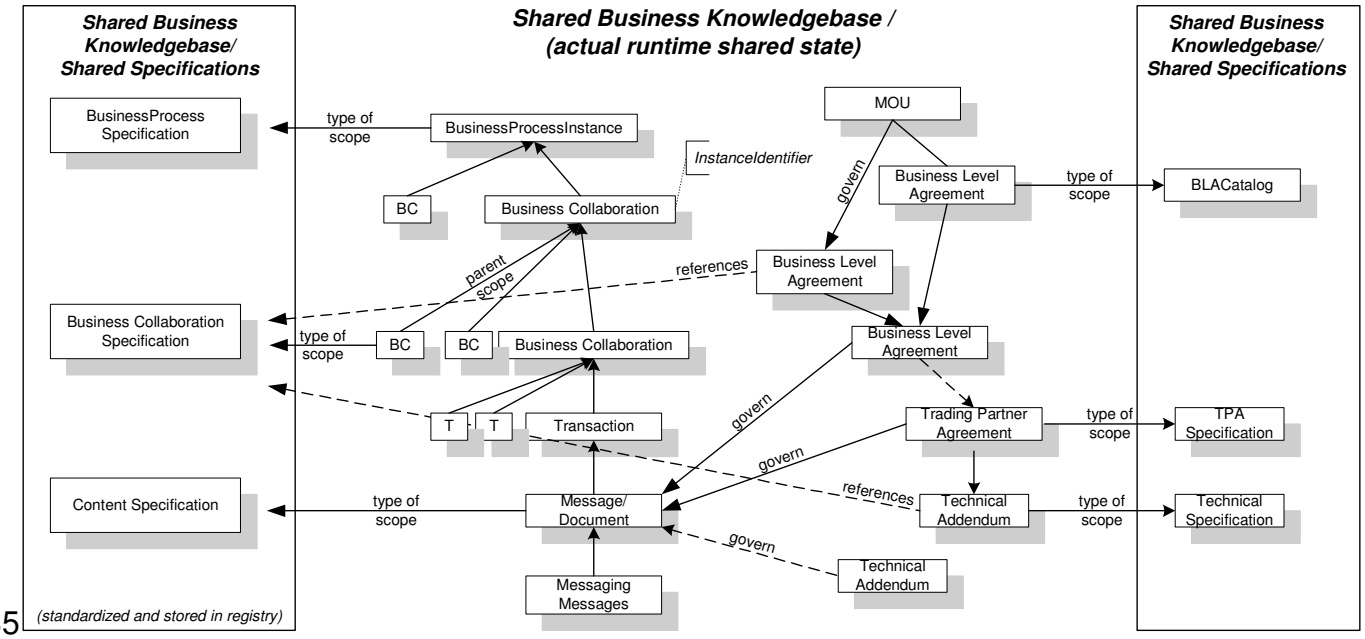


Figure C.2. Shared Business Knowledgebase

C.1 The Commonly Occurring Perspectives of Business Scope

There are 3 commonly occurring perspectives of scope and contexts:

1. Protocol:
When exchanging business information and documents, only the lowest level, smallest, innermost scope is needed or required. All upper level, governing parent scopes are accessed implicitly through knowledge of previously exchanged information and specifications. This view corresponds to a protocol stack where knowledge about upper layers should (must) not be required explicitly.

2. All scopes must be specified:
In order to successfully and deterministically process an exchange of business information all governing scopes must be available in every exchange.

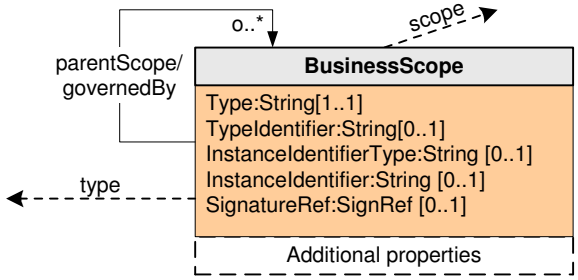
3. Interest based:
Only the scope information that the parties agree to or the parties deem interesting should be exchanged.

Information about a particular perspective may be specified in a Profile. (see optional parts below)

C.2 Meta model

The meta model adds simple yet dynamic scoping to the header construct: The model specifies a directed acyclic graph (DAG) of governing scopes and contexts that covers a large set of frequently occurring business cases.

This meta model of scope and context specification allows for great flexibility for business partners to use in ways we today cannot foresee. Yet it is predictable, composable and deterministic.



BusinessScope contains [1..1] [

Scope consists of [1..*] [
ScopeType:String [1..1] - type of scope:
Examples are UN/CEFACT Transaction, BCF:BusinessCollaboration, BusinessProcess, ebXML:BusinessService, BusinessServiceAction, BCF:AuthorizedRole

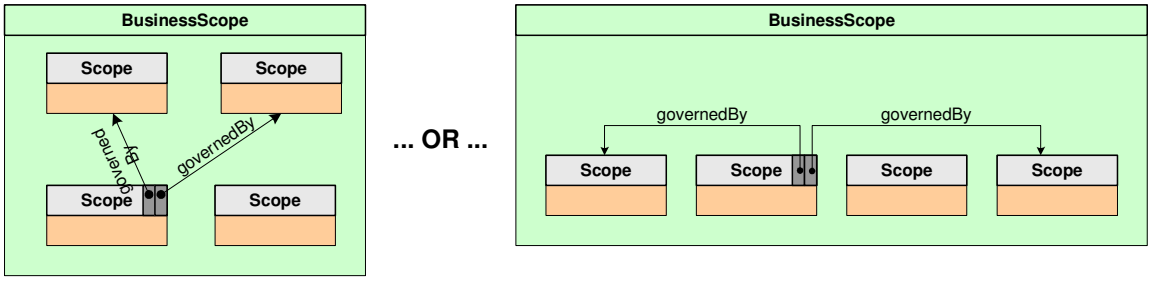
TypeIdentifier: String[0..1] – optional unique identifier that references the type of governing scope (e.g. process model, document specification). Example; “bpss:dropship”

InstanceIdentifierType: String [0..1] – identifiers the type of instance identifier. Examples: URL, GUID, ID, IdentifierString;

InstanceIdentifier: String [0..1] – unique identifier that references the instance of that scope (e.g. process execution instance, document instance) Example; “bpss:dropship:id-abcd123”

ScopeSignatureReference: SignRef [0..1] – a optional signature reference to the (governing) scope. In order to provide additional security a signature reference that point to governing scope may be defined.

GovernedBy contains [0..*] [
ParentInstanceIdentifier:String [1..1] - optional Most of the time scopes forms a natural governance hierarchy and often a message exchange is governed by more than one parent agreement. This element references another Scopes InstanceIdentifier.
]
]



C.3 Wellformedness rules

[1] It is not mandatory to put all intermediary scopes in a generic header. Only those that the parties agree to are needed. The following examples are all relevant: [transaction], [transaction, business process], [business process], [transaction, collaboration, collaboration, business process].

[2] A Profile may be used to group wf-rules together.

[3] The generic meta model specifies that cycles must not be present, i.e. by following the GovernedBy relationship one must not return to the same scope.

C.4 Optional parts

[1] An addition to above meta model: It is to possible add extra properties that contain additional information about the scope and context. This information is most likely to be redundant but may be used to control and verify state synchronization. If the Scope is modelled using UML or similar modelling language then additional properties may be captured in subclasses to Scope.

[2] It possible to add a Profile concept to Business Scope wellformedness rules so that various combinations of mandatory ScopeType requirements may be

grouped together. A profile is an expression of a particular perspective of Business Scope.

[3] It is possibility to add an extra property to the Governance element which specifies that the parent and child lifecycles are related and that when a parent ends its lifecycle the child also end its lifecycle.

[4] It is possible to add an information element in the GovernedBy element in order to indicate governance details. An example is an element that defines superiority rules regulating overlapping rules in child scope versus parent governing scope.

[5] The generic meta model specifies that cycles must not be present, i.e. by following the GovernedBy relationship one must not return to the same scope. This restriction may be relaxed by adding above Superiority rule and allowing cycles.

C.5 NOTES

[1] The parent child relationship between scopes is not the same as a lifecycle relationship. When a parent scope ends the child scope may still be active. However in many use cases the scope relationship is linked to lifecycles but in this generic meta model this dependency is implicit.

[2] Several methods may be use to identify scopes: Global identifiers (GUID, …) , relative identifiers (role name sequence number, local name, ..)

[3] In many type of specifications, business rules in a parent scope determine processing rules of child scopes. Dynamic composition of specification and the usage of business context such as in Core Component make it difficult to extract information from one source, one specification document in order to determine the final set of processing rules.

[4] In the future TPA, Contracts and technical agreements should be added as governing scopes when defined within UBAC project.

[5] It is also possible have a Role-Party as a scope type. Could be used to indicate role reversal.

[6] Business processes are important to organization but most business systems don’t keep track of them explicitly.

[7] Processing nodes between the sender and receiver may add and remove scopes at the lowest lever without disturbing higher level governing parent scopes. An example is a communication service that adds transport specific

2470 scopes before forwarding messages to lower lever transports and removes it
2471 when forwarding messages to upper lever business data receiver application.

Appendix D Relationship Between the SBDH and Other Standards

Cross-Section of Areas of Potential Interest between SBDH and other UN/CEFACT and ebXML Standards

		Boundaries	Integration Points	Dependencies
AS2	**Near-Term**	SBDH supplements AS2 technology	SBDH integrates only at the Communication Software Application level	AS2 utilizes the SBDH for routing
	Long-Term	SBDH will continue to supplement AS2 technology	SBDH will continue to integrate at the Communication Software Application level	AS2 will continue to utilize the SBDH for routing
ATG NDR	**Near-Term**	SBDH follows its own syntax Naming and Design Rules	**--**	**--**
	Long-Term	SBDH syntax will be subsumed by ATG Naming and Design Rules	**--**	SBDH is dependent upon the ATG Naming and Design Rules for interoperability
BPSS	**Near-Term**	SBDH will supplement BPSS technology	SBDH integrates at the Parser/Translator or Middleware level	BPSS is not dependent upon some generic header technology but may optionally use it
	Long-Term	SBDH will supplement or be subsumed by BPSS technology	SBDH will continue to integrate at the Parser/Translator or Middleware level	BPSS is not dependent upon some generic header technology but may optionally use it
ebMS	**Near-Term**	SBDH may supplement ebMS technology	SBDH integrates at the Communication Software Application level	ebMS is not dependent upon some generic header technology but may optionally use it
	Long-Term	SBDH may supplement ebMS technology	SBDH will integrate at the Communication Software Application level	

EDI	**Near-Term**	SBDH will supplement EDI technology	SBDH will integrate at the Parser/Translator or Middleware Application level	EDI is not dependent upon some generic header technology but may optionally use it, especially for Service Information and Correlation Information
	Long-Term	SBDH will supplement EDI technology	SBDH will continue to integrate at the Parser/Translator or Middleware Application level	EDI is not dependent upon some generic header technology but may optionally use it, especially for Service Information and Correlation Information
UBAC	**Near-Term**	SBDH will supplement or be subsumed by UBAC specifications	SBDH will integrate at the Business Transaction View and Business Service View levels	UBAC is dependent upon some generic header technology such as the SBDH
	Long-Term	To be determined	To be determined	If another technology becomes available, UBAC could use the new technology or the SBDH
UMM	**Near-Term**	SBDH will supplement UMM	SBDH will integrate at the Business Transaction View and Business Service View levels	SBDH is dependent upon the UMM meta-model
	Long-Term	To be determined	To be determined	SBDH continues to be is dependent upon the UMM meta-model