Chapter Three Research Approach

3.1 Research Model

Liang 1997 summarizes the MIS research methods, he states that MIS scholars held a series of conferences on research methods in 1989, and identified the five primary research methods including (1) case study (2) survey (3) experiment (4) model driven (5) prototyping.

After comparing the characteristics of these five methods, we realize case study and prototype are applicable to this research.

Yin 1994 suggests that case studies are a preferable strategy when “how” or “why” questions are being posed, when the investigator has little control over events, and when the focus is on a contemporary phenomenon within some real-life context. Besides, a case study is an empirical inquiry that investigates a contemporary phenomenon within its real-life context, especially when the boundaries between phenomenon and context are not clearly conspicuous. The case study inquiry relies on multiple sources of evidence.

According to Benbasat et al., 1987, they provide a list of the key characteristics of case studies:

1. Phenomenon is examined in a natural setting.
2. Data are collected by multiple means.
3. One or few entities (person, group, or organization) are examined.
4. The complexity of the unit is studied intensively.
5. Case studies are more suitable for the exploration, classification and hypothesis development stages of the knowledge building process; the investigator should have a receptive attitude towards exploration.
6. Case research is useful in the study of “why” and “how” questions because they deal with operational links to be traced over time rather than with frequency or incidence.

This research intends to investigate a contemporary phenomenon and “what” and “how” questions**.** It is an exploratory study having to choose sites to interview or observe directly, needing multiple sources of data to represent the comprehensive process model, and conforming to the characteristics and principles of case study. Therefore, we decide to use in-depth case study as an empirical inquiry method to observe, collect data, and analyze data.

As for the verification of processes, we analyze the result of the interviews with experts and examine secondary data in order to know whether these public processes are common and representative in this example industry, and revise the processes to become a common process model of an example industry.

Moreover, Liang suggests prototype is also a popular research method in MIS papers. When a creative idea is come up with, researchers will construct the prototype system to verify the feasibility and validity of this idea or theory. Therefore, this research use prototype, and apply the results of interviews to construct and implement a prototype system to verify the feasibility of process model.

In case study, according to Benbasat and Yin, the following components of a research design should be considered.

1. Study questions.
2. Study propositions.
3. Unit of analysis.
4. Single-case or multiple-case and site selection.
5. Data collection methods.
6. Data analysis and exposition.
7. Quality of research design.

Study questions have been identified without study propositions in the research. The following are detailed contents of research design.

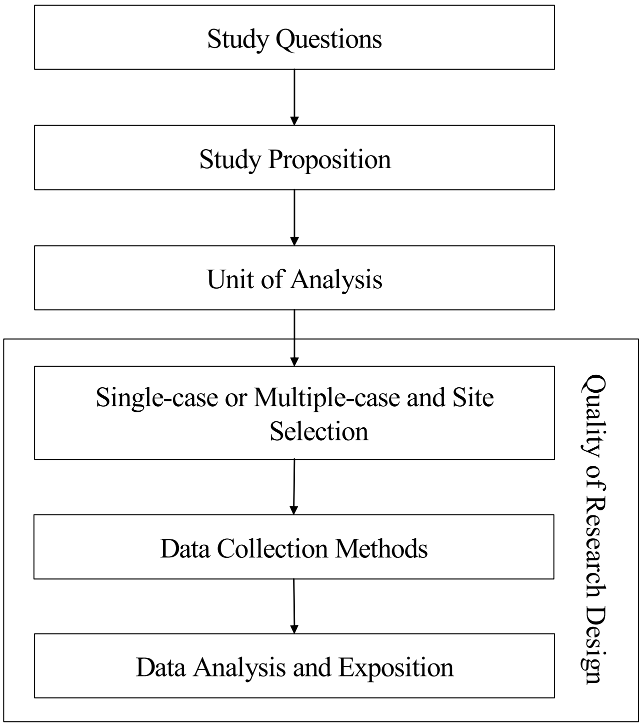

Study Questions
Study Proposition
Unit of Analysis
Single-case or Multiple-case and Site Selection
Data Collection Methods
Data Analysis and Exposition
Quality of Research Design

Figure 3. 1 Components of Research Model

1. Unit of analysis

The unit of analysis is related to the fundamental problem of defining what the ”case” is. Selecting appropriate unit of analysis results from your accurately specifying the primary research questions (Yin, 1994). When the case is an event, a project, an activity, or a series of related logic things, we should identify research scope as clearly as possible. Based on the objectives of this research, the unit of analysis is defined as business processes and models, including business entities, participant roles, transaction messages, and documents, et al.

2. Single-case or multiple-case and site selection

A single case study, as Yin supposes, is a revelatory case, and may be conducted as a prelude to further study, such as the use of case studies as exploratory devices. The logistics industry, for example, may have common but inaccessible commercial transaction models. Therefore, we decide to use single-case due to the limited time and resource.

Since the scope of business-to-business C-Commerce is rather enormous, we choose a company with multiple partners as the case in our study, and hope this research is a great contribution to the goal of global logistics center of the government. We decide to study global logistics industry thoroughly.

Business processes are complex in global logistics industry. There are a wide variety of participants such as shippers, carriers, consignees, forwarders, agents, etc, and complicated documents and information flows.

The Industrial Technology Department of The Ministry of Economic Affairs executes the C plan, D plan, and E plan just now, and integrates related companies to establish a global logistics and management e-alliance for the information technology (IT) industry. Our research chooses one of these global logistics companies as our in-depth case study company. The case study company was established in November 1999 as Taiwan's first comprehensive global logistics enterprise. The main businesses of the case study company can be categorized into the following divisions:

a. Order Management
b. Warehousing Management
c. Distribution and Consolidation
d. Cargo Management
e. One-Stop Documentation
f. Bundled Service

We hope to find the global operation situation, logistics strategies, logistics processes, and logistics management of the case study company, and go more deeply into its information and operation processes which carry out data exchange and transactions from ordering to goods arriving. By reviewing literatures and secondary data, we validate the collected data and try to identify and specify an appropriate operation and transaction model in this industry.

3. Data collection methods

Evidence for case study may come from documents, archival records, interviews, direct observation, participant-observation, and physical artifacts. Because of single-case, limited resource and time, and the pleasure of case study company, we decide to adopt interviews.

In the course of data collection, there is not any data source that can represent all facts or the whole phenomenon. Different data sources may be validation and complementary to one another. Better case studies rely on a wide variety of sources (Yin, 1994). Therefore, this research use interviews, documents, and internal data of the case study company as data sources. In order to thoroughly understand the case study company and draw up the study questions, we collect related secondary data which includes pertinent reports on a special topic from magazines or journals before interviewing. We interview the chief of the information department and related personnel to find the whole operational processes and collect internal data of the case study company. Finally, we organize all data and verify it. If the data conflicts, we ask related personnel again.

4. Data analysis and exposition

The quality of data analysis relies on integration ability of the researcher. The researcher must organize, analyze, and interpret finds that come from a variety of sources. Case study analysis has two analytic strategies: “relying on theoretical propositions”, “developing a case description” (Yin, 1994), as shown in Table 3.1. According to Yin, this research belongs to “developing a case description”. We use a case description as the way of conducting case study analysis.

Table 3. 1 Difference between Two Case Study Analysis Strategies

Case study analysis strategy	Characteristic	Goal
Relying on Theoretical Propositions	An example of a theoretical orientation guiding the case study analysis. Clearly, the proposition helps to focus on certain data and to ignore other data.	Theoretical propositions about causal relation—answers to “how” and “why” questions—can be very useful in guiding case study analysis in this manner.
Developing a Case Description	Developing a description framework for organizing the case study.	Intending to identify the attributes of a phenomenon entirely.

Data source: Yin, 1994

Since we intend to discover, analyze, and design business process model of the global logistics industry, we record the findings in informal flow diagrams and tables in detail in order to gather and organize the findings quickly and comprehensively. Regarding the representation of analysis results, we use UML and ebXML Worksheets as data analysis aids to fill out ebXML worksheet forms according to former informal flow charts and tables, which is to develop and model the process model and framework according to the ebXML business process standard.

The ebXML Business Process Worksheets are a set of business process design aids, used with the UMM as a reference. The worksheets are intended to be extensible to capture all the bits of information that the complete description of a business process requires so that it can be registered, classified, discovered, reused and completely drive the software. It will be more useful to the electronic business community to have an approach that does not require such analysis and modeling expertise.

5. The quality of research design

There are four tests to judge the quality of research design. They are "construct validity", "internal validity", "external validity", and "reliability" (Yin, 1994). Due to the objectives of this research and the usage of single-case, we emphasize construct validity, external validity, and reliability. We try to increase the quality of research design by means of the following ways. About construct validity, we use a variety of data sources such as interviews, documents, and internal data of a company, and have a draft case study report reviewed by key informants to avoid subjective notions and wrong interpretations. As for external validity, after finishing data collection and interviewing with domain experts, we also ask domain experts to scrutinize the analysis results of the case study and give some advice. If the research analysis results do not conform to the general process model of global logistics industry, we should correct it. About reliability, in order to identify exactly what data should be sought, we write down a list of case study questions before data collection. Table 3.2 is a summary of the above.

Table 3. 2 Case Study Tactics for the Design Tests

Tests	Description	Tactic
Construct validity	Knowing whether a case study investigator develops a sufficiently operational set of measures and that "subjective" judgments aren't used to collect the data.(avoiding "subjective" judgments when collection data)	During data collection, using multiple sources of data, and having the draft case study report reviewed by key informants.
External validity	Knowing whether a study's findings are generalizable beyond the immediate case study, and trying to establish a realm domain to which a study's findings can be generalized.	Using expert interviews, asking domain experts to go through the analysis of the case study and give some advice.
Reliability	If a later investigator follows exactly the same procedures as described by an earlier investigator and conducts the same case study all over again, the later investigators he should arrive at	Listing case study questions relevant to the research objectives.

	the same case over again.	

3.2 ebXML BPSS (Business Process Specification Schema)

3.2.1 BPSS Overview

The goal of the ebXML Business Process Specification Schema is to provide a bridge between e-business process modeling and the specification of e-business software components. The BPSS provides a standard framework for business process specification.

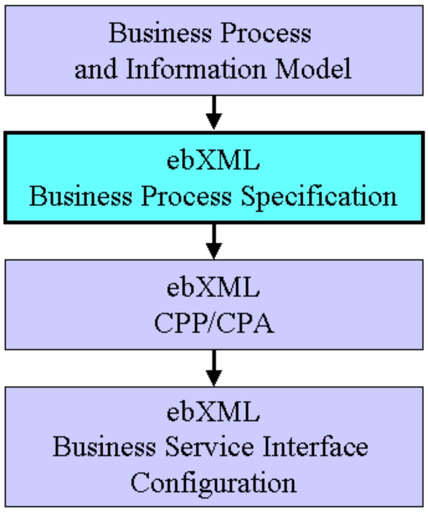


Figure 3. 2 Business Process Specification and Business Service Interface Configuration

Data source: ebXML Business Process Specification Schema

First, by using the UMM methodology and the UMM metamodel, the user may thus create a complete Business Process and Information Model. The UMM metamodel describes business semantics that allows trading partners to capture the details of a specific business scenario (a business process) through a consistent modeling methodology. The details of UMM methodology and UMM metamodel is discussed in the next sections.

Based on the Business Process and Information Model, and the ebXML BPSS**,** the user can then extract and format the nominal set of elements necessary to configure an ebXML runtime system in order to execute a set of ebXML business transactions. The result is an ebXML Business Process Specification. An ebXML Business Process Specification contains the specification of Business Transactions and the choreography of Business Transactions into Business Collaborations. This ebXML Business Process Specification is then the input to the formation of ebXML trading partner Collaboration Protocol Profiles (CPP) and Collaboration Protocol Agreements (CPA). Finally, the ebXML CPPs and CPAs in turn serve as configuration files for ebXML Business Service Interface software.

The architecture of the ebXML BPSS consists of the following functional components:

1. UML version of the BPSS
2. XML version of the BPSS
3. Production Rules defining the mapping from the UML version to the XML version of the BPSS.
4. Business Signal Definitions

These components combined together allow us to fully specify all the run time aspects of a business process model, which are shown (inside the dotted box) in Figure3.3 below.

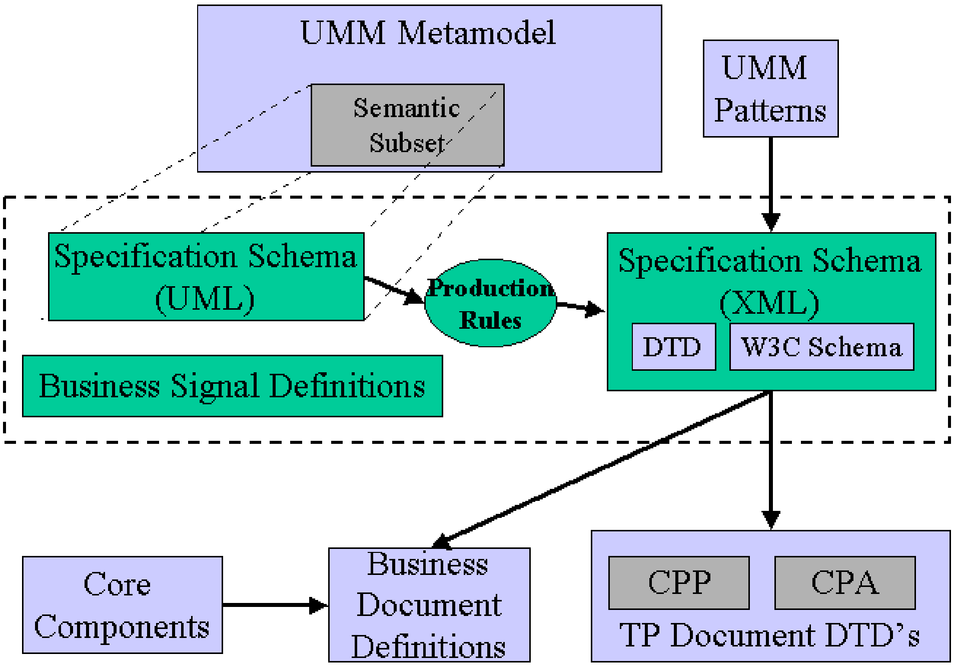

UMM Metamodel
Semantic Subset
UMM Patterns
Specification Schema (UML)
Production Rules
Specification Schema (XML)
DTD
W3C Schema
Business Signal Definitions
Core Components
Business Document Definitions
CPP
CPA
TP Document DTD's

Figure 3. 3 Relationship of ebXML Business Process Specification Schema to UMM, CPP/CPA and Core Components

Data source: ebXML Business Process Specification Schema

The ebXML BPSS is a semantic subset of the UMM metamodel. Each Business Transaction can be implemented when one of many available standard patterns is used. These patterns determine the actual exchange of messages and business signals between partners to achieve required electronic commerce transaction.

The UML version of the ebXML BPSS, a semantic subset of the UMM metamodel, is merely a UML Class Diagram. It is not intended for the direct creation of the ebXML BPSS. Rather, it is a self-contained statement of all the specification elements and relationships required to be able to create an ebXML compliant Business Process Specification.

The XML version of the ebXML BPSS provides the specification for XML-based instances of the ebXML Business Process Specifications and serves as a target for production rules from other representations. Thus, a user may either create a Business Process Specification directly as an XML document, or choose to use some other means of specification first and then apply production rules to become the XML document version.

A set of production rules define the mapping from the UML version to the XML version of the ebXML BPSS.

Business signals, application-level documents that ‘signal’ the current state of business transaction, have specific business purposes and are different from lower protocol and transport signals.

The Business Process Specification is therefore incorporated with or referenced by ebXML trading partner Collaboration Protocol Profiles (CPP) and Collaboration Protocol Agreements (CPA). Each CPP declares its support for one or more roles within the Business Process Specification. Within these CPP profiles and CPA agreements are then added further technical parameters resulting in a full specification of the run-time software at each trading partner.

The BPSS itself does not support the definition of Business Documents. Business Document Specifications may be based on the ebXML Core Components specifications.

Figure3.4 shows the semantics of a UML class diagram, which contains the whole UML version of the ebXML BPSS. All the specification elements in the UML version can be grouped as shown in Table 3.3.

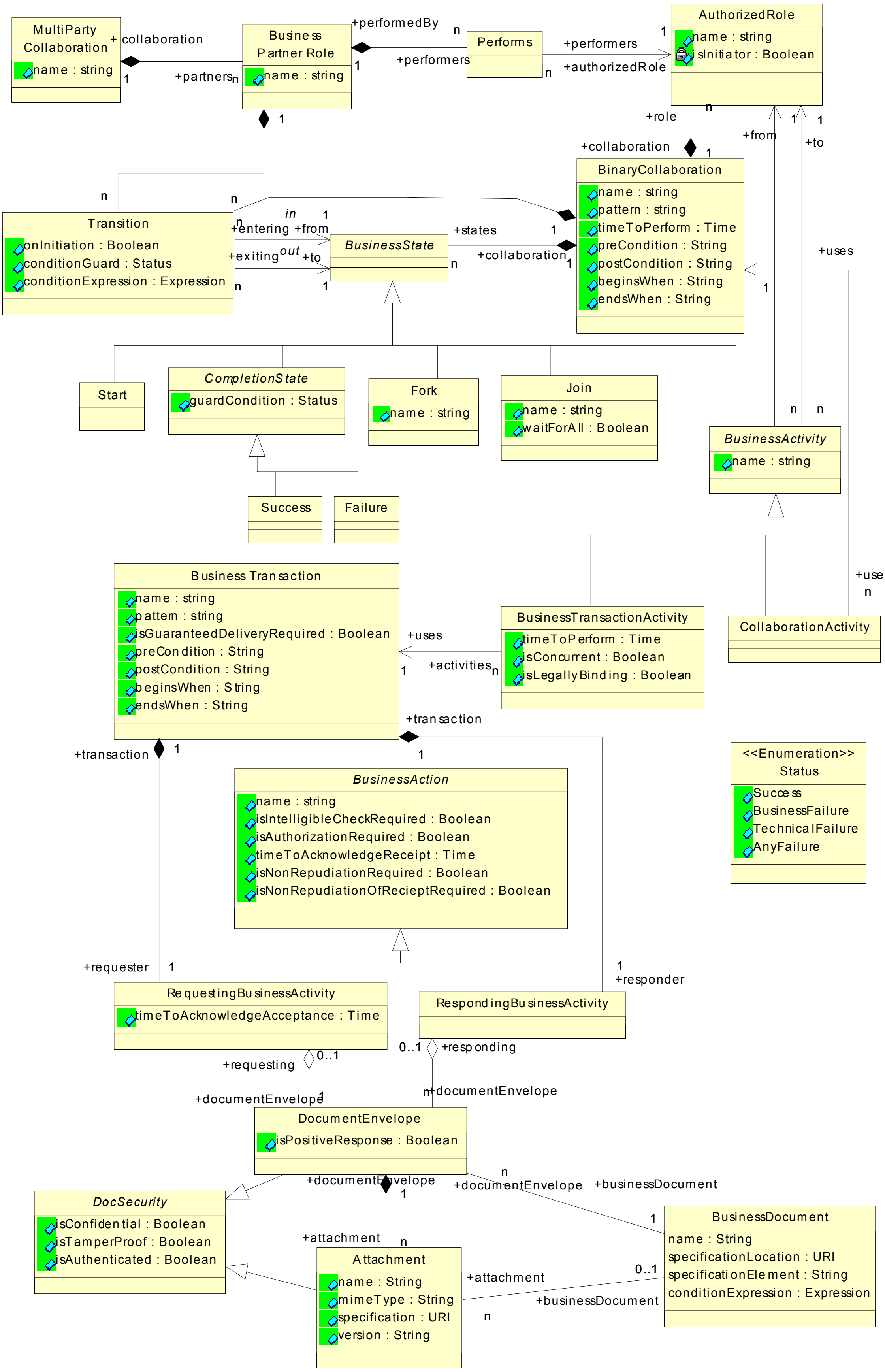


Figure 3. 4 Overall ebXML Business Process Specification Schema as UML Class Diagram

Data source: ebXML Business Process Specification Schema

Table 3. 3 Specification Groups in the UML Version

Group	Description
Business Collaborations	A business collaboration consists of a set of roles collaborating through a set of choreographed transactions by exchanging business documents. The ebXML BPSS supports two levels of business collaborations, “Binary Collaborations” and “Multiparty Collaborations”.
Business Transactions	A business transaction is an atomic unit of work in a trading arrangement between two business partners
Document flow	A business transaction is realized as business document flows, pertaining to the business transaction, between the requesting and responding roles.
Choreography	The business transaction choreography describes the order and transitions between business transactions or sub-collaborations within a binary collaboration. It can be done by using a UML activity diagram.

1. Business Collaborations

a. Multiparty Collaborations

Multiparty collaborations contain more than two roles, but are always synthesized from two or more binary collaborations. A multiparty collaboration consists of a number of business partner roles. Each business partner role performs one authorized role in one of the binary collaborations, or perhaps one authorized role in each of several binary collaborations. This is modeled by using the “Performs” element.

The “Performs” linkage between a business partner role and an authorized role is the synthesis from binary collaborations into multiparty collaborations. Implicitly the multiparty collaboration consists of all binary collaborations in which its business partner roles play authorized roles. Within a multiparty collaboration, you may choreograph transitions between business transaction activities in different binary collaborations.

b. Binary Collaborations

A binary collaboration is always implemented between two roles, which are called authorized roles because they represent the actors authorized to participate in the collaboration. A binary collaboration consists of one or more business activities, which are always conducted between two authorized roles of the binary collaboration. In each activity, one of the two roles is assigned to be the

InitiatingRole (from) and the other to be the RespondingRole (to).

A business activity can be either a business transaction activity or a collaboration activity. A business transaction activity is the performance of a business transaction. Business transactions are reusable relative to business transaction activity. The same business transaction can be performed by multiple business transaction activities in different binary collaborations, or even by multiple business transaction activities in the same binary collaboration.

A collaboration activity is the performance of a binary collaboration, possibly within another binary collaboration. Binary collaborations are reusable relative to collaboration activity. The same binary collaboration can be performed by multiple collaboration activities in different binary collaborations, or even by multiple collaboration activities in the same binary collaboration.

2. Business Transactions

A business transaction consists of a requesting business activity, a responding business activity, and one or two document flows between the two activities. A business transaction may be additionally supported by one or more business signals that govern the use and meaning of acknowledgements and related matters in the transaction.

Implicitly there is a requesting role performing the requesting business activity and a responding role performing the responding business activity. Both roles become explicit when the transaction is used within a business transaction activity in a binary collaboration.

There is always a request document flow. Whether a response document is required is part of the definition of the business transaction. Some business transactions need these types of request and response, typically for the formation of a contract or agreement. Other business transactions are more like notifications, and have only a request document flow. An abstract superclass, a business action, is the holder of attributes common to both requesting business activity and responding business activity.

3. Document Flow

A document flow is not modeled directly. Rather it is modeled indirectly as a document envelope sent by one role and received by the other. The document envelope is always associated with one requesting business activity and one responding business activity to model the flow.

Document envelopes are named. A requesting activity always has only one named document envelope. And a responding activity may have zero, one, or more mutually exclusive named document envelopes. The document envelope represents the document flow between the activities. Each document envelope carries exactly one primary business document.

A document envelope can optionally have one or more attachments, all related to the primary business document. The document and its attachments in essence form one transaction in the payload in the ebXML Message Service message structure.

4. Choreography

The choreography is specified in terms of business states, and transitions between those business states. A business activity is an abstract kind of business state. Its two subtypes, business transaction activities and collaboration activities are concrete business states. The purpose of choreography is to order and sequence business transaction activities and/or collaboration activities.

There are a number of auxiliary business states that facilitate the choreography of business activities, including a start state, a completion state (which comes in a success and failure flavor), a fork state and a synchronization state. These are all equivalent to diagramming artifacts on a UML activity chart.

Transitions conducted between business states, can be gated by guards. Guards can refer to the status of the document envelope that causes the transition, the type of the document sent, the content of the document, or postconditions in the prior state.

3.2.2 UMM Methodology and UMM Metamodel

The UN/CEFACT Modeling Methodology (UMM) is a methodology for business process and information modeling. The UMM business process and information modeling technique is based on the Unified Modeling Language (UML) from the Open Management Group (OMG). The UMM is based on configuring the Unified Process methodology developed by the Rational Corporation to meet UN/CEFACT needs for modeling business processes in addition to objects.

Focusing on technologically and protocol independent steps of the normal software engineering process, the UMM metamodel facilitates the specification of reusable, reproducible process models that are technologically and protocol insensitive.

The focus of the UMM developed by UN/CEFACT is predominately on the technologically neutral intersection of Inception and Elaboration of the UP (Unified Process) phases and the Software Engineering project workflows of Business Modeling, Requirements, Analysis and Design as shown in Figure 3.5.

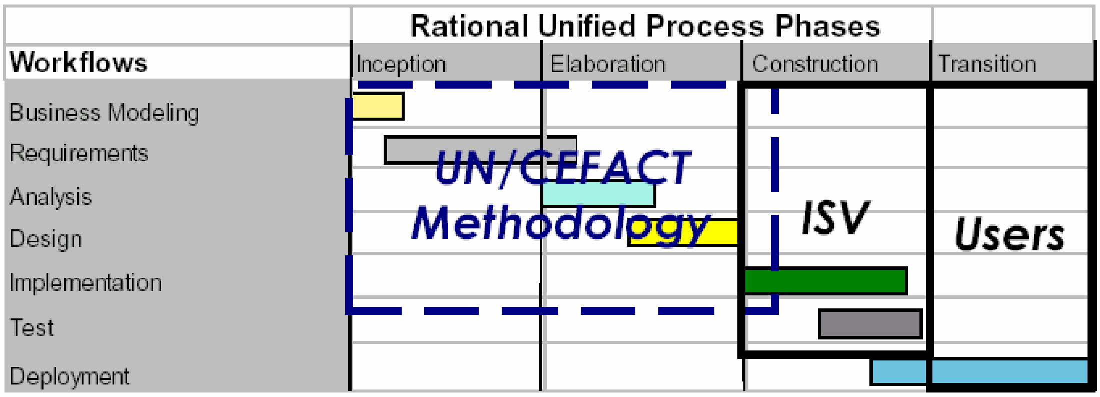


Figure 3. 5 Phases and Workflows

Data source: UN/CEFACT Modeling Methodology, 2001

The UMM metamodel supports a set of business process viewpoints providing a set of semantics (vocabulary) for each viewpoint, and forms the basis of specification of the semantics and artifacts required to facilitate business process and information integration and interoperability.

The UMM metamodel is based on a precise definition of the UML metamodel extension that facilitates the expression of business processes as an object-oriented model. It is organized into the following views so that each business process and information model can be viewed from a number of perspectives.

1. The Business Operations Map (BOM) metamodel: corresponding to the business modeling workflow, the BOM is an organizational view of the Business Areas and Process Areas as defined by domain taxonomy.

2. The Business Requirements View (BRV) metamodel: corresponding to the requirements workflow, a view of a business process model captures the use case scenarios, inputs, outputs, constraints and system boundaries for business transactions and their interrelationships.

3. The Business Transaction View (BTV) metamodel: corresponding to the analysis workflow, the view of a business process model captures the semantics of business information entities and their flow of exchange between roles as they perform business activities.

4. The Business Service View (BSV) metamodel: corresponding to the design workflow, the view of a business process model specifies the network component services and agents and their message (information) exchange as interactions are necessary to execute and validate a business process.

The Business Operations Map, Business Requirements View, Business Transaction View and Business Service View are produced as a result of transforming business process requirements into an object-oriented business process model. According to these object-oriented business process models several worksheets can help users to capture all bits of information required to completely describe a business process. The details of the worksheets are described in the next sections.

3.2.3 UMM Transaction Pattern

Patterns are reusable, generalized business process abstractions that can be applied to many domains. Patterns are applications of the metamodel to common business process and information representations. While patterns can be expressed for business processes at various levels, the UMM currently includes patterns for business transaction activities and their associated service collaborations. The following six property-value conventions for business transactions have been proven useful in the application of the metamodel to existing business requirements. These conventions are applied by stereotyping the requesting business activity with the syntax shown in Table 3.4.

Table 3. 4 Business Transaction Stereotypes

Business Transaction	Stereotype
Commercial Transaction	<<CommercialTransactionActivity>>
Request / Confirm	<<RequestConfirmActivity>>
Query / Response	<<QueryResponseActivity>>
Request / Response	<<RequestResponseActivity>>
Notification	<<NotificationActivity>>
Information Distribution	<<InformationDistributionActivity>>

Data source: UN/CEFACT Modeling Methodology, 2001

1. Commercial Transaction Pattern (Contract formation, e.g., place order)

This design pattern is best used to model the "offer and acceptance" business transaction process that results in a residual obligation between both parties to fulfill the terms of the contract. The pattern specifies an originating business activity sending a business document to a responding business activity that may return a business signal or business document as the last responding message. Figure 3.6 illustrates the commercial transaction design pattern. The intent of this business transaction pattern is to model the formation of an offer and acceptance business contract.

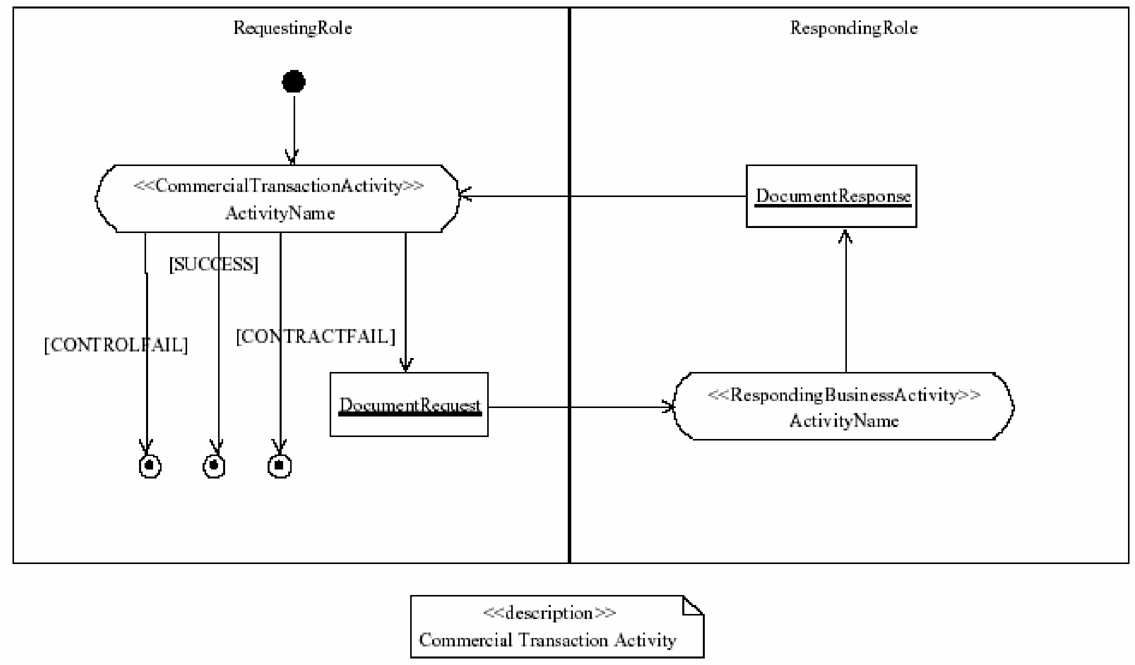


Figure 3. 6 Commercial Transaction Activity Design Pattern

Data source: UN/CEFACT Modeling Methodology, 2001

2. Query/Response Pattern (Static information, e.g., obtain catalog)

It is best used to query for information that a responding partner already has, for example, a fixed data set that resides in a database. Figure 3.7 illustrates the query/response design pattern, which specifies one business document as output and one business document as input.

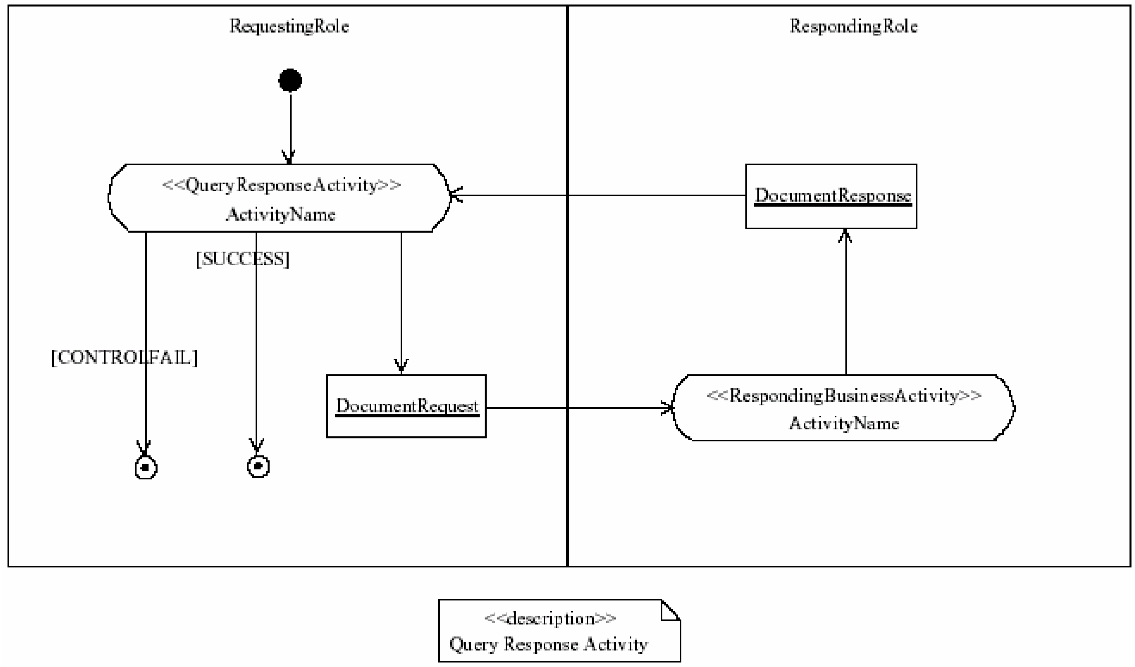


Figure 3. 7 Query/Response Activity Design Pattern

Data source: UN/CEFACT Modeling Methodology, 2001

3. Request/Response (Dynamic information, e.g., obtain Buyer ID, obtain quote)

It is best used for business contracts when an initiating partner requests information that a responding partner already has and when the request for business information requires a complex interdependent set of results. Namely, this response requires some business processing on a query before a response is returned to the requestor. This pattern specifies the exchange of a requesting and responding business document. Figure 3.8 illustrates the request/response design pattern.

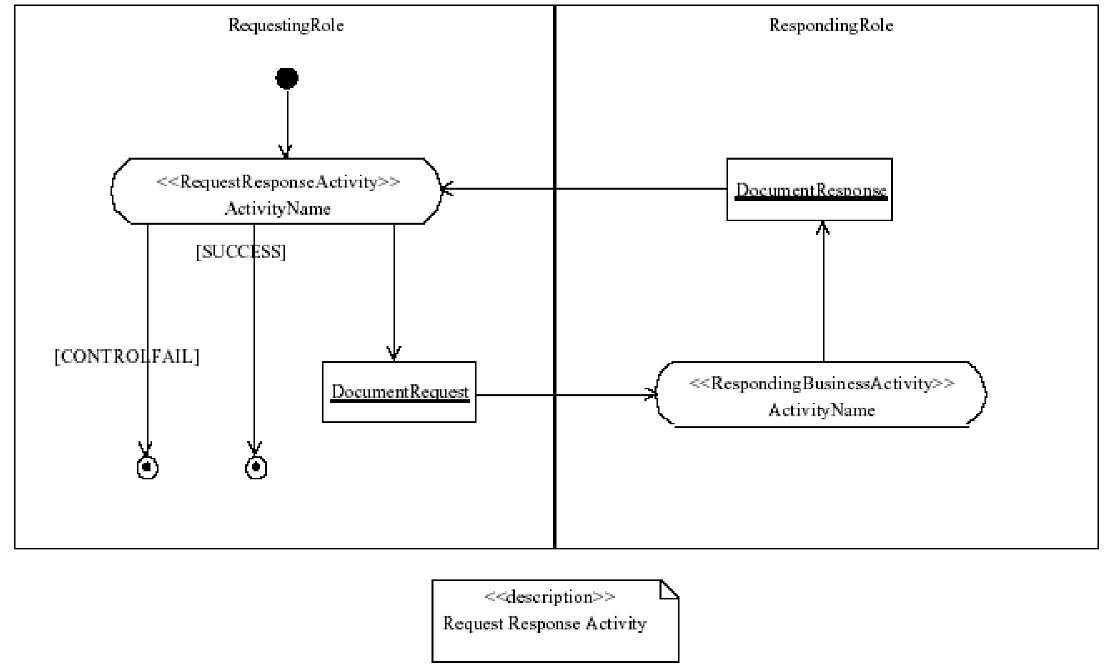


Figure 3. 8 Request/Response Activity Design Pattern

Data source: UN/CEFACT Modeling Methodology, 2001

4. Request/Confirm Pattern (Status information, e.g., obtain order status)

It is used for business contracts where an initiating partner requests confirmation about their status with respect to previously established contracts or with respect to business rules of a responding partner.

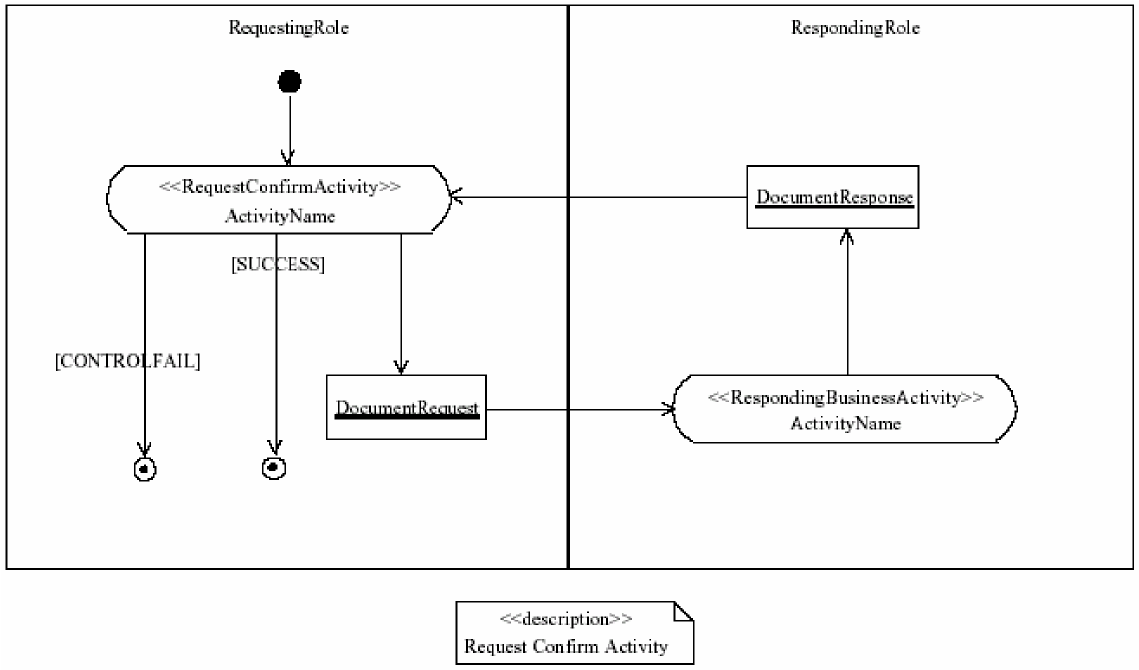


Figure 3. 9 Request/Confirm Activity Design Pattern

Data source: UN/CEFACT Modeling Methodology, 2001

5. Information Distribution Pattern

It is used to model an informal information exchange of business transaction that therefore has no non-repudiation requirements. This pattern specifies the exchange of a requesting business document and the return of an acknowledgement of receipt business signal. Figure 3.10 illustrates the information distribution design pattern.

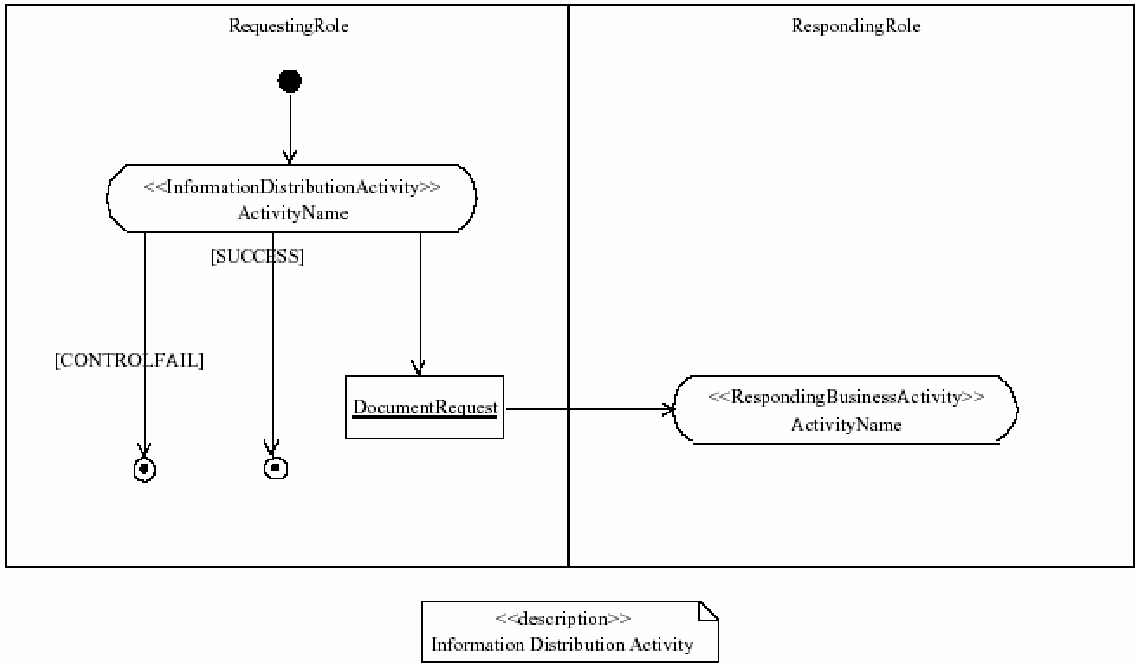


Figure 3. 10 Information Distribution Design Pattern

Data source: UN/CEFACT Modeling Methodology, 2001

6. Notification Pattern

It is used to model a formal information exchange business transaction that therefore has non-repudiation requirements. This pattern specifies the exchange of a notifying business document and the return of an acknowledgement of receipt business signal. Figure 3.11 illustrates the notification design pattern.

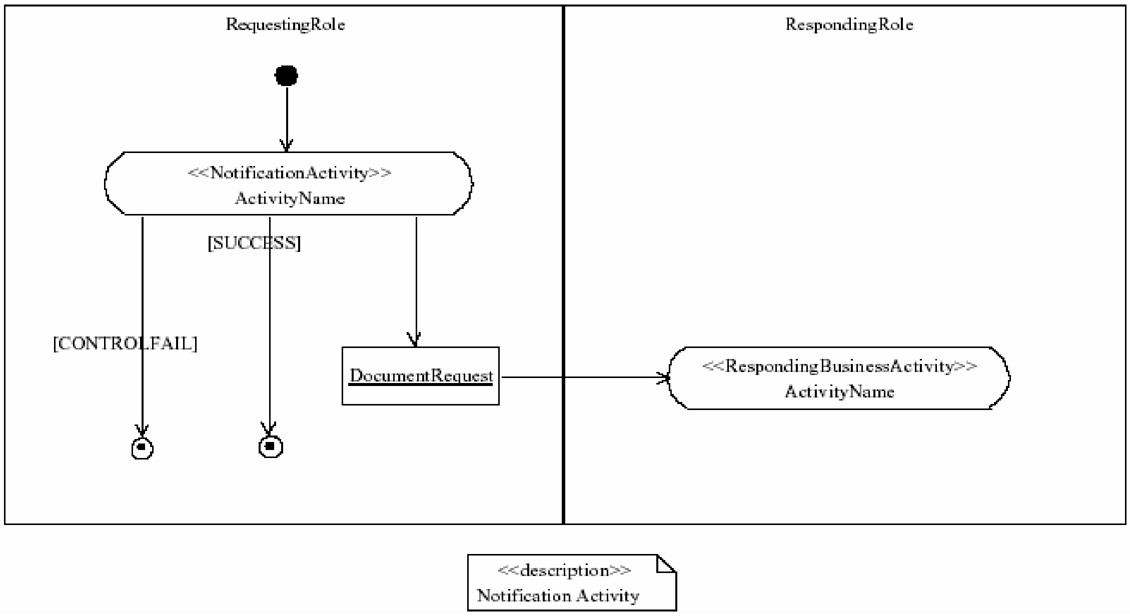
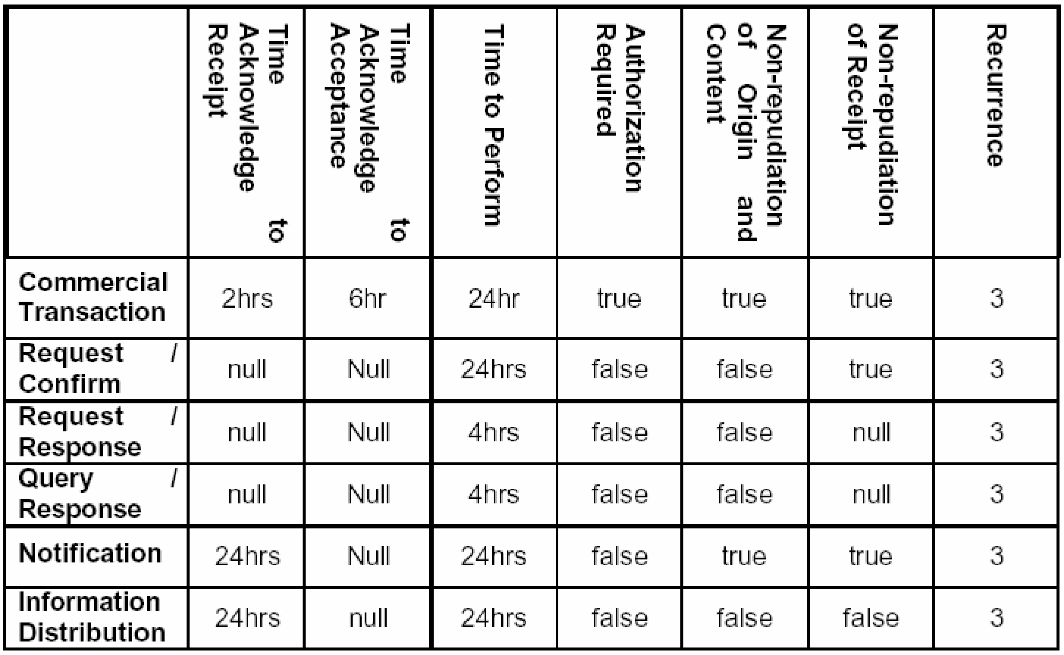


Figure 3. 11 Notification Design Pattern

Data source: UN/CEFACT Modeling Methodology, 2001

The following tables specify the property-values of requesting and responding business activities for each of the business transaction stereotypes.

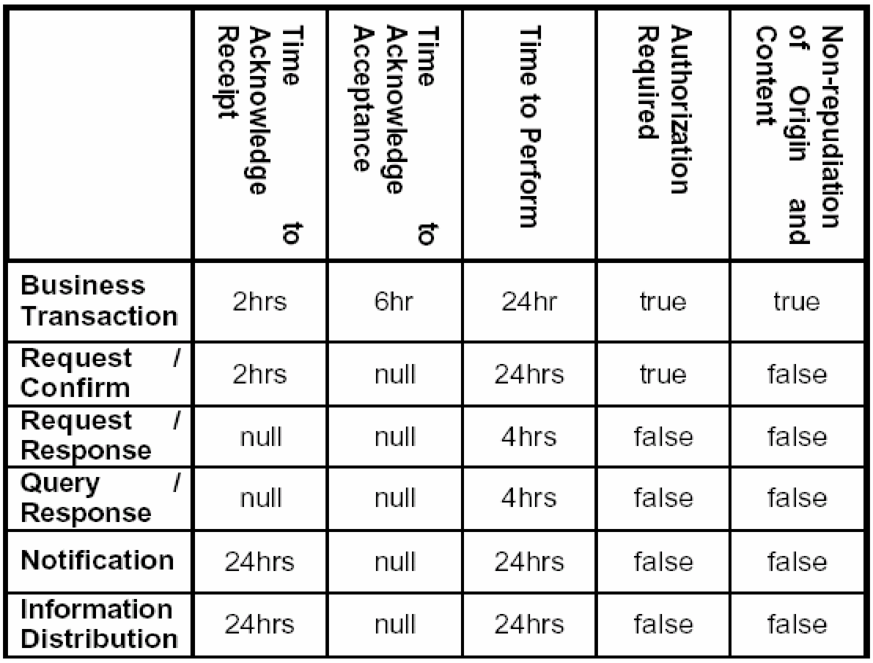
Table 3. 5 Requesting Business Activity Property-values



	Time to Acknowledge Receipt	Time to Acknowledge Acceptance	Time to Perform	Authorization Required	Non-repudiation of Origin and Content	Non-repudiation of Receipt	Recurrence
Commercial Transaction	2hrs	6hr	24hr	true	true	true	3
Request / Confirm	null	Null	24hrs	false	false	true	3
Request / Response	null	Null	4hrs	false	false	null	3
Query / Response	null	Null	4hrs	false	false	null	3
Notification	24hrs	Null	24hrs	false	true	true	3
Information Distribution	24hrs	null	24hrs	false	false	false	3

Data source: UN/CEFACT Modeling Methodology, 2001

Table 3. 6 Responding Business Activity Property-values



	Time to Acknowledge Receipt	Time to Acknowledge Acceptance	Time to Perform	Authorization Required	Non-repudiation of Origin and Content
Business Transaction	2hrs	6hr	24hr	true	true
Request / Confirm	2hrs	null	24hrs	true	false
Request / Response	null	null	4hrs	false	false
Query / Response	null	null	4hrs	false	false
Notification	24hrs	null	24hrs	false	false
Information Distribution	24hrs	null	24hrs	false	false

Data source: UN/CEFACT Modeling Methodology, 2001

3.3 ebXML Business Process Analysis Worksheets

The ebXML Business Process Analysis Worksheets are a set of business process design aids, used with the UMM as a reference, the worksheet-based approach provides an easier way of applying the UMM and the UMM metamodel. To develop company business processes for an ebXML compliant electronic trading relationship, we use the UMM as a reference guideline plus the ebXML Business Process Analysis Worksheet to create the necessary business process models. The following diagram sketches out a more detailed mapping from the Worksheets Model to the metamodel defined by the UMM as shown in Figure 3.12. The leftmost column is the selection of the main elements the Worksheets need to specify or edit. The rightmost column shows significant metamodel elements. The middle column is the other elements that are part of the Worksheets.

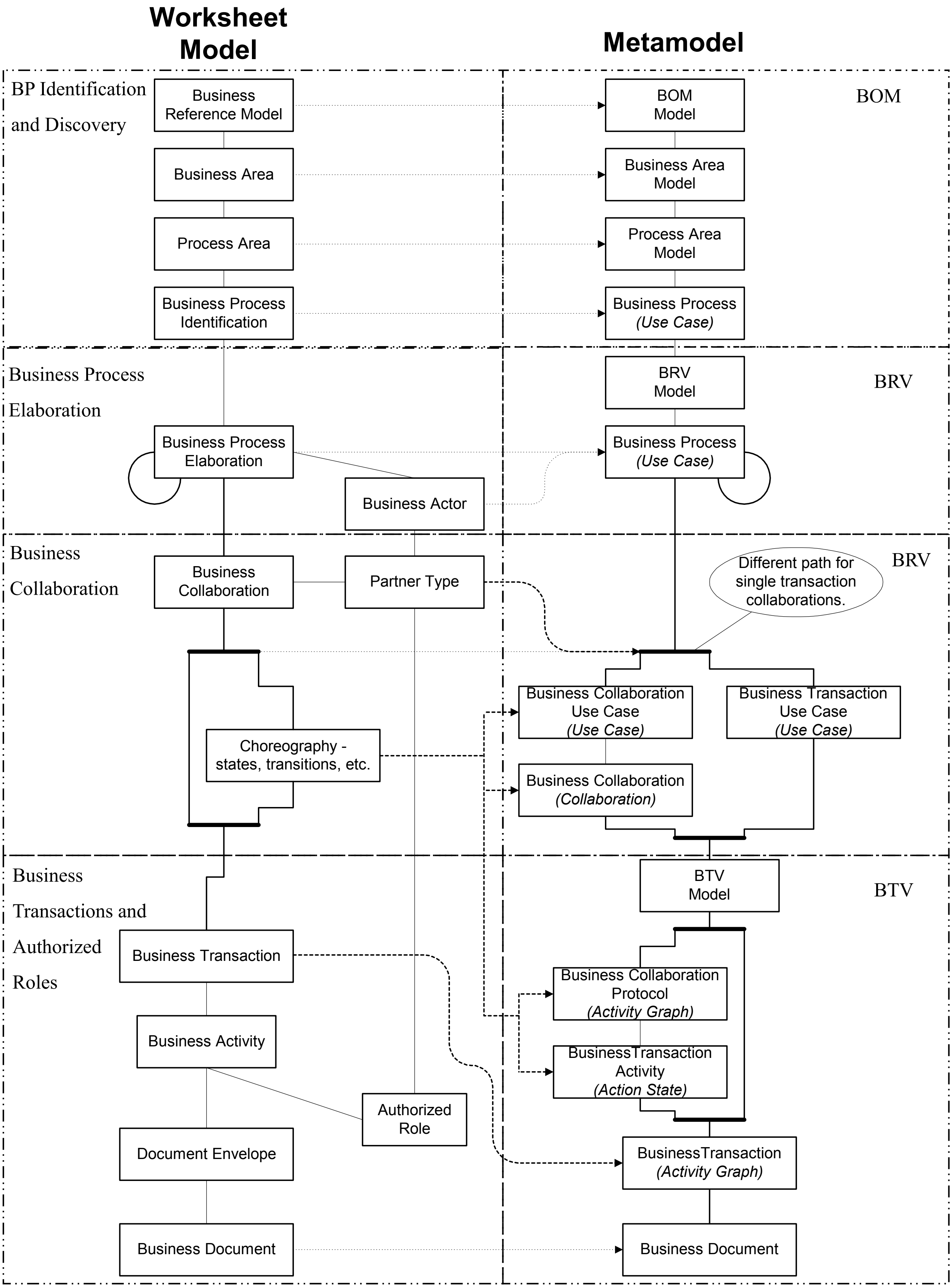

Worksheet Model
Metamodel
BP Identification and Discovery
Business Reference Model
Business Area
Process Area
Business Process Identification
BOM Model
Business Area Model
Process Area Model
Business Process (Use Case)
BOM
Business Process Elaboration
Business Process Elaboration
Business Actor
BRV Model
Business Process (Use Case)
BRV
Business Collaboration
Business Collaboration
Partner Type
Different path for single transaction collaborations.
BRV
Choreography - states, transitions, etc.
Business Collaboration Use Case (Use Case)
Business Transaction Use Case (Use Case)
Business Collaboration (Collaboration)
Business Transactions and Authorized Roles
BTV Model
BTV
Business Transaction
Business Activity
Authorized Role
Document Envelope
Business Document
Business Collaboration Protocol (Activity Graph)
BusinessTransaction Activity (Action State)
BusinessTransaction (Activity Graph)
Business Document

Figure 3. 12 Worksheets to Metamodel Mapping

Data source: Business Process Analysis Worksheets & Guidelines

3.3.1 Business Process Identification and Discovery

The first set of worksheets helps the user begin formalizing the domain they are trying to model processes in.

1. Business Reference Model: to define a "frame of reference" for business processes, and this frame of reference might define basic terms accepted by the given industry segment. UML packages can be created to categorize key concepts and business models.

Table 3. 7 Business Reference Model

Form: Describe Business Reference Model	
Form ID	[Provide an ID for this form so other forms can reference it.]
Business Reference Model Name	[Provide a name for the reference model. You can use an existing reference model such as the Supply Chain Council or the Porter's Value Chain or creates your own name.]
Industry Segment	[Provide the name of the industry segment that this business applies to. Search the business process library for a list of possible industry segments. If the industry segment does not exist, then provide an appropriate name/label for the industry segment.]
Domain Scope	[Provide a high level statement that encapsulates the scope of all the business areas.]
Business Areas	[List the business areas within the scope. A business area is a collection of process areas. A process area is a collection of business processes. You may wish to refer to the ebXML Catalog of Business Processes that provides a list of normative categories that may be used as business areas.]
Optional for ebXML	
Business Justification	[Provide the business justification for the collection of business processes.]

Data source: Business Process Analysis Worksheets & Guidelines

2. Business Area: to group business processes according to the primary business function. UML packages can be created to categorize key concepts and business models.

Table 3. 8 Business Area

Form: Describe Business Area	
Form ID	[Provide an ID for this form so other forms can reference it.]
Business Area Name	[Provide a name for the business area. This should be listed in the Business Areas section of at least one Business Reference Model.]
Description	[A brief summary of this functional area.]

Scope	[Provide a high level statement that encapsulates the scope of all the business areas. The scope of the business area must be within the scope of the encompassing business reference model. Typically the scope of the business area will be more constrained or limited than the scope of the business reference model.]
Boundary of the Business Area	[Describe the boundary of the business area. This defines the entities that interact in this business area; actors, organizations, possibly systems]
References	[Any external supporting documentation.]
Constraints	[Identify any constraints on the process areas (and, thus, business processes) within this business area.]
Stakeholders	[Identify the practitioners that care about the definition of this business area. At this level, this is likely to be some participants in an industry group (perhaps a standards body or an enterprise). These are the people who will define the BRV.]
Process Areas	[List the process areas within the scope. A process area is a collection of business processes. You may wish to refer to the ebXML Catalog of Business Processes that provides a list of normative process groups that may be used as process areas.]
Optional for ebXML	
Objective	[Describe the objective of this business area.]
Business Opportunity	[Describe the business opportunity addressed by this business area.]

Data source: Business Process Analysis Worksheets & Guidelines

3. Process Area: a process area consists of a sequence of processes that are combined to form the "value chain" of the given business area. UML use case diagram can be created to visualize each process area.

Table 3. 9 Process Area

Form: Describe Process Area	
Form ID	[Provide an ID for this form so other forms can reference it.]
Process Area Name	[Provide a name for the process area. This should be listed in the Process Areas section of at least one Business Area.]
Objective	[Describe the objective of this process area.]
Scope	[Provide a high level statement that encapsulates the scope of all the business areas. The scope of the business area must be within the scope of the encompassing business reference model. Typically the scope of the process area will be more constrained or limited than the scope of the corresponding business area.]
References	[External supporting documentation.]
Boundary of the Process Area	[Describe the boundary of the process area. The communicating services.]

Constraints	[Identify any constraints on the business processes within this process area.]
Stakeholders	[Identify the practitioners involved in this process area.]
Business Processes	[List the business processes within the scope of this process area. You may wish to refer to the ebXML Catalog of Business Processes that provides a normative list of business processes.]
Optional for ebXML	
Business Opportunity	[Describe the business opportunity addressed by this process area.]

Data Source: Business Process Analysis Worksheets & Guidelines

3.3.2 Business Process Elaboration

At this stage we begin to move from requirements analysis to design analysis. A business process is a use case used to gather requirements about business processes. Inputs to the business process must be specified in the pre-conditions and outputs from the business process must be specified in the post-conditions.

Table 3. 10 Business Process Use Case

Form: Business Process Use Case	
Form ID	[Provide an ID for this form so other forms can reference it.]
Business Process Name	[Provide a name for the business process. This should be a name identified on the form "Identify Business Process" and on a "Describe Process Area" form. If you are starting with this form, you may wish to refer to the ebXML Catalog of Business Processes that provides a normative list of business processes.]
Identifier	[This is a unique identifier that follows the Business Process Identifier Naming Scheme. This can be provided when the business process description is submitted to a business process library.]
Actors	[List the actors involved in the use case.]
Performance Goals	[A specification of the metrics relevant to the use case and a definition of their goals. Non-functional requirements may be a source of performance goals. For each performance goal, provide a name of the performance goal and a brief description of the performance goal.]
Preconditions	[Preconditions are constraints that must be satisfied starting the use case.]
Begins When	[Describe the initial event from the actor that starts a use case.]
Definition	[A set of simple sentences that state the actions performed as part of the use case. Include references to use cases at extension points.]
Ends When	[Describe the condition or event that causes normal completion of the use case.]

Exceptions	[List all exception conditions that will cause the use case to terminate before its normal completion.]
Postconditions	[Post-conditions are states that must be satisfied ending the use case.]
Traceability	[These are the requirements covered.]

Data source: Business Process Analysis Worksheets & Guidelines

3.3.3 Business Collaboration

To capture the detailed user requirements specified by the stakeholders for the business-to-business project, this workflow develops the Business Requirements View (BRV) of a process model that specifies the use case scenarios, input and output triggers, constraints and system boundaries for business transactions (BTs), business collaboration protocols (BCPs) and their interrelationships. It may make sense to use UML activity diagram to convey some of this information.

Table 3. 11 Business Process Use Case

Form: Business Collaboration	
Form ID	[Provide an ID for this form so other forms can reference it.]
Identifier	[This is a unique identifier that follows the Business Process Identifier Naming Scheme. This can be provided when the business process description is submitted to a business process library.]
Description	[Provide a descriptive overview of the collaboration.]
Partner Types	[This is a list of entities that participate in the collaboration. These participants exchange the events that form the collaboration.]
Authorized Roles	[These are the roles that a partner must be authorized to play to issue specific transactions in the collaboration (by sending certain signals).]
Legal Steps/Requirements	[If any step in the collaboration has any legal standing, it should be captured here.]
Economic Consequences	[If any step in the collaboration has and economic consequence, it should be captured here.]
Initial/Terminal Events	[List the events that initiate this collaboration and how it terminates.]
Scope	[Specify the set of business actions this collaboration encapsulates.]
Boundary	[Specify the systems and users that communicate with each other over he course of this collaboration.]
Constraints	[Spell out any special constraints that are relevant to this collaboration (e.g. business scenario, pre-conditions.)]

Data source: Business Process Analysis Worksheets & Guidelines

Table 3. 12 Business Collaboration Protocol

				Form: Business Collaboration Protocol Table
Form Id	[Provide an ID for this form so other forms can reference it.]			
Identifier	[Enter the Identifier from the associated Business Collaboration form.			
From Business Activity (Transaction)	**Initiating Partner Type**	**To Business Activity**	**Responding/ Receiving Partner Type**	**Transition Condition**
[START for the first activity or the name of originating business activity.]	[Partner type name or NOT-APPLICABLE.]	[Name of destination business activity.]	[Partner type name or NOT-APPLICABLE.]	[A boolean expression defining or describing the condition for the transition or NONE.]
[Name of an activity.]	NOT-APPLICABLE	SUCCESS	NOT-APPLICABLE	[A boolean expression defining or describing the condition for the transition.]
[Name of an activity.]	NOT-APPLICABLE	FAILURE	NOT-APPLICABLE	[A boolean expression defining or describing the condition for the transition.]

Data source: Business Process Analysis Worksheets & Guidelines

3.3.4 Business Transactions and Authorized Roles

The goal of the worksheet is to identify the individual transactions that implement the workflow of a Business Collaboration. A transaction is made up of several activities. Each activity has an authorized role that the signaler must have in order to initiate that activity. The modeling artifacts generated as a result of the worksheet is the Business Transaction Activity Diagram.

Table 3. 13 Business Transactions

Form: Business Transaction	
Form ID	[Provide an ID for this form so other forms can reference it.]
Description	[Provide a descriptive overview of this transaction.]
Pattern	[If you have chosen to follow one of the canonical transaction patterns in the UMM (or elsewhere) denote it here. If not and you have special

	semantics (as mentioned above), describe them here.]
Business activities and associated authorized roles	[List each activity (along with its initiator) and the role required to perform that activity]
Constraints	[Any constraints should be listed here.]
Initiating/Requesting Partner Type	[Partner type from collaboration.]
Initiating/Requesting Activity Role	[These are the roles that a partner must be authorized to play to issue specific transitions in the transaction (by sending certain signals).]
Initiating/Requesting Activity Document	[Document initiating the transaction. Might reference a standard document (e.g. an X12 document).]
Responding Partner Type	[See above.]
Responding Activity Role	[See above.]
Responding Activity Document	[See above.]

Data source: Business Process Analysis Worksheets & Guidelines

Complete the following property values if they differ from the defaults defined in the UMM transaction patterns for requesting and responding business activities.

Table 3. 14 Business Transaction Property Values

Form: Business Transaction Property Values							
Form Id	[Provide an ID for this form so other forms can reference it.]						
	Time to Acknowledge Receipt	**Time to Acknowledge Acceptance**	**Time to Perform**	**Authorization Required**	**Non-repudiation of Origin and Content**	**Non-Repudiation of Receipt**	**Recurrence**
Requesting Business Activity	[time]	[time]	[time]	[true or false]	[true or false]	[true or false]	[whole number]
Responding Business Activity	[time]	[time]	[time]	[true or false]	[true or false]	NOT-APPLICABLE	NOT-APPLICABLE

Data source: Business Process Analysis Worksheets & Guidelines

Provide a Business Transaction Transition Table if needed. An UMM compliant activity diagram (UML) can be created or a Business Transaction Transition Table can be used to convey the same information.

Table 3. 15 Business Transaction Transition

					Form: Business Transaction Transition Table
Form Id	[Provide an ID for this form so other forms can reference it.]				
From Activity	**From Role**	**Document**	**To Activity**	**To Role**	**Guard Condition**
[Name of the “from” activity. The keyword START shall be used for the first activity.]	[A Requesting/Initiating Activity Role or NOT-APPLICABLE. NOT-APPLICABLE is to be used when the From Activity is START.]	[Document name or NONE.]	[Name of the destination activity or keyword END or keyword CONTROL-FAILED.]	[A Responding Activity Role or NOT-APPLICABLE.]	[A boolean expression defining or describing the condition for the transition or NONE.]
[Name of the last activity before the END state]	[Appropriate role name.]	NONE	END	NOT-APPLICABLE	[Expression of the guard condition.]
[Name of the last activity before the CONTROL-FAILED state.]	[Appropriate role name.]	NONE	CONTROL-FAILED	NOT-APPLICABLE	[Expression of the guard condition.]

Data source: Business Process Analysis Worksheets & Guidelines

3.3.5 Business Information Description

The goal of this set of worksheets is to identify the information requirements for the business documents specified in the business transactions.

1. Business Information Context: the Business Information Context form is provided as convenience for aggregating contextual values that affect the analysis of business information. It is intended that the information is obtained from other forms. For example, Industry Segment is specified in the Business Reference Model form. If there is no appropriate value for an entry, enter NOT-APPLICABLE or NONE whichever is appropriate.

Table 3. 16 Business Information Context

	Form: Business Information Context
Form Id:	[Provide an ID for this form so other forms can reference it.]
Industry Segment	
Business Process	
Product	
Physical Geography /Conditions /Region	
Geo-Political Legislative/ Regulatory/ Cultural	
Application Processing	
Business Purpose /Domain	
Partner Role	
Service Level (profiles – not preferences.)	
Contracts/Agreements	

Data source: Business Process Analysis Worksheets & Guidelines

2. Document Content Description: It describes each element or groups of elements in the document. Logically related elements can be placed in separate forms. For example, a document may have logically three parts, a header, body which may have further logical partitioning, and summary. Possible values for Occurs include: 1 (one instance), 0..1 (zero on one instance), 0..* (zero or more instances), 1..* (one or more instances), or n..m (n to m instances and n is less than m). Information "looping" is specified through appropriate Occurs values. Possible values for Data Type include primitive data types – such as integer, string, and date-type – or a Form Id of another Content Description Form.

Table 3. 17 Content Description

Form: Content Description					
Form Id:	[Provide an ID for this form so other forms can reference it.]				
Element/Component Name	**Occurs**	**Data Type**	**Field Width**	**Semantic Description**	**Notes**
[Provide a name for the element/component. For example, "Order Summary" or "Issued Date."]					

Data source: Business Process Analysis Worksheets & Guidelines

3. Content Mapping: This information is very important as it shows that the documents have a basis in existing standards. Furthermore, the information will be used to create document transformations.

Table 3. 18 Content Mapping

Form: Content Mapping		
Form Id:	[Provide an ID for this form so other forms can reference it.]	
Content Description Form Id	[Provide the identifier of the associated Content Description form.]	
Standard	[Name of the standard. For example, UN/EDIFACT]	
Version	[Standard version number. For example, D.01A]	
Element/Component Name	**Mapping/Transformation**	**Note**
[Enter element/component name from corresponding Content Description form]	[Mapping or transformation. If the element/component is a complex structure, this entry should reference the appropriate Content Mapping form.]	[Any useful mapping notes.]

Data source: Business Process Analysis Worksheets & Guidelines

3.4 eXtensible Markup Language (XML)

3.4.1 SGML, XML, and HTML

eXtensible Markup Language (XML) was developed by an XML Working Group (originally known as the Standard Generalized Markup Language (SGML) Editorial Review Board) formed under the auspices of the World Wide Web Consortium (W3C) in 1996. XML is an application profile or restricted form of SGML, the Standard Generalized Markup Language (W3C, October 2000).

SGML forms the basis for HyperText Markup Language (HTML), XML, SGML is a very sophisticated metalanguage designed for large and complex documentation. However, full SGML contains many optional features that are not needed for Web applications and have been proven to have a cost/benefit ratio unattractive to current vendors of Web browsers (Bosak, 1997).

HTML, on the other hand, is a specific markup language implemented using SGML. A markup language defines its own set of tags, such as <h1> and <p>.

As a subset of SGML, XML uses only the most-commonly features of it, and inherits the characteristics of self-describing, “separation of representation from content”, and extensibility from SGML but gets rid of the shortcomings of SGML. XML is not, however, an integration solution in itself - it is just a data definition language. In other words, XML is a metalanguage which is a mechanism for representing other languages in a standardized way. Unlike HTML, XML tags identify the data and specify the semantics (meaning) of the data, rather than specify how to display it. The following figure clearly indicates the relationship among SGML, XML, and HTML.

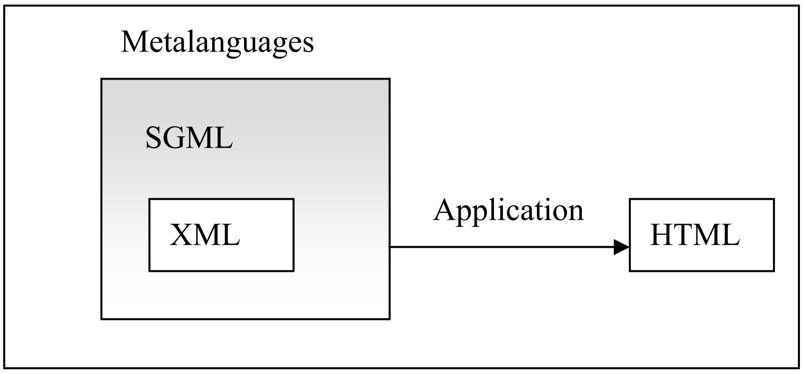


Figure 3. 13 Relationship among SGML, XML, and HTML

3.4.2 Core Concepts of XML

1. Tags, elements, and attributes

There are three common terms used to describe the components of an XML document: tags, elements, and attributes. Here is a sample document that illustrates the terms (Tidwell, 2002):

```
<address>
    <name>
        <title>Mrs.</title>
        <first-name>Mary</first-name>
        <last-name>McGoon</last-name>
    </name>
    <street>1401 Main Street</street>
    <city state="NC">Anytown</city>
    <postal-code>34829</postal-code>
</address>
```

a. A tag is the text between the left angle bracket (<) and the right angle bracket (>). There are starting tags (such as <name>) and ending tags (such as </name>).

b. An element is the starting tag, the ending tag, and everything between them. In the sample above, the <name> element contains three child elements: <title>, <first-name>, and <last-name>.

c. An attribute is a name-value pair inside the starting tag of an element. In this example, state is an attribute of the <city> element.

2. Well Formed and Valid

A "Well Formed" XML document is a document that conforms to the XML syntax rules that described below. The XML specification requires a parser to reject any XML document that does not follow the basic rules. A parser is a piece of code that attempts to read a document and interpret its contents. A "Valid" XML document is a "Well Formed" XML document, which also conforms to the rules of a Document Type Definition (DTD) or schema. Here are some of the basic rules:

a. All XML documents must have a root element.
b. XML elements cannot overlap.
c. All XML elements must have a closing tag.
d. XML elements are case sensitive.
e. Attributes must have quoted values.

3.4.3 Document Type Definition (DTD) and XML Schema

1. DTD

The purpose of a DTD is to define the legal building blocks of an XML document. It defines the document structure with a list of legal elements. A DTD can be declared inline in the XML document, or as an external reference. We can use a DTD to verify our own data. With a DTD, independent groups of people can also use a common DTD for interchanging data. The computer application systems can use a standard DTD to verify if the data received from outside world is valid. Here's a DTD that defines the basic structure of an address document example (Tidwell, 2002):

```
<!-- address.dtd -->
<!ELEMENT address (name, street, city, state, postal-code)>
<!ELEMENT name (title? first-name, last-name)>
<!ELEMENT title (#PCDATA)>
<!ELEMENT first-name (#PCDATA)>
<!ELEMENT last-name (#PCDATA)>
<!ELEMENT street (#PCDATA)>
<!ELEMENT city (#PCDATA)>
<!ELEMENT state (#PCDATA)>
<!ELEMENT postal-code (#PCDATA)>
```

This DTD defines three basic things:

a. An <address> element must contain a <name>, a <street>, a <city>, a <state>, and a <postal-code>. All of those elements must appear, and all of them must appear in the order.

b. A <name> element contains an optional <title> element (the question mark means the title is optional), followed by a <first-name> and a <last-name> element.

c. All of the other elements contain text. (#PCDATA stands for parsed character data; you cannot include another element in these elements.)

2. XML Schema

A schema can define all of the document structures that put in a DTD and it can also define data types and more complicated rules than a DTD can. Although XML Schema is an XML-based alternative to DTD, it has several advantages over DTD (Tidwell, 2002):

a. XML Schema uses XML syntax: An XML Schema is an XML document.

b. XML schema supports data types: XML schema supports all of the original data types from DTD (things like IDs and ID references). It also supports integers, floating point numbers, dates, times, strings, URLs, and other data types useful for data processing and validation.

c. XML Schema is extensible: You can also create your own schema, and you can derive new data types based on other data types.

d. XML Schema has more expressive power: You can define the value constraints of any attribute, or the value of any element must match the regular expression.

e. XML Schema supports namespaces.

Here's an XML Schema that matches the original name and address DTD. Although the schema is much longer than the DTD, it expresses more clearly.

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <xsd:element name="address">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element ref="name"/>
        <xsd:element ref="street"/>
        <xsd:element ref="city"/>
        <xsd:element ref="state"/>
        <xsd:element ref="postal-code"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>

  <xsd:element name="name">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element ref="title" minOccurs="0"/>
        <xsd:element ref="first-Name"/>
        <xsd:element ref="last-Name"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>

  <xsd:element name="title" type="xsd:string"/>
  <xsd:element name="first-Name" type="xsd:string"/>
  <xsd:element name="last-Name" type="xsd:string"/>
  <xsd:element name="street" type="xsd:string"/>
  <xsd:element name="city" type="xsd:string"/>

  <xsd:element name="state">
    <xsd:simpleType>
      <xsd:restriction base="xsd:string">
        <xsd:length value="2"/>
      </xsd:restriction>
    </xsd:simpleType>
  </xsd:element>
  <xsd:element name="postal-code">
    <xsd:simpleType>
      <xsd:restriction base="xsd:string">
        <xsd:pattern value="[0-9]{5}(-[0-9]{4})?"/>
      </xsd:restriction>
    </xsd:simpleType>
  </xsd:element>
</xsd:schema>
```

3.4.4 Related XML Techniques

1. XLINK

The XLink (XML Linking Language) specification describes constructs that may be inserted into XML resources to describe links between objects. It can describe simple links like those in HTML, as well as more sophisticated multi-ended, typed links. The basic form of address is a Uniform Resource Identifier (URI), which is a more general form of resource location than the Web's URL (Uniform Resource Locator).

2. XPath

The XPath (XML Path Language) is a language for addressing parts of an XML document. It operates on the abstract, logical structure of an XML document.

3. XPointer

The XPointer (XML Pointer Language) specification supports addressing into the internal structures of XML documents. It is an extension XPath to address points, ranges, and nodes; to locate information by string matching; and to use addressing expressions in URI-references as fragment identifiers.

4. XSL

The Extensible Style Language (XSL) is a specification by the World Wide Web Consortium for applying formatting to XML documents in a standard way. The XSL specification is divided into two main documents. The XSL Transformations (XSLT) specification describes how to transform one XML document into another, and an XML vocabulary specifies formatting semantics. An XSL stylesheet specifies the presentation of a class of XML documents by describing how an instance of the class is transformed into an XML document that uses the formatting vocabulary.

5. Namespace

A namespace is a collection of names that may be used in an XML document as elements or attribute names. It can associate names with a particular domain and thus avoid redundancy or allow the use of the same name with two different meanings. Namespaces in XML are identified by a Uniform Resource Indicator (URI), which allows each namespace to be unique.

3.5 Advantages of XML in B2Bi Standard

According to Wangler (2001), B2B integration requires a standards-based approach enabling organizational partnerships to be flexible in adding or removing applications and partners depending on the organizational needs. Standards thus facilitate the integration of new members in network organizations. Quite a few standardization efforts for inter-organizational processes are currently being developed. Most of these efforts use the XML as a basis. XML is designed to be an application-independent and flexible way of representing organized or structured data.

XML provides a robust, human-readable information exchange standard that is not just a consensus choice but a unanimous one. It can support the exchange of application semantics and information content, providing an application-level mechanism for producing business information that other application can use without needing to understand anything about the transmitting applications (Linthicum, 2001). From the development team's point of view, this means they no longer have to use proprietary mechanisms for formatting data passed between applications or stored in files or databases. From the vendors' and business partners' points of view, this means they can easily exchange information (IBM Corporation, 2000). Table 3.19 summarizes the relationship between main XML features and e-business applications.

Table 3. 19 Relationship between XML Features and e-Business Applications

XML Feature	Relationship in the inter-business applications
Sharing data across platforms and applications	Data tagged with XML in one application can be understood in another. The XML tags permit content sharing across the company/organization Intranet as well as Extranet.
Reusing information efficiently	Company information created in XML and stored in a document management system has a longer life span — helping to ensure future readability as well as reuse today and tomorrow.
Improving customer satisfaction	The XML metadata capabilities help software to personalize end-user experiences without complex programming. You can modify data presentation for different client devices.
Using XML in an open, standard environment	XML is a critical component to completing the technology foundation necessary for e-business, removing direct dependencies between data and the software that uses it.

Data source: IBM Corporation, 2000

As explained above, we can say that XML plays an important role in B2Bi standard; it provides a common mechanism for data exchange and application integration. Following is a list of the main advantages that XML provides in B2Bi standard:

1. Self-describing

It means that data in an XML document identifies itself using element and attribute names. A company can define its data formats, ways of process data in an XML document. Another company can know the data it received through parsing and interpreting it. Therefore, XML supports the exchange of application semantics and information content. The data model and process model of XML-based standard can be easily accepted and interpreted.

2. Open

XML provides an open, cross-platform way to transact, manage, and share information. Companies no longer have to use proprietary mechanisms for formatting data passed between applications. This characteristic contributes to the popularity of XML-based standard.

3. Simple

It is a text-based tag language that people and computers can easily understand. Comparing with EDI, XML-based standard document is human-readable. Comparing with HTML, XML is machine-readable.

4. Extensible

A company can invent custom tags and vocabularies for any purpose and share them across interested groups without much cost. Conforming to XML-based standard specification, companies can modify the basic rules according to their business requirements.

5. Linkable

XLINK defines two-way links, multiple-target links, "expanding" links, and links between two existing documents which are defined in a third. Thus, using XLINK, XML-based standard document can link related and most-used pieces of information together as a logical block.

6. International

Built-in support for Unicode, an international language encoding standard, supports all scripts in the world today. This makes transnational data accessible to programmers.

XML has the potential to be the standard language of inter-enterprise business integration: if company information systems, applications, and so on are XML-compliant, then data can be easily exchanged between them. However, XML alone cannot be the solution to the business integration problem. XML itself is not a business data description language; rather, it is a specification language that can be used to build the former (IBM Corporation, 2000).

Most importantly, for XML messages to be interpreted by all companies participating in B2Bi they need to agree on a common XML-based B2B standard, which will define the document formats, allowable information, and process descriptions (Samtani and Sadhwani, 2002).