First Draft
Business Process Project Team
Technical Specification Document
Draft Version 1.0 5/26/00

Table of Contents

The ebXML Business Process Metamodel First Draft
Business Process Project Team
Technical Specification
Draft Version 1.0 5/26/00

The ebXML Business Process Metamodel

Introduction

This document is a first draft technical specification for review by the ebXML Plenary. Comments are welcome. When registering your comment, please provide the following information:

- Your name,
- Your email address,
- The document line number(s) associated with your comment,
- Your comment,
- Rationale for the comment, and,
- Your recommended action for resolution of the issue or any recommended document add/change/delete modifications.

Please e-mail comments to Marcia McLure, marcia.mclure@mmiec.com within two weeks following the official posting date of May 26, 2000.

This document includes the following sections:

- The ebXML Business Process Metamodel Class Diagram
- Metamodel Sub-groupings
- Descriptions of the Metamodel Sub-groupings
- Metamodel Sub-grouping Class Diagrams
- Class Definitions
- Scenarios for the Use of the ebXML Business Process Metamodel
- Automobile Component Procurement Example
- Issues

Suggestions for document improvement are welcome. Thank you, in advance for your comments.

ebXML Business Process Team

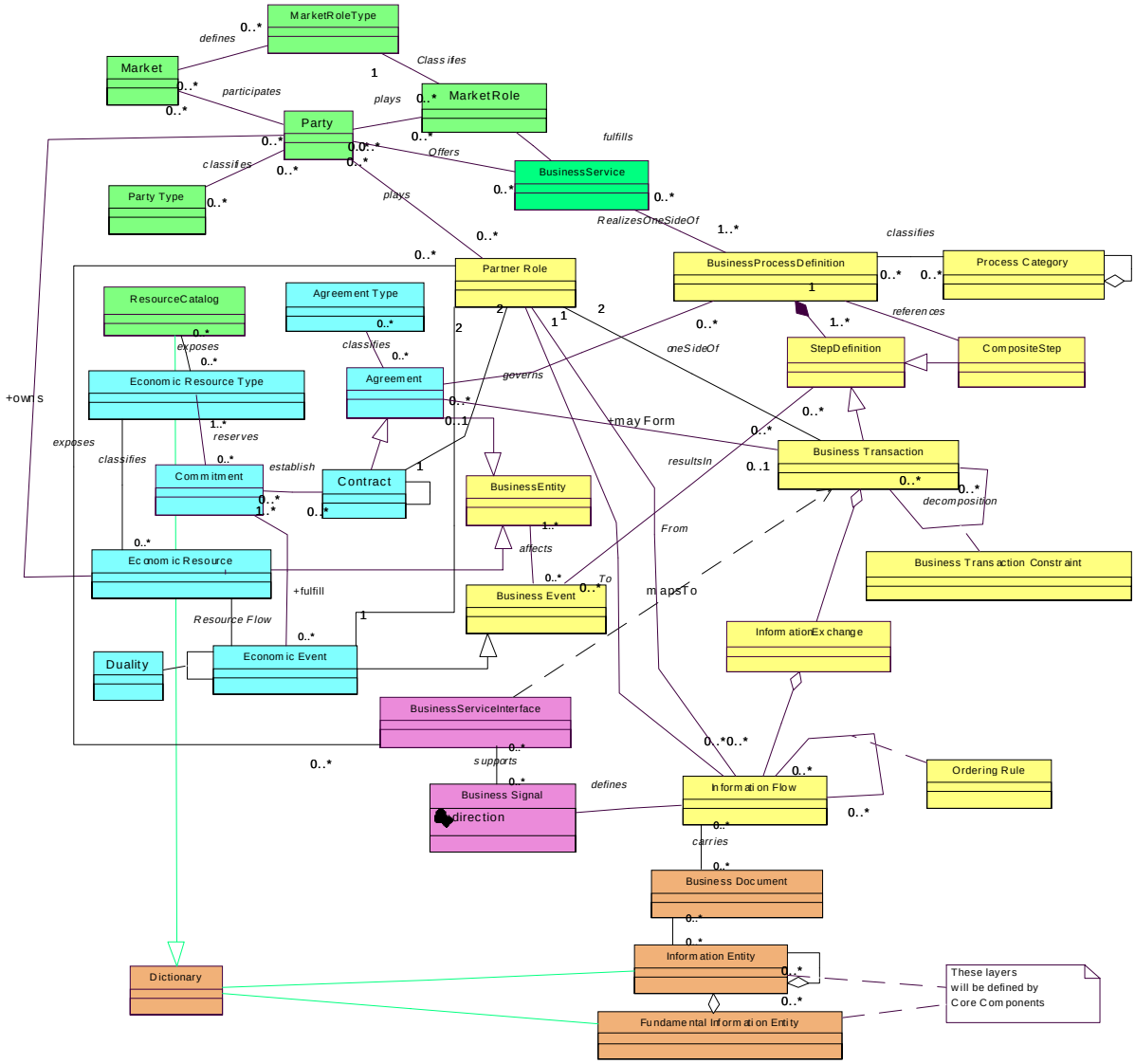
The ebXML Business Process Metamodel

This is the ebXML business process metamodel that more fully defines the contract/commitment section. This ebXML business process metamodel also enables re-usability of process definitions. We refer to this state of the metamodel as Version 1.0.

The model consists of the following logical sub-groupings:

1. Resources and Contracts (color coded in blue),
2. Markets and Communities (color coded in green),
3. Business Processes and Rules (color coded in yellow),
4. Business Service Interfaces and Communication (color coded in purple), and the
5. Information Model (color coded in brown).

The ebXML Business Process Metamodel



Metamodel Sub-groupings

The metamodel consists of the following logical sub-groupings:

1. **Resources and Contracts**
   This is a high level economic model, adapted from REA (Resources, Events, and Agents). It creates a very useful anchor point for the ebXML model, and establishes a pattern for how economic events should be transacted using this model.

2. **Markets and Communities**
   This is the part of the model that allows organizations to register themselves relative to the markets they perform in and the types of services they offer. This aligns with the first four of the seven layers of the eCO framework. Once a number of organizations have registered themselves, other organizations can start discovering new business partners by navigating among the layers of the markets and communities submodel.

3. **Business Processes and Rules**
   This is the part of the model that describes the actual business processes that support the services offered by a given organization. It also describes the interactions required between the partners in order to obtain/perform the services offered.

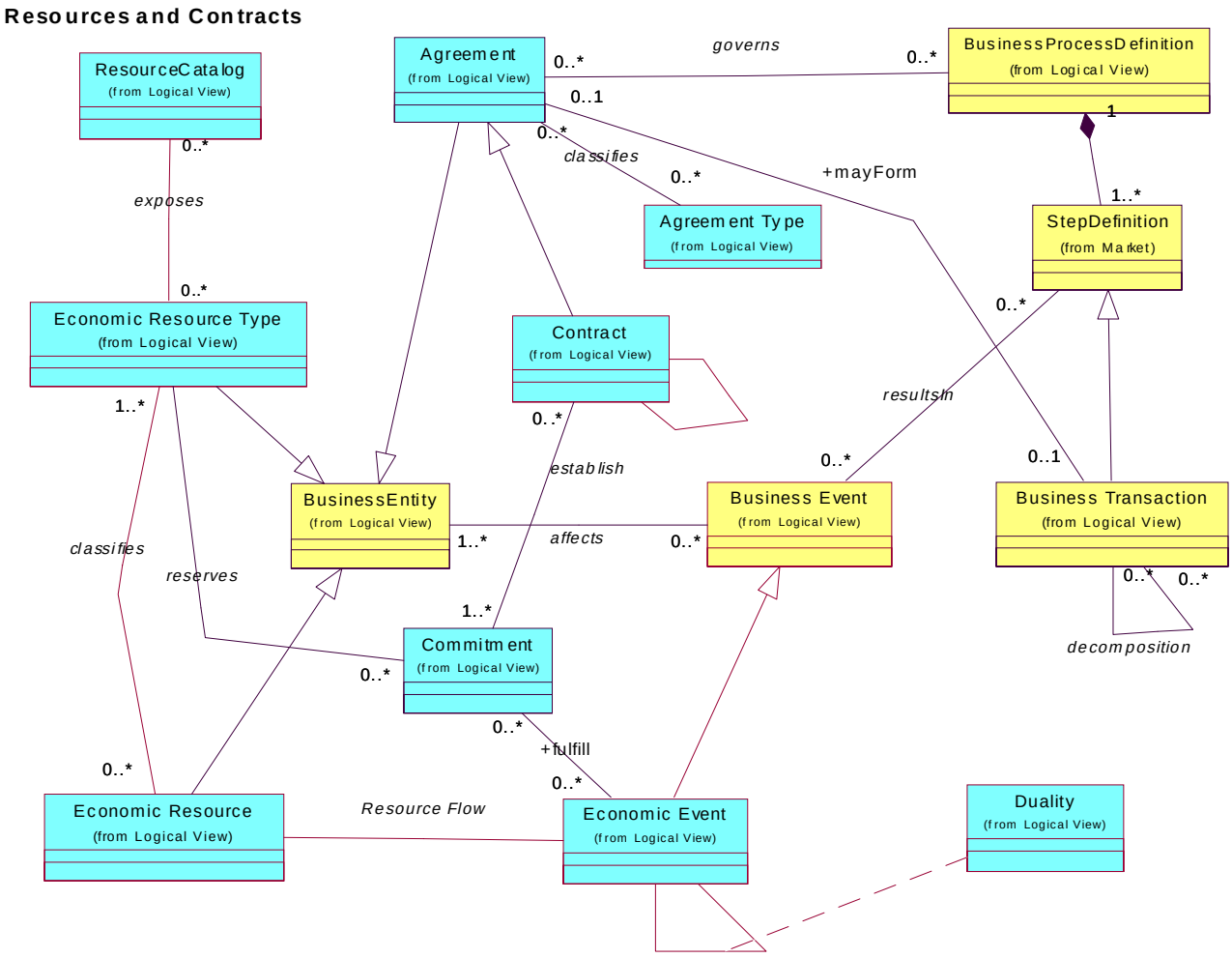
4. **Business Service Interfaces and Communication**
   This is the part of the model that describes the 'interface' that the partners expose, against which the 'opposing' partner can interact, typically by sending business signals consisting of business documents. Document is a broad term that covers both complete documents in the traditional sense, i.e. a sales order, but also descriptions of business events relevant to the service obtained/performed.

5. **Information Model**

Illustrations of the Metamodel Sub-groupings

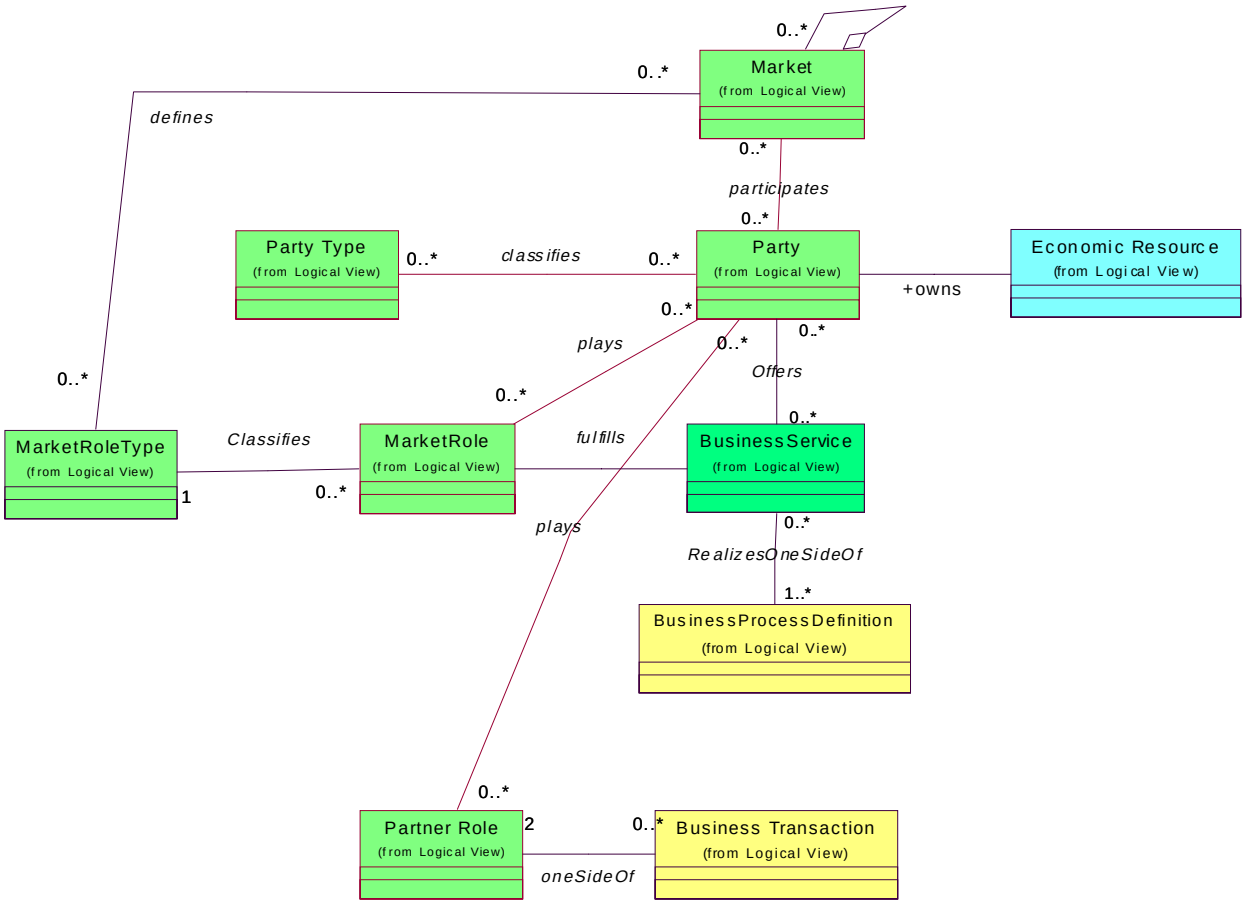
The exact boundaries of each sub-grouping is subject to revision. The metamodel sub-groupings are as follows:

1. Resources and Contracts

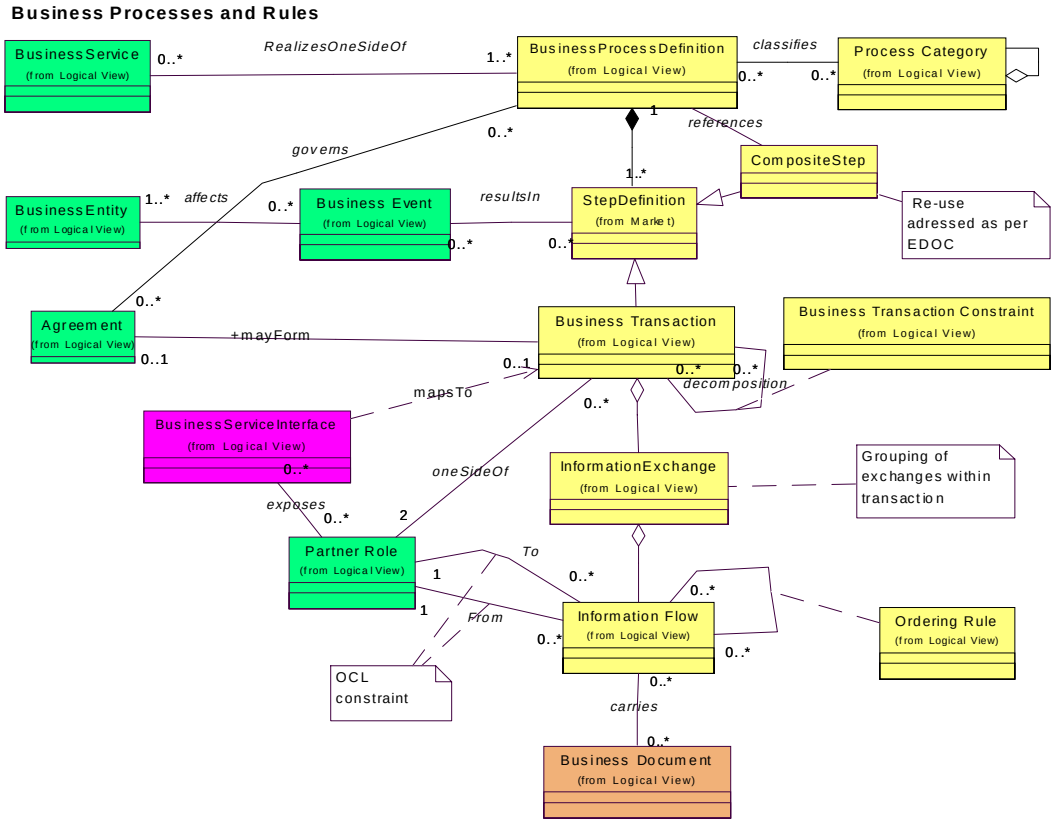


2. Markets and Communities

Markets and Communities

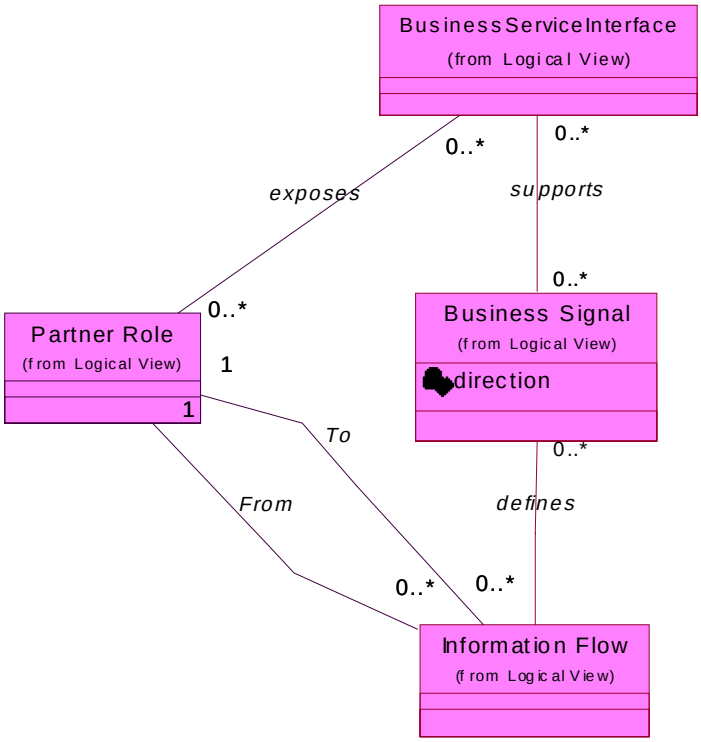


3. Business Processes and Rules



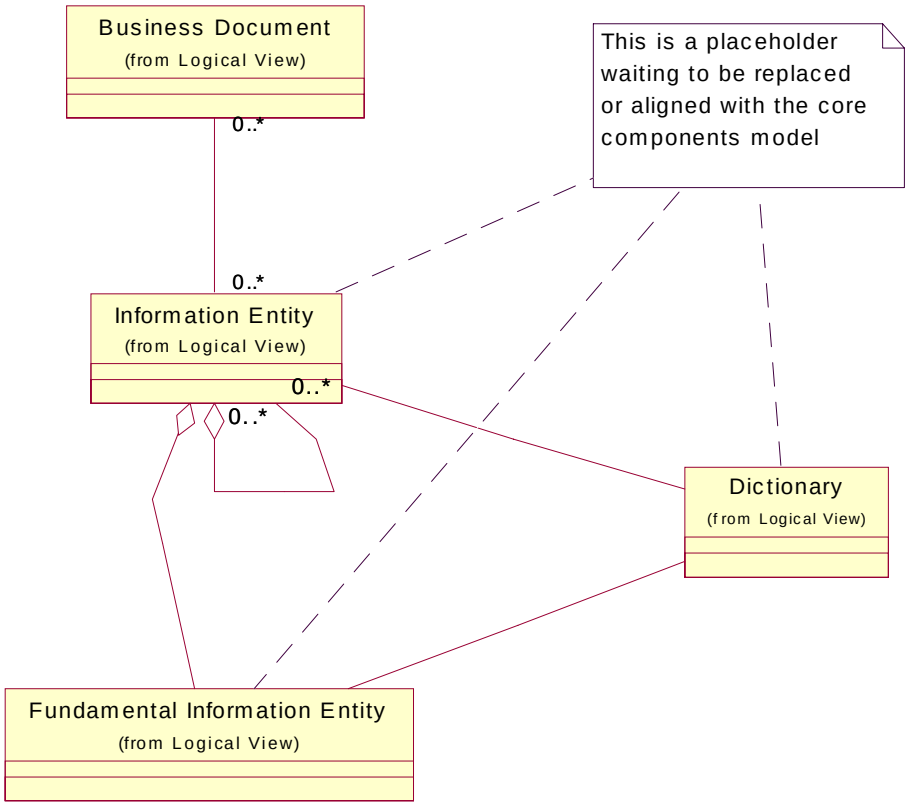
4. Business Service Interfaces and Communication

Business Service Interface



5. Information Model

Information Model



Class Definitions

Definitions of each of the classes are as follows:

Agent.
An **agent** is a particular kind of business process interface that represents an individual.

Agreement
An **agreement** is an arrangement between two parties that specifies in advance the conditions under which they will trade (terms of shipment, terms of payment, expectations of quotations and pricing, etc.) An agreement does not imply specific economic commitments.

Agreement Type
An **agreement Type** is the abstract classification of different types of agreements. Examples might include front-end agreements and yearly contracts.

Business Activity.
A **business activity** is used to represent the state of the business process of one of the partners. For instance the requester is either in the state of sending the request, in the state of waiting for the response, or in the state of receiving (and processing) the response.

Business Document.
A **business document** is the description of a particular entity within a business, or the description of an agreement between organizations, or the description of a business event. The document is never the 'real' thing, just a description of it. A business document is the central component of any information exchange among partner roles.

Business Event
A **business event** is an activity that a business decision- maker needs to monitor or evaluate. In most cases, a business event is performed with the objective of making progress toward a specific business goal within the context of a business process. However, some business events simply exchange or synchronize information between parties. Business event examples might include “obtain a quote” or “make an engineering change.”

Business Process
A **business process** is a collection of business events that are required to achieve a business goal. Normally, such objectives imply the execution of a business transaction or a related set of business transactions that are intended to accomplish a value-added entrepreneurial purpose.

Business Process Definition
A **business process definition** specifies the choreography of business transactions needed to complete a business process.

Business Process Interface.
A **business process interface** is the definition of how to interact with one partner role in order to make him/her perform a desired service. For example, a partner role can expose a business process interface for 'quotation service'. It will describe precisely what kind of business signal (i.e. message) you need to send, what you will get back, and what you may expect to have happen as a result of the exchange.

Business Rule.
A **business rule** is a very generic term to describe rules that govern how we conduct business. In this context a business rule is a rule that guides and constrains the execution of steps within a business process.

Business Signal.
A **business signal** is a message sent between the business process interfaces of two partner roles. A business signal fulfills the information flow requirements between request activity and response activity. A business signal contains business documents(s).

Business Transaction
A **business transaction** is a logical unit of business conducted by two or more parties. The community, the partners, and the process, are all in a definable, and self-reliant state prior to the business transaction, and in a new definable, and self-reliant state after the business transaction. In other words if you are still 'waiting' for your business partner's response or reaction, the business transaction has not completed. A business transaction in our model is reflected as the required exchange or series of exchanges of information between two (or more) partner roles in order to complete the transaction. For example, the exchange could consist of a request for quote and the return either of the actual quote, or of the confirmation that the request had been received. It would not make sense to have the transaction (interaction) consist of the request only

Commitment
A **commitment** is an obligation to perform an economic event at some future point in time. Commitment are fulfilled or executed by economic events.

Community.
A **community** is a collection of parties that have formed a set of mutual partnerships in support of a shared goal. Within a community a party takes on a particular role, and is now distinguished as being a 'partner' as opposed to just a 'party'. Communities often, but not always, form as subsets of markets. What communities have in common are shared interests and shared processes. Examples of communities are: A given company's entire supply chain, An alliance or joint venture of a number of companies to collaborate to offer complete solutions, A company and all its customers.

Contract
A **contract** is a mutual arrangement between parties that some actual economic exchanges will occur in the future. Contracts can have recursive relationships with other contracts, for example, yearly contracts with monthly releases and weekly or daily shipping schedules. Contracts are

containers for collections of commitments. For example, a purchase order is a contract wherein the line items are commitments.

Contract Type.
A **contract type** is the abstract classification or definition of a contract. Examples might be service contracts, orders, and committed-plans.
As in other type objects, contract types are not just categories, they can also define the rules and processes governing contracts of the type.

Dictionary.
The **dictionary** should contain data types, re-usable components, and the templates (DTD's) of the business documents, but not the documents themselves.

Document Envelope.
A **document envelope** is the wrapper of an information flow between partner roles.
It is not completely within the scope of the business process project team, rather it belongs to the transport project team, but we need to all have a common agreement on the business aspects and core component aspects of what goes in the envelope.

Duality.
Duality is a relationship between Economic Events, where one is the legal or economic consideration of the other. Examples include a payment for a product or service.

Economic Event
An **economic event** is the transfer of control of an Economic Resource from one party to another party. Examples would include sale, cash-payment, shipment, and lease.

Economic Resource
An **economic resource** is a quantity of something of value that is under the control of an enterprise. Examples are cash, inventory, labor service and machine service.

Economic Resource Type
An **economic resource type** is the abstract classification or definition of an Economic Resource. For example, in an ERP system, ItemMaster or ProductMaster would represent the Economic Resource Type that abstractly defines an Inventory Item or Product. Economic Resource Types may have recursive relationships, so that for example broad classifications like "product" could group smaller classifications like "product family", which in turn could have as members the specific "product masters" with SKU numbers.

Fundamental Information Entity.
A **fundamental information entity** is in essence a data type. In business contexts we might need many more 'data types' with business semantics beyond the standard data types of 'int', float' etc.

Information Entity.
An **information entity** is a primitive or complex data structure. We haven't defined this yet, but it may be that the difference between a data structure and an information entity is that the information entity also contains business rules about the data.

Information Flow.
An **information flow** is a flow of information between partner roles, related to a specific set of business activities within a specific business transaction. Often the information flow will specify a particular business document to be exchanged between the partners before the interaction or the general process can proceed.

Market.
A **market** is a 'meeting place' where organizations and individuals can exchange services or products. A market is defined in terms of the types of services and products that are likely to be exchanged. The "Yellow Pages" in a telephone book is an example of classifications of products and services, e.g. 'Legal Services', or 'Air condition products'. A person can then anticipate the existence of a 'Legal Services' market and an 'Air Conditioning' market.

Partner.
A **partner** is a participant in a community. It is defined in terms of it's generic partner type and its particular partner role(s) within processes within the community.

Partner Role
A **partner role** is the role a party plays in a specific business transaction.

Partner Type.
A **partner type** is a broad classification of the kind of role an organization or individual is able to play within a community. Examples would be 'supplier', 'agent', 'consultant', 'administrator', 'consumer'.

Party
A **party** is any organization or individual that participates in exchanges of products or services in one or more markets. A party is established first as an absolute entity and then in terms of the roles it plays in a market and in terms of the role it plays in a business transaction.

Party Role (Party Role is shown only as “Role” in the diagram.)
A **party role** is the role a party plays within a given market. Examples would be 'manufacturer', 'whole-saler', 'consultancy', 'logistics'. A party can have one or more roles within each market.

Party Type.
A **party type** is a broad classification of the kind of organization or individual. Examples are 'University', 'Corporation', 'Individual', 'Government'.

Process Category.
A **process category** is a broad classification of business processes. At a macro level this classification could be like the "Yellow Pages" classification of services. At a finer level, processes could be classified to more functional groupings such as 'quotation', 'scheduling', The metamodel does not constrain the kinds of classification of processes.

Resource Catalog
A **resource** catalog is basically a navigable guide to offered products and services (Economic Resource Types). It is the market equivalence of a company's product catalog. It would be intended for narrowing down the particular kind of product or service you are looking for, hopefully leaving you with multiple possible sources for that product or service.

Service.
A **service** is a particular kind of business process interface that represents an organization.

Step.
A **step** is a decomposition of a process that has a dependency on another decomposition. That dependence could be a predetermined sequence, or it could be otherwise determined through simple or complex business rules. A step is always either an action taken by a single partner role or an interaction among partner roles.

Scenarios for Use of the ebXML Business Process Metamodel.

The objective of ebXML is to "create a single global electronic market" that enables organizations to find each other and conduct business together through the exchange of information in the form of XML based business documents.

From this statement we can glean the following layers of importance to the Business Process Metamodel:

It must be support the definition of a "market", the definition of processes for "conducting business", the definition of required "exchanges of information", and the definition of the "business documents" themselves.

Therefore the following LAYERS of the business process must be supported by the metamodel:

A. Market (for categorizing and organizing parties and their processes/services)
B. Business Process (for conducting business)
C. Information Exchange (in support of a business process)
D. Business Document (for structuring information)

Note: There is an alignment of these layers to the packages of the metamodel.
The alignment is as follows:

- The market layer uses the 'Markets and Communities' package and the resource part of the "Resources and Contracts" package.
- The Business Process layer uses the "Business Process and Rules" package and the "Resources and Contracts" package.
- The "Information Exchange" layer uses the "Information Flow and Communication" package.
- The "Business Document" layer uses the business document and information entity part of the "Information Flow and Communication" package.

We also divide the scenarios for usage of the metamodel into the following 'STAGES':

1. Designing/Describing markets, business processes, information exchanges and business documents.
2. Implement system to execute in conformance with described business processes, information exchanges and business documents
3. Registering markets, business processes, information exchanges and business documents.
4. Discovering markets, business processes, information exchanges and business documents.
5. Actual execution of a business process through the exchange of business documents.

So we can organize the scenarios as follows:

	1. Design/Describe	2. Implementation	3. Register	4. Discover	5. Execution
A: Market	Market-Design	N/A	Market-Registration	Market-Discovery.	N/A
B: Business Process	Process-Design	Process-Implementation	Process-Registration	Process-Discovery	Process-Execution
C: Information Exchange	Exchange-Design	Exchange-Implementation	Exchange-Registration	Exchange-Discovery	Exchange-Execution
D: Business Document	Document-Design	Document-Implementation	Document-Registration	Document-Discovery	Document-Execution

In the following we describe, for each of the table entries above, how the user and/or a tool provider will make use of the metamodel, and how each of the other pieces of the ebXML architecture are related.

For ease of understanding, we divide this discussion into the following distinct types of scenarios.
'From Scratch design' – An organization designing, implementing, registering a brand new market and process.
'Conversion' – An organization converting an existing market and process design, and adjusting an existing implementation.
"Discovery and adaption" – An organization discovering an existing partner and process and adapting their existing implementation to interoperate.
"Actual communication" – Two organizations actually conducting business by exchanging messages.

Brand new business model.

This scenario assumes for simplicity that none of the parts of the business model are yet in the repository and that the organization(s) designing it are willing to retrofit their applications to fit the new model.

The stages the organization would go through are:
.. When this is working they would register the market, party, partner-role, business process, information exchange and business documents and register themselves as capable of supporting this new model.

1. Design: (For the organization would either use established modeling tools and convert the output to DTD/XML compliant with the ebXML metamodel, or they would use newer lightweight ebXML front end tools to produce ebXML compliant DTD/XML directly)
    a) Market-Design: Determine and describe the market in terms of its domain and it's parties.

b) Process-Design: Determine and describe the business process in terms of its partner roles and business transactions
c) Exchange-Design: Determine and describe each business transaction in terms of its required messages exchanged.
d) Document-Design: Determine and describe each business document in terms of its attributes

2. Implementation. (This may be accomplished using new lightweight adaptor tools to front-end their applications)
    a) Market implementation is not relevant
    b) Process-Implementation: Design and implement a Business Process Service that covers all the business transactions specified in 1.b. above.
    c) Exchange-Implementation: Design and implement Information Exchange Handlers that cover all the Information Exchanges specified in 1.c. above.
    d) Document-Implementation: Design and implement mappings from the documents specified in 1.d. above.

3. Registration: Registration takes place by using a web-based front end to the ebXML repository and/or sending a model compliant xml file using the ebXML message exchange.
    a) Market-Registration: Register each market and party specified in 1.a.
    b) Process-Registration: Register business process specified in 1.b. and its associated business transactions and business rules.
    c) Exchange-Registration: Register for each business transaction specified in 1.b. the required information exchanges as specified in 1.c.
    d) Document-Registration: Register each business document specified in 1.d. above.

The process and site-implementation for this "brand new" business process is now ready for business, next step would be "discovery and adaptation" by potential business partners (see below)

Conversion

This scenario assumes for simplicity that the company already has a complete model design described in some other format and protocol.

The stages the organization would go through are:

1. Design. (or in this case convert the existing explicit or implicit design)
    a) Market-Design: Extract and convert from existing model the market in terms of its domain and it's parties. This conversion should yield an ebXML metamodel compliant XML based model ready for registration.
    b) Process-Design: Extract and convert from existing model the business process in terms of its partner roles and business transactions. This conversion should yield an ebXML metamodel compliant XML based model ready for registration.

c) Exchange-Design: Extract and convert from existing model each business transaction in terms of its required messages exchanged. This conversion should yield an ebXML metamodel compliant XML based model ready for registration.
d) Document-Design: Extract and convert from existing model each business document in terms of its attributes. Since many "libraries" of standard based document designs already exist, and since the metamodel here is very flexible, it is anticipated that little or no conversion be needed for standards based documents. Rather there would just be a qualification attribute of the exchange-design in 1.c. above as to which of several standards the documents involved belong to.

2. Implementation. (This may be an activity of creating wrappers around the existing system to enable the sending and receiving of messages).
    a) Market implementation is not relevant
    b) Process-Implementation: Design and implement a Business Process Service that covers all the business transactions specified in 1.b. above.
    c) Exchange-Implementation: Design and implement Information Exchange Handlers that cover all the Information Exchanges specified in 1.c. above.
    d) Document-Implementation: Design and implement mappings from the documents specified in 1.d. above.

3. Registration: Registration takes place by using a web-based front end to the ebXML repository and/or sending a model compliant xml file using the ebXML message exchange.
    a) Market-Registration: Register each market and party specified in 1.a.
    b) Process-Registration: Register business process specified in 1.b. and its associated business transactions and business rules.
    c) Exchange-Registration: Register for each business transaction specified in 1.b. the required information exchanges as specified in 1.c.
    d) Document-Registration: Register each business document specified in 1.d. above. Since your document may already be specified in another industry standard protocol, you may register just a hyper-link to where the specification is found in an ebXML compliant format.

The process and site-implementation for this "converted" business process is now ready for business, next step would be "discovery and adaptation" by potential business partners (see below)

Discovery and adaption

This scenario assumes for simplicity that an organization can find a partner with an appropriate process and only needs to make adjustments to its applications in order to 'play'. In this scenario the discovery comes first (so we have changed the sequence, but left the numberings intact as a reference back to the matrix). Once discovery has yielded an acceptable partner, process, information exchange, and document structure, the organization has only to adapt its applications.

The stages the organization would go through are:

4. Discovery: (This is done using web frond ends to the ebXML repository, or by sending XML 'query' documents through the ebXML message facility).
    a) Market-Discovery: Using appropriate keywords and wildcards find the market of interest. Starting from the market find possible parties who may be possible partners.
    b) Process-Discovery: Starting from each possible partner discover his/her role in various processes. Find a process that matches the business transactions you need to transact.
    c) Exchange-Discovery: Starting from each business transaction discover if you are capable of producing and consuming the required information exchanges in the specified protocols.
    d) Document-Discovery: Starting from each information exchange, discover if you are capable of mapping into and out of the specified business documents.

1. Design. Not applicable, in essence this organization is using a design already done by another organization.
    a. Market design was discovered in 4.a. above
    b. Process design was discovered in 4.b. above
    c. Exchange design was discovered in 4.c. above
    d. Document design was discovered in 4.d. above

2. Implementation. (This may be an activity of creating wrappers around the existing system to enable the sending and receiving of messages).
    a. Market implementation is not relevant
    b. Process-Implementation: Design and implement a Business Process Service that covers all the business transactions specified in 1.b. above.
    c. Exchange-Implementation: Design and implement Information Exchange Handlers that cover all the Information Exchanges specified in 1.c. above.
    d. Document-Implementation: Design and implement mappings from the documents specified in 1.d. above.

3. Registration: Not required unless you want to establish a more formal 'trading partner agreement'
    a. Market registration already done
    b. Process registration already done
    c. Exchange design may involve the registration of your business process interface to handle your end of the process. This may be validated against the business processes already registered for handling the other end.
    d. Document registration may involve the registration of your document handler interfaces to handle the incoming and outgoing messages. At this point it may be possible to send a series of "test messages" that traverses the whole process and proves that the two parties can in fact live up to the implicit or explicit 'trading partner agreement'.

Note: The described kind of registration of business process interfaces and document handler interfaces may not initially be part of ebXML scope, rather – initially - an eCO style self-registration on your own site might be workable.

This business partner is now ready to do business with the partner/process previously registered.

Actual communication

This scenario assumes that we have already designed, registered and implemented as per above.

1. Design: Already done above
2. Implementation: Already done above
3. Discovery: Already done above
4. Registration: Already done above
5. Execution: The model drives the execution in the sense that the business transaction sequence within a process is (optionally) specified, and the message exchange sequence within a business transaction is (optionally) specified. So one could envision an implementation that actually accesses the ebXML repository to figure out what needs to happen next. More likely the parties implement their ebXML process compliant message handlers, and the exchanges happen directly between these message handlers, using message formats prescribed in the repository. These message handlers may themselves handle the mapping into or out of the organizations applications, or may interact with "wrappers" specifically designed for this purpose. In either case, the ebXML end of the mapping is prescribed by the registered documents.

Automobile Component Procurement Example

Introduction

This is the first ebXML-BP metamodel example. More will come, including some that are much simpler than this one, which is deliberately complex in order to test the metamodel.

This example is not "final". The intention is for this example to develop along with the ebXML project until it is fully populated with functional test data, and also to be accompanied by several other examples illustrating different scenarios.

The reasons for starting with this particular process include:

- it is a supply chain component procurement example, instead of the usual office supply purchase;
- the business practices cover most of the metamodel;
- the business practices are well documented by an industry-wide group, AIAG (Automotive Industry Action Group);and
- the business practices are similar to supply chain relationships in other industries, e.g. appliances and retail.

Sections

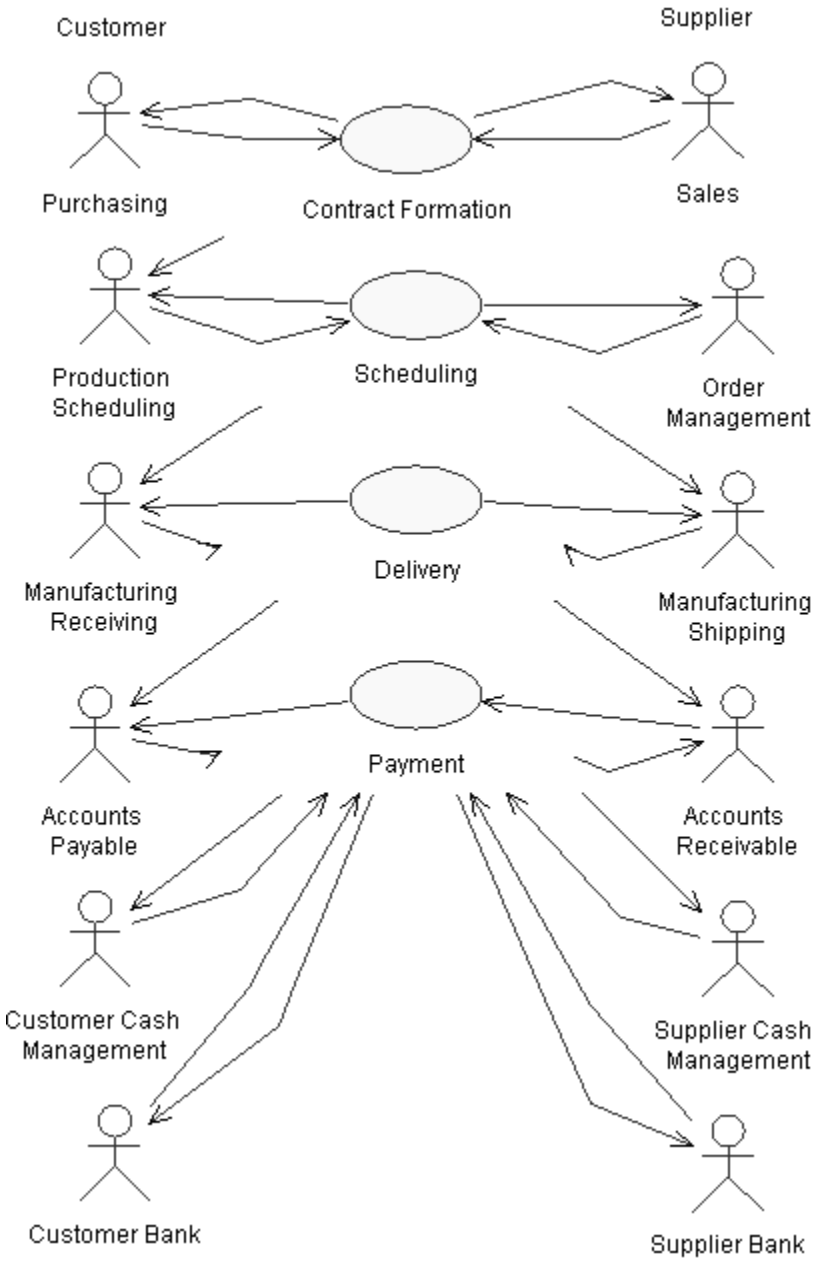
1. **UML Use Cases**, with no reference to ebXML metamodel classes or technology.

2. **UML Collaboration Diagrams** mapping the use cases to the current ebXML metamodel classes. (Note: not every detail of the use cases is shown in collaboration diagrams. Some sections were omitted as being repetitive, with no new mappings.)

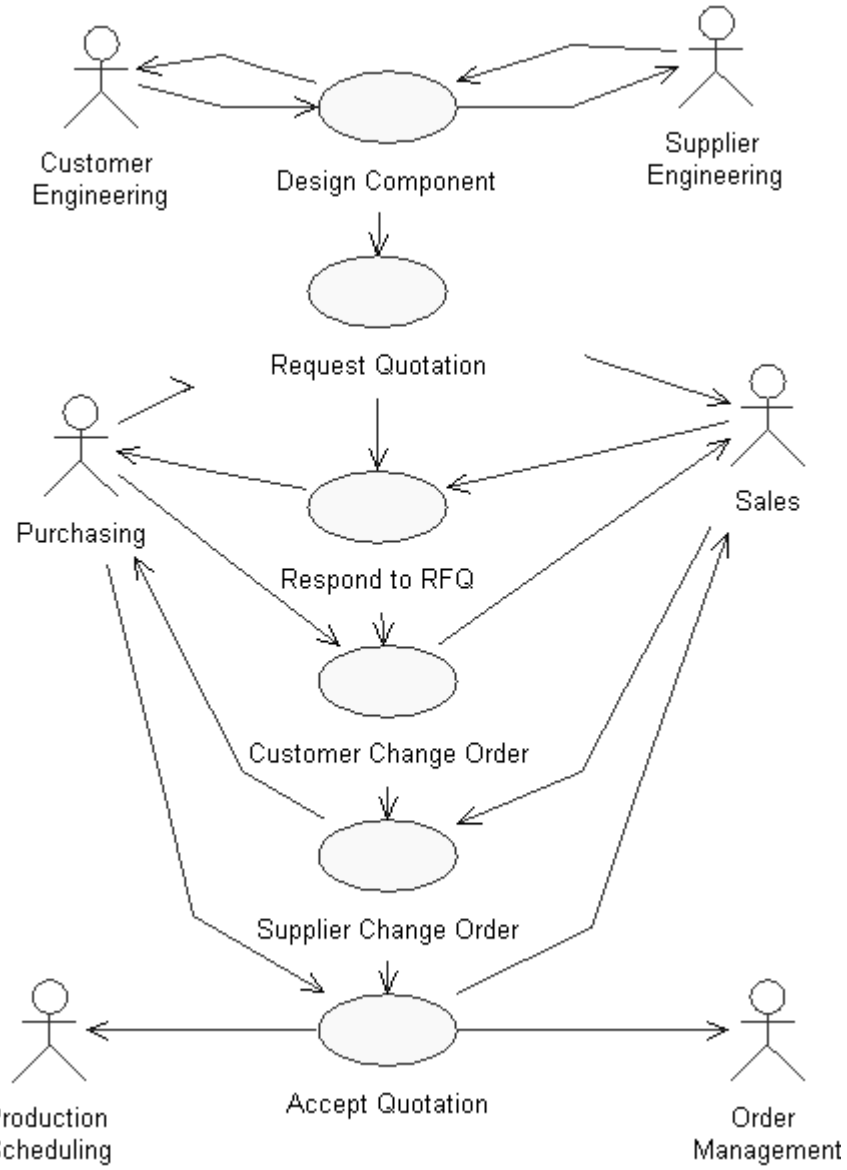
3. **Auto Supply Chain Procurement Practices Not Captured in Current Use Cases** - not yet included in the current use cases.

4. **ebXML Metamodel Issues**, that is, places where the use cases did not map cleanly to the current metamodel.

Use Cases - Overview

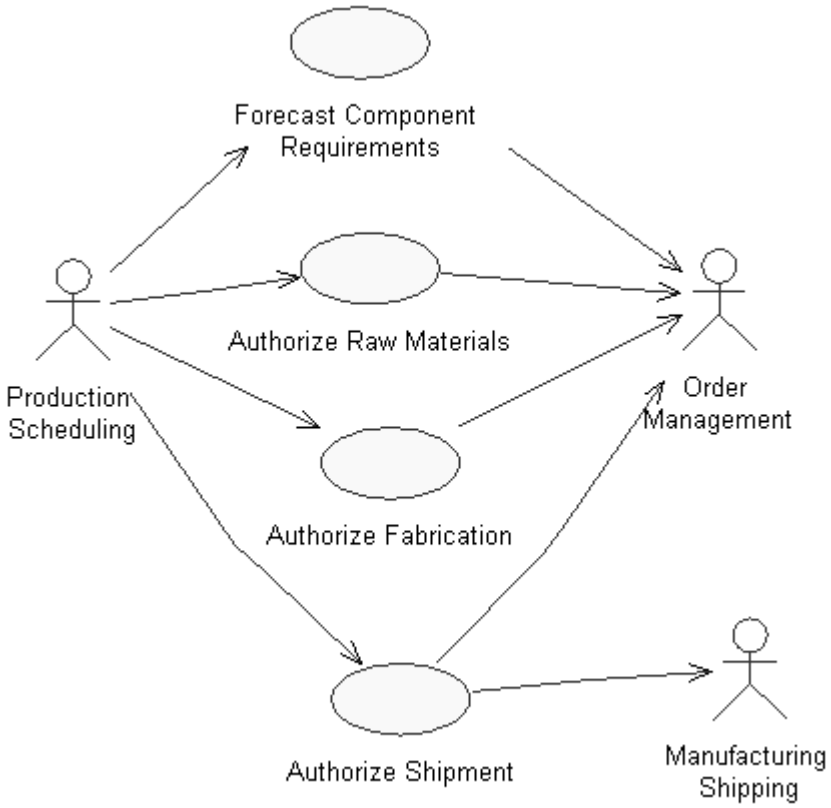


693 **Contract Formation**

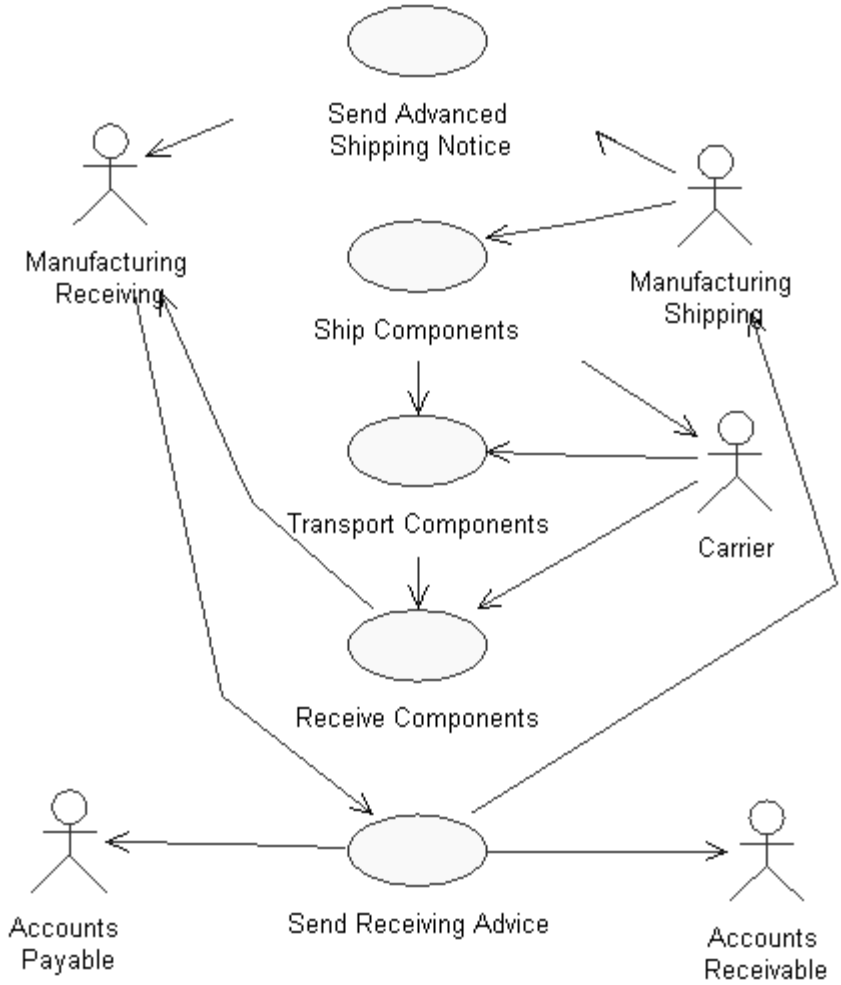


694
695
696
697

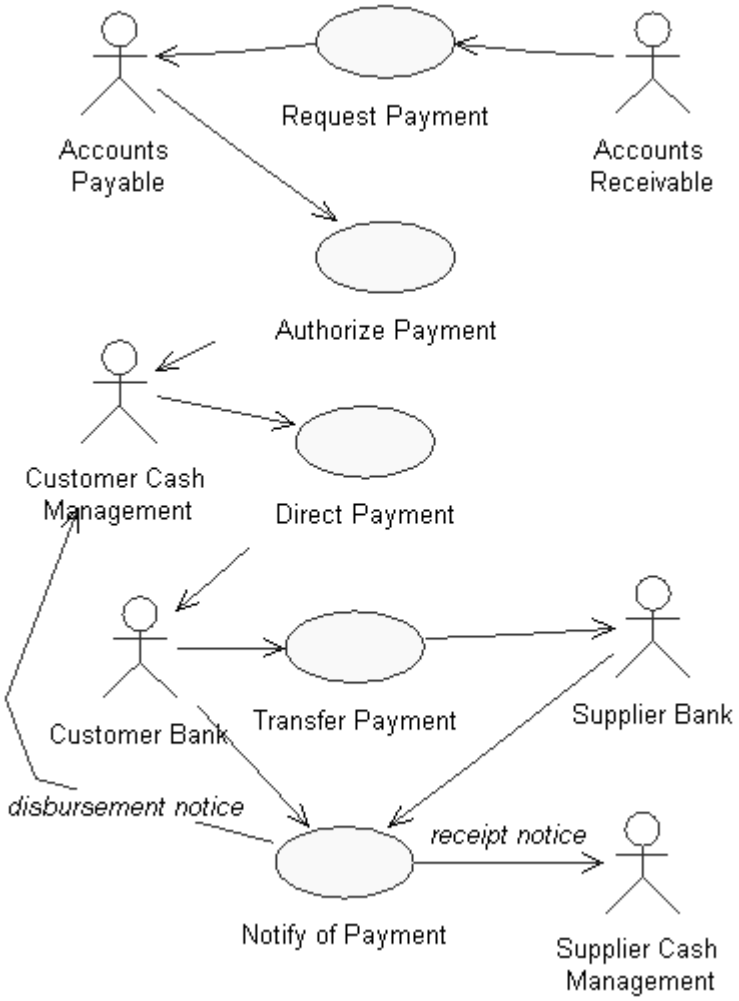
Scheduling


Forecast Component Requirements
Authorize Raw Materials
Production Scheduling
Order Management
Authorize Fabrication
Authorize Shipment
Manufacturing Shipping

Delivery


Send Advanced
Shipping Notice
Manufacturing
Receiving
Manufacturing
Shipping
Ship Components
Transport Components
Carrier
Receive Components
Accounts
Payable
Send Receiving Advice
Accounts
Receivable

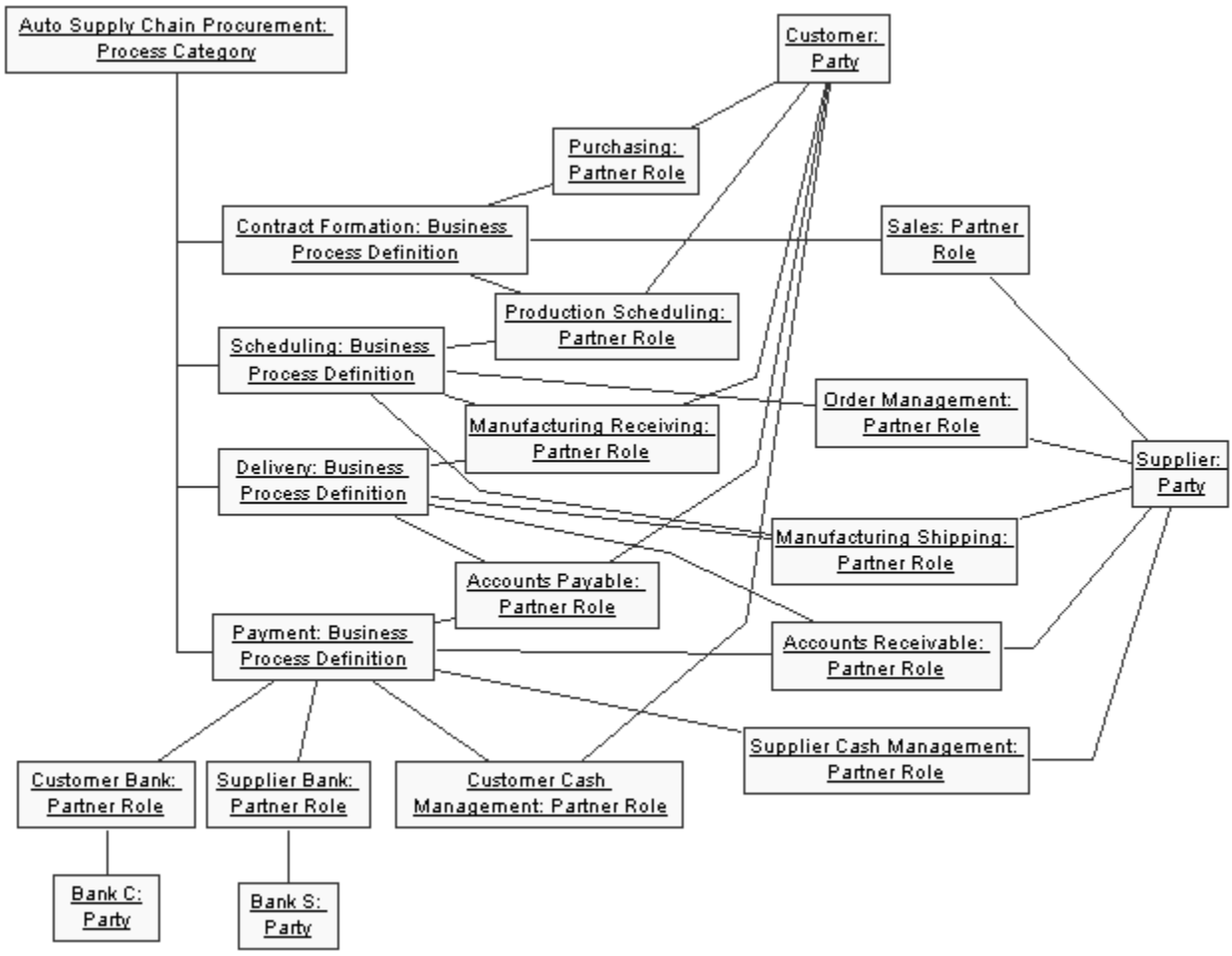
Payment


Request Payment
Accounts Payable
Accounts Receivable
Authorize Payment
Customer Cash Management
Direct Payment
Customer Bank
Transfer Payment
Supplier Bank
disbursement notice
receipt notice
Notify of Payment
Supplier Cash Management

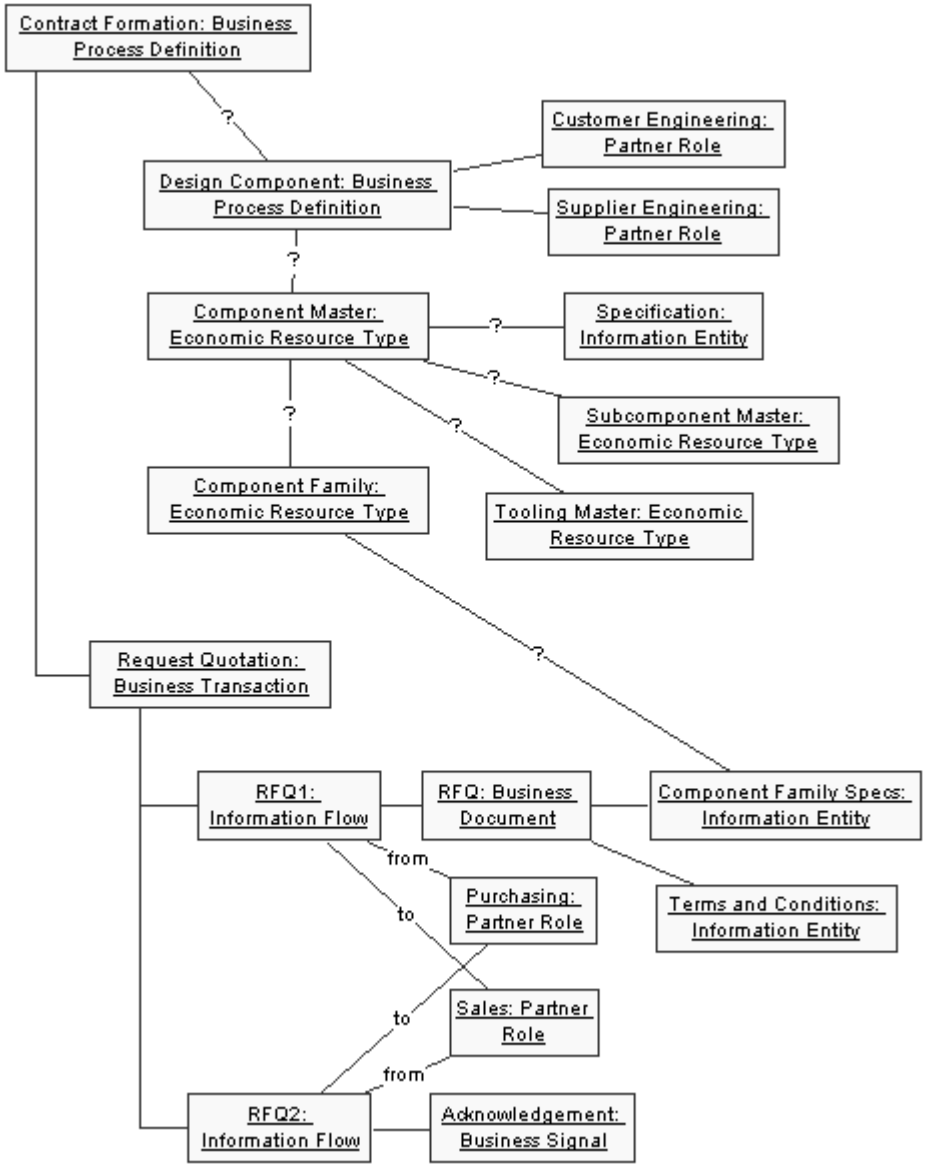
Corresponding Collaboration Diagrams

In each rectangle, an object name is followed by an ebXML metamodel class name, e.g. Object: Class.

Overview

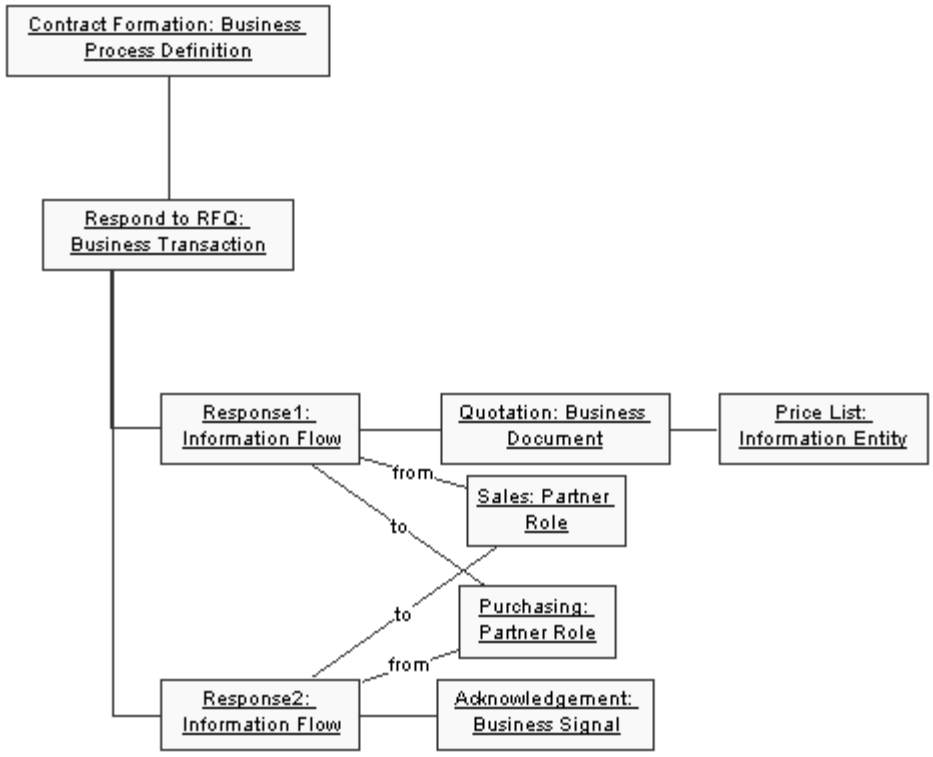


Contract Formation Step 1

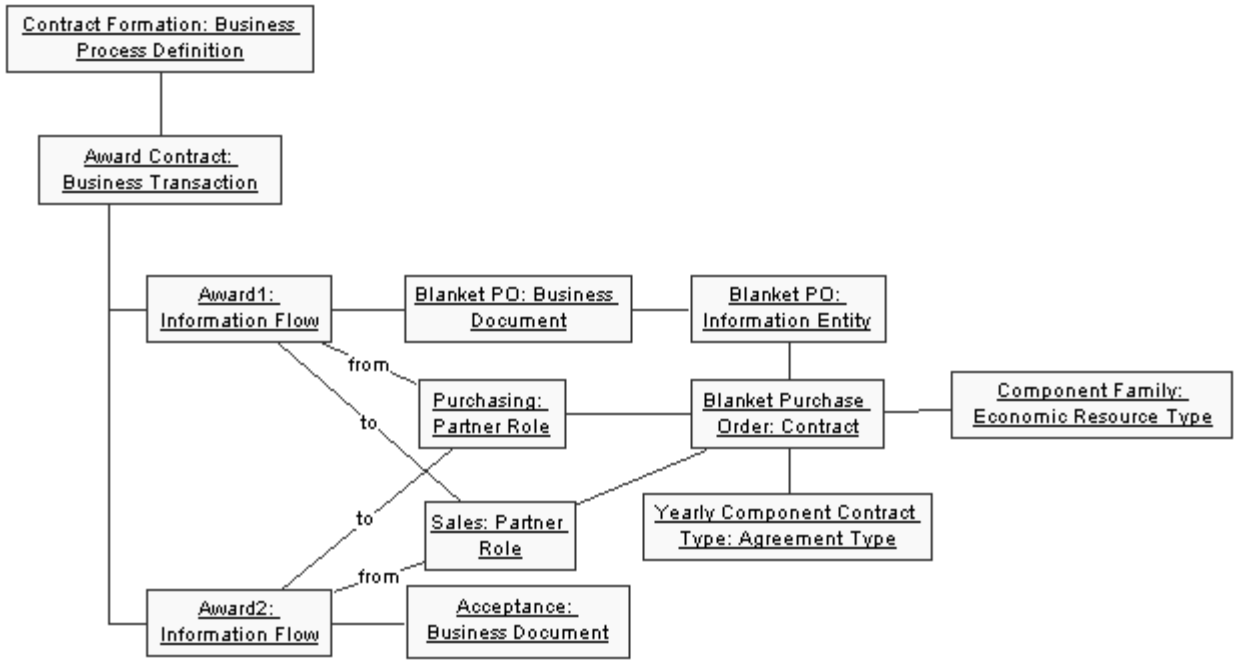


The question marks in this diagram represent relationships that do not exist in the current ebXML metamodel.

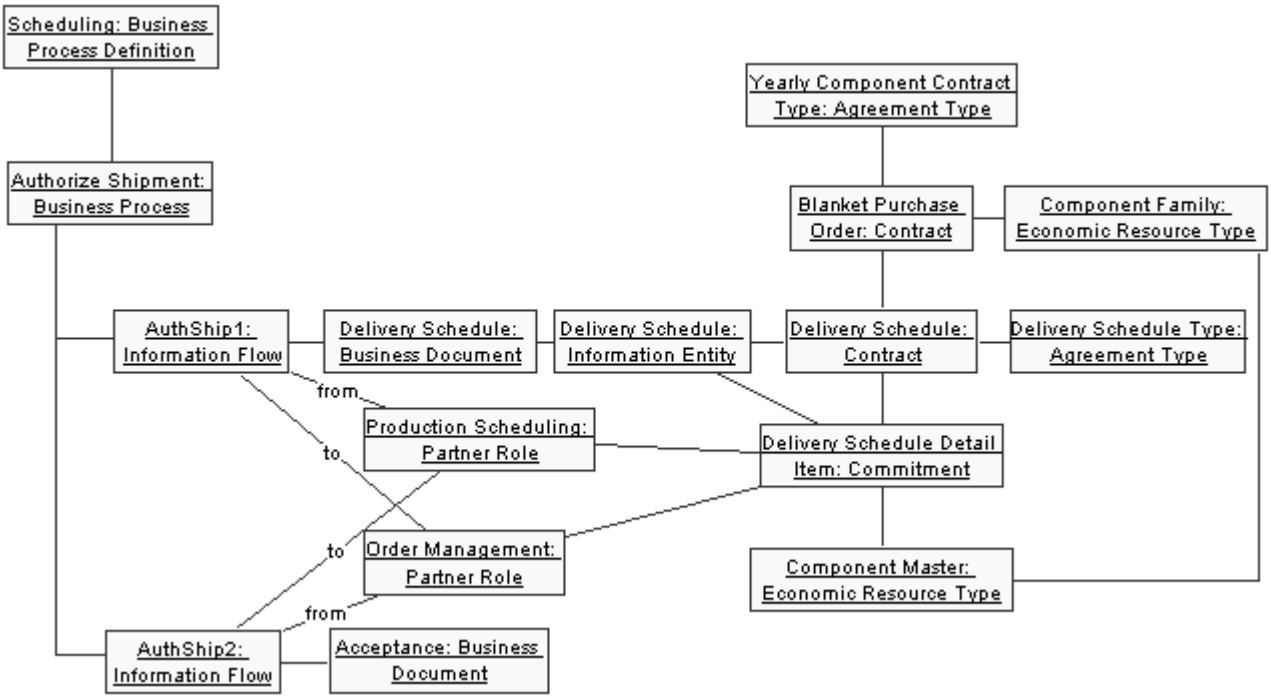
Contract Formation Step 2



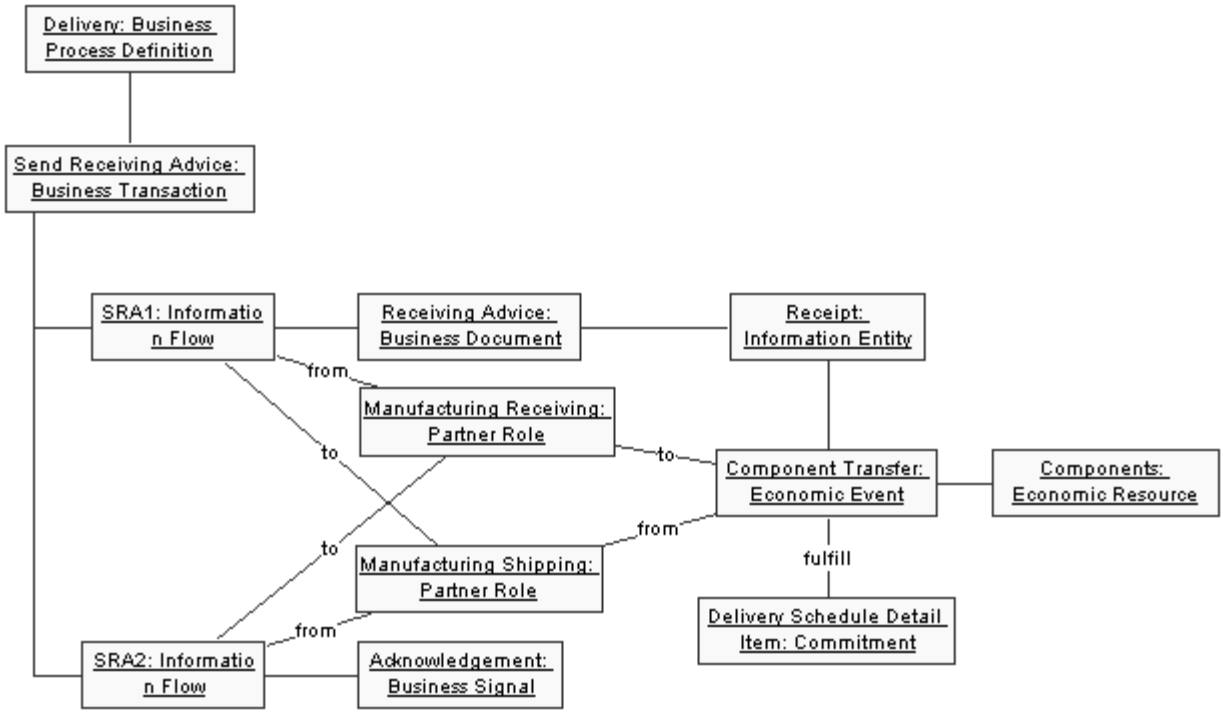
Contract Formation Final Step



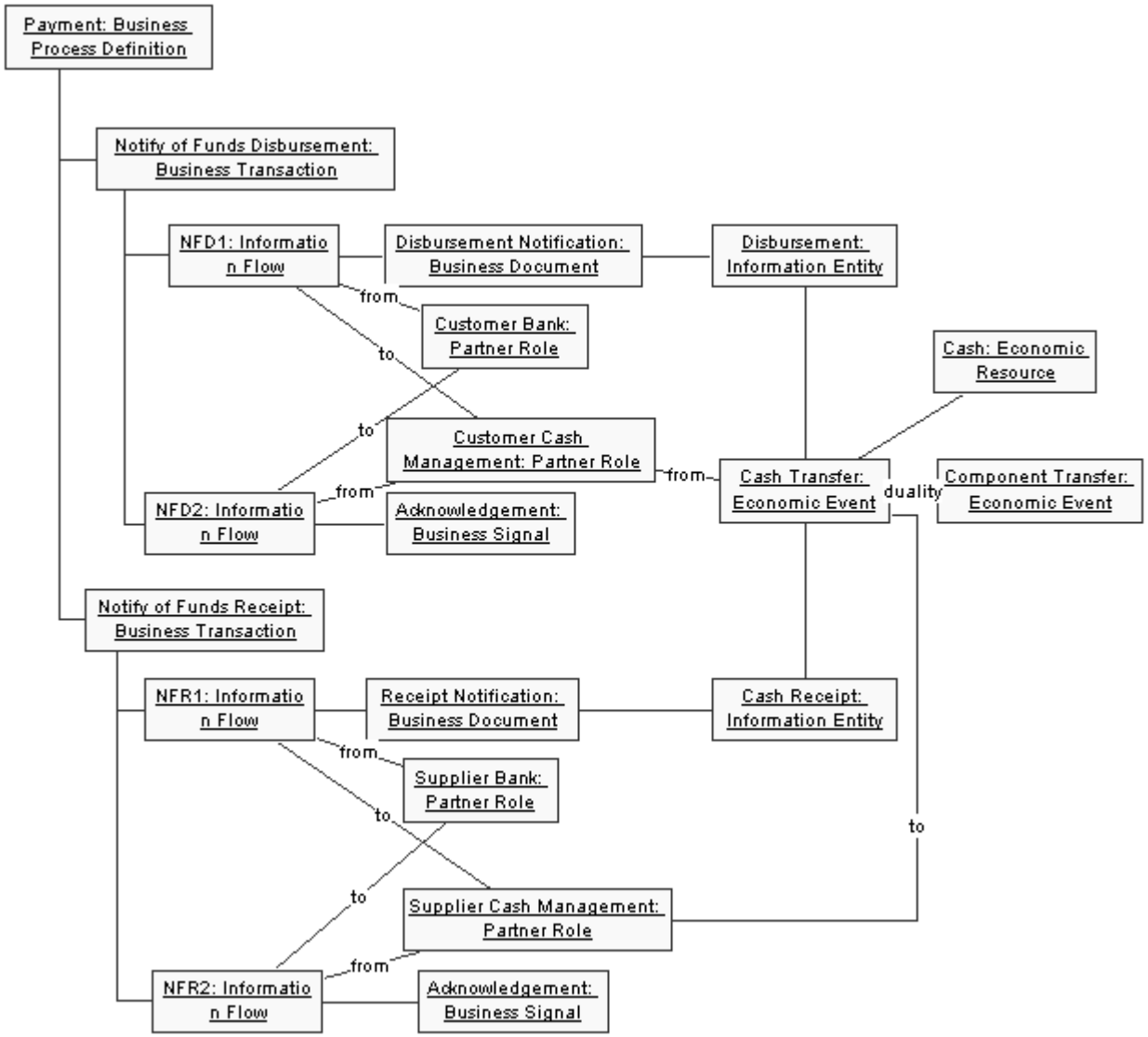
Scheduling Final Step



Delivery Final Step



Payment Final Steps



Auto Supply Chain Procurement Practices Not Captured in Current Use Cases

1. **Preliminary trading partner agreements** may be formed before contracts are negotiated. These agreements may not carry any economic commitments. They would be mapped to the Agreement class in the latest ebXML metamodel.

2. **Intermediate consignees** may be used in the Delivery use case, to pool components before delivery to the point of production, and/or to perform outside services.

3. **Variations in delivery authorization** include regular purchase orders, delivery schedules, sequenced delivery schedules, and electronic Kanbans or JIT pull signals.

4. **Variations in payment authorization** include evaluated receipts settlement, pay on production, pay to the ASN, and invoices.

5. **Variations in payment**

Metamodel Design Issues

These are places where the use cases do not map cleanly to the current ebXML metamodel classes and relationships.

1. There are several missing relationships in the **Contract Formation Step 1** collaboration diagram:
    1.1. Recursive relationship from [**Business Process Definition**] to nested [**Business Process Definition**].
    1.2. Recursive relationship from [**Economic Resource Type**] to subcomponent [**Economic Resource Type**] (e.g. Bill of Materials relationships and Tools).
    1.3. Relationships between higher and lower level [**Economic Resource Types**] (e.g. Component Family and Component Master).
    1.4. Relationships from [**Economic Resource Types**] to [**Information Entities**] (e.g. Specifications).
2. It is clear that [**Agreement Types], [Agreements]** and [**Contracts]** are complex objects. Likewise [**Economic Resource Types]**. Should their decomposition be part of the metamodel, or deferred to Common Components? (Same issue applies to any other complex objects.)
3. The relationships between [**Information Entities]** and other metamodel classes are not specified in the metamodel.