



# ebXML Technical Architecture

김 룡 원

[rwkim@etri.re.kr](mailto:rwkim@etri.re.kr)

한국전자통신 연구원  
전자상거래 연구부

내용

1. 개요
2. ebXML Architecture Model
3. ebXML Functional Phases
4. ebXML Infrastructure

## 개 요

## 탄생배경

### ◆ EDI

- ▶ 고비용이 요구되므로 주로 대기업에서 활용
- ▶ 중소하청업체는 대기업 시스템에 통합된 모형

### ◆ XML

- ▶ 전자 거래의 새로운 데이터 교환의 수단으로 등장
  - ▶ More open, more loosely-coupled more object- or component-oriented
- ▶ “EDI is dead”를 표방
  - ▶ 비즈니스 및 기술적측면을 고려하지 않은 성급한 판단

### ◆ EDI를 쉽게 포기하지 못하는 이유

- ▶ EDI는 실제 비즈니스에 많이 이용되고 검증된 부분이 많음
- ▶ EDI를 위한 그 동안의 많은 투자비용
- ▶ XML을 이용한 비즈니스 메시지 설계 및 의미의 표준화가 이루어지지 않음

## Goals

### ◆ Goal for ebXML

- ▶ ebXML Specification은 비즈니스 처리를 위해 EDI에 투자한 가치를 유지하면서 새로운 XML의 기술적 능력을 이용할 수 있게 하는 구조를 제공한다.
- ▶ ebXML Mission
  - ▶ To provide an open XML-based infrastructure enabling the global use of electronic business information in an interoperable, secure and consistent manner by all parties.

### ◆ Goal for TA

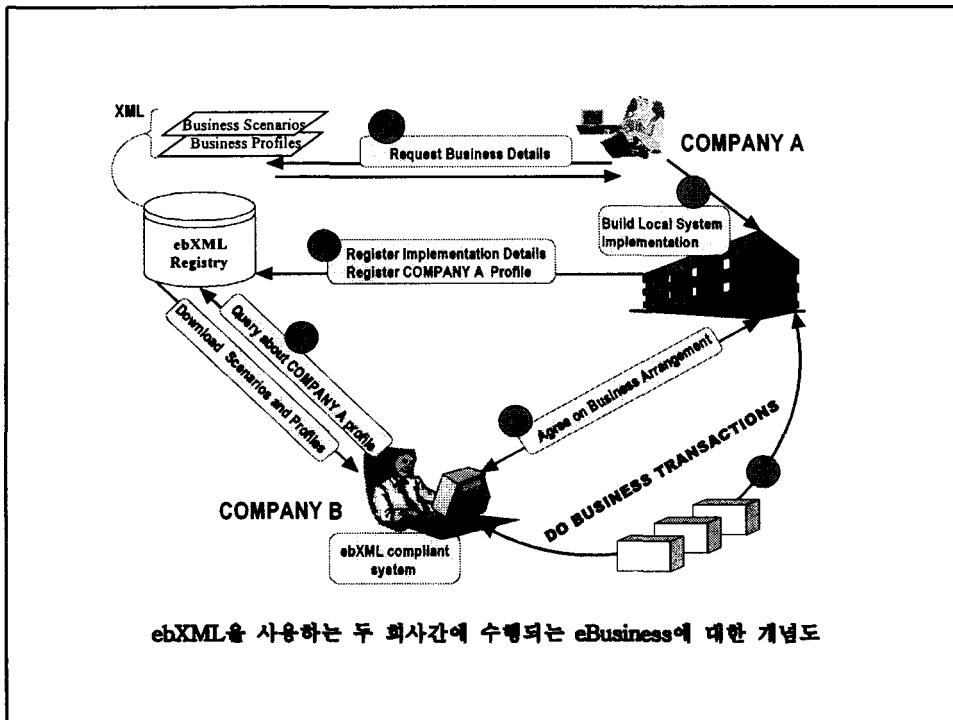
- ▶ TA Specification은 ebXML의 high level overview를 제공하는 것이다.
  - ▶ ebXML 어플리케이션, 컴포넌트 그리고 관련 서비스를 만들기 위해 요구되는 세부사항은 각각의 프로젝트팀 스펙을 참조


## TA Specification에서 소개되는 개념(1)

1. Business Process와 Information Model을 기술하기 위한 표준 메커니즘
2. Business Process와 Information Meta Model이 공유되고 재사용되도록 등록하고 저장하기 위한 메커니즘
3. 참여자들 각각에 대한 정보의 등록과 검색
  - 참여자들이 제공하는 Business Process
  - Business Process를 제공하기 위한 Business Service Interface
  - 각각의 Business Service Interface들 사이에서 교환되는 Business Message
  - 전송, 보안 그리고 엔코딩에 대한 프로토콜 설정
4. 3에서 기술된 내용을 기반으로 참여 파트너들이 business arrangement에 대한 동의(agreement)를 기술하는 방법

## TA Specification에서 소개되는 개념(2)

5. 두 참여자 사이의 상호 운용성, 보안 그리고 신뢰할 수 있는 메시지 교환을 가능하게 하는 표준화된 Business Messaging Service
6. Business arrangement에 정의된 제약사항과 일치하도록 Business Process에 관련된 각각의 Messaging Service에 대한 설정 메커니즘

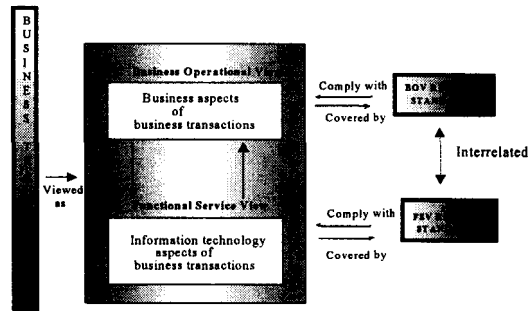




# ebXML Architecture Model

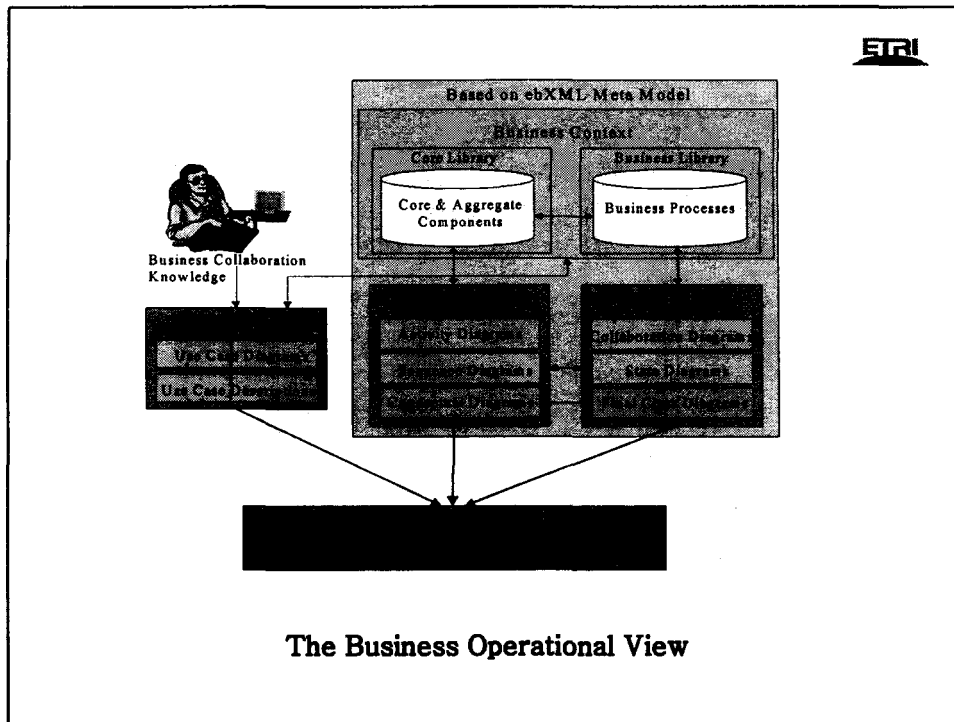
## ebXML Architecture 모델(1)

- ◆ Open-edi 참조 모델, ISO 14662에 기반
  - ▶ 시스템의 상호운용성과 호환성을 극대화
  - ▶ BOV와 FSV를 명확히 구분하는 것이 목적



## ebXML Architecture 모델(2)

- ◆ BOV(Business Operational View)
  - ▶ 트랜잭션과 데이터 교환에 사용되는 business data의 semantics
  - ▶ Business transaction에 포함되는 사항들
    - ▶ operational convention
    - ▶ agreements and arrangements
    - ▶ mutual obligations and requirements
  - ▶ ebXML 호환 어플리케이션과 컴포넌트를 생성하는데 필요한 business object를 제공



**ETRI**

## ebXML Architecture 모델(3)

---

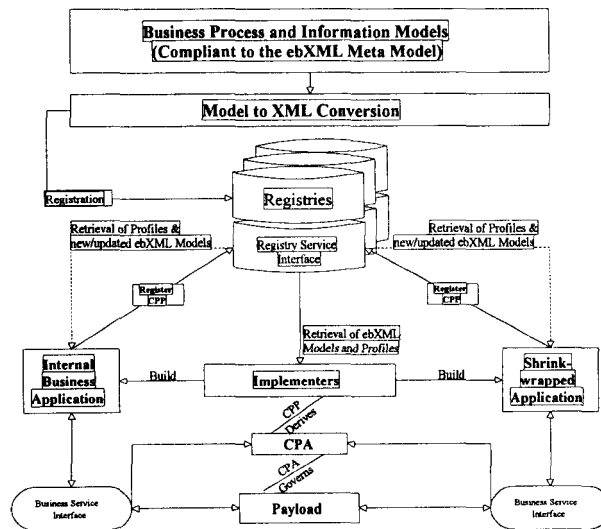
◆ FSV(Functional Service View)

- ▶ Information technology에 초점이 맞추어져 있음
  - ▶ Functional capabilities (어떤 기능이 있는가)
    - ◆ 구현, 검색, 배치 그리고 실행 시나리오에 대한 기능
  - ▶ Service Interface
    - ◆ 사용자 어플리케이션에 대한 인터페이스
    - ◆ 데이터 전송 하부구조에 대한 인터페이스
  - ▶ Protocols and Messaging Service
    - ◆ 상이한 조직의 XML 어휘들과 상호운용을 가능하게 하는 protocol

● ETRI Proprietary      e-Business Solution Fair @ World 2001      12

## ebXML Architecture 모델(3)

- ◆ BOV의 결과인 Business Process and Information Models은 Registry에 저장되고 검색된다. => FSV
  - ▶ Registry는 사용자에 대한 효율적인 인터페이스를 제공해야 한다.
  - ▶ XML 기반 business information은 통일된 방법으로 atomic data 수준까지 검색이 될 수 있는 방법으로 표현되어야 한다.
    - ▶ Item에 대해서 Unique Identifier(UID)를 반드시 가지게 한다.
    - ▶ UID 키들은 다양한 방법의 물리적인 XML구문으로 구현될 수 있다.
      - A pure explicit reference mechanism (example: URN:UID method)
      - A referential method (example: URI:UID / namespace:UID)
      - An object-based reference compatible with W3C Schema (example: URN:complextype name)
      - A datatype based reference (example : ISO 8601:2000 Date/Time/Number datatyping and then legacy datatyping)



ebXML Functional Service View

## ebXML Functional Phases

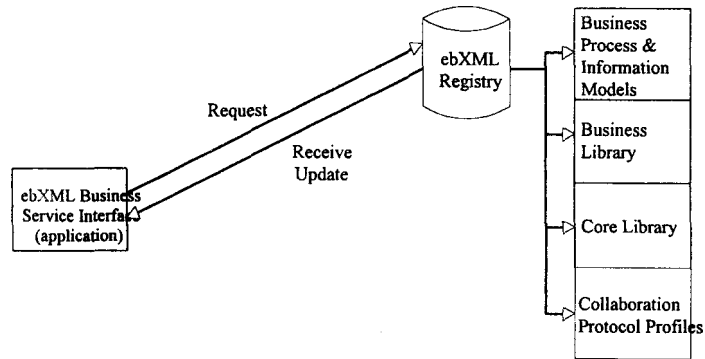
### ebXML Functional Phases(1)

---

#### ◆ Implementation Phase

- ebXML Infrastructure를 위한 어플리케이션 생성 절차를 다룬다.
- ebXML Compliant transaction을 행하고자 할 경우
  1. Trading partner는 가장 먼저 ebXML Framework 명세를 획득
  2. 명세를 검토한 다음에 Core Library와 Business Library를 다운로드
  3. 분석과 검토를 위해서 다른 Trading Partner의 Business Process(business profile에 저장)를 요구
  4. Trading Partner는 자신의 Business Process 정보를 Registry 시스템에 제출



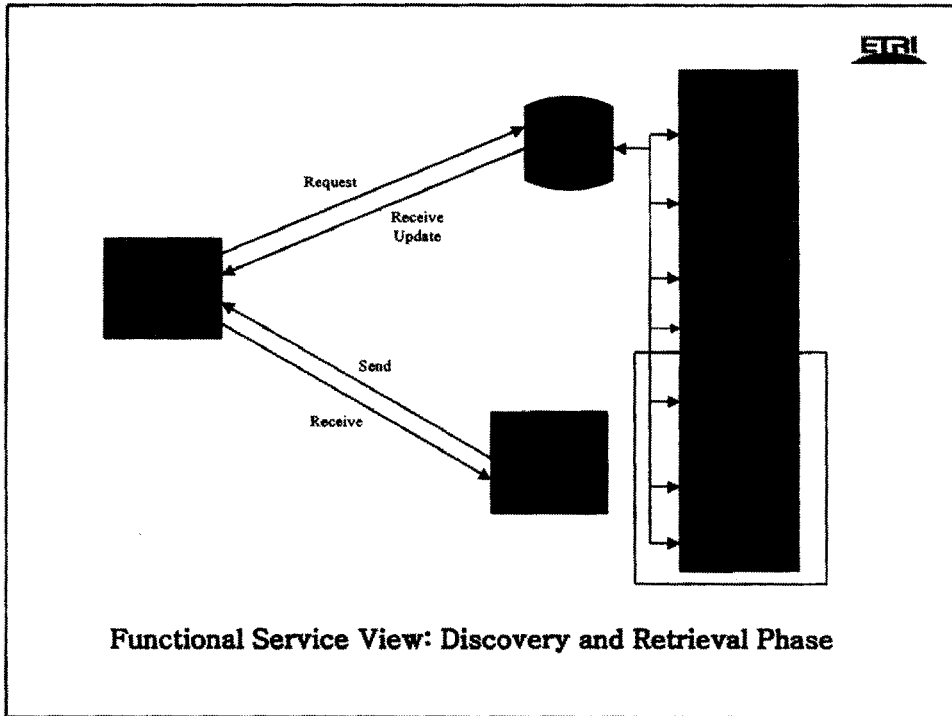


**Functional Service View: Implementation Phase**

## ebXML Functional Phases(2)

### ◆ Discovery and Retrieval Phase

- Trading Partner가 다른 Trading Partner의 business information의 의미를 찾는 것
- ebXML 관련 자원 검색에 대한 모든 범위를 다룬다.
  - ▶ Trading Partner Profile 검색
  - ▶ Core Libraries, Business Object Libraries 갱신
  - ▶ 새로운 Business Process 등록



## ebXML Functional Phases(3)

---

◆ Run Time Phase

- 이 단계에서는 Trading Partner들 사이에 ebXML Message가 ebXML Message Service를 통하여 교환된다.
- 이 단계에서 Registry에 대한 요청이 요구되면 discovery and retrieval 단계로 되돌아 가는 것이다.

The diagram shows two rectangular boxes, each labeled 'ebXML Business Service Interface (application)'. An arrow labeled 'Send' points from the left box to the right box, and an arrow labeled 'Receive' points from the right box back to the left box.

● ETRI Proprietary
e-Business Solution Fair @ World 2001
20

## ebXML Infrastructure

1. Trading Partner Information [CPP and CPA's]
2. Business Process and Information Modeling
3. Core Component and Core Library Functionality
4. Registry Functionality
5. Messaging Service Functionality

### Trading Partner Information[CPP & CPA's](1)

#### ◆ CPP(Collaboration Protocol Profile)

- Trading Partner에 대한 정보를 표현
  - ▶ Contact information
  - ▶ Industry classification
  - ▶ 제공하는 Business Process
  - ▶ Business Service Interface의 요구사항을 표현
- Business Information 교환에 대한 capability
- Trading Partner는 하나이상의 CPP를 Registry에 등록할 수 있다.
- ebXML Message Header를 생성하는데 이용되는 binding 세부 사항들을 포함해야 한다.

## Trading Partner Information[CPP & CPA's](2)

### ◆ CPA (Collaboration Protocol Agreement)

- ▶ Business document의 전송과 수신을 위한 agreement를 기술
  - ▶ CPA는 협상이 이루어진 두 CPP에 대한 intersection이다.
  - ▶ 2 또는 그 이상의 Trading Partner에 의해서 동의를 이루어진 Business Process와 Messaging Service Interface 요구사항을 기술
- ▶ CPA는 Registry에 등록될 수 있다.(필수사항은 아님)



3 Level view of CPA

In scope  
for ebXML

Scope for CPA

```

<CollaborationProtocolProfile>
  <Party>
    <PartyDetails>
      ...
    </PartyDetails>
    <Role>
      <ServiceBinding name="alfa"
        collaborationid="
          channelid="
        </Role>
      <DeliveryChannel
        transport="
        docExchange="
      </DeliveryChannel>
      <DocExchange>
      ...
    </DocExchange>
    <Transport>
      ...
    </Transport>
  </Party>
  <CollaborationProtocol href="this_BP"
    This_BP
  </CollaborationProtocol>
</CollaborationProtocolProfile>

```

## Business Process and Information Modeling(1)

- ◆ Business Process는 Trading Partner들의 상호 작용을 위한 세부사항을 기술한다.
- ◆ Role들 사이의 상호작용은 Business Transaction의 집합으로 나타난다.
- ◆ 각각의 Business Transaction은 Business Document의 교환으로 표현이 된다.
- ◆ Business Document들은 재 사용 가능한 Business Information Component(Core Component)들로 구성이 된다.
- ◆ Registry System내의 Business Process는 UID와 GUID를 가지고 있다.

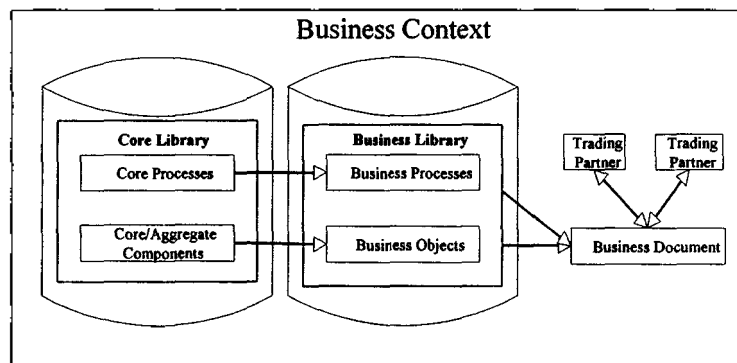
## Business Process and Information Modeling(2)

- ◆ Business Process Specification Schema
  - ▶ Business Transaction과 Business Transaction 집합인 Business Collaboration을 제공한다.
  - ▶ 각각의 Business Transaction은 가용한 여러 표준적 패턴 중 하나의 형태로 실행되어 진다.
    - ▶ 패턴은 partner들 사이에 message 와 signal들의 교환형태를 의미한다.
    - ▶ Service-Service, Agent-Service-Service, Service-Service-Agent, Service-Agent-Service, Agent-Service-Agent
  - ▶ Collaboration Profile과 Collaboration Agreement 생성을 위한 입력이 된다.

```

<EbXmlProcessSpecification>
.....
<document name="doc1">....</document>
<document name="doc2">....</document>
...
<business-transaction name="trans1">
  <request type="doc1"/>
  <response type="doc2" status="success"/>
  <response type="doc3" status="failure"/>
</business-transaction>
<business-transaction name="trans2"> ... </business-transaction>
....
<binary-collaboration name="bc1" initiator="buy" responder="sell">
  <business-transaction-activity name="trans1" from="buy" to="sell"/>
  <business-transaction-activity name="trans2" from="sell" to="buy"/>
  <business-transaction-activity name="trans3" from="buy" to="sell"/>
  <start to="trans1"/>
  <transition from="trans1" to="trans2"/>
  <transition from="trans2" to="trans3"/>
  <success from="trans3"/>
  <failure from="trans1" condition="failure"/>
</binary-collaboration>
.....
</EbXmlProcessSpecification>

```



ebXML Business Process and Information Modeling layer

## Core Components and Core Library Functionality(1)

- ◆ Core Component는 real world 비즈니스 개념에 대한 정보와 개념들 간의 관계와 context정보를 담고 있다.
  - 하나의 business information를 Core component라 하고 이러한 business information의 집합을 Aggregate Component라 한다.
- ◆ Context는 Core 또는 Aggregate Component가 어떻게 특정 ebXML scenario에 사용되는지를 기술한다.
- ◆ ebXML Core Component project team은 Core Component 초기 집합을 정의한다.

## Core Components and Core Library Functionality(2)

- ◆ Core Component는 ebXML Registry 메커니즘을 사용하여 저장되고 검색될 수 있어야 한다.
  - 유일하게 식별 가능
- ◆ Core Component는 사업적, 기술적 요구를 만족하는 정보의 최소집합을 가지고 있어야 한다.
- ◆ XML 구문으로 표현이 가능해야 한다.
- ◆ Core Component는 다른 XML vocabulary의 XML 엘리먼트와 인터페이스를 가질 수 있다.

## Context

- ◆ 비즈니스 영역을 구분하기 위한 계층적 표현방법
- ◆ 일반적 의미의 Core Component를 특정 비즈니스 영역에 적용하기 위한 방법을 제공한다.
  - 예) 자동차 공장에서 화학공장에서부터 페인트를 구입하고자 할 때 Context 값은 다음과 같다.

Industry(buyer)	Automotive
Industry(Seller)	Chemicals
Process	Procurement
Product Classification	Paint
Region(buyer)	France
Region(seller)	U.S.

```

<ContextRules>
  <Rule>
    <Condition Test="Region='United States' "/>
    <Action ApplyTo="Address">
      <Require>
        <Field Name = "State" />
      </Require>
    </Action>
  </Rule>
  <Rule Priority= "10">
    <Condition Test="Process='RFQ' and Industry='insurance' "/>
    <Action ApplyTo="Party">
      <Add>
        <Field Name = "QualifyingInfo" Type="QualifyingInfo" />
      </Add>
    </Action>
  </Rule>
  <Rule>
    <Condition Test="Industry='Automotive' "/>
    <Action ApplyTo="QualifyingInfo">
      <Add>
        <Field Name = "DrivingRecord" Type="DrivingRecord" />
        <Field Name = "CarDescription" Type="CarDescription" />
        <Field Name = "DrivingHabits" Type="DrivingHabits" />
      </Add>
      <Require>
      </Require>
    </Action>
  </Rule>
</ContextRules>

```



## Registry(1)

- ◆ Registry는 등록된 아이템의 metadata에 대한 인터페이스를 유지하는 컴포넌트이다.
- ◆ ebXML Registry에 대한 접근은 Registry Service에 의해 드러난 인터페이스(API)를 통해서 제공되어진다.
- ◆ Registry는 application 인터페이스와 human 인터페이스를 모두 가지고 있어야 한다.
  - Content 기반 검색
- ◆ 적절한 보안 프로토콜이 요구된다.

## Registry(2)

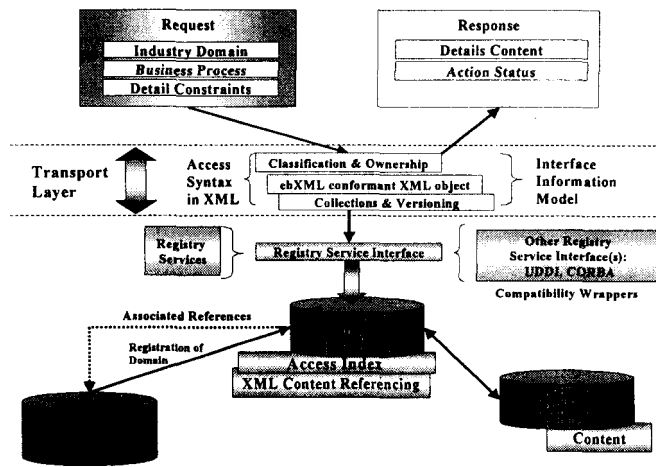
- ◆ Registry는 multibyte 문자 집합을 사용하는 구문으로 표현된 item의 저장소를 제공해야 한다.
- ◆ Submitting Organization에 의해서 정의된 각각의 granularity에 의해서 각각의 Registry item은 유일하게 식별가능 해야 한다.
- ◆ Registry item은 item들에 대한
  - [식별, 명명 그리고 서술에 관련된 정보를 제공하기 위해서]
  - [Item의 관리와 접근 상태를 제공하기 위해서]
  - [영구성과 변화 가능성을 정의 하기 위해서]
  - [이미 정의된 분류 구조에 따라 분류하기 위해서]
  - [파일 표현 타입을 선언하기 위해서]
  - [submitting and responsible organization을 식별하기 위해서]
 구조화 되어야만 한다.

## Registry(3)

- ◆ Registry access 매커니즘에 대한 질의구문은 물리적 시스템에 독립적이어야 한다.
- ◆ Messaging service를 통해 Registry와 Communication이 이루어진다.
- ◆ XML 엘리먼트는 Registry 서비스를 통해서 관리되는 아이템에 대한 표준 메타 데이터를 제공한다.
  - 이것은 Registry는 분산되고 다른 Registry와 상호 참조가능하도록 한다.

## Registry(4)

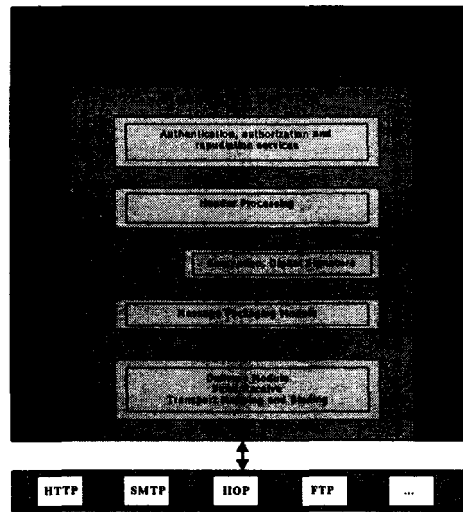
- ◆ 비표준적인 구현 참고사항
  - Registry 내에 있는 Business Process and Information Model은 다양한 분류 체계에 따라 저장 될 수 있다.
  - ISO11179/3은 Registry 구현 참조 모델이 될 수 있다.
  - Registry item과 metadata는 직접 접근을 위해 http만을 사용하는 XML 기반 URI addressing이 가능하다.
  - Registry service는 Registry interface만 따르면 여러 정보 모델을 이용해도 상관없다.
  - GUID는 DC 128 GUID와 같은 표준적인 알고리즘을 사용하는 것이 좋다.



Overall Registry / Repository Architecture

## Messaging Service Functionality(1)

- ◆ Business message를 교환하는 표준적인 방법을 제공한다.
- ◆ ebXML Message는 Header와 Payload 부분으로 나누어 진다.
- ◆ ebXML Messaging Service의 3단계
  - **Abstract Service Interface**
  - Functions provided by the **Messaging Service Layer**
  - Mapping to underlying **transport service(s)**.



The Messaging Service Architecture

## Messaging Service Functionality(2)

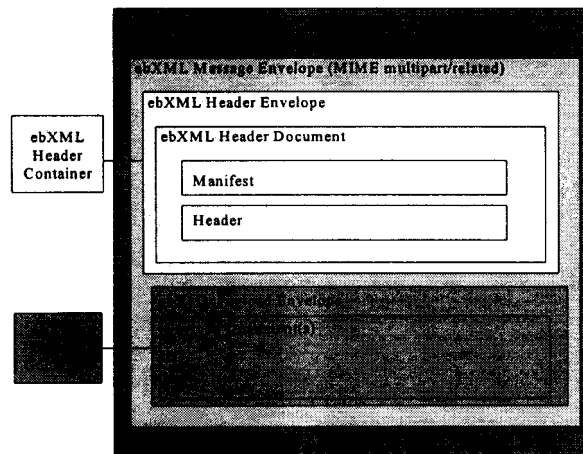
### ◆ Formal Functionality

#### ▶ ebXML Messaging Service는

- ▶ Message교환에 있어서 secure, consistent and reliable 메커니즘을 제공
- ▶ Registry 에서 application 까지 분산되어 있는 ebXML 컴포넌트들 사이의 모든 Message 형식을 정의한다.
- ▶ Payload 내에는 어떤 제약사항도 기술하지 않는다.
- ▶ Simplex방식의 통신을 한다.
- ▶ Sequence of payload 또는 multiple message를 지원해야 한다.
- ▶ CPA에 정의된 "rules of engagement"를 준수해야 한다.

## Messaging Service Functionality(3)

- ◆ Abstract level의 abstract interface
  - Send : Message를 보냄
  - Receive : Message를 받음
  - Notify : expected 또는 unexpected event에 대한 통보
  - Inquire: Message 교환 상태에 대한 질의 방법
- ◆ Interface with internal system
  - 내부 시스템으로 도착된 Message에 대한 routing
  - Error notification



ebXML Message Structure

## References

---

- ◆ <http://www.ebXML.org>
- ◆ [http://www.ebxml.org/project\\_teams/technical\\_arch/private/](http://www.ebxml.org/project_teams/technical_arch/private/)
- ◆ <http://lists.ebxml.org/archives/>