

e-비즈니스 통합을 위한 시맨틱 웹의 활용

An Application of the Semantic Web for e-Business Integration

장태우(Tai-Woo Chang)*, 신기태(Kitae Shin)**, 박진우(Jinwoo Park)***

초 록

본 연구에서는 시맨틱 웹을 구성할 수 있도록 하는 지식표현, 온톨로지, 에이전트를 통해 급변하는 전자거래 환경을 극복하고 이질적이고 분산되어 있는 정보들을 통합할 수 있도록 하기 위한 방법론을 제시하고자 한다. 이를 위해 모델링의 기반이 되는 구문론적/의미론적 통합 및 에이전트의 운용방법론에 대해 논하며, 메타 모델링에 바탕을 두는 온톨로지의 구축에 중점을 두고 설명한다. IRDS (Information Resource Dictionary Standard) 프레임워크에 기반한 구조에 따라 OWL(Web Ontology Language)을 사용하여 온톨로지를 구체화하며, 프로세스 표현을 위해 PSL(Process Specification Language) ontology의 용이들을 중간 개체로 구성하여 차용한 예를 제시한다. 제시한 프레임워크를 통해 정보 자원들과 비즈니스 프로세스를 정의하고 공유하며 관리함으로써 중복성과 불일치성을 제거할 수 있을 것이다.

ABSTRACT

In this study, we propose a methodology of making it easy to integrate the heterogeneous and distributed information of e-Business environments with the semantic web components, knowledge representation, ontology and agents. It is based on a collaborative interoperable environment, which should be constructed by meta-modeling and employing agents, and the syntactic/semantic integration. We adopt OWL(Web Ontology Language) to specify ontology in a layered architecture for each agent to use, which is based on the IRDS (Information Resource Dictionary Standard) framework. And we give an illustrative example using the terms from PSL(Process Specification Language)-ontology. By defining, managing and sharing resources and business-processes on the proposed framework, it could be possible to get rid of the cause of redundancy and inconsistency.

키워드 : e-비즈니스 통합, 시맨틱 웹, 온톨로지, 에이전트

e-Business Integration, Semantic Web, Ontology, Agents

* 한국전사통신연구원 우정기술연구센터 선임연구원

** 대전대학교 산업시스템공학과 부교수

*** 서울대학교 산업공학과 교수

1. 서 론

전자거래에 관련한 정보기술의 생명주기가 짧아지고 정보시스템을 구축하기 위한 인력 및 프로젝트의 특성이 다양화 되어가는 등 e-비즈니스 환경이 변화하고 있다. 또한 이질적이고 분산되어 있는 전자상거래 상의 정보들은 그 내용 또한 기계가 이해하기에는 어려움이 많다. 이러한 환경적 변화와 정보기술의 부족함을 극복하기 위해서는 비용요소를 줄일 수 있는 자기기술적(self-describing) 규약이나 소프트웨어 설계 방법들이 필요하게 되었다. 전자거래 환경, 특히 기계가 이해할 수 있는 웹 콘텐츠의 설계를 위해 시맨틱 웹이 도입될 필요가 있고[2]. 이러한 관점에서 시맨틱 웹을 구성할 수 있도록 하는 지식표현, 온톨로지, 에이전트에 대한 통합 프레임워크가 요구된다. 특히 자기기술적 규약이라 할 수 있는 XML이 웹 문서들의 문법(Syntax)을 담당하여 왔다고 한다면, 기계가 이해할 수 있는 웹 콘텐츠의 통합적인 구현을 위해서는 지식과 의미체계(Semantics)에 대한 명확한 정의가 요구되는데, 주어진 문제영역(domain)에 대한 지식과 의미체계의 구체적 표현과 정의 및 공유를 위해서는 온톨로지와 온톨로지에 기반한 분석이 필요하다[4].

또한 전자거래 상의 소프트웨어 에이전트가 질의를 하거나 질의를 통한 결과를 내놓기 위해서는 개체와 개체간의 관계 및 기능에 대한 개념적 모델을 파악할 수 있어야 하며, 개념적 모델에 대한 명확한 설계사양, 즉 온톨로지가 있어야 에이전트의 운용이 가능한 것이다. 서로간의 거래에서 통용되는 개체, 개체

간의 관계와 기능, 조건 및 규칙들을 통합적인 온톨로지로 체계적으로 구축함으로써 정확하고 자동화된 거래를 구현할 수 있으며, 거래를 이루는 프로세스 역시 온톨로지를 통하여 의미를 공유할 수 있게 된다.

본 연구에서는 메타 모델링 프레임워크를 바탕으로 온톨로지를 구축할 수 있도록, 데이터 및 프로세스에 대한 개념적 모델을 네 계층의 스키마 구조를 통해 구체화 방법을 제안하며, XML 및 그 어플리케이션인 OWL(Web Ontology Language)을 이용한다. 프로세스 모델링 과정에서는 PSL-ontology에서 정의된 용어들을 사용한다.

이하 2절에서는 정보시스템 통합 및 온톨로지에 대한 기존의 연구현황과 본 연구에서 사용한 표준 및 기술에 대해 간략히 소개한다. 3절에서 온톨로지와 에이전트를 중심으로 하는 전자거래 통합 프레임워크에 대해 설명하고, 온톨로지 구축 방법론에 대해 언급한다. 4절에서 지식표현으로써 간단한 e-비즈니스 시나리오를 제시하고 그에 따른 온톨로지와 이를 구체화한 예를 들고, 5절에서 결론을 내린다.

2. 관련 연구

컴퓨터 및 인터넷 등을 포함하는 IT 분야에서는 상호운용성(interoperability)을 바탕으로 하는 통합에 대한 요구가 점점증하고 있다. 기업의 모델링 및 통합의 단계 및 양상에 대해 비즈니스 통합, 어플리케이션 통합, 물리적 시스템에 대한 통합의 세 가지로 구분할

수 있다[15].

전자상거래 분야 역시 기업들의 비즈니스 영역 자체에 대한 통합과 어플리케이션에 대한 통합 등을 위해 각종 방법론 및 프레임워크들이 발생하게 되었고, 특히 분산된 데이터에 대한 통합의 필요성이 증대되고 있다. 분산된 데이터의 처리 방법들 중 이종 시스템간의 데이터 공유와 시스템 자율성이 중요한 다중 데이터베이스에 대한 결합 모형에서는 데이터베이스의 스키마를 통합할 필요가 존재하며, 데이터베이스 구성 관점(이름, 포맷, 단위 등)의 불일치, 의미론적 문제, 스키마 추상화 레벨의 불일치 등과 같은 문제점들이 있어 통합에 어려움이 따른다[3].

정보시스템에 기반하는 전자상거래 환경에서는 국제적 표준으로서의 프레임워크들이 자리잡고 있으며, 로제타넷(RosettaNet), ebXML, OAGIS, BizTalk 등이 그 대표적인 예라 할 수 있다. 하지만, 이들 전자상거래 프레임워크들은 추진 주체와 대상 산업분야에 따라서 별도의 표준으로 받아들여지고 있고, 이들에 대한 통합의 필요성이 여전히 존재하고 있으나, 각 프레임워크에는 비즈니스 프로세스에 대한 모형화 또는 메타 모형화가 되어 있지 않은 경우도 있고 의미론적으로 기초적인 기능을 제공하는 사전과 같은 것을 제공하지 않는 경우도 있어 통합에 어려움이 따를 것으로 예상된다. 로제타넷 표준에서는 거래에서 사용되는 모든 용어들에 대해서 업무 및 기술 사전(Business and Technical Dictionaries)에 정의해 놓았으나 용어들간의 관계는 정의되지 않았으며 여타 표준들에서 정의된 용어와 의미가 일치하지 않는 등의 단점을 가지고

있어서 의미체계가 통합된 전자상거래를 구현하기 위해서는 온톨로지와 같이 정형화된 도구가 필요하다고 할 수 있다. 이러한 분제는 OAGIS의 사전에서도 마찬가지로 나타나고 있다.

Smith와 Poulter 역시 전자상거래를 지원하기 위한 에이전트 기반 시스템이 오해 없이 상호작용할 수 있도록 하려면 공통된 온톨로지가 필요함을 주장하고 있다[12]. 예를 들어, 판매자와 구매자가 어떤 공통적인 속성을 가지는지, 응답제한시간과 실행제한시간은 어떤 공통적인 의미를 가지고 계산되는 지 등에 대한 의미체계가 없다면 각 소프트웨어마다 구현 구조가 달라지게 될 수 있기 때문에, 공유할 수 있는 의미체계와 명확한 상세 설계인 온톨로지가 필요한 것이다.

인공지능 분야에서 재조명되어 온 개념인 온톨로지는 일반적으로 '개념화(conceptualization)에 대한 명시적인 구체화(specification)'라고 정의되는데[6], 흔히 사용되는 분류학(Taxonomy)과 다른 점은 풍부한 내부 구조를 가지고 있고 함의를 반영하는 것이라 할 수 있다[13]. Mena 등은 온톨로지가 에이전트를 위한 개념 및 관계들에 대한 기술이라고 정의하여 정보시스템에서 온톨로지의 필요성에 대해 역설하고 있다[10].

온톨로지를 정의하고 구축하기 위한 연구들이 많이 있었지만, 전자상거래에 대한 영역에 대한 것은 근래에 들어 늘어나고 있다. Wang의 연구에서는 전자상거래 중 입찰에 대한 프로세스에 영역을 한정하여 지식표현 언어의 일종인 Telos를 이용하여 온톨로지를 구축하는 방법을 제안하였으나[16], 프로세스

의 표현에 대한 온톨로지는 언급하지 않고 있다. Kayed와 Colomb은 온톨로지의 구축을 하나의 생명주기로 보고 Building, Manipulation, Maintenance의 세 단계로 뭉뚱그려서 설명하고 있으며, 입찰에 대한 시나리오에 대해서 EDI 메시지와 함께 실례를 제시하고 있다[9]. 두 연구 모두 일반화되어 있지 않은 기술 방법을 사용하고 있다는 단점을 가지고 있다.

본 연구에서는 앞서 언급한 단계들 중 비즈니스 및 어플리케이션에 대한 통합에 초점을 맞추고, 비즈니스 통합을 위해 프로세스 및 데이터 모형화를, 어플리케이션 통합을 위해 정보의 재사용/공유와 자료의 의미 및 형식에 대한 온톨로지의 구축을 주 연구 과제로 삼는다. 이를 위해 본 연구에서 사용하는 프로세스 기술 표준인 PSI,은 프로세스를 위한 온톨로지와 표현방식을 제공하는 표준(ISO 18629)으로 미국 NIST에서 추진중이다[11]. 프로세스와 관련된 다양한 어플리케이션들을 통합할 수 있도록 하기 위해 교환되는 내용을 중립적으로 표현할 수 있도록 개발된 것이다. 현재는 설계 및 제조 생명주기에서의 프로세스를 포함하는 영역에 연구가 제한되어 있는 상태이지만, 개발된 프로세스 기술방법의 표준성과 중립성, 의미표현가능성 등은 충분히 비즈니스 프로세스를 기술하는 데에도 이용할 수 있다. PSL-ontology에 포함되는 술어(predicate)들에는 object, activity, activity-occurrence, timepoint와 같은 기본적인 개체와 beginof, endof와 같은 함수 기호, before, occurrence-of, participates-in과 같이 요소들간의 관계를 표현하는 것들이 있다.

3. 전자거래 통합 프레임워크

기업간 거래의 상호운용성을 가능하게 하기 위해 많은 노력들이 있었으나, 현재의 방법들에서는 비즈니스 프로세스에 대한 확인 및 검증이 어려울뿐더러 표준의 변화에 따른 시스템의 변경과 예외처리 등의 변화가 운용성을 저하시키는 요인이 되고 있다. 이를 해결하기 위해서는 메타 모델링 및 에이전트 모델을 주축으로 해서 협업이 가능한 환경을 구축하는 것이 우선되어야 한다.

3.1 협업 환경

각종 문서 및 프로세스에서 사용되는 용어들의 의미와 관계에 대한 메타 모형이 우선적으로 갖추어져야 한다. 메타데이터는 데이터에 대한 데이터라 할 수 있으며, 이러한 메타데이터에 대한 모형이 메타 모형이라 할 수 있다.

국제 표준화 단체인 ISO/IEC에서는 메타 모델링 프레임워크로 IRDS(Information Resource Dictionary Standard; ISO/IEC 10027:1990)를 제정하였으며, 이 프레임워크에서의 어플리케이션 환경과 개발 환경은 다음과 같은 네 단계로 구성된 형태를 가진다[7].

- IRI Definition Schema Level : 하위 단계인 IRD Definition Level의 객체들에 대해 기술할 수 있도록 하는 메타-메타 모형을 규정
- IRD Definition Level : 스키마나 어플리케이션 프로그램이 표현되는 언어를 규정

- IRD Level : DB 스키마, 어플리케이션 프로그램, 워크플로우와 같은 활동 규정
- Application Level : 어플리케이션 데이터를 포함

Jarke 등의 연구에서는 메타 모형에 기반한 상호운용성 및 적용성을 분석하기 위해 다이나몬드 모형으로 불리는 프레임워크를 개발했다[8]. 복표를 이루기 위해 세 가지의 측면에서 메타 모형이 구축된다고 보고 있는데, 프로세스는 분제영역의 활동들에 대한 집합을, 온톨로지는 의미론적인 문제에 대한 것을, 표기법은 구문론적인 문제에 대한 것을 의미한다.

이들 프레임워크들에 따른 메타 모델링의 결과물로 일관된 표기법에 따르는 온톨로지가 생성된다고 할 수 있으며, 이것들이 Jarke 등의 연구에서 언급한 메타 모형의 기능들(클래스, 속성, 제약사항, 공통 용어의 정의 등)을 제공한다는 것은 4절에서 설명하도록 한다.

일반적으로 에이전트가 상호 작용하는 환경은 외부 환경, 내부 환경, 그리고 에이전트 간의 정보구조를 갖는다[14]. 이러한 환경에서 에이전트들은 통신, 내·외부 인터페이스 정보지원관리의 세 가지 단계로 구분할 수 있을 것이다. 에이전트들을 메시지 관리/데이터베이스 연결 및 질의 관리/워크플로우 관리/모니터링 등의 기능에 따라서 3단계로 구분하여 각각 할당하여 모형화해야 한다. 에이전트들의 구분에서는 인터페이스 단계의 에이전트들이 통합 환경에서 중요한 역할을 차지하게 되며, 데이터베이스와 연결할 수 있는 또 다른

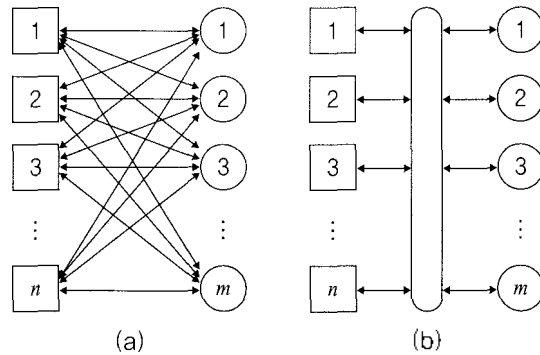
에이전트와의 통신과 구축된 온톨로지를 이용한 의미 관리의 기능을 담당해야 한다.

3.2 구문론적·의미론적 통합

XML은 자기-기술적 규약(self-describing protocol)으로 받아들여지고 있으며, 현재 추진중인 전자상거래 표준들도 모두 구문론적 도구로 XML을 사용하여 작성되고 있다. 본 연구 역시 모든 구체화 도구로 XML을 사용한다.

2절에서 전자상거래 프레임워크의 현황과 의미론적 통합의 필요성을 언급한 바 있다. 의미론적 통합을 위해서는 개념모형에 대한 명시적인 구체화라 정의되는 온톨로지에 대한 구축이 필요하며, 구축된 온톨로지를 통해 정보의 공유 및 재사용이 가능하게 될 것이다. 온톨로지는 메타 모델링에 바탕을 두고 구축하게 되며, 이를 기반으로 에이전트를 운용할 수 있게 해야 할 것이다. 구축된 온톨로지를 이용하면 에이전트들은 <그림 1>에서 보는 바와 같이 'n * m' 가지 의미를 알아야 하는 대신 'n + m' 가지의 의미만을 알면 되므로 중복성과 불일치성을 줄일 수 있게 된다.

분산데이터의 이름 및 유형의 통합적 관리에 의한 데이터베이스 스키마의 통합을 통해 온톨로지의 기초를 구축하는 것도 가능하며, 일부 연구에서 기초적인 방법으로 이를 수행한 바가 있다[1]. 즉 스키마 통합의 과정에서 데이터들의 속성 및 유형을 통해 개체들 간의 관계(동의어, 유의어, 부모/자식 등)를 파악할 수 있게 되며 이를 기반으로 초기 온톨로지의 구축이 가능하게 되는 것이다.



〈그림 1〉 온톨로지의 존재 여부에 따른 비교

본 연구에서 제시하는 모델링 방법은 저자들의 선행연구[5]에서 설명한 시스템공학적 접근방법에 따라 크게 물리적인 모델링 단계(1~4)와 논리적인 모델링 단계(5~7), 모델링 이후 단계(8~10)로 구분할 수 있으며, 다음과 같은 단계에 따라 진행된다.

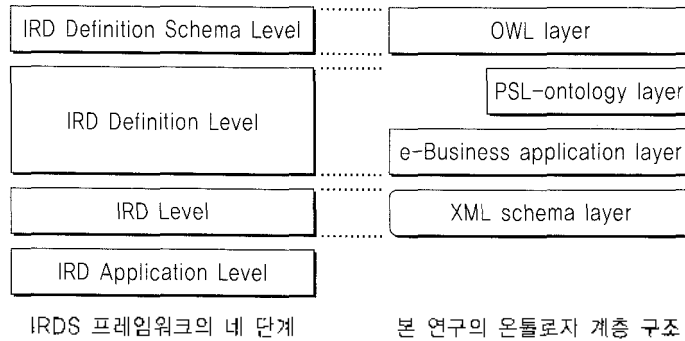
1. 비즈니스 범위 설정
2. 범위에 포함되는 용어 식별
3. 개체 및 개체들 간의 관계 식별
4. 개체들 간의 기능 식별
5. 데이터의 논리적 구성
6. 프로세스 도출
7. 온톨로지 구성
8. 온톨로지 평가, 유지, 보수
9. 문제영역 지식의 축적
10. 온톨로지를 참고하는 에이전트의 기능 구현

단계 2의 결과와 단계 3의 식별 과정을 거쳐 단계 5를 통해서, 전자상거래 상에서 사용

되는 자원 및 데이터에 대해 식별하고 데이터 간의 관계를 논리적으로 구성하는 것이 기존의 데이터 모형화와 관련이 있으며, 단계 4, 단계 6을 통해 대상 시스템을 이루는 개체들 간의 기능을 식별하여 기능들간의 관계와 기능들의 연결인 프로세스를 도출하는 것이 기존의 프로세스 모델링과 관계가 있다.

단계 6까지가 사람들 간의 정보 공유와 이해까지 포괄한 내용에 중심을 뒀다면, 단계 7은 기계 또는 소프트웨어 에이전트가 이해할 수 있는 내용의 기술에 중심을 둔다고 할 수 있다. 개념적 모델을 구체화하는 단계로 데이터 부분을 먼저 구성하고 프로세스와 연관된 데이터 및 메타데이터를 식별한 후, 프로세스에 대해 구성하는 하위 단계를 거친다.

온톨로지의 전체적인 구조는 OWL 계층, PSL-ontology 계층, e-Business 응용 계층, XML schema 계층으로 나누어 종적으로 구성한다. 본 연구에서 구성한 프레임워크는 앞서 언급한 IRDS의 프레임워크와 같이 단계별로 구분되어 있으나, 온톨로지를 구축하기



〈그림 2〉 온톨로지 계층 구조와 IRDS 프레임워크 단계들과의 비교

위한 OWL 및 PSL과 같은 기술 표준의 사용과 구분으로 되어 있어 단계별로 동일하게 구성되어 있지는 않다. 각 프레임워크와 구조는 〈그림 2〉와 같이 비교하여 구분할 수 있다. 본 연구의 어플리케이션 계층에는 실제 어플리케이션에 적용하기 위해 사용되는 메타 데이터가 구성되기 때문에 인스턴스 데이터가 저장되는 IRD Application Level과는 차이점을 갖는다.

온톨로지의 계층 구조는 구체화 단계에서는 owl, psl, eb 등의 XML 이름공간(namespace)으로 구별될 것이다.

3.3 통합 프레임워크

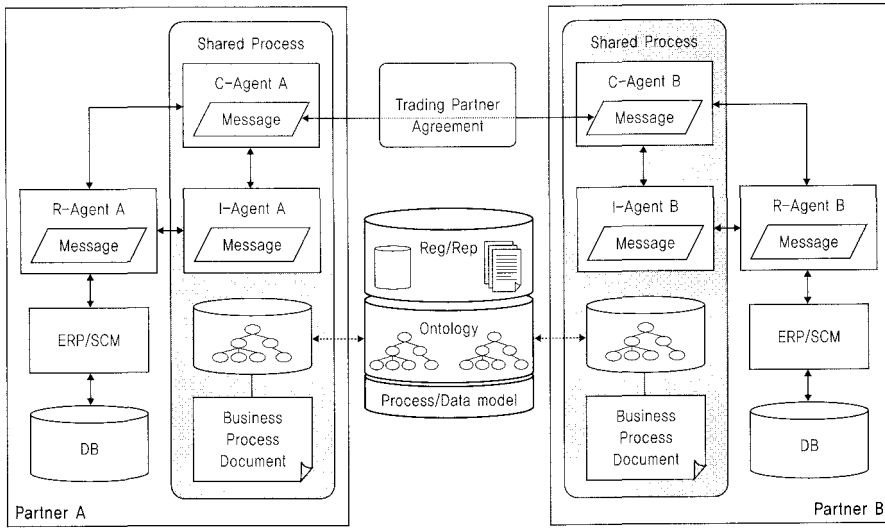
3.1절에서 설명한 협업 환경을 위한 온톨로지와 에이전트의 운용에 대한 통합 프레임워크는 〈그림 3〉과 같은 구조를 갖게 된다. 양측 거래당사자 A와 B의 전자거래를 위한 공유 프로세스 및 데이터들의 전체적인 모습을 보이고, 온톨로지와 에이전트의 위치에 대해 나

타내도록 하였다.

프로세스 및 데이터 모델을 표현하기 위한 온톨로지와 온톨로지에 기반을 두고 작성된 모델이 함께 등록소/지장소에 함께 저장된다. 저장된 온톨로지는 거래당사자 모두가 함께 공유하게 되며, 서로간의 통신은 메시지에 기만하게 된다.

운용되는 에이전트들은 통신(C-), 인터페이스(I-), 자원(R-) 에이전트로 구분되며, 이들은 상대편 거래당사자의 에이전트, 저장된 공유 온톨로지, 기존 시스템 및 데이터베이스에 각각 연결하여 작업을 수행하게 한다. 각 거래당사자의 C-에이전트는 거래협약에 따라 메시지를 상호 교환함으로써 거래를 진행하고, R-에이전트는 ERP/SCM 같은 기존 시스템과 연결하여 개별 데이터베이스에 인스턴스들을 저장할 수 있도록 한다. I-에이전트는 통합된 거래 환경에서의 프로세스 진행을 위해 공유되는 온톨로지를 참조하여 C-에이전트 및 R-에이전트의 활동을 지원하도록 한다.

이러한 구조를 통해 온톨로지의 사용 및 에

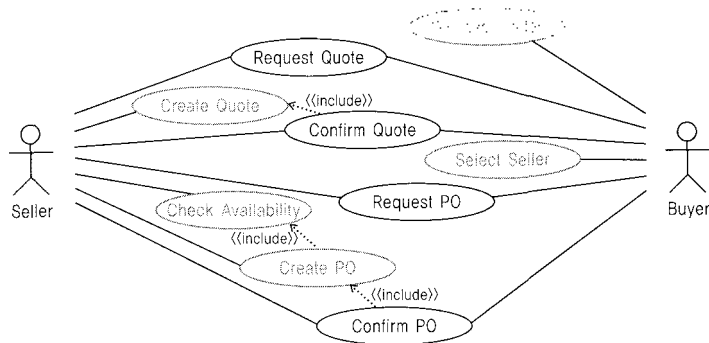


〈그림 3〉 통합 프레임워크의 전체적인 구조

이전트의 운용을 통한 통합된 전자거래가 가능하게 된다. 전체 프레임워크의 기반이 되는 온톨로지의 구체화 방법 및 표현 구조에 대해서는 다음 절에 설명한다.

4. 온톨로지의 구체화

전자상거래의 구매주문 관련 예를 통해 OWL로 표현하는 온톨로지에 대해 구체화한 결과들을 간략하게 보이도록 한다. 먼저 구매주문과 관련한 판매자와 구매자간의 시나리



〈그림 4〉 견적 및 주문에 관련한 유스케이스 다이어그램

오를 유스케이스(Use-case) 다이어그램으로 간략하게 나타내면 <그림 4>와 같다. 공유되지 않는 개별 프로세스는 본 연구에서 대상으로 하는 공유 온톨로지로 구성할 필요가 없으며, 회색으로 표현해 놓았다.

견적이나 주문에 대한 요청 및 확정 전에는 응답확인(Acknowledgement)가 있어야 하며, 확인까지의 시간(time to acknowledgement)과 실행까지의 시간(time to perform) 등이 제약조건으로 존재한다.

이러한 시나리오에 대한 데이터 및 프로세스 모델링이 우선적으로 필요하며, 데이터 모델링은 의미망(Semantic network)이나 UML 클래스 다이어그램으로 구성할 수 있으며, 프로세스 모델링은 UML 시퀀스 다이어그램 등으로 구성할 수 있다. 이러한 과정은 4절에서 제시한 접근방법 단계들 중 5-6단계에서 수행하게 된다.

모델링한 결과물에 따라 4절에서 제시한 구조에 따라 온톨로지를 구성하게 되는데 본 연구에서는 OWL을 이용한다. RDF에서 제시한 메타데이터 모델링 프레임워크에 시맨틱을 추가한 사양으로 더욱 다양한 의미를 포함할 수 있고, W3C에서 웹 온톨로지의 표준으로 고려되고 있기 때문이다. 구체적인 OWL 문서에 대해서는 데이터 모델과 관련한 표현을 먼저 상술하고, 프로세스 모델과 그 구성요소 및 표현을 나중에 열거하기로 한다.

4.1 데이터 표현

거래당사자(판매자, 구매자)와 같은 개체에 대한 데이터 모델의 OWL 표현은 다음과

같다. 개체들은 owl:Class로서 정의된다.

```
<owl:Class rdf:ID = "Partner"/>
<owl:Class rdf:ID = "Seller">
  <rdfs:subClassOf rdf:resource = "#Partner"/>
</owl:Class>
<owl:Class rdf:ID = "Buyer">
  <rdfs:subClassOf rdf:resource = "#Partner"/>
</owl:Class>
```

거래당사자의 속성에 대한 예로 고유번호(ID)에 대한 표현은 다음과 같다. 속성은 기업 식별기호 체계인 DUNS에 따르는 값을 갖도록 지정하였다.

```
<owl:DatatypeProperty rdf:ID = "partnerId">
  <rdfs:domain rdf:resource = "#Partner"/>
  <rdfs:range rdf:resource =
    "http://147.46.236.48:88/eBusiness/eb-
    du#DUNS"/>
  <rdf:type rdf:resource =
    "http://www.daml.org/2001/03/daml+oil
    #UniqueProperty"/>
</owl:DatatypeProperty>
```

DUNS를 데이터 형식으로 갖는다고 했을 때 이에 대한 표현은 owl:DatatypeProperty로 정의하고 rdfs:range으로 속성을 지정해주며, DUNS에 대한 XML Schema 표현은 아래와 같이 패턴을 고려하여 별도로 정의해준다. 참고로, DUNS 코드는 '00-825-574 4'와 같이 8자리에 오류 체크를 위한 1자리를 더한 형태로 구성된다.

```

<xsd:schema xmlns:xsd =
"http://www.w3.org/2000/10/XMLSchema">
<xsd:simpleType name = "DUNS">
  <xsd:restriction base = "xsd:string">
    <xsd:pattern value = "\d{2}-\d{3}-\d{3}-
\d{1}" />
  </xsd:restriction>
</xsd:simpleType>

```

이와 같이 표현된 데이터 모델과 형식 정의는 <그림 5>의 (a)와 같은 의미망으로 동일하게 나타낼 수 있으며, 이를 (b)의 UML 클래스 다이어그램과 비교할 수 있다. 그림에서 볼 수 있듯이, UML의 경우에는 그래픽 위주의 표현은 풍부하나 텍스트로 표현되는 표준화된 문법구조가 없고, 구체화(specification)와 의미공유가 어렵다는 단점이 있다.

다른 비즈니스 영역에서 사용되는 Party, Supplier, Customer와 같은 개체들을 위한 데이터 모델에 대한 OWL 표현 역시 상기 설명

한 바와 같이 정의되며, C-에이전트가 사용하게 될 공유 온톨로지는 다음과 같이 표현될 수 있다.

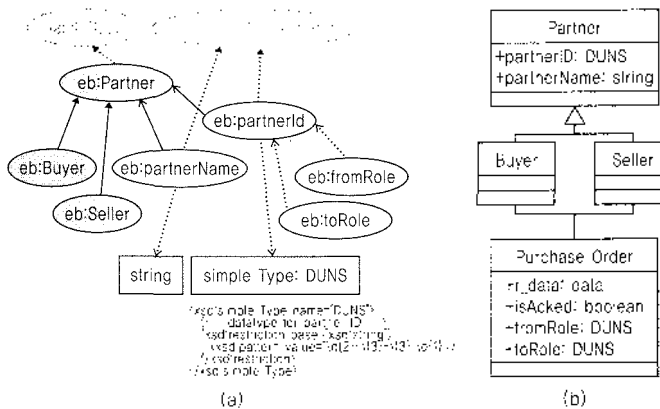
```

<owl:Class rdf:ID = "Party">
  <owl:sameAs rdf:resource = "# Partner" />
</owl:Class>
<owl:Class rdf:ID = "Supplier">
  <owl:sameAs rdf:resource = "# Seller" />
</owl:Class>
<owl:Class rdf:ID = "Customer">
  <owl:sameAs rdf:resource = "# Buyer" />
</owl:Class>

```

4.2 프로세스 표현

프로세스에 관련된 용어들은 PSL-ontology의 용어들을 차용하여 그 의미를 명확히 할 수 있으며, 전자상거래에 관련된 용어들은 <그림 3>에서 나타낸 바와 같이 PSL-ontology



<그림 5> 본 연구의 데이터 표현과 UML과의 비교

계층의 하위 계층으로 연결시켜 표현하도록 하는 것이다. 프로세스 표현을 위해서 프로세스를 활동(Activity), 활동의 발생 및 순서(Occurrence), 제약조건(Constraint)들로 구성되는 또 다른 활동으로 가정한다. 즉 다음과 같이 표현한다. 참고로, 본 연구에서는 프로세스에 관련해서 로제타넷의 표준에 대해 적용해보았으며 PIP로 정의된 프로세스를 여러 활동들의 집합으로 보고 구체화 단계를 진행하였다.

```
<owl:Class rdf:ID = "Activity"/>
<owl:Class rdf:ID = "PIP">
  <owl:equivalentClass rdf:resource =
    "# Activity"/>
  <owl:unionOf rdf:parseType =
    "owl:collection">
    <owl:Class rdf:about =
      "# ActivityContainer"/>
    <owl:Class rdf:about =
      "# OccurrenceContainer"/>
    <owl:Class rdf:about =
      "# ConstraintContainer"/>
  </owl:unionOf>
</owl:Class>
<owl:Class rdf:ID = "ActivityContainer">
  <owl:intersectionOf rdf:parseType =
    "owl:collection">
    <rdfs:subClassOf>
      <owl:Restriction>
        <owl:minCardinality rdf:datatype =
          "&xsd:nonNegativeInteger"
          >1</owl:minCardinality>
```

```
<owl:Class rdf:resource =
  "# Activity"/>
</owl:Restriction>
</rdfs:subClassOf>
</owl:intersectionOf>
</owl:Class>
```

활동의 발생과 그 순서에 관련된 용어들 역시 다음과 같이 PSL-ontology에 정의된 용어들과 연결시켜주도록 한다. 속성들 중 다른 개체를 그 값으로 가지는 것들은 owl:ObjectProperty로 정의하도록 한다.

```
<owl:Class rdf:ID = "Occurrence"/>
<owl:ObjectProperty ID = "after_start">
  <rdfs:domain rdf:resource =
    "# Occurrence"/>
  <rdfs:range rdf:resource = "# Occurrence"/>
</owl:ObjectProperty>
<owl:Class rdf:ID = "Timepoint">
<owl:ObjectProperty ID = "begin">
  <rdfs:domain rdf:resource =
    "# Occurrence"/>
  <rdfs:range rdf:resource = "# Timepoint"/>
</owl:ObjectProperty>
<owl:ObjectProperty ID = "end">
  <rdfs:domain rdf:resource =
    "# Occurrence"/>
  <rdfs:range rdf:resource = "# Timepoint"/>
</owl:ObjectProperty>
```

제약조건에 관련된 용어들에 대해서도 마찬가지로의 방법으로 지정하며, 응답제한시간

에 관련된 "time to acknowledgement"와 "time to perform"은 다음과 같이 Timeduration의 인스턴스로 지정해줄 수 있다.

```

<owl:Class rdf:ID = "Timeduration"/>
<owl:DatatypeProperty ID = "t_value">
  <rdf:domain rdf:resource =
    "#Timeduration"/>
  <rdf:range rdf:resource =
    "http://www.w3.org/2000/10/XMLSchema
    #integer"/>
</owl:DatatypeProperty>
<owl:Class rdf:ID = "Time2ack">
  <rdf:type rdf:resource = "#Timeduration"/>
  <rdf:domain rdf:resource = "#PIP"/>
</owl:Class>
<owl:Class rdf:ID = "Time2perform">
  <rdf:type rdf:resource = "#Timeduration"/>
  <rdf:domain rdf:resource = "#PIP"/>
</owl:Class>

```

이러한 방법에 따라 표현된 프로세스 온톨로지를 바탕으로 표현되는 프로세스 역시 XML로 표현하며, 전체적인 문서의 구조는 다음의 XML DTD에서 정의된 것과 같은 형태를 갖게 된다. 전술한 바와 같이 프로세스며 활동, 발생 및 순서, 제약조건으로 구성되며, 각 클래스에 대한 집합(Container)으로 표현되도록 하였다.

```

<!ELEMENT PIPContainer (PIP*)>
<!ELEMENT PIP (comment,
ActivityContainer*, OccurrenceContainer*,

```

```

ConstraintContainer*)>
<!ELEMENT ActivityContainer
(Activity*)>
<!ELEMENT Activity (fromRole, toRole)>
<!ELEMENT OccurrenceContainer
(Occurrence*)>
<!ELEMENT Occurrence (#PCDATA)>
<!ELEMENT ConstraintContainer
(Time2ack*, Time2perform+, RetryCount)
.....
<!ELEMENT RetryCount (#PCDATA)>

```

거래당사자의 I-에이전트는 위의 구성에 따라 실제 값이 입력된 메시지 인스턴스를 R 에이전트에게 건네주고, R-에이전트는 그 값들을 내부 데이터베이스에 저장하게 된다.

이와 같이 XML 및 OWL로 표현하여 구체화한 각 프로세스 및 데이터의 메타데이터 및 모형들을 데이터베이스에 등록하고 저장해야 한다. 온톨로지에 대한 사용을 위해 먼저 <그림 3>에서 제시한 통합시스템의 전체적인 구조에서 공유되는 온톨로지의 등록 및 저장의 역할을 하기 위한 기능이 필요한 것이다. 이러한 기능을 간단히 구현한 것이 <그림 6>와 <그림 7>이며, 전자는 메타데이터에 대한 정의와 모형들을 기술한 XML 메시지들을 등록하고 저장하는 화면을, 후자는 저장된 메시지들에 대한 목록에서 검색하는 화면을 보여준다.

이렇게 저장된 메시지들은 각각의 에이전트들이 아래의 예와 같은 질의문을 통해 검색하게 할 수 있다.

참 고 문 헌

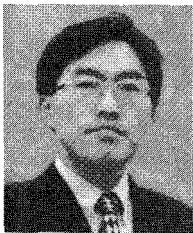
- [1] 오상경. 이중 관계형 데이터베이스의 스키마 추출을 통한 포괄적 RDF 스키마 생성과 데이터베이스 관리. 서울대학교 산업공학과 석사학위논문, 2002
- [2] Berners-Lee, T., Hendler J., and Lassila, O., "The Semantic Web". Scientific American, 2001
(<http://www.scientificamerican.com/2001/0501issue/0501berners-lee.html>)
- [3] Breitbart, Y., "Multidatabase interoperability", ACM SIGMOD Record, Vol. 19, No. 3, 1990, 53-60
- [4] Chandrasekaran, B., Josephson, J. R., and Benjamins, V. R., "What Are Ontologies, and Why Do We Need Them?", IEEE Intelligence Systems, Vol. 14, No. 1, 1999, pp.20-26
- [5] Chang, T.-W., Shin, K., Pillai, B., and Park, J., "A development methodology for e-Work ontology using RDF/RDFS and PSL", Production Planning and Control, Vol. 14, No. 8, 2003, pp.766-777
- [6] Gruber, T. R., "Toward Principles for the Design of Ontologies Used for Knowledge Sharing", International Journal of Human-Computer Studies, Vol. 43, 1995, pp.907-928
- [7] ISO/IEC, Information Technology -- Information Resource Dictionary System (IRDS) framework, ISO/IEC 19027, 1990
- [8] Jarke, M., et al., Meta Modeling: A Formal Basis for Interoperability and Adaptability, in Kramer, B., Papazoglou, M., and Schmidt, H-W., Information Systems Interoperability, Research Studies Press Ltd., 1998
- [9] Kaye, A., and Colomb, R. M., "Extracting ontological concepts for tendering conceptual structures", Data and Knowledge Engineering, Vol. 40, 2002, pp.71-89
- [10] Mena, E., Illarramendi, A.: Ontology-based Query Processing for Global Information Systems, Kluwer Academic Publishers, 2001
- [11] Schlenoff, C., Gruninger, M., Tissot, F., Valois, J., Lubell, J., and Lee, J., The Process Specification Language (PSL): Overview and Version 1.0 Specification, NISTIR 6459, National Institute of Standards and Technology, Gaithersburg, MD, 2000
- [12] Smith, H., and Poulter K., Share the Ontology in XML-based Trading Architectures, Communications of the ACM, Vol. 42, No. 3, 1999, pp.110-111
- [13] Studer, R., Benjamins, V. R., and Fensel, D., "Knowledge Engineering: Principles and methods", Data and Knowledge Engineering, Vol. 25, 1998, pp.161-197
- [14] Szczerbicki, E., "External environment of an autonomous manufacturing agent: dynamics and representation", International

Journal of Systems Science, Vol. 27, No. 12, 1996, pp.1211-1218

[15] Vernadat, F. B., Enterprise Modeling and Integration: principles and applications. Chapman & Hall, 1996

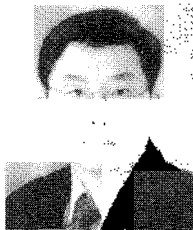
[16] Wang, H., "A Conceptual model for virtual markets". Information & Management, Vol.32, 1997, pp.147-161

저 자 소 개



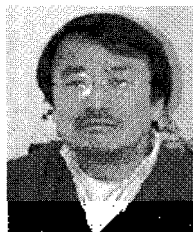
장태우

(E-mail : keenbec@etri.re.kr)



신기태

(E-mail: ktshin@daejin.ac.kr)



박진우