

◆특집◆ 지능형 설계 기술

온톨로지 기반 설계 패러다임에 관한 고찰

김용세*, 강무진*, 양정진**

A Review on Ontology-Based Design Paradigm

Yong Se Kim*, Mujin Kang* and Jungjin Yang**

Key Words : Ontology (온톨로지), Design Knowledge (설계 지식), Workflow Management (업무 흐름 관리), Knowledge Representation (지식 표현)

1. 서론

제품의 라이프사이클은 요구조건 분석, 개념 설계, 제조, 사용 및 유지보수 등 여러 단계를 포함한다. 각 단계는 다른 단계에서 오는 정보를 사용하여 새로운 정보를 생성하면서 여러 종류의 제품 정보를 취급한다. 제품의 설계 지식은 제품의 설계 과정과 관련된 제품 정보들로 이루어진다. 설계자는 설계시에 기하학적인 자료만을 필요로 하는 것이 아니라, 명세, 설계 규칙, 제한점, 이론적 설명등 설계와 설계과정에 관련된 일반적인 지식의 활용을 필요로 한다. 현대의 제품들은 복잡성이 커지면서 설계가 점점 더 지식 집약적이 되고 협업이 필요하게 되었다. 이에 따라 다수의 설계자들이 분산된 환경에서 공동된 작업을 할 수 있도록 설계자 사이에서 공유되는 지식의 표현과 관리 그리고 사용을 지원하는 것이 중요하게 되었다. 이는 Schematics, CAD 모델, 문서 조사등을 제공하는 전통적인 설계 데이터 베이스 뿐만 아니라 가공되는 지식을 표현하고 저장하여 이러한 지식

이 처리될 수 있는 형태를 갖춘 기본 방식이 요구되는 것이다. 설계자는 현재 구상하고 있는 설계와 유사한 설계가 과거에 있었는지, 바로 사용할 수 있는 표준 부품 또는 특정 부품이 CAD 시스템 내에 있는지 알고 싶을 경우가 많다. 특히 설계하는 제품이 매우 복잡하거나 설계 데이터베이스가 방대하다면 설계자는 답을 얻는 데에 많은 노력을 소요해야 할 것이다. 한편, 지식을 관리하기 위해서는 서로 다른 여러 출처(Source)로부터의 지식을 통합해야 할 필요가 있고, 따라서 여러 다른 지식 표현 모델들 간의 의미론적 관계(Semantic Relationship)를 식별하여 정리하는 일이 선행되어야 한다.

설계자들이 설계시에 적용하는 지식의 형태는 크게 다음과 같이 나눌 수 있다.

- 제조 공정 모델
- 공정 및 조립 계획
- 기존 설계 및 설계 이력
- 설계 경험, 기록, 기업의 레거시 데이터

이러한 지식은 설계자 입장에서 적용 가능한 새로운 처리방식을 찾거나, 현재 설계도와 유사한 설계의 검색, 다른 설계자의 해결방법에 관한 노하우 처리 및 조합을 위한 계획과 관련된 정보를 찾는데 활용된다. 이에 필요한 지원을 하기 위하여, 현재의 설계 패러다임이 지식 중심적이기 보다는 자료 중심적이고 인공물의 제한을 내포하고 있는 전통적인 설계 데이터 베이스에서 설계 지식

* 성균관대학교 기계공학부

Tel. 031-290-7452/7441, Fax. 031-290-5849

Email yskim@me.skku.ac.kr, mjikang@skku.edu

** 가톨릭대학교 컴퓨터정보공학부

Tel. 02-2164-4377, Fax. 02-2164-4777

Email jungjin@catholic.ac.kr

을 통일된 형식으로 표현하고 검색, 공유, 재사용을 가능케 하는 지능형 지식기반 설계 모델링 시스템으로의 체제 변환이 대두되고 있으며 이는 설계 저장소 구축, 지식처리 및 관리에 필요한 온톨로지 활용 등이 포함된다.

현재 공학에서 대부분의 설계는 사례 기반 설계로 이루어지고 있는 반면, 복잡한 형태로 이루어진 도형과 공학의 정보를 논리적으로 추론하고, 저장된 형태의 지식을 효과적으로 검색하고 재사용할 수 있는 처리 방법은 미숙한 편이다. 설계분야에서의 처리 방법 향상은 설계 지식의 저장소 구축뿐 아니라 추론을 통한 저장형 공학 지식의 활용도 향상을 꾀함과 함께 이러한 설계 및 제조 작업이 네트워크를 통하여 공동으로 이루어질 수 있도록 개념상의 이해를 위한 온톨로지 개발 등을 통하여 이루어 질 수 있다.

2. 설계 온톨로지

설계 저장소는 인공물 표현의 세 가지 근원적인 속성들 - 물리적인 구조(Structure)와 그 전반적인 효과(Function) 그리고 사용의 원인이 되는 근거(Behavior) - 을 통합하고자 하는 지능적 지식기반 설계 인공물 모델링 시스템이다. 이 FBS (Function-Structure-Behavior) 모델을 기반으로 하면 설계 지식을 네 가지 종류로 분류할 수 있다.^{1,2,3,4} 즉, 인공물의 기능, 인공물의 거동, 인공물 구조, 그리고 구조와 거동과 기능 간의 인과관계가 그것이다. 기능 관련 지식은 인공물의 목적이 무엇인가 하는 것이고, 거동에 관한 지식은 인공물의 상태(State) 변화에 관한 것이며, 구조 지식은 부품의 집합과 그 결합 관계이고, 인과관계 지식은 설계 제한(Constraints), 의도, 물리적 원리, 경험적 규칙 등에 관한 것이다. 예를 들어, 헤드가 두 개인 전기 면도기를 설계하는 데, 재충전이 가능해야 하고 충전 후 30 분까지 사용할 수 있어야 하며 트림 유니트가 있고 외형 크기는 60 x 25 x 100 mm 이내라야 한다는 요구 조건을 만족시켜야 한다고 하자. 이 경우, 재충전하여 30 분 사용 가능하고 트리밍할 수 있는 면도기라는 것은 기능 지식에 해당되고, 두 개의 헤드와 트림 유니트는 구조 지식에 속하며, 외형 크기 제한은 인과관계 지식에 속하는 것이다.

설계 프로세스의 모델을 Formulation, Synthesis,

Analysis, Evaluation, Reformulation, Presentation 으로 보고, 각각의 설계 과정을 확장된 FBS 모델에 비추어 설명하면 다음과 같다. F를 기능, D를 설계 표현(Design Description), S를 구조, B_S를 구조의 거동(Behavior of Structure), B_e를 기대 거동(Expected Behavior)라 하면, 설계의 본연의 목적은 (F → D) 매핑이다. 즉, 요구되는 인공물이 그 기능을 만족시키는 방법에 대한 표현을 찾는 것이 설계인 것이다. 설계 표현은 인공물을 구성하는 요소들과 그들간의 관계를 포함하는 데, 이것은 (S → D)로 나타낼 수 있고, 기존의 전산제도시스템이 구조를 설계 표현으로 변환하는 과정을 잘 지원하고 있다. 설계의 Formulation(Specification) 과정은 기능을 기대 거동으로 바꾸는 것이므로 (F → B_e)로 나타낼 수 있다. Synthesis 과정은 기능으로부터 유도된 기대 거동을 구조의 거동에 관한 지식을 이용하여 구조를 선택하고 조합하는 단계이므로 (B_e → S(B_S))로 나타낼 수 있다. Analysis는 구조의 거동을 결정하는 것으로 (S → B_S) 매핑이고, Evaluation은 (B_e ↔ B_S)으로 기대 거동과 시스템 거동을 비교하는 것이다. 그 비교의 결과가 불만족스러운 경우에는 Synthesis의 결과로 얻은 구조의 거동에 대한 기대 거동의 Superset 으로부터 변경된 새로운 기대 거동을 결정하게 되는 데, 이것이 Reformulation 이다 (S(B_S) → B_e). Design Presentation은 확정된 구조의 표현으로 (S → D)로 나타낸다.³

설계는 기능적인 요구 사양을 구체적인 사물의 속성값들로 변환하는 창조적인 행위이다. 개년 설계 과정에 사용되는 설계 노우하우는 그 주관성과 모호성으로 인해 암묵적으로 남겨지곤 한다. 이와 같은 주관적이고 암묵적인 지식이 어떤 커뮤니티 안에서 공유되기 위해 명시적으로 만들어져야 할 필요성은 대단히 크다. 설계 지식을 공유할 수 있다면 설계 과정을 현저하게 개선할 수 있을 것이기 때문이다. 이를 위하여는 좀처럼 공유되기 어려웠던 개념 설계 지식을 수집하고 표현할 수 있는 일반적인 프레임워크를 개발해야 한다.

두말 할 필요도 없겠지만, 설계 과정에는 대단히 풍부한 양의 지식이 관여한다. 기존의 지식 공학은 전문가 시스템 실패 사례에서도 볼 수 있듯이 지식 관리를 잘 지원해 주지 못하였다. 지식 공학의 차세대 기술인 온톨로지 공학은 전통적인 지식 공학이 해결할 수 없었던 문제점을 극복할

수 있을 것으로 보인다. 온톨로지 공학의 도움을 받아서 설계 지식 배경에 숨어 있는 개념적인 구조를 명시적으로 표현할 수 있다면 지식 체계의 상호 작용과 지식의 공유 및 재사용에 크게 기여하게 될 것이다.

기본적인 개념의 체계라 할 수 있는 온톨로지는 어떤 지식 베이스의 배경 지식 체계로서 대상 세계의 개념화를 명시적으로 표현함으로써 폭넓게 사용될 수 있는 공유 가능한 지식 베이스를 구축할 수 있는 단단한 기초를 제공한다. 공학 설계 영역에서의 온톨로지는 작업 온톨로지(Task Ontology)와 영역 온톨로지(Domain Ontology)로 분류할 수 있다. 전자는 설계 활동의 프로세스를 의미한다. 후자는 설계되는 사물, 즉 설계 목표와 관계되는 것으로, 물리적인 시스템만을 생각한다면 제품 설계의 경우라면 제품이, 생산 설계라면 생산 공정이 설계 목표가 될 것이다. 영역 온톨로지는 일반적으로 설계 목표 자체 - 구조나 형상 등 - 또는 그 물리적 속성의 변화 - 거동 및 기능 - 를 표현하고자 하는 것이다. 그러나 지식을 표현한다는 것만으로는 충분하지 않고, 지식이 적합한 어휘로 그리고 적절한 구조로 조직(organized)되어야 한다. 온톨로지 공학은 지식을 표현하고 또한 체계적으로 서술할 수 있는 전산 처리 가능한 어휘를 제공한다.

설계의 목표 지식은 제품 지식, 즉 설계 대상 지식과 공정 지식으로 이루어지며, 공정 지식은 Domain-dependent 한 것과 Domain-independent 한 것으로 나눌 수 있다. 지금까지는 설계 지식의 체계화와 관련하여 주로 Domain-dependent 한 지식을 해석적이고 정량적으로 조사하는 연구가 많았다. 그러나 그 대부분의 경우에는 설계 프로세스를 정적(static)으로 보아 설계과정의 Input 과 Output 을 가지고 수학적으로 모델링함으로써 설계 과정 내부의 서브 프로세스를 무시하는, 즉 동적인 특성을 무시하는 결과를 낳는다. Tomiyama⁵ 는 로직과 인공지능을 이용한 설계 프로세스 모델링 연구에서 설계 과정은 개연적 귀납(abduction)이라는 사실에 주목하여, 설계 대상에 대한 Domain-independent 한 지식을 조사하여 설계 과정의 고유한 특성을 밝히려 시도하였다. 그는 인공물의 설계 과정을 안내(guide)하는 프레임워크와 일관성 있고 공유 가능한 방식으로 기능 지식을 서술할 수 있는 근본 개념을 제안하였다 (Fig. 1). 엔터티,

프로세스, 시간 등의 *Top level ontology* 위에 기능 지식의 계층적 체계를 보여주며, 기본적으로 각 층에서의 지식은 그 하위 층에서의 개념을 사용하여 서술된다. *Extended device ontology* 는 인공물을 일관성 있게 설명할 수 있는 공통 관점을 제공하기 위해 개발된다. 이 *Top level ontology* 와 *Extended device ontology* 는 상위 층에서 일관성 있게 지식을 구축할 수 있는 기반이 된다. *Functional concept ontology* 는 기능적 개념을 *Device ontology* 에 정의된 "function" 개념의 인스턴스로 명시한다. 그 정의는 장치(Device)나 영역, 또는 구현 방법에 의존하지 않으므로 보편적이고, 폭넓은 분야에서 사용될 수 있다. 많은 이론과 물리학의 원리 그리고 추상적 부품 라이브러리 등이 *General concept layer* 라 불리는 이 지식 클래스에 속한다. 기능적 개념이 그 기능이 "무엇"을 달성하고자 하는가에 관한 것이라면, *Way of function achievement* 는 "어떻게" 기능을 얻을 것인가에 관한 지식이다.

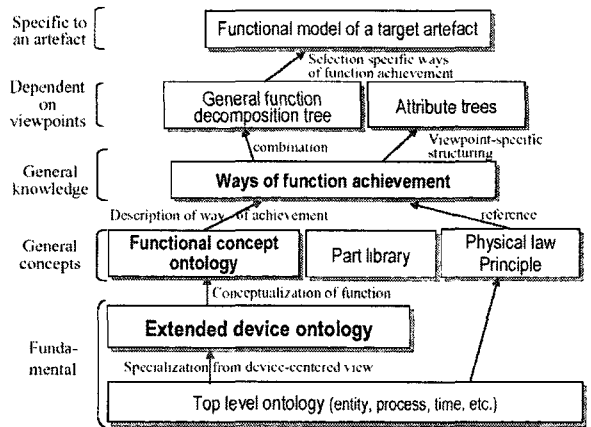


Fig. 1 Hierarchy of ontology and knowledge of function

이것은 기능 전개 지식과 유사하지만, 상이한 역할과 원리의 체계로 이루어져 있어 훨씬 더 풍부한 내용을 담고 있다. 이러한 지식의 고유한 구조는 *is-a* 계층구조로 조직된다. *is-a* 구조는 여러 영역에 거쳐 공유되고 적용될 수 있도록 각 방법 (way)의 고유한 특성을 식별하여 주의 깊게 설계된다. 중요한 점은 대상 사물의 고유한 특성이 무엇인지를 염두에 두어 *is-a* 관계와 *part-of*, *is-achieved-by* 등의 다른 관계를 명확하고 일관되게

구별해야 한다는 것이다.

3. 설계에서의 온톨로지 공학

3.1 온톨로지 기반 시스템

온톨로지 기반 설계 지식 처리의 대표적인 예로, NIST Design Repository Project⁶ 는 설계 지식의 재사용을 지원하기 위하여 설계에 필요한 이질적인 지식과 자료의 설계 저장소 구축에 관한 기술적인 기초 연구를 시도하였다. NIST 프로젝트가 포함하고 있는 이슈로는 전통적인 CAD 시스템보다 포괄적인 지식 표현을 제공하는 모델링을 지원 방법, 상호운용성과 연구효과를 높이기 위하여 표준화된 표현의 사용, 처리 효과 증대를 위하여 인간과 기계가 동시 해석 가능한 형태의 정보 표현, 정보사용의 효율성과 일관성을 제공하는 표준화된 용어의 분류체계, 쉽고 효과적으로 디자인 저장소를 사용할 수 있는 인터페이스 개발 등이 있다.

온톨로지의 철학적 의미는 존재론을 뜻한다. 정보기술 분야에서의 온톨로지는 프로그램과 같은 일종의 설계된 가공물이라 할 수 있다. 온톨로지는 사물이나 사건이 존재하는 세계의 구조를 표현하고 이를 체계적으로 설명해 내기 위한 것이라 할 수 있다. 다시 말해서, 온톨로지(ontology)는 개념(concept)들의 집합들을 나타내는 것으로 개념간에 존재하는 관계를 명백하게 기술하여 개념과 그러한 관계 성립의 공유를 목적으로 사용된다. 공유된 지식으로의 온톨로지는 연관 관계 또는 관련 지식을 찾아내는데 활용되고, 다양한 층의 온톨로지 개발을 통하여 연관관계 및 메타데이터 추출의 자동화를 추구한다. 여기서 자동화는 사람과 기계를 통하여 처리 가능한 형태를 의미하며 이를 RDFS 나 OWL 과 같이 표준화된 언어로 기술함으로써 더욱 가능하게 된다.⁷

온톨로지의 다양한 정의들 중에서 가장 본질적인 정의를 들어 온톨로지를 설명하자면 “온톨로지는 공유하기(shared) 위한 개념들의 개념화(conceptualization)를 형식적(formal)이고, 명백하게(explicit) 설명해 놓은 명세서(specification)”라고 할 수 있다.⁸ 여기서 개념화(conceptualization)는 어떤 현상에 대해서 관련 있는 개념들을 식별하여 그 현상을 추상화한 모델, 즉, 관련 있는 개념들만 모아 재사용이 가능한 일반화된 모델로 설명하는 것이다. 명백하게(explicit) 표현한다는 것은 사용된

개념의 종류와 개념들이 추상화된 모델과 갖는 관계들, 그리고 그 개념들의 사용에 있어서 주어지는 제한점(constraints)들을 명백하게 정의하여 사람과 기계를 통하여 처리 가능한 형태를 의미한다. 형식적(formal)인 표현은 온톨로지가 기계 판독 가능한 형태이어야 한다는 것을 의미한다. 끝으로 공유된(shared) 개념은 개별적이거나, 사적인 용도의 온톨로지만이 아니라 그룹의 의견수렴을 통해 그룹이 동의한 일치된 지식을 담고 있는 것을 말한다. 온톨로지는 데이터보다 복잡한 형태의 지식과 관련이 있고, 특수한 개별적 사실에 대한 표현이 아닌 보편적인 개념의 표현이라는 차원에서 기존의 데이터 베이스 그리고 지식 베이스와의 차별성을 가지며 이러한 특성으로 온톨로지는 지식의 공유와 재사용을 가능케 한다.

근래 대규모의 소프트웨어나 정보시스템은 초기부터 온톨로지에 대한 공약을 우선으로 시도하고 있다. 최신의 대규모 서버 소프트웨어는 상호적인 연결을 위하여 XML 기반과 관계형 데이터베이스를 기본층으로 하여 구성되어 있다. 이러한 기본층들이 구문론(syntactic)에 관련된 낮은 층의 문제를 담당함으로써 syntax 이상의 개념적인 면을 구분하여 정보시스템에 활용토록 고려하는 것이 가능하게 되었다.

개념적인 면을 강조하여 구축된 온톨로지는 지식을 표현하고 가공하여 만든 명세서로서, 공학적인 특성을 적용하여 디자인하고 평가하도록 되어 있다. 공학적으로 고안된 온톨로지는 주어진 기능적인 목표와 제약점을 만족하도록 제작된다. 이를 위하여 공학적으로 온톨로지를 관리하고 검증하여주는 도구들이 필요하게 되고, 상표나 권한에 의한 것이기 보다는 목표기준에 맞추어 협력적으로 또는 경합적으로 사용되는 다중의 온톨로지를 다룰 수 있어야 한다.

3.2 온톨로지 구축 및 관리

온톨로지 공학의 목적은 온톨로지 개발과 사용을 온톨로지 라이프사이클 - 디자인, 평가, 검증, 유지, 전개, 매핑, 통합, 공유와 재사용 - 과정을 거쳐 효과적으로 지원하는 것에 있다. 온톨로지 개발은 시간과 비용이 많이 들고 쉽지 않은 작업이다. 디자인의 목적이 자동화된 추론에 적용되기 위한 것이라면 형식화하는 과정을 포함하여 더욱 그러하다. 온톨로지를 개발하는데 있어서 정석이

란 없다. 도메인 모델링과 마찬가지로 적절한 온톨로지 개발은 고려 중에 있는 애플리케이션과 그 애플리케이션에서 예측되는 확장성에 달려있다.

다루어지는 문제의 상황에 맞게 온톨로지가 적용될 수 있기 위해서는 우선 다양한 규모와 내용의 온톨로지들이 라이브러리에 저장되고 가장 적절한 온톨로지를 발견할 수 있어야 하며 필요에 따라서 여러 온톨로지들을 결합 내지 변형, 통합이 가능하여야 한다. 온톨로지 통합을 위한 자동화 및 반자동화된 도구 개발이 중요시된다.⁹ 온톨로지 도구의 구조를 살펴보면, 편집기, 브라우저, 온톨로지 리파지토리, 주석데이터 리파지토리, 추론 엔진, 온톨로지 미들웨어로 구분지을 수 있다. 온톨로지 관리를 위한 도구로는 대표적으로 Protégé-2000 을 위시하여 OLEd, OntoEdit, Ontolingua, OpenKnoME, WebODE, VOM¹⁰ 등을 들 수 있다.

향후 온톨로지 기반 지식 관리체계의 발전방향은 아래와 같이 언급되어 질 수 있다.

- 온톨로지 표현 언어의 지속적인 상위계층 구현과 추론 기능의 지원 보강
- 관리도구: 에이전트를 활용한 통합된 메모리 관리, GUI 기반 지식 변환도구, 통합적인 온톨로지 관리 시스템
- 방법론: 협동적인 온톨로지 공학과 도메인 중심 내지 도구 중심 방법론
- 비즈니스 응용: 정보 검색, 관리 기술, 조직간의 지식교환 등이다.

4. 온톨로지 기반 Workflow 관리

대부분 공학적인 처리 과정은 자세한 부분은 시작 단계에서 불분명하게 명시되어 있고, 이를 플랜의 형태로 기획하고 실행할 시에는 TASK의 복잡성 내지는 인력의 변동 문제로 기획된 플랜의 변경이 불가피하다. 처리과정이나 그 결과물들은 표준에 명시된 바에 준하여 그러한 동적 요소들을 포함할 필요가 있으며, 이는 공학도간의 협업적인 관계나 TASK 간의 서로 상호 의존적인 정도가 반영이 되어야 한다. 워크플로어는 다수의 참여자들이 정하여진 규칙에 따라 서로 협업적으로 문서, 정보, TASK등을 전달하는 자동화된 비즈니스 처리의 논리적인 단계들을 의미한다.¹¹ 워크플로어

관리 시스템(Workflow Management System-WFMS)은 TASK 실행이 워크플로어 로직에 따라 계획된 처리과정에 준하여 이루어지도록 실행자체를 정의, 제작, 관리하여 주는 시스템으로 이는 필요에 따라서 IT 도구나 애플리케이션, 처리규정 해독 및 참여자와의 상호작용을 통하여 분산된 소프트웨어들을 통하여 이루어진다. Fig. 2 는 워크플로어 관리시스템의 일반적인 아키텍처를 나타낸다.

WFMS 는 개발시 기능(Build-time function), 실행시 제어기능(Run-time control function), 실행시 상호작용(Run-time interaction)으로 구분하여 WFM 을 지원한다. 워크플로어와 처리 모델링을 위한 메타언어에는 WPD(Workflow Process Definition Language), PIF(Process Interchange Framework), PSL(Process Specification Language), GPSG(Generalized Process Structure Grammars), UML(Unified Modeling Language) 등이 있다.

최근 이러한 제어 흐름에 초점을 둔 워크플로우 패턴 연구도 대두되고 있으며, 이와 함께 온톨로지 기반 워크플로우 관리가 요구되고 있는 실정이다. 온톨로지 기반 워크플로우 관리에 필요한 요구사항으로는 기능적인 부분과 구조적인 부분이 있다. 기능적인 요구사항으로는 대표적으로

- 목적의 명시를 목표 온톨로지 내지 원하는 결론상태로 나타낸다.
- 가능한 서비스 선택은 명시된 목적의 관련성 여부와 서비스에 대한 선호도를 고려하여 선택할 수 있어야 한다.
- 서비스 요청 형식과 응답 판단방법은 기술된 서비스 처리방식에 따라서 이루어 져야 한다.
- 요청자와 응답자가 다른 온톨로지를 활용하여 통신을 하는 경우 상호운용적으로 의미적 내용들을 바꾸어 줄 수 있어야 한다.
- 서비스를 호출할 수 있으면서도 동적으로 서비스들을 조합할 수 있어야 한다.

등이 있다. 구조적인 요구사항으로는 보편적으로 공유된 온톨로지를 활용하는 아래와 같은 표준화된 프로토콜들이 필요하다.

- 온톨로지 통역 서비스
- 처리 중재 서비스
- 처리의 스케줄링과 조합 서비스
- 처리 실행과 상태 기록 서비스

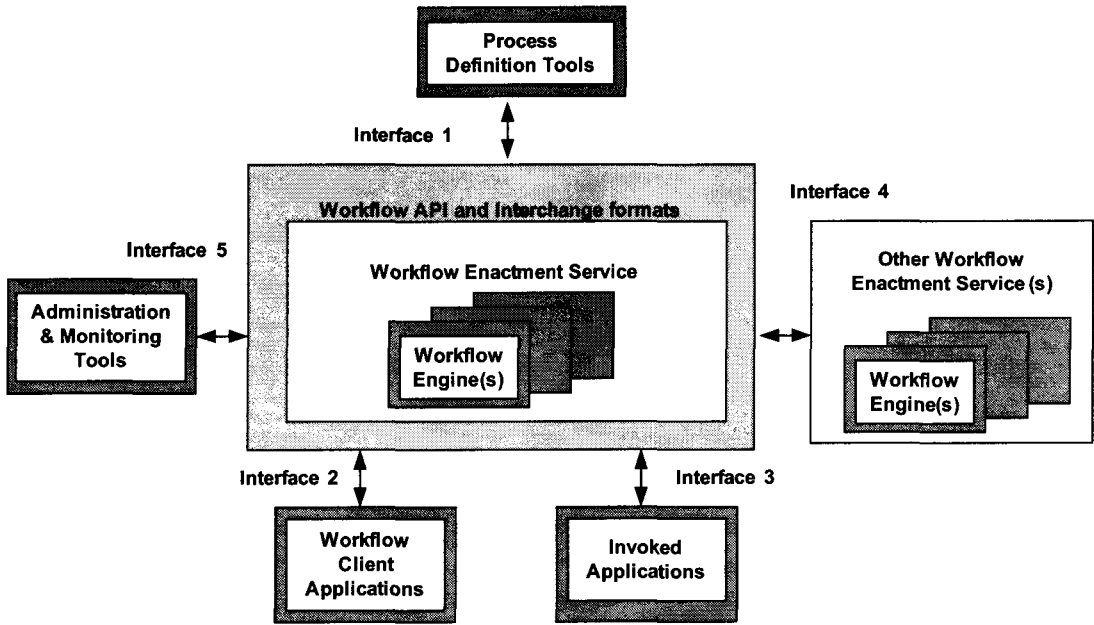


Fig. 2 Workflow management system architecture

WSFL(Web Service Flow Language)과 XLANG의 연합언어인 BPEL4WS(Business Process Execution Language for Web Services)는 PDL(Process Definition Language)의 대표적인 언어로, 실행 가능한 비즈니스 프로세스와 추상화된 프로세스를 표현하는데 사용된다. 구조적으로, 워크플로우를 서술하는 BPEL4WS 파일은 참여자명시, 실행되어야 하는 서비스들, 이벤트들간의 가능한 발생순서 등을 나타내고 있다. BPEL4WS은 웹 서비스용으로 고안되었고, 웹 서비스는 그 전 상태를 기억하지 않기 (stateless) 때문에 설계자는 진행중인 워크플로어에 상태정보를 추가하는 방법이 필요하다.^{12,13} 이러한 목적으로 이전 메시지의 저장과 워크플로어의 현재 상태를 안내할 수 있는 “컨테이너”라는 일관성 있는 저장소를 제공하는 방법을 취한다. BPEL4WS는 UML에서 표현될 수 있는 것보다 더 복잡한 것, 즉, 우선순위 제한요소와 같은 것을 표현할 수 있는 것으로 “링크”라는 개념을 포함하고 있다. BPEL4WS 처리모델은 WSDL 서비스 모델에 탑재되어 있고 모든 기본 행동들이 WSDL portType 기술로 표현가능함을 전제로 하여 WSDL 규정대로 기술된 메시지들로 이루어져 있다.

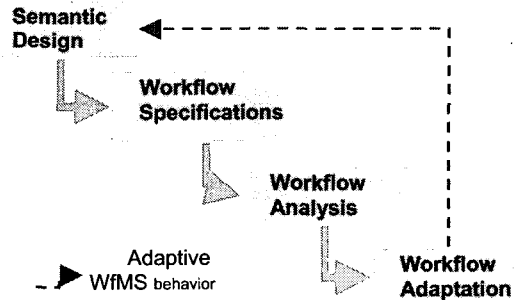


Fig. 3 Ontology-based workflow design

구조적인 요구사항으로는 보편적으로 공유된 온톨로지를 활용하는 아래와 같은 표준화된 프로토콜들이 필요하다.

의미론/온톨로지 기반 워크플로어 디자인은 Fig. 3에서와 같이 동적인 요소를 포함하여 적응력있는 워크플로어를 가능케 할 것이다. 이러한 방법론은 타스크의 의미적 타당성을 제시하여 줌으로써 다음 사항들에 도움이 될 수 있다.

- 구성(organization) 간의 복잡한 처리인 워크플로어 디자인

- 서로 다른 구성(organization)들 내에서 이질적인 타스크를 필요에 따라 연결
- 디자인 단계에서 사용자 지원

의미론/온톨로지 기반 워크플로어 디자인이 당면한 문제로는 다음과 같이

- 명세서: 워크플로어 처리의 설계를 용이하고 자동화 하기위해 의미적인 내용을 워크플로어 타스크 명세서에 명시하는 것
- 선택/재사용: 구문론(syntactic)과 의미론적(semantic)인 방법을 적용하여 디자인 저장소로부터 필요한 타스크의 검색, 순위, 선택을 향상하는 방법
- 조립(composition): 데이터 분석과 타스크 간의 도식적인 충돌을 해결하도록 의미론적용 방법 등이 있다.

종합적으로 보면, 웹 서비스는 사용자에게 원하는 정보를 제공하는 서비스에서 점차 사용자 소프트웨어 에이전트(스스로 환경 변화를 인지하고 변화에 반응하며 경험을 바탕으로 학습하는 기능을 갖는 프로그램)를 중심으로 사용자를 대신해 필요한 정보를 요구해 자동으로 처리해주는 서비스로 변해가고 있다. 이와 함께 언제, 누가 어떠한 일을 하는가가 고정적으로 명시되어 있는 처리중심의 워크플로어는 동적인 환경과 이에 맞추어진 처리가 필요한 적응력있는 워크플로어의 전환이 요구된다. 이러한 워크플로어의 장래성은 BPM(Business Process Mangement)과 PDL, 웹 서비스, 시맨틱 웹, 에이전트 기반 소프트웨어 공학등의 앞으로 진척되는 진화에 통합적으로 영향을 받을 것이다.

5. 결론

설계 과정에는 대단히 풍부한 양의 지식이 관여한다. 지식 공학에서 지식 관리에 관한 연구가 많이 있었지만 Domain-dependent 한 부문에 국한되었었고 실용적인 시스템으로 발전하는 데에 한계가 있었다. 온톨로지 공학은 이 미해결 과제들을 극복하고 지능형 설계의 새장을 여는 데에 기여할 것으로 기대된다. 온톨로지를 이용해 설계 지식 배경에 숨어 있는 개념적인 구조를 명시적으로 표현할 수 있다면 지식 체계의 상호 작용과 지식의 공유 및 재사용에 크게 기여하게 될 것이다.

후 기

본 연구는 과학기술부 핵심연구개발사업 “미세형상 설계 지능화 기술 개발” 과제 수행의 일환으로 이루어진 것임을 밝히며, 지원 기관에 감사드립니다.

참고문헌

1. Tomiyama, T., Umeda, Y. and Yoshikawa, H., "A CAD for a Functional Design," Annals of the CIRP, Vol. 42, pp. 143 - 146, 1993.
2. Ma, Z. and Mili, F., "Knowledge Comparison in Design Repositories," Engineering Applications of Artificial Intelligence, Vol. 16, pp. 203 - 221, 2003.
3. Gero, J., "Design Prototypes: A Knowledge Representation Schema for Design," AI Magazine, Vol. 11, No. 4, pp. 26 - 36, 1990.
4. Kitamura, Y. and Mizoguchi, R., "Ontology-based Systemization of Functional Knowledge," Journal of Engineering Design, Vol. 15, No. 3, pp. 321 - 341, 2004.
5. Tomiyama, T., "A Theoretical Approach to Synthesis," Proceedings of 2000 International Symposium on Modeling of Synthesis, pp. 25 - 64, 2000.
6. NIST Design Repository Project, <http://edge.mcs.drexel.edu/repository/>
7. Ontology Languages, <http://w3c.org/sw>
8. Gruber, T., "A Translation Approach to Portable Ontology Specifications," Knowledge Acquisition Journal, Vol. 5, pp. 199 - 220, 1993.
9. Musen, M.A., "Domain Ontologies in Software Engineering: Use of Protégé with the EON Architecture, Methods of Informations in Medicine.
10. VOM, <http://www.sandsoft.com>
11. Workflow Management Coalition(WfMC), <http://www.wfmc.org/>
12. W3C, SOAP, <http://www.w3.org/TR/SOAP>, 2000.
13. W3C, WSDL, <http://www.w3.org/TR/wsdl>, 2001