

능동문서: 서식설계자의 의도가 내장된 프로그램

남철기*, 배재학*, 유해영**

*울산대학교 컴퓨터·정보통신공학부

**단국대학교 정보컴퓨터학부

e-mail : cholki.nam@oracle.com, hjhbae@ulsan.ac.kr, yoohy@dankook.ac.kr

Active Documents: Another Kind of Program Implying Intention of Form Designers

Chul-Ki Nam*, Jae-Hak J. Bae*, Hae-Young Yoo**

*School of Computer Engineering and Information Technology
University of Ulsan

**Division of Information and Computer Science
Dankook University

요 약

인터넷 기반의 많은 응용이 문서중심의 컴퓨팅 모델로 전환되고 있다. 문서는 문서설계자의 의도를 함축하고 있으며 이는 업무처리 과정의 자동화에 적극적으로 활용될 수 있다. 이러한 문서의 본질 파악을 통해 본 논문에서는 문서를 실행 가능한 컴퓨터 프로그램의 시각으로 접근한다. 이를 위해 서식, 데이터, 규칙 그리고, 질의로 구성되는 능동문서 모델을 제시한다. 모델의 각 요소는 문서의 재사용성과 상호 운영성을 위해 XML 로 일관되게 표현된다. 능동문서는 사용자 인터페이스를 제공하는 수동적인 역할 뿐만 아니라 문서설계자가 의도하는 문서처리 절차와 업무규칙을 기계가 읽고 추론하여 처리할 수 있는 문서이다. 이를 통해 문서와 기계가 상호작용을 할 수 있으며 다른 응용 프로그램과 협력할 수도 있다. 이러한 능동문서의 적용 가능성을 보이기 위해 기업간거래(B2B) 시스템에서 구매주문 처리의 예를 보였다. 요컨대 본 논문에서 제시한 능동문서는 지식표현 및 처리기능이 내장되어 있는 바, 문서중심의 지능적인 응용프로그램 개발을 가속화하는 토대를 마련할 수 있을 것으로 기대한다.

1. 서론

월드와이드웹(이하 웹)은 1990 년대 정보통신 발전에 견인차 역할을 하였다. 이러한 웹의 사용자와 정보의 양이 증가함에 따라 이들 정보를 표현, 교환, 그리고 저장하기 위한 수단으로써 웹 문서의 중요성은 더욱더 부각되고 있다. 현재 웹 문서를 표현하기 위해 주로 사용되는 언어는 HTML(Hypertext Markup Language)이다. HTML 로 표현된 문서는 내용중심의 정보표현보다는 인간이 보기 편하도록 외형을 제시하는데 중점을 두었다. 이러한 이유로 에이전트가 처리할 문서의 내용은 단지 데이터에 불과하다.

본 논문에서는 문서에 대한 일반적인 시각을 달리하여 문서의 본질을 파악하고 이를 통해 지능적인 응용프로그램의 구현 가능성을 도출해 내고자 한다. 문서는 행위자체에 초점을 둔 기능적인 관점과, 문서의 장(Chapter), 절(Section), 문단(Paragraph) 등과 같은 외형에 주안점을 둔 물리적인 관점으로 나누어 볼 수 있다[1]. 문서의 기능적인 관점에서 볼

때, 문서는 수동문서(Passive Document)와 능동문서(Active Document)로 대별된다[2]. 수동문서는 내용과 구조화된 표현만을 가지고 있는 문서이다. 능동문서는 내용과 구조화된 표현 뿐만 아니라, 문서의 동적인 특성을 포함하는 문서이다[3].

능동문서라는 용어는 1988 년에 스피라드(Spinrad)에 의해 처음 언급되었다. 그 후 여러 연구에서 능동문서의 개념을 사용하였다[4]. 하지만, 능동문서의 응용 영역에 따라 서로 다른 개념을 사용한다. 접근방법에 의해 나누어 볼 때 클라이언트 영역에서 능동문서 처리를 하는 연구와 서버 영역에서 처리하는 연구로 나누어 볼 수 있다. 또한, 전자출판, W/MS, 그리고 메일 시스템에 능동문서의 개념을 도입하기도 하였다. 능동문서 시스템은 문서를 중심으로 사용자 상호작용이나 통신 등을 지원하는 시스템이다. 이는 내용과 행위정의를 내포하는 행동 가능한 문서를 통해 지능적인 어플리케이션을 구현하고자 하는 접근 방법이다.

본 논문에서는 인지 과학적인 측면에서 문서의 능동성을

파악하고, 이를 통해 서식문서를 능동적으로 실행 가능한 컴퓨터 프로그램의 관점에서 접근하였다. 즉, 일반적으로 자료처리의 대상이 되는 문서는 수동적인 것이 아니고, 그 문서의 설계자가 문서 작성자와 상호작용을 할 때 필요한 지식을 함축하고 있다는 점에서 능동적이라고 본다. 여러 종류의 문서 중에서 서식문서를 대상으로 하였다. 그 이유는 서식문서의 경우, (1) 사람이 단순히 문자를 해독하여 그 문서를 작성하는 것이 아니고, (2) 주어진 서식에 함축되어 있는 바, 서식설계자가 지향하는 업무처리과정 및 업무규칙을 추론하여, (3) 그의 의도를 능동적으로 재구성해 가는 이해 과정이 필요하다는 것이 잘 드러나는 문서이기 때문이다.

제안하고자 하는 능동문서는 그 내부에 선언적 지식을 포함하며 추론엔진을 통해 문서제어와 처리에 대한 자동화를 지원하는 컴퓨터 프로그램이다. 이를 위해 문서를 프로그램의 관점에서 조명하였으며, 이를 통해 문서의 모델을 재정의하였다. 능동문서 모델은 서식, 데이터, 규칙 그리고, 질의로 구성된다. 모델의 각 요소를 문서의 재사용과 상호운용성을 위해 XML 로 일관되게 표현하였다. 이 중에서 규칙은 Prolog 규칙으로 표현한 것을 XML 로 변환하였다. 이를 ERML (Executable Rule Markup Language)이라 칭한다. ERML 은 XML 의 장점을 그대로 가지고 있을 뿐만 아니라, 규칙의 교환과 저장도 가능한 마크업 언어이다. 이처럼 본 논문의 능동문서는 동적인 특성을 가지고 있을 뿐만 아니라, 응용프로그램 인터페이스와 유사한 상호 작용하는 요소를 프로그램 형태로 포함하고 있다. 일반적으로 프로그램과 문서는 서로 다른 객체로 취급되어져 왔지만, 능동문서는 이러한 이질적인 두 가지 특성을 함께 내포하고 있다.

2. 능동문서에 대한 새로운 접근법

본 논문에서 제안하는 능동문서는 의형상으로는 일반적인 형태의 서식문서다. 서식문서는 특정한 업무목적에 위해 만들어진 문서이다. 수많은 유형의 서식들이 존재하는데, 이러한 문서들은 대개의 경우 사용자 인터페이스만 제공하는 역할을 하고 있다. 일상적으로 사용하고 있는 업무용 서식 문서에는 문서 설계자가 문서에 반영하고자 하는 의도가 함축되어 있다. 이러한 의도는 (1) 문서작성규칙, 업무규칙과 그리고 문서처리 절차와 같은 명시적 지식과, (2) 서식설계자의 배경지식과 같은 암묵적 지식이다. 이러한 지식은 문서에 명시적으로 표현되어 있지 않을 수도 있다. 따라서 업무용 서식문서를 작성하기 위해서는 문자 언어로 기록한 텍스트 속에 포함된 서식설계자의 의도를 이해하고 서식문서 사용자가 가지고 있는 기존의 지식과 새로 얻은 지식을 통해 새로운 의미 체계를 재구성하는 단계가 필요하다.

한편, 서식설계자의 의도를 내장한 문서에 대해 지능적인 문서처리를 하기 위해서는 효과적인 지식표현방법이 필요하다. 요컨대 서식설계자의 의도를 명시적으로 문서에 포함시켜 이를 업무처리 과정의 자동화에 적극적으로 활용함으로써 문서처리의 효율성을 기대할 수 있다. 주지하는 바와 같이 지식 표현이란 실세계의 지식을 기계와 사람이 동시에 이해할 수 있는 형태로 표현하는 것을 의미한다. 이는 목적달성에 부합되는 구조를 가져야 할 뿐만 아니라 추론의 효율성, 지식 획득의 용이성, 저장의 간결성 및 표현의 정확성, 그리고 다양성 등을 갖추어야 한다.

웹의 창시와 발전을 주도한 팀 버너스-리 (Tim Berners-Lee)는 시맨틱 웹[5]을 위한 언어의 요건[6]으로 다음과 같이 다섯 가지 특성을 언급한 바 있다: (1) 간결한 문법 (Compact Syntax), (2) 잘 정의된(Well-defined) 시맨틱, (3) 지식을 표현하기에 충분한 표현력, (4) 효과적이고 강력하며 기계가 이해하여 처리 할 수 있는 추론 메커니즘을 가져야 하며, 그리고 (5) 대용량의 지식베이스를 구축할 수 있어야 한

다는 것이다. 이 중에서 특성 (3)과 (4)는 서로 상반되는 특성이므로 동시에 만족시키기 어렵다.

위의 다섯 가지 특성은 단지 지식표현과 계산 (Computation)에 주안점을 두고 있다. 지식의 교환 및 공유를 위한 특성은 빠져있다. 따라서 다음과 같은 특성이 추가되어야 한다: (6) 지식은 이종의 여러 시스템에서 교환 및 공유가 가능한 표현이어야 한다. 위 (1), (2), (3), (4)의 특성을 만족시키기 위해 본 논문에서는 선언적 지식표현을 하는 Prolog 를 사용하였다. 그리고 (5)의 특성을 지원하기 위해 객체관계형 데이터베이스를 이용하였다. 마지막으로 (6)의 특성을 위해 Prolog 로 표현되어 있는 규칙을 변환기를 통해 XML 형식의 규칙인 ERML 로 변환하였다.

3. 실행가능 규칙 마크업 언어(ERML)

ERML 은 Prolog 규칙의 XML 표현이며 서식설계자의 의도를 소프트웨어 에이전트가 읽고 이해할 수 있는 형태의 언어이다. 또한, 능동문서 처리 프레임워크[7, 8, 9] 기반 하에서 실행 가능한 규칙 마크업 언어이다. 이로써 지식의 상호운용성을 제공할 수 있는 기반을 마련할 수 있다. 광의의 의미에서 보면 XML 기반으로 규칙을 표현하고자 한다는 점에서 ERML 도 규칙 마크업 언어의 한 시도로 볼 수 있다. 그러나 ERML 은 단순히 규칙을 표현하는 데 그치는 것이 아니다. 즉, 서식문서에 내포되어 있는 서식설계자의 지식을 식별하여 명시적으로 표현하고 이를 업무처리과정에 적극적으로 활용한다. 또한, ERML 은 Prolog 추론엔진에서 실행가능하고 또한, 워크플로우 관리시스템과 연동되어 처리 될 수 있다는 점에서 기존의 규칙 마크업에 관한 연구 [10]들과는 구별된다.

XML 문서는 요소(Element)로 구성되어 있고 각각의 요소는 <태그> ... </태그>의 형식이다. Prolog 규칙을 XML 형식으로 변환하기 위해서는 Prolog 에서 사용하는 상수 (Constants), 변수(Variables), 구조체(Structures)와 같은 Herbrand Term 과 사실(Fact), 규칙(Rule)과 같은 Horn Clause 를 XML 요소로 변환하기 위한 명명규칙이 필요하며 [표 1] 과 같다.

[표 1] Prolog-XML 요소 명명규칙

Herbrand Term	Element	Horn Clause	Element
원자	<atom>	술어	<functor>
숫자	<number>	관계기호	<functor>
논리변수	<variable>	사실	<hn><relationship>
구조체	<structure>	규칙	<hn><relationship>

한 예로 구매관련 B2B 사이트에서 고객에 대한 할인율 정책이 다음과 같다고 하자: 만약 고객이 우수 고객이면서 총 구매금액이 1000 만원 이상이면 15%를 할인한다. 이와 같이 대부분 IF-THEN 형식의 자연어로 기술된 업무규칙은 다음 그림 1 의 *과 같이 Prolog 규칙으로 표현할 수 있다. 이 규칙은 지식의 교환 및 공유를 위해 Prolog2XML 변환기를 통해 *와 같이 적격환(Well-formed) XML 문서인 ERML 로 자동적으로 변환 가능하다. ERML 을 이종의 규칙 기반 시스템에서 사용하기 위해서는 해당 규칙 기반 시스템에 적용 가능한 규칙으로 변환되어져야 한다.

본 논문에서는 실험의 효율을 위해 역방향 추론 엔진을 가지고 있는 SWI-Prolog[11]를 사용하였다. ERML 을 Prolog 의 추론엔진에서 실행하기 위해서는 원래의 Prolog 규칙으로 변환하는 작업이 필요하다. 이는 DCGs(Definite Clause Grammars)를 이용하여 만들어진 XML2Prolog 변환기를 통해 변환한다. 그림 1 의 *에서는 리스트의 하나의 요소로 원래의 할인율 규칙(discountRate)이 표현됨을 확인할 수 있다.

이때 각각의 변수는 SWI-Prolog 에서 할당된 내부 변수로 표현된다. 리스트의 요소를 표현된 규칙은 질의 처리를 위해 Prolog 지식베이스로 저장된다.

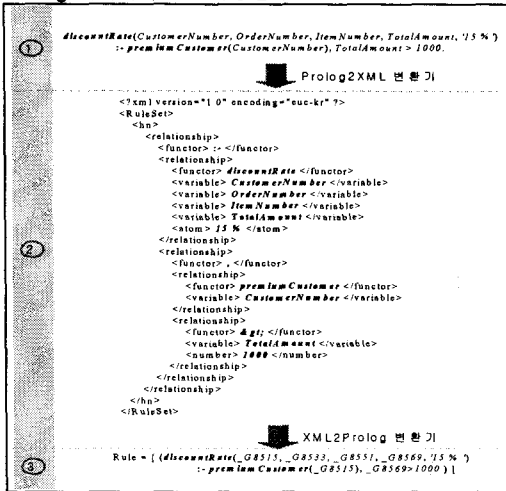


그림 1. Prolog 규칙과 ERML의 변환 예

4. 컴퓨터 프로그램으로서의 능동문서

일반적인 의미로 컴퓨터 프로그램이란, 특정한 문제를 해결하기 위해 기계가 사용하는 일련의 지시 및 명령을 말한다. 이러한 컴퓨터 프로그램을 통해 사용자에게 특정한 서비스를 제공하는 응용프로그램을 제공할 수 있다. 응용프로그램은 세 가지 구성요소로 이루어진다: (1) 데이터, (2) 규칙 또는 로직, 그리고 (3) 사용자 질의이다. 전술한 바와 같이 본 논문에서는 서식설계자가 의도하는 지식을 표현하기 위해 Prolog 를 사용하였다. 한 예로, Prolog 로 기술한 간단한 기업간거래(B2B) 응용프로그램은 다음과 같이 구성된다: (1) 판매 가능한 제품의 목록 또는 서비스, (2) 가격 할인정책과 환불규칙과 같은 업무규칙, 그리고 (3) 건적요청과 제품주문과 같은 사용자 질의이다. 다음은 도서 구매주문과 관련된 프로그램을 Prolog 로 간결하게 기술하였다.

- 판매제품: Prolog 지식베이스
book('8972806584', 'Oracle9i XML Applications', 'Scott', '30').
book('3974806649', 'Prolog in Java', 'Eddy', '25').
- 업무규칙: Prolog 규칙 (*activerule.pl*)
 (규칙 1) 할인율: 전체 구매액이 1000 만원보다 클 경우에는 할인율이 15.0%이며, 그 외의 경우에는 10.0%이다.
discountRate(CustomerNumber, OrderNumber, ISBN, Quantity, TotalAmount, '15.0 %') :- TotalAmount > 1000.
discountRate(CustomerNumber, OrderNumber, ISBN, Quantity, TotalAmount, '10.0 %').
 (규칙 2) 최소구매금액: 최소 구매금액은 300 만원이다.
minimumPurchase(CustomerNumber, OrderNumber, ISBN, Quantity, TotalAmount, '최소구매금액은 300 만원 입니다.') :- TotalAmount > 300.
- 사용자 질의: Prolog 질의 (*goal.pl*)
 % 200 번 고객이 Oracle9i XML Application 을 200 만원 구매할 경우의 할인율을 질의한다.
 :- (**discountRate**(200, 'PO-100', '8972806584', 10, 200, Rate), **format**("-q-n", [Rate])).

그림 2. Prolog 프로그램의 예

본 논문에서 제안하는 능동문서는 그 내부에 선언적 지식을 포함하며 문서제어와 처리에 대한 자동화를 지원하는 컴퓨터 프로그램이다. 문서를 프로그램적인 관점에서 접근하기 위해 문서의 모델을 재검립하였다. 그림 3 은 능동문서의 모델을 나타낸다.

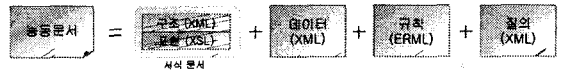


그림 3. 능동문서의 모델

문서모델을 구성하는 각각의 요소는 다음과 같이 다섯 가지로 구성된다: (1) 구조(Structure)는 문서의 논리적인 구조이다. 즉, 서식의 각각의 필드집합과 그 각 집합을 구성하는 필드이름, 필드형태와 필드길이를 정의하는 부분이다. (2) 표현(Presentation)은 XML 로 표현된 문서의 논리적 구조를 시각적으로 표현하는 XSL 문서이다. 구조를 표현하는 XML 문서에 표현을 나타내는 XSL 문서를 적용하여 HTML 서식문서를 생성한다. 이 서식문서는 사용자 인터페이스의 역할을 한다. (3) 데이터는 Prolog 추론엔진이 추론할 때 필요한 지식베이스를 제공한다. (4) 규칙은 업무문서에 함축되어 있는 업무규칙과 문서에 대한 처리방법을 의미하며, 일반적인 계산 로직, 업무규칙과 같이 업무 프로세스를 정의하고 제어하는 규칙, 데이터 검증에 필요한 데이터 무결성의 제약조건, 그리고 워크플로우의 호출 등이 있다. 이러한 규칙은 우선 Prolog 규칙으로 표현된 후에 Prolog 규칙의 XML 표현인 ERML 로 변환된다. 그리고 마지막으로 (5) 질의는 사용자가 서식에 입력한 데이터 검증 및 문서처리를 위한 것이다. 이들 각각의 구성요소는 XML 로 일괄되게 표현되어 서식문서 내에 함께 포함된다.

능동문서를 생성하는 단계에서는 크게 네 가지 작업이 이루어진다. 먼저, 본 논문에서 구현한 서식 디자이너를 이용하여 업무전문가가 WYSIWYG 환경에서 서식문서를 디자인 및 생성한다. 그림 4 에서 *orderform.xml* 는 서식문서 구조를 저장하고 있다 이 문서에 *orderform.xml* 스타일시트를 적용하면 HTML 기반의 서식문서를 생성하게 된다. 다음으로, 서식 디자이너에 포함되어 있는 규칙 편집기를 사용하여 업무규칙을 기술한다. 규칙 편집기는 업무전문가가 포인트와 클릭(Point & Click)과 목록상자(Listbox)와 같은 직관적인 인터페이스를 사용하여 쉽게 규칙을 만들 수 있다. 규칙편집기를 통해 작성된 규칙은 그림 2 와 같이 *activerule.pl* 파일에 저장된다.

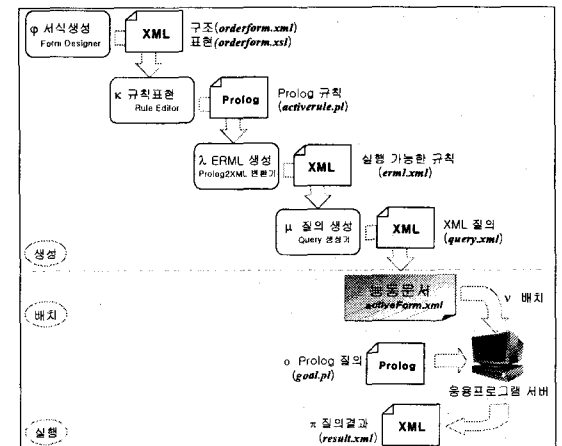


그림 4. 능동문서의 생성 및 실행과정

그림 4 의 4 번째 단계인 ERML 의 생성은 Prolog2XML 변환기를 통해 자동적으로 이루어진다. 또한, 질의는 4 번째 단계에서 생성된 서식필드의 데이터형태와 규칙에 따라 질의 생성기로 자동적으로 생성된다. 마지막으로 각 단계에서 생성된 파일은 그림 5 와 같은 activeForm.xml 에 저장된다. 즉, activeForm.xml 파일은 orderform.xml 과 erml.xml, 그리고 query.xml 을 포함한다. 각각의 엘리먼트중에서 <RuleSet> 엘리먼트는 지식 엘리먼트인 <hn>로 각각의 규칙을 나타낸다. 위와 같이 능동문서 프로그램을 나타내는 activeForm.xml 은 물리적으로는 3 개의 XML 문서로 구성되어 있지만 논리적으로는 하나의 문서이다.

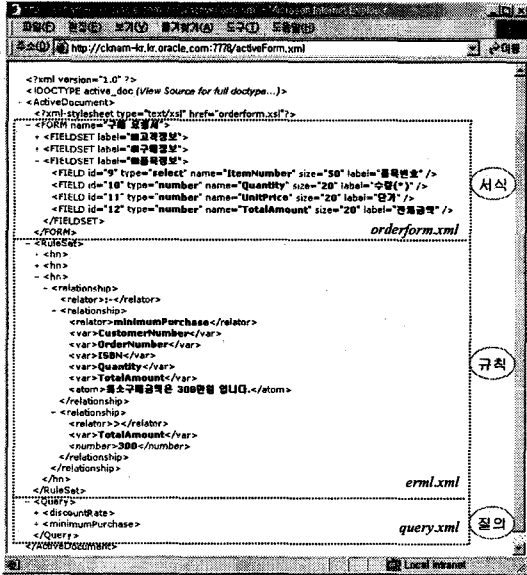


그림 5. 능동문서 프로그램(activeForm.xml)

능동문서를 실행하는 단계에서는 크게 두 단계의 과정을 거친다: 1 단계에서는 ERML 문서를 Prolog 규칙으로의 변환하는 단계이다. 이 단계에서는 서식문서의 데이터 검증과 처리를 위해 XML2Prolog 변환기를 사용하여 erml.xml 로부터 원래의 Prolog 규칙을 변환하여 Prolog 의 추론엔진에 저장한다. 그리고 2 단계에서는 사용자가 서식에 기입한 데이터에 근거해서 Prolog 질의를 수행한다. 이는 Prolog 추론엔진을 통한 데이터 검증 및 문서처리를 하기 위함이다.

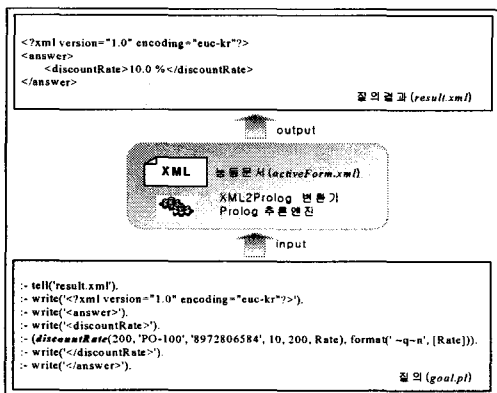


그림 6. 질의 및 결과

그림 6 과 같이 능동문서에 질의를 수행한 결과는 result.xml 로 저장된다. 이 경우에 그림 2 의 할인율정책에 의해 현재 고객은 전체구매금액이 200 만원이므로 10.0%의 할인율을 적용 받게 된다.

5. 결론

본 논문에서는 문서를 수행 가능한 컴퓨터 프로그램의 시각으로 접근하였다. 이를 위해 서식, 데이터, 규칙 그리고, 질의로 구성되는 능동문서 모델을 제시하였다. 모델의 각 요소는 문서의 재사용과 상호 운영성을 위해 XML 로 일관되게 표현된다.

본 연구에서 제안하고 있는 문서에 대한 컴퓨터 프로그램적인 접근법을 통해 기대할 수 있는 점은 다음과 같다: (1) 지금까지 문서의 역할은 단순히 정보의 표현 및 저장, 그리고 교환에 국한되었었다. 여기에서 한 단계 더 나아가 문서에 내포되어 있는 문서설계자의 지식을 식별하여 다양한 지식의 사용자가 공유할 수 있도록 하는 것은 문서의 진화를 위한 새로운 발판을 마련할 것이라 기대한다. 또한, (2) 서식설계자의 의도를 명시적으로 문서에 포함시켜 이를 업무처리 과정의 자동화에 적극적으로 활용함으로써 문서처리의 효율성을 기대할 수 있다. (3) 능동문서에 대한 본 논문의 접근법은 문서와 프로그램에 대한 시각을 통합 (Unification) 하였으며, 또한 문서처리와 프로그램실행, 그리고 질의처리에 대한 관점을 통합하였다. 요컨대 능동문서는 문서중심(Document-Centric)의 응용프로그램 개발을 가속화 하는 토대를 마련할 수 있을 것으로 기대한다. 마지막으로 (4) 능동문서 모델을 XML 로 일관되게 표현할 수 있기 때문에 이상적인 전자문서의 요건을 갖추고 있는 XML 의 장점을 그대로 가질 수 있다.

본 논문에서 논의한 능동문서는 지식표현 및 처리기능이 내장되어 있는 바, 의미 웹(Semantic Web)에서 의미 웹문서(Semantic Web Document)의 역할을 할 수 있을 것이라 기대한다.

참고문헌

- [1] Buckland, M. K., "What is a document?", Journal of the American Society for Information Science, Vol.48, pp. 804-809. 1997.
- [2] Paolo, C. and Robert, T. and Franco, Z., "Coordination Middleware for XML-centric Applications", 16th ACM symposium on Applied Computing, 2002.
- [3] Nam, C.-K. Jang, G.-S and Bae, J.-H. J, "An XML - based Active Document for Intelligent Web Applications", Journal of the Expert System with Applications (Forthcoming), 2003.
- [4] Helena Ahonen et al, Intelligent Assembly of Structured Documents. Technical Report C-1996-40, University of Helsinki, Department of Computer Science, June 1996.
- [5] W3C Semantic Web, <http://www.w3.org/2001/sw/>.
- [6] T. Berners-Lee, "The Semantic Web as a Language of Logic", <http://www.w3.org/DesignIssues/Logic.html>, 2000.
- [7] Nam, C.-K. and Bae, J.-H. J., "A Framework for Processing Active Documents", The 6th Russian-Korean International Symposium On Science and Technology, Proc. KORUS-2002, Vol.1, pp. 122-125, 2002.
- [8] 남철기, 장길상, 배재학, "능동문서에 대한 새로운 접근법과 그 응용", 한국정보과학회 논문지, 제 30 권, 제 4 호, 2003.
- [9] 남철기, 배재학, "업무규칙을 포함한 능동문서", 한국정보과학회 춘계학술발표대회, 제 29 권, 제 1 호, 2002.
- [10] RuleML Homepage, <http://www.dfki.uni-kl.de/ruleml/>.
- [11] SWI-Prolog Home, <http://www.swi-prolog.org/>.