

UML/XML 기반의 비즈니스 프로세스 정의 도구

한관희[†] · 황태일

경상대학교 산업시스템공학부/공학연구원

An UML/XML-Based Business Process Definition Tool

Kwan-Hee Han · Tae-Il Hwang,

Department of Industrial & Systems Engineering, Gyeongsang National University, Jinju, 660-701

Proposed in this paper is a standard-compliant business process modeling tool which is based on the UML(Unified Modeling Language) activity diagram and produces an XPDL(XML Process Definition Language) file as an output. The XPDL is a standard process definition exchange format by WfMC (Workflow Management Coalition). To develop an UML/XPDL-based modeling tool, the mapping of modeling elements between activity diagram and XPDL format is conducted after the detailed analysis of each modeling specification. As a result of this mapping, it is revealed that modeling elements of each activity diagram and XPDL must be expanded. So new modeling elements are identified and added to each specification. Based on this mapping, the prototype system is developed, and the usefulness of the developed system is shown through the case study.

Keywords: activity diagram, XPDL, UML, XML, business process modeling

1. 서론

최근의 급속한 기업 환경의 변화를 요약하면 공급자에서 고객으로의 힘의 이동, 국경을 초월한 기업 간의 경쟁, 기술과 시장의 급격한 변화로 특징지어질 수 있다(Hammer and Champy, 1993). 이러한 극심한 환경 변화 아래서 생존하기 위해 각 기업들은 프로세스 혁신(PI: Process Innovation)이나 BPR(Business Process Reengineering) 등을 통해 경쟁력을 확보하려 총력을 기울이고 있다. 이 과정에서 각 기업들은 자사가 수행하고 있는 프로세스의 정확한 분석과 설계의 중요성을 인식하게 되어 현재 대다수의 기업들이 BPR 수행시 비즈니스 프로세스 모델링 도구들을 사용하고 있는 실정이다. 프로세스 모델링의 결과는 BPR을 구현하는 데 직접 사용되거나 BPR 구현을 지원하기 위한 ERP(Enterprise Resource Planning) 시스템이나 워크플로 관리 시스템을 개발하거나 혹은 도입하는 데 필요한 입력 요소로 활

용된다. 특히, 최근에는 워크플로 관리 시스템이 BPR 구현의 핵심 정보 기술로 인식되면서 워크플로 관리 시스템의 운영을 전제로 한 비즈니스 프로세스 모델링 사례가 증가하고 있는 실정이다.

그러나 지금까지 기업에서 사용하고 있는 비즈니스 프로세스 모델링 도구는 몇 가지 중요한 문제점을 노출하고 있다. 첫째, 현재 대부분의 기업에서는 하나의 기업 내에서도 서로 다른 프로세스 모델링 도구들을 사용하고 있으며 이 도구들은 각기 고유의 모델링 방법과 저장 형식들을 사용하고 있어서 도구 간의 상호호환이 어렵고 구성원 간의 의사소통을 어렵게 하고 있다. 또, 이 때문에 사용자 입장에서는 여러 가지 도구들의 사용법을 중복적으로 익혀야하는 입장이다. 이러한 비효율성을 해결하기 위해서는 국제표준이나 산업표준을 준수하는 모델링 방법과 이를 근거로 한 모델링 도구의 개발과 사용이 시급한 과제라 하겠다.

본 연구는 산업자원부 전자상거래기술개발 사업(주문적응형 부품산업을 위한 웹기반 통합 제조 정보시스템 개발)의 지원으로 수행되었음.

[†]연락처 : 한관희 교수, 경남 진주시 가좌동 900 경상대학교 산업시스템공학부/공학연구원, Fax : (055)762-6599,

e-mail : hankh@nongae.gsnu.ac.kr

2003년 2월 접수, 1회 수정 후 2003년 4월 게재 승인

둘째, BPR 시에 사용되는 비즈니스 프로세스 모델링 도구와 워크플로 관리 시스템 간의 통합에 있어서 현재까지는 대부분의 워크플로 관리 시스템이 고유의 프로세스 정의 형식을 사용하고 있어서 프로세스 정의 도구에서 산출한 프로세스 정의 데이터를 수정 없이 워크플로 관리 시스템에서 해석하여 실행하기 어려운 구조를 갖고 있다. 그래서 대부분의 BPR 프로젝트에서 생성된 프로세스 재설계 데이터를 워크플로 관리 시스템에서 실행하기 위해서는 워크플로 관리 시스템이 가지고 있는 별도의 워크플로 정의 도구를 가지고 다시 모델링 해야 하는 낭비가 발생하고 있는 실정이다.

본 연구에서는 위에서 언급한 현행 비즈니스 프로세스 모델링 도구 사용에 있어서의 두 가지 중요한 문제점을 해결하기 위해, 비즈니스 프로세스 모델링 시 객체지향 모델링의 산업 표준인 UML(Unified Modeling Language) 모델링 방법을 사용하고 모델링 결과를 워크플로 관련 표준화 단체인 WfMC(Workflow Management Coalition)가 제정한 표준 프로세스 교환 포맷인 XPDL(XML Process Definition Language) 형식으로 산출하여 XPDL을 해석할 수 있는 워크플로 관리 시스템에서는 모델링 결과의 재정의 없이 바로 프로세스 실행을 가능케 하는 모델링 도구를 개발하고 사례 연구를 통해 그 유용성을 보이고자 한다.

2. 관련 연구

지금까지 비즈니스 프로세스를 모델링 하는 데 주로 사용된 도구로는 IDEF3(ICAM Definition 3)(Mayer *et al.*, 1995), ARIS EPC(Event-driven Process Chain)(Scheer, 1999), 페트리 넷(van der Aalst, 1998; Salimifard and Wright, 2001) 등을 들 수 있으며, Yim *et al.* (1998)과 Bosilj-Vuksic *et al.*(2000)에서는 IDEF 모델링 방법과 페트리 넷 방법과의 비교분석에 관한 연구가 이루어졌다. 최근에는 객체지향 기술의 발달에 따라 정보 시스템의 개발에 있어 객체지향 모델링 표준인 UML이 광범위하게 사용되는 추세인데, Marshall(2000)과 Eriksson and Penker(2000)에서는 UML이 표현력의 풍부함과 사용자 편의성 및 정보 시스템과의 통합성 등의 특성으로 인해 정보 시스템 개발에서뿐만 아니라 기업 프로세스 모델링에도 적합하다는 것을 보였고, Wirtz *et al.* (2000)와 Hurby(1998)에서는 UML을 비즈니스 프로세스 모델링에 보다 적합하게 만들기 위해 모델링 요소들을 확장하는 연구가 수행되었다. UML 사양의 일부인 OCL(Object Constraint Language)은 대상 영역 내의 제약 조건이나 규칙 등을 기술하기 위한 용도로 발표되었으며 UML 그래픽 표현의 한계를 보완하기 위한 언어로 사용되므로(Eriksson and Penker, 2000) 기업 프로세스와 같은 동적인 흐름을 모델링하기에는 부적합하다고 할 수 있다. UML 요소 중 하나인 활동 다이어그램은 기업 활동의 동적 흐름 표현, 이벤트 표현, 객체지향 표준 준수 및 사용자 편의성이라는 측면에서 기업 프로세스를 모델링 하는 데 적합한

방법으로 인식되어 활동 다이어그램의 장·단점을 분석하고 이를 보완하려는 연구가 다수 이루어졌으나(Bastos and Ruiz, 2002; Dumas and ter Hofstede, 2001), 최근까지 UML 활동 다이어그램을 기반으로 기업 프로세스를 모델링하려는 도구 개발에 관한 연구는 발표되지 않고 있는 실정이다.

비즈니스 프로세스의 정의와 정의된 프로세스 데이터의 시스템 간 교환을 위한 언어로는 WPDL(Workflow Process Definition Language), PSL(Process Specification Language), PIF(Process Interchange Format) 등이 있으며 zur Muehien and Becker(1999)에서는 이들 언어에 대한 상호 비교분석이 이루어졌다. 최근에는 XPDL(WfMC, 2002)과 BPML(Business Process Modeling Language)(Arkin, 2002)이 발표되었는데, XPDL은 XML 스키마를 기반으로 WPDL을 재정의한 것으로 WPDL에 비해 프로세스 정의 데이터의 유효성을 사전 검증할 수 있다는 점에서 주목을 받고 있으며 BPML 역시 XML 스키마를 기반으로 하고 있으나 기업의 내부 프로세스를 모델링 하는 데 초점이 맞추어져 있어서 시스템 간 프로세스 정의 데이터 교환을 위한 표준으로 사용하기에는 XPDL에 비해 부적합하다. 그리고 정의된 프로세스 데이터의 워크플로 관리 시스템에서의 실행 가능성 측면에서 볼 때 현재 많은 상용 워크플로 관리 시스템들이 WfMC에서 제정한 XPDL 포맷 해석 기능을 제공하거나 제공할 계획으로 있어서 실행 가능한 프로세스 교환 포맷으로는 XPDL이 다른 교환 포맷에 비해 적합하다고 하겠다. 보다 일반적인 형태의 모델링 도구 간 데이터 교환을 위한 XML 기반 사양으로는 OMG(Object Management Group)의 XMI(XML Metadata Interchange)(OMG, 2002)를 들 수 있는데, 이 XMI 사양은 UML 클래스 모델을 XML DTD(Document Type Definition)로 변환하거나 객체 다이어그램을 XML 문서로 변환하는 사양을 정의하고 있다. 그러나 XMI는 XML 스키마를 생성하지 못하고 UML 활동 다이어그램과 같은 행위 모델을 대상에서 제외했다는 점에서(Carlson, 2001) 동적인 기업 프로세스 데이터 교환 포맷으로 사용하기에는 부적합하다.

표준적인 기업 프로세스 정의 데이터를 산출해내는 도구에 관한 연구로는 Lee *et al.*(2001)과 Shin and Shin(2000)에서 고유의 기업 프로세스 모델링 방법으로 모델링하고 그 결과를 WPDL 파일로 변환하여 저장하는 방법에 대한 연구가 이루어졌고, Kim *et al.*(2000)에서는 모델링 도구로 ICN(Information Control Net)(Ellis, 1979)을 이용하여 모델링하고 그 결과를 WPDL 파일로 저장하는 연구가 이루어졌으나, 최근까지 모델링 결과를 XPDL 파일로 산출해내는 모델링 도구 개발에 관한 연구는 발표되고 있지 않은 실정이다.

3. 프로세스 정의 도구 기능 설계

3.1 요구 사항 분석

본 연구에서 목표로 하는 비즈니스 프로세스 정의 도구는 사

용자가 그래픽 기능을 이용하여 시각적으로 손쉽게 프로세스를 모델링 할 수 있어야 하고 모델링 과정에서 실제 기업 프로세스의 동적인 흐름을 정확하게 표현할 수 있어야 하며, 또 워크플로 관리 시스템에서 실행 가능한 워크플로 프로세스 정의 데이터를 산출할 수 있어야 한다. 이러한 요구 사항은 크게 모델링 기능, 그래픽 기능, XPDL 파일 관리 기능으로 나눌 수 있는데 이를 정리하면 아래와 같다.

첫째, 모델링 측면에서, 1) 프로세스의 동적인 흐름을 표현하는 순차, 분기, 병합, 반복 경로들이 표현될 수 있어야 하며 분기와 병합은 Hollingsworth(1995)에서 제시한 유형들이 모두 표현될 수 있어야 하는데 <표 1>에서는 여러 분기, 병합 유형을 UML 활동 다이어그램으로 매핑한 것을 나타내고 있다. <표 1>에서는 세 개 이상의 대안 경로 중에서 하나의 경로를 선택하는 'multi-choice OR'을 복수 개의 'XOR-split'으로 표현한다는 것을 보이고 있다.

2) 계층적 프로세스 표현을 위해 하나의 주 프로세스 내부 프로세스를 정의할 수 있고 이 부 프로세스가 주 프로세스와 동기적으로 실행되는지 비동기적으로 실행되는지를 구분하여 표현할 수 있어야 한다.

3) 하나의 프로세스에서 여러 번 사용되거나 반복하여 사용되는 활동들을 그룹으로 묶어서 사용할 수 있게 하기 위해 복수 개의 단위 활동들을 하나로 묶는 집합 기능이 필요하다.

4) 일반적인 정보 객체, 특히 단위 활동의 수행에 필요하거나 단위 활동 수행의 결과로서 산출되는 문서 정보와 문서의 흐름을 표현할 수 있어야 한다.

5) 각 단위 활동들이 어느 참여자 혹은 조직/역할에 의해 수행되는지를 구분하여 표현할 수 있어야 하고 이를 시각적으로 파악할 수 있어야 한다.

6) 하나의 활동에서 다른 활동으로 전이하기 위해 필요한 전이 조건을 명시적인 형식으로 표현할 수 있어야 한다.

7) 워크플로 관리 시스템 하에서 참여자가 실행하는 업무의

표 1. 활동 다이어그램의 분기/병합 표현

유형	활동 다이어그램의 표현
AND split	
AND join	
XOR split	
XOR join	
multi-choice OR	

유형은 ① 담당자가 수작업으로 업무를 진행하는 수작업 업무 (Manual), ② 담당자가 수작업으로 업무를 진행하되 애플리케이션의 참조가 필요한 애플리케이션 참조 업무 (Reference) 및 ③ 시스템에서 자동으로 애플리케이션을 수행하여 담당자의 개입 없이 업무를 완료하는 자동 업무(Application)로 구분될 수 있는데 모델링 시, 이를 명시적으로 정의할 수 있어야 한다.

둘째, 그래픽 기능에서는 모델링 요소를 표현하는 그래픽 심벌들의 드래그 앤 드롭, 크기 조정 등이 가능해야 하고 작성된 그래픽 모델의 자동 스크롤이나 자동 정렬 및 줌인/줌아웃 기능 등이 제공되어야 한다.

셋째, 산출된 XPDL 파일 관리 기능으로서 그래픽 모델을 XPDL 파일로 변환하여 저장하고 XPDL 파일의 유효성을 검증할 수 있는 기능과 XPDL 파일을 저장소에서 불러, 이를 그래픽 모델로 변환하는 기능이 제공되어야 한다. 이러한 기능적 요

표 2. 시스템의 기능적 요구사항

모델링 기능	1) Routing	Sequence
		Branch
		Merge
		Iteration
	2) Sub-Process	Sub Process 정의
		Sub Process의 동기/비동기적 실행
	3) activity set 정의	
4) Information Object	Document 정의	
	Document flow 표현	
5) Swimlane		
6) 명시적 전이 조건		
7) 실행 업무 유형 표현		
그래픽 기능	1) 심벌	Drag & Drop
		Resizing
	2) 전체 모델	Auto-scroll
Auto-alignment		
Zoom in/out		
파일 관리	1) XPDL 파일 save/load	
	2) XPDL 파일 유효성 검증	

구 사항을 테이블 형태로 정리하면 <표 2>와 같다.

이와 같은 기능적 요구사항을 만족시키는 모델링 도구를 설계하기 위해서는 우선 UML 활동 다이어그램 사양(OMG, 2001)에서 정의한 모델링 요소들이 XPDL 요소들과 상호 매핑이 가능한지를 분석해보고 매핑이 안 되는 요소들을 판별해 내야 한다. 또, 기존의 활동 다이어그램과 XPDL 포맷이 수용하지 못하는 요구 사항들을 판별해야 한다. 이들 매핑 불가능 요소들과 수용 불가능 요구 사항들을 충족시키기 위해서는 활동 다이어그램 모델링 요소나 XPDL 프로세스 정의 포맷을 확장하여야

한다. <표 3>에서는 활동 다이어그램과 XPDL 구성 요소 간 매핑을 나타내고 있다. <표 3>에서 보듯이 활동 다이어그램의 ‘객체(object)’ 구성 요소는 XPDL 구성 요소에는 존재하지 않으며, XPDL의 ‘패키지(package)’와 ‘활동 집합(activity set)’ 구성 요소는 활동 다이어그램에는 매핑할 수 있는 요소가 없다.

3.2 활동 다이어그램/XPDL의 기능 확장

<표 2>, <표 3>을 참조하여 아래와 같이 활동 다이어그램과 XPDL의 모델링 요소들을 각각 확장하였다.

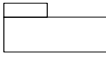

3.2.1 활동 다이어그램의 확장

XPDL 요소인 ‘패키지’와 ‘활동 집합’을 표현하기 위해 각각 해당되는 심벌을 추가하였고, XPDL에서 표현되는 부 프로세스의 동기/비동기적 실행을 표현하기 위해 활동 다이어그램 부 프로세스 심벌의 색으로 구분하였으며, 활동 다이어그램에서는 전이 조건에 대한 규정화된 형식론이 없으나 이를 명시적 표현이 가능하도록 규정화된 표현 형식을 도입하였다. 즉, 모든 전이 조건은 (operand1, operator, operand2)의 세 개의 요소로 표현한다. 또, XPDL에서의 실행 업무 유형 표현은 단위 활동 심벌의 색으로 구분하도록 하였다. 이를 요약하면 <표 4>와 같다.

3.2.2 XPDL의 확장

활동 다이어그램의 ‘객체’를 표현하기 위해 XPDL의 확장 속성을 가지고 구성 요소를 확장하였다. 활동 다이어그램의 ‘객체’는 물리적이거나 추상적인 것을 모두 포함하는 일반적인 개념인데 Bastos and Ruiz (2002)에서는 활동을 수행하는 자원이나 사람 혹은 관련 데이터를 표현하기 위해 ‘객체’ 개념을 사용하고 있으나 본 연구에서는 이들을 ‘활동’ 속성의 일부로 정의하고 있으며 ‘객체’는 활동과 관련된 객체 중에서 문서의 흐름을 명시적으로 표현하기 위하여 사용하고 있다. XPDL에서도 활동 다이어그램에서와 같이 전이 조건에 대한 규정화된 형식론이 없으므로 이를 규정화하였다. 그리고 XPDL에서는 실행 업무 유형을 수작업 업무와 자동 업무 두 가지로만 분류하는데 최근의 기업 업무 수행에는 담당자가 수작업으로 업무를 진행하되 애플리케이션의 참조가 필요한 ‘애플리케이션 참조 업무 (Reference)’가 많이 발생하여 이를 수작업 업무의 특수 유형으로 분류하고 확장 속성을 이용하여 정의하였다. 이를 요약하면 <표 5>와 같다.

표 4. 활동 다이어그램의 확장

요구 기능	확장 내용
XPDL Package 표현	 심벌 추가
XPDL Activity Set 표현	 심벌 추가
XPDL의 Sub Process의 동기/비동기적 실행	Sub-activity state 심벌 색으로 구분
명시적 전이 조건 표현	(Operand1 , Operator, Operand2)
XPDL의 실행 업무 유형 표현	Action state 심벌 색으로 구분

체’는 물리적이거나 추상적인 것을 모두 포함하는 일반적인 개념인데 Bastos and Ruiz (2002)에서는 활동을 수행하는 자원이나 사람 혹은 관련 데이터를 표현하기 위해 ‘객체’ 개념을 사용하고 있으나 본 연구에서는 이들을 ‘활동’ 속성의 일부로 정의하고 있으며 ‘객체’는 활동과 관련된 객체 중에서 문서의 흐름을 명시적으로 표현하기 위하여 사용하고 있다. XPDL에서도 활동 다이어그램에서와 같이 전이 조건에 대한 규정화된 형식론이 없으므로 이를 규정화하였다. 그리고 XPDL에서는 실행 업무 유형을 수작업 업무와 자동 업무 두 가지로만 분류하는데 최근의 기업 업무 수행에는 담당자가 수작업으로 업무를 진행하되 애플리케이션의 참조가 필요한 ‘애플리케이션 참조 업무 (Reference)’가 많이 발생하여 이를 수작업 업무의 특수 유형으로 분류하고 확장 속성을 이용하여 정의하였다. 이를 요약하면 <표 5>와 같다.

표 3. 활동 다이어그램과 XPDL의 매핑











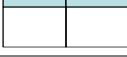
활동 다이어그램	심벌	XPDL 요소	비고
Action state		Activity	단위 활동
Sub-activity state		Workflow process	 로 대체 사용
Object		-	객체
Decisions		Route Activity	XOR split
Synch state		Route Activity	AND split/join
Start		Route Activity	프로세스의 시작을 표현
Finish		Route Activity	프로세스의 종료를 표현
Object flow		Transition	객체의 전이
Transition		Transition	활동 간의 전이
Swimlane		Participant	활동을 수행하는 주체 구분
-	-	Package	연관 프로세스들의 집합
-	-	Activity Set	활동들의 집합

표 5. XPDL의 확장

요구 기능	확장 내용
객체(Object) 표현	Extended Attributes에서 정의
실행 업무 유형 중 참조 업무 표현	수작업 업무의 특수 유형으로 표현, Extended Attributes에서 정의

3.3 분석 모델

기능적 요구사항을 만족시키기 위해 필요한 문제 영역 내의 객체와 객체가 갖고 있는 속성 및 객체 간의 관계를 표현한 모델을 분석 모델이라 하며 UML 클래스 다이어그램으로 나타내는데, 이는 <그림 1>에 나타나 있다. 패키지는 서로 연관성이 높은 복수 개의 프로세스를 묶는 관리 단위이며 하나의 프로세스는 여러 개의 단위 활동(activity)으로 구성된다. 하나의 프로세스 내에는 부 프로세스가 존재할 수도 있어서 단위 활동이 프로세스를 호출할 수도 있다. 프로세스는 활동 집합 클래스를 상속받는데, 활동 집합은 일반적인 활동들의 묶음을 의미하며 프로세스는 하나의 명시적인 시작과 종료를 갖는 특수한 활동 집합이다. 하나의 단위 활동은 활동과 활동 사이의 이동을 표현하는 전이 클래스와 업무에서 사용하는 애플리케이션 도구와 관련 데이터(relevant data) 및 업무를 수행하는 참여자(participant) 클래스와 연결되어 있다. 참여자는 활동을 수행하는 주체를 의미하며 사람 이외에 소프트웨어, 자원, 조직, 역할 등으로 지정할 수 있다. 관련 데이터는 프로세스 내에서 생성

되거나 사용되는 데이터를 의미하며 전이 조건을 표현하거나 프로세스 혹은 애플리케이션을 호출할 때 사용된다.

그리고 애플리케이션이나 부 프로세스 및 활동 집합을 호출하기 위한 별도의 중간 클래스들이 존재한다(Activity Set Invocation, Application Invocation, Sub-Flow Invocation). 주요 구성 요소인 패키지, 프로세스, 활동, 전이, 관련 데이터, 애플리케이션 클래스 들은 모두 ‘프로세스 정의’ 슈퍼 클래스의 상속을 받는 구조로 모델링 되어 있어서 공통 속성이나 메서드들을 재사용할 수 있는 구조를 갖고 있다.

4. 프로세스정의 도구구현 및 사례 연구

4.1 시스템 구조

3장에서 도출한 기능적 요구사항을 만족시키는 프로세스 모델링 도구를 개발하기 위해 개발 언어로는 플랫폼 독립적인 자바를 사용하였으며 JDK(Java Development Kit)는 J2SE v1.4를 사용하였다. J2SE v1.4에는 자체적으로 XML을 파싱(parsing) 하는 클래스가 내장되어 있어 XPDL을 파싱 할 때 이를 사용하였다. 객체지향 개발을 위해 UML 모델링 도구로 Rational Rose를 이용하였으며 그래픽유저 인터페이스를 위해서는 자바 스윙을 사용하였다.

본 연구에서 개발한 시스템(이하 C³-P:DEF라 칭함)의 기능 구조를 <그림 2>에서 나타내고 있는데, 크게 실제 비즈니스

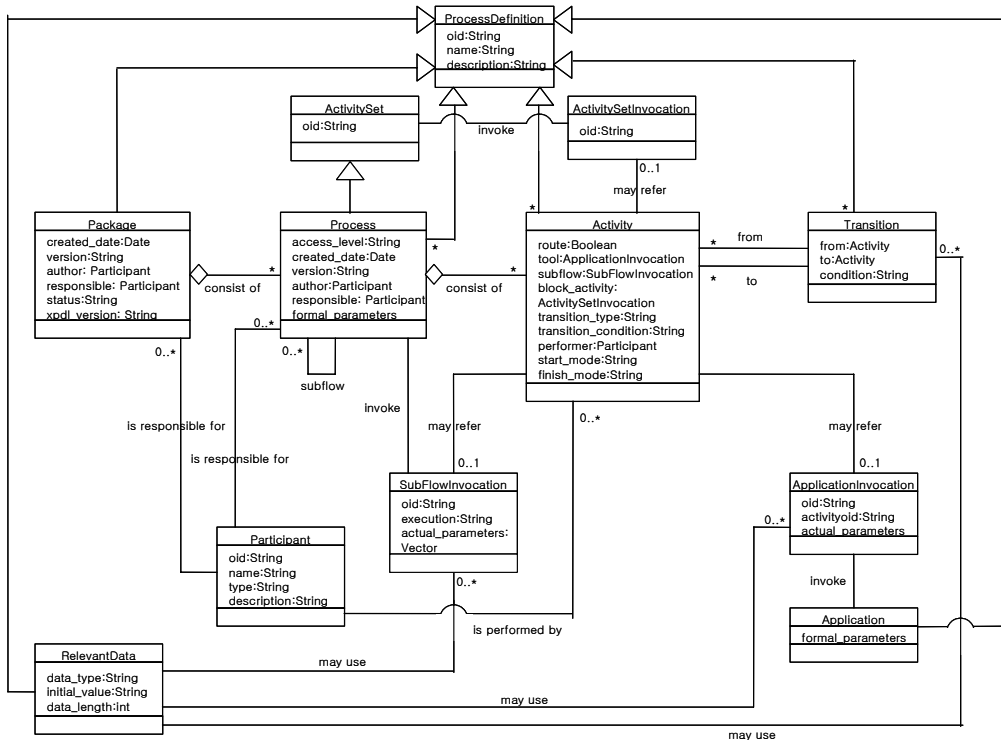


그림 1. 클래스 다이어그램.

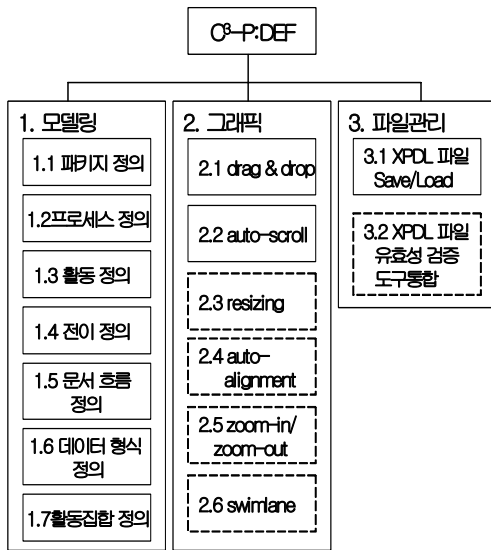


그림 2. C³-P:DEF 기능 구조도.

프로세스를 활동 다이어그램으로 모델링 하는 프로세스 모델링 기능과 모델링 시 그래픽 심벌들과 전체적인 그래픽 모델의 형상을 관리하는 그래픽 기능 및 모델링 결과 산출된 XPDL 파일을 저장하거나 불러오는 파일 관리 기능으로 나누어져 있다. 대상 비즈니스 프로세스를 그래픽 심벌들을 이용하여 활동 다이어그램으로 모델링하고 이 모델링 결과를 시스템에 저장할 때 활동 다이어그램이 XPDL 포맷 파일로 변환된다. 즉, 이 때 활동 다이어그램과 XPDL 간의 매핑이 발생한다. <그림 2>에서 점선 부분은 현재 구현되지 않은 기능을 표시한다. 모델링 기능은 패키지, 프로세스, 활동, 전이 및 문서 흐름 정의 기능이 기본 기능이며 이들 기능은 다음 절 사례 연구에서 상세하게 설명된다. 데이터 형식 정의는 정수형, 스트링 형 등 기본적인 데이터 형식 이외에 사용자가 원하는 특별한 데이터 형식을 정의하여 사용할 수 있게 하는 기능이며, 활동 집합 정의는 하나의 프로세스에서 여러 번 사용되거나 반복하여 사용되는 활동들을 하나로 묶어서 사용할 수 있게 하는 기능이다. 그래픽 기능은 현재 드래그/드롭 기능과 자동 스크롤 기능만 구현되어 있다. 파일 관리 측면에서는 그래픽으로 모델링 된 결과를 XPDL 파일로 저장하기 전에 파일의 유효성(validity)을

검증할 필요가 있다. XML 문서는 적격 (well-formed) 문서와 유효한(valid) 문서로 구분할 수 있는데 적격 문서란 어떤 문서가 XML 문법에 맞게 작성되었다는 의미이며, 유효한 문서는 적격일 뿐만 아니라 해당 문서를 규정하는 XML DTD나 스키마 규칙들을 만족하는 문서를 말한다(Boumphrey, 1999). 이러한 XML 문서의 적격성과 유효성 체크 기능은 상용 XML 편집기들이 일반적으로 제공하고 있다. 생성된 XML 파일이 XPDL에서 규정한 문법(syntax)과 의미(semantics)에 맞는가를 체크하는 유효성 검증은 현재는 사용자가 C³-P:DEF 시스템에서 산출된 XPDL 파일을 상용 편집기인 XMLSPY (altova, 2002)를 이용하여 수동으로 작업을 수행하고 있는데 향후에는 전용 검증 도구를 개발하거나 제3의 도구를 시스템 내에서 직접 통합하여 사용자 개입 없이 시스템에서 자동으로 검증을 수행하는 방식으로 확장되어야 한다.

4.2 사례 연구

대상이 되는 사례는 기업 정보 시스템 등과 같은 소프트웨어를 개발하는 데 있어서 개발 생산성 제고와 유지 보수를 향상시키기 위해 프로그램 작성에 사용되는 각종 데이터들을 표준화하여 여러 개발 프로젝트에서 표준화된 데이터만을 사용하도록 하는 ‘데이터 사전 관리’ 중의 일부 프로세스이다(Han, 2002). 여기서 데이터라 함은 프로그램 작성 시 사용되는 각종 클래스 명이나 클래스 내의 속성 명/메서드 명이나 데이터베이스 테이블 명/칼럼 명 등을 지칭한다. 대상 프로세스는 데이터 사전 관리 중 단위 데이터를 데이터 사전에 등록하는 데 요구되는 ‘단위 데이터 등록’ 프로세스이며 이를 활동 다이어그램으로 나타내면 <그림 3>과 같다. 데이터 사전에 없는 단위 데이터를 신규로 사용할 필요가 있는 개발자는 승인 과정을 거쳐 신규 단위 데이터를 중앙 데이터 사전에 등록한 후에 사용해야 한다. 프로세스가 시작되면 첫 번째 업무가 ‘승인 요청’이고 승인 요청 이후에는 두 가지 단위 업무가 병렬적으로 수행되는데, 하나는 이미 존재하는 유사 데이터를 대신 사용할 수는 없는지를 판단하는 ‘관련 데이터 분석’이고 다른 하나는 승인 요청된 단위 데이터가 데이터 생성 규칙에 부합하는지를 판단하는 ‘문법 체크’ 업무이다. ‘관련 데이터 분석’ 업무를 위해서는

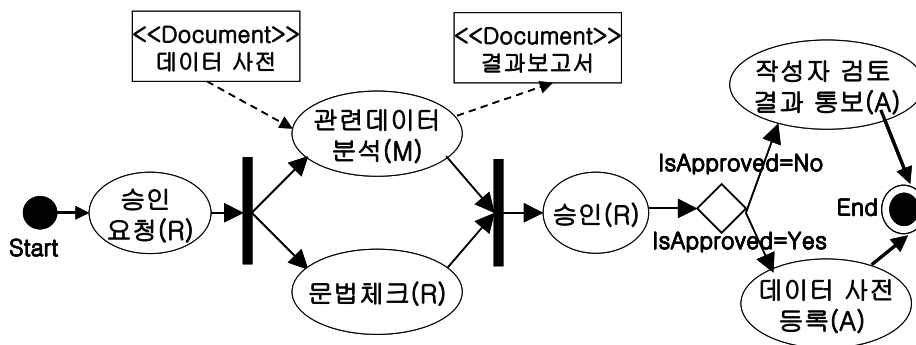


그림 3. 단위 데이터 등록 프로세스 시나리오.

‘데이터 사전’ 문서가 입력으로 필요하고 업무 수행의 결과 ‘결과 보고서’ 문서가 산출된다. 이 두 업무가 모두 완료되어야만 다음 ‘승인’ 업무가 가능한데 이 업무의 결과에 따라 승인이면 자동으로 데이터 사전 데이터베이스에 등록되며 부결되면 승인 요청자에게 반려 통지가 이메일로 발송된다. <표 6>은 각 활동을 정의하는 데 필요한 실행 유형, 필요 입출력 문서 및 사용하는 애플리케이션 정보를 정리한 것이다.

표 6. 각 활동별 필요 정보

활동명	실행 유형	입력 문서	출력 문서	사용 애플리케이션
승인 요청	참조(R)	-	-	단위데이터 등록
관련 데이터 분석	수동(M)	데이터 사전	결과 보고서	-
문법 체크	참조(R)	-	-	단위 데이터 등록
승인	참조(R)	-	-	단위 데이터 등록
결과 통보	자동(A)	-	-	메시지 통보
DB 등록	자동(A)	-	-	DB 등록

C³-P-DEF를 사용하여 예제 사례 프로세스를 정의하기 위해서는 패키지 정의 → 프로세스 정의 → 활동 정의 → 전이 정보 정의 → 문서 흐름 정의의 과정을 거친다. <그림 4>는 이와 같은 과정을 거쳐 예제 사례를 모델링 한 최종 화면을 나타내고 있다. 이하에서는 예제 사례를 모델링 하는 각각의 단계를 살펴보기로 한다.

4.2.1 패키지 (Package) 정의

서로 연관성 높은 프로세스들을 하나로 묶어 각 프로세스에

서 공통적으로 사용하는 애플리케이션이나 관련 데이터 등을 한 번만 정의하여 재사용하기 위한 프로세스 관리 단위를 ‘패키지’라 하며 XPDL 활동 표현의 최상위 요소가 되는데, 이를 정의하기 위해서는 <그림 4> 상단 톨 바에 있는 패키지 심벌을 클릭하면 주 화면에 패키지 심벌이 생기면서 정보를 입력하는 <그림 5>의 창이 뜨게 된다. 이 창에 요구되는 일반 정보를 입력하고 공통으로 사용되는 워크플로 관련 데이터나 애플리케이션들은 별도의 창에 입력한다. 예제 사례는 데이터 사전 관리 중의 일부 프로세스이므로 ‘데이터 사전 관리’ 패키지를 정의하게 되는데, 사용되는 XPDL 버전, 작성자, 책임자, 생성일, 파일 상태 등을 정보로 입력한다.

4.2.2 프로세스 (Process) 정의

‘데이터 사전 관리’ 패키지 내에 단위 데이터 등록 프로세스를 정의하기 위해 필요한 일반 정보와 해당 프로세스 내에서

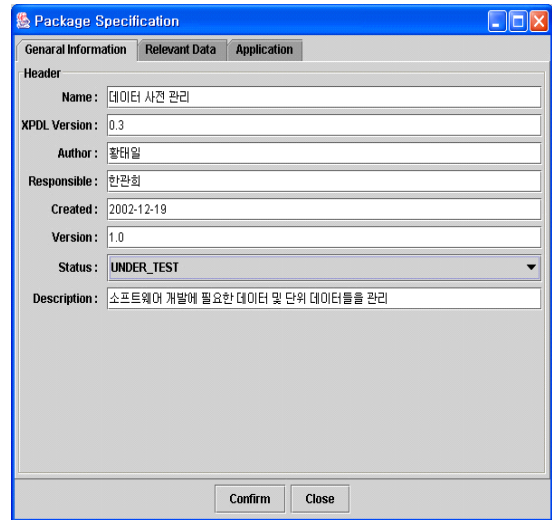


그림 5. 패키지 정의 화면.

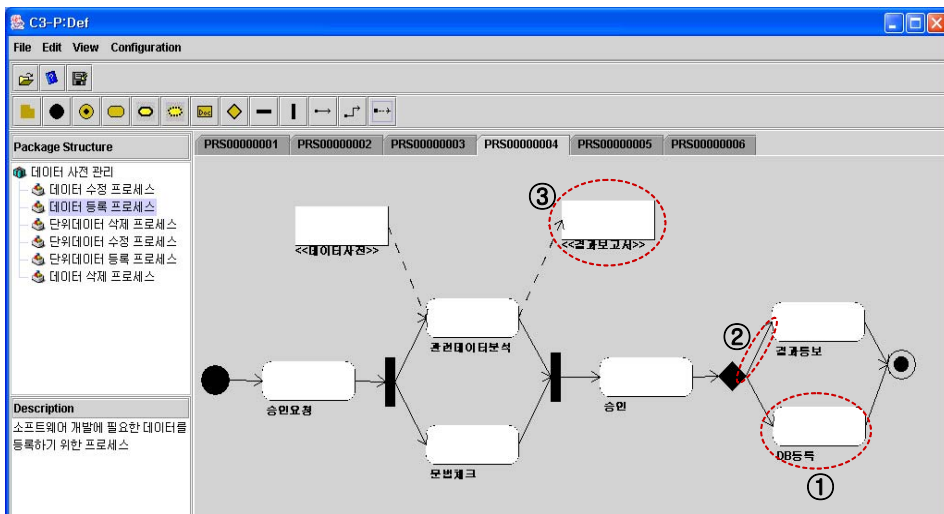


그림 4. 예제 사례 모델링의 최종 화면.

만 사용되는 워크플로 관련 데이터 및 애플리케이션 도구 정보를 추가로 입력하게 된다. <그림 6>은 프로세스의 일반 정보를 입력하는 화면인데 작성자, 책임자, 생성일, 버전 등의 정보를 정의하고 있다. <그림 7>은 ‘데이터 등록’ 프로세스에서 사용하게 될 워크플로 관련 데이터를 입력하는 화면이다. 워크플로 관련 데이터는 데이터 이름, 데이터 타입, 기본값, 데이터 크기 등의 정보를 입력하며, 이는 워크플로 실행 시에 부 프로세스 혹은 애플리케이션을 호출하기 위한 파라미터 값이나 전이 조건을 나타내는 정보로 사용된다.

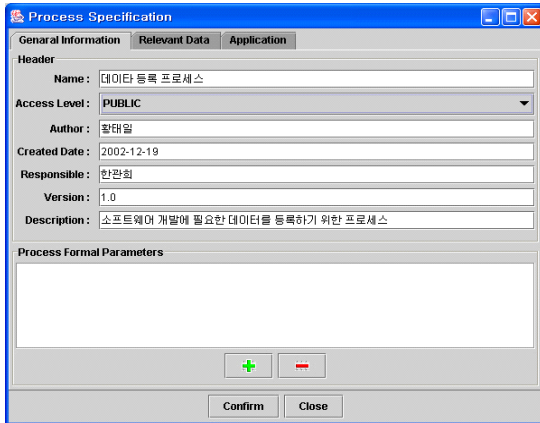


그림 6. 프로세스 일반 정보 정의.

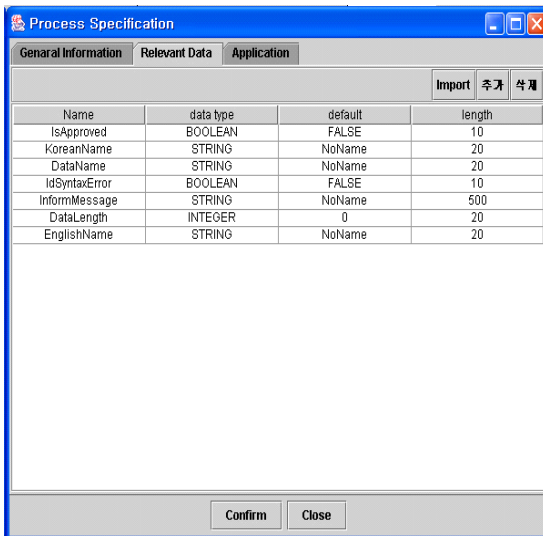


그림 7. 워크플로 관련 데이터 정의.

<그림 8>은 ‘데이터 등록’ 프로세스에서 사용하게 될 애플리케이션에 대한 정보를 입력하는 화면인데 예제 사례에서는 3 개의 애플리케이션을 등록하고 있다. 각 애플리케이션을 등록하기 위해서는 우선 애플리케이션의 이름을 입력한 후 애플리케이션을 호출하기 위해 필요한 파라미터를 입력해야 한다. 여기에서 입력하는 파라미터를 형식 파라미터(formal parameter)라 하며 파라미터에 대한 데이터 유형만을 정의한다. 파라

미터의 구체적인 값은 ‘활동’ 정의 시에 애플리케이션 호출시 필요한 실제 파라미터 값을 워크플로 관련 데이터에서 선택하게 되는데 이 때 사용되는 워크플로 관련 데이터를 실제 파라미터(actual parameter)라고 한다. 즉, 애플리케이션 호출은 이러한 형식 파라미터와 실제 파라미터와의 매핑을 통해 이루어지게 된다.

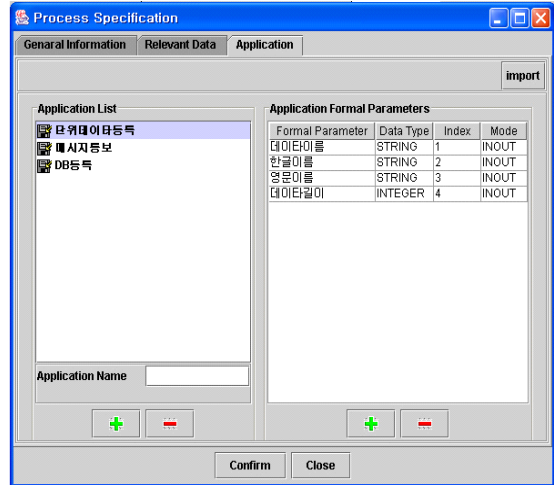


그림 8. 애플리케이션 정의.

4.2.3 활동(Activity) 정의

‘데이터 등록’ 프로세스가 생성이 되고 나면 <그림 3>에 나타난 각 활동을 정의해야 하는데 이를 위해 툴 바에서 ‘활동’ 심볼을 클릭하여 주 화면에 가져다 놓으면 활동에 필요한 정보를 입력하는 창이 뜨게 된다. <그림 9>는 <그림 4>에서 ①로 표시된 실행 업무 유형이 자동 업무인 ‘DB 등록’ 활동 정의를 보여준다. 정의에 필요한 정보로는 활동 이름, 실행 업무 유형, 참여자 유형, 시작/종료 모드 및 실행 유형이 애플리케이션인 경우 애플리케이션 호출에 필요한 파라미터를 정의하기 위해 형식 파라미터와 실제 파라미터 간의 매핑이 이루어진다.

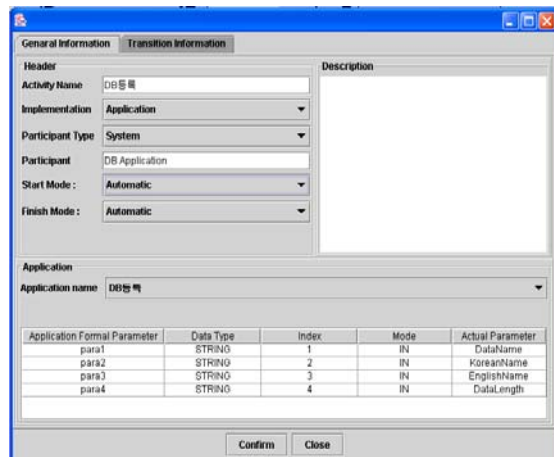


그림 9. 활동 정의.

4.2.4 전이(Transition) 정의

활동의 흐름은 전이를 통해 이루어지게 되며 전이의 유형은 순차, 분기, 병합, 반복으로 나눌 수 있고 이 중에서 분기와 병합은 <표 1>의 경우를 모두 모델링 할 수 있어야 한다. 분기의 경우 필요한 분기 조건을 정의해야 하는데 전이 조건은 (operand1, operator, operand 2)의 형식으로 정의된다. operand 1은 워크플로 관련 데이터를 사용하며 operator는 ==, >, <, >=, <= 중의 하나를 사용한다. operand 2를 입력할 때는 반드시 operand 1의 데이터 타입과 일치하는 값을 입력해야 한다. <그림 10>은 <그림 4>의 ②에 해당하는 전이를 정의하는 화면으로 상단에 전이의 시점/종점 활동 정보를 나타내고 오른쪽 하단에 ‘승인이 기각되면’이라는 조건을 (‘IsApproved’, ‘=’, ‘FALSE’)로 표현하고 있다.

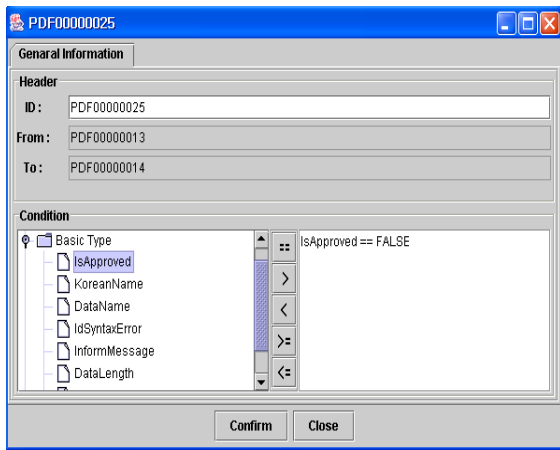


그림 10. 활동 전이 정의.

4.2.5 문서 흐름 표현

예제 사례에서 ‘관련 데이터 분석’ 활동은 문서의 입출력을 필요로 하고 있으며 이를 정의하기 위해서는 우선 필요로 하는 입출력 문서 내용을 정의하고 이를 객체 흐름선을 이용하여 관련 활동과 연결한다. <그림 11>에서는 <그림 4>에 ③으로 표시된 출력 문서인 ‘결과 보고서’의 정보를 나타내고 있다.

4.2.6 XPDL 파일 저장 및 불러 오기

<그림 4>와 같이 프로세스의 모든 정의가 완료되고 나면 정의된 프로세스를 저장할 수 있으며 저장 형식으로 WfMC의 XPDL 형식을 따른다. 또, 이미 저장된 XPDL 파일을 불러들여 활동 다이어그램 형태로 변환할 수 있다. <표 7>에서는 예제 사례에 나타난 단위 활동 중에서 ‘DB 등록’ 활동의 XPDL 포맷 산출 결과를 나타내고 있다. 이 파일에는 모델링 시 정의한 활동 이름, 실행 업무 유형, 애플리케이션 호출에 필요한 실제 파라미터, 시작/종료 모드, 그래픽 모델에서의 해당 심벌의 좌표 값 등이 나타나 있다.

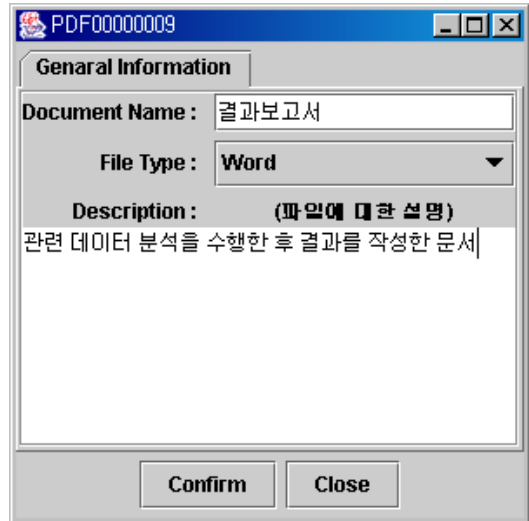


그림 11. 문서 정의.

4.2.7 산출된 XPDL 파일의 워크플로 실행

C³-P:DEF를 이용하여 예제 사례를 모델링 한 결과 <표 7>과 같은 XPDL 파일이 산출되었는데, 이 프로세스 정의 데이터가 워크플로 관리 시스템에 입력으로 들어가 실제 각 활동들이 실행되는가를 검증하기 위해 Han and Kim(2002)에서 개발된 C³-P:EXE 워크플로 관리 시스템을 이용하여 시험해 본 결과 성공적으로 수행됨을 검증하였다. <그림 12>는 C³-P:EXE 워크플로 관리 시스템의 화면으로서 상단 화면은 예제 사례인 ‘단위 데이터 등록’ 프로세스를 포함하여 현재 시스템에서 실행되고 있는 프로세스 목록을 보여주고 있으며 하단 화면은 예

표 7. 산출된 XPDL 파일 포맷

```

<Activity Id="PDF00000015" Name="DB등록">
<Description/>
<Implementation>
<Tool Id="DB등록" Type="APPLICATION">
<ActualParameters>
<ActualParameter>DataName</ActualParameter>
<ActualParameter>KoreanName</ActualParameter>
<ActualParameter>EnglishName</ActualParameter>
<ActualParameter>DataLength</ActualParameter>
</ActualParameters>
</Tool>
</Implementation>
<StartMode><Automatic/></StartMode>
<FinishMode><Automatic/></FinishMode>
<ExtendedAttributes>
<ExtendedAttribute Name="Coordinates">
<smi:Coordinates xpos="656.0" ypos="362.0"
Type="Activity"/>
</ExtendedAttribute>
</ExtendedAttributes>
</Activity>
    
```

제 사례 내의 각 활동들의 현재 상태 및 시작 시간, 완료 시간 등의 정보를 상세하게 나타내고 있다.

실행 프로세스 목록					
번호	프로세스명	관리자	상태	시작 시간	완료 시간
1	unit 단위 데이터 등록	김강용	RUNNING	2003/01/07 10:45:38	
2	Workflow 데이터 등록 프로세스	장태일	RUNNING	2003/01/08 05:03:05	
3	Data 단위 데이터 등록 프로세스	장태일	COMPLETE	2003/01/06 10:26:59	2003/01/07 15:22:12

실행 액티비티 목록						
번호	액티비티명	Dummy액티비티	담당자	상태	시작 시간	완료 시간
1	승인요청		김강용	COMPLETE	2003/01/07 10:45:38	2003/01/07 10:46:12
2		AND-SPLIT			2003/01/07 10:46:12	2003/01/07 10:46:12
3	관련데이터부식		관리자	RUNNING	2003/01/07 10:46:12	
4	문법체크		장태일	COMPLETE	2003/01/07 10:46:12	2003/01/07 10:50:12
5		AND-JOIN				
6	승인		한관희	NOT READY		
7		XOR-SPLIT				
8	결과통보			NOT READY		
9	DB등록			NOT READY		

그림 12. C³-P:EXE 시스템 화면.

5. 결론 및 추후 연구과제

본 연구에서는 현재 기업에서 비즈니스 프로세스 모델링 도구를 사용하는 데 있어서의 문제점을 분석하고 이를 해결하기 위해서는 표준을 준수하는 모델링 도구의 개발 및 사용이 필요함을 보였다. 모델링 도구를 개발하기 위해 기능적 요구 사항을 도출하였고 이를 구현하기 위해 UML 활동 다이어그램과 XPDL 구성 요소 간의 기능 매핑과 기능 확장을 하였다. 또, 개발된 시스템의 유용성을 실사례 연구를 통해 제시하였으며 산출된 XPDL 파일이 실제 실행됨을 증명하기 위해 본 연구의 사례 연구 결과로 산출된 XPDL 파일을 Han and Kim(2002)에서 개발된 C³-P:EXE 워크플로 관리 시스템에서 실행해 본 결과 성공적으로 수행됨을 알 수 있었다.

그러나 본 연구에서 개발한 시스템은 아직 프로토타입 수준이어서 실제적으로 기업에서 사용하기 위해서는 보완해야 할 점이 많이 있다. <그림 2>의 기능 구조도에서도 나타나듯이 사용자가 시각적으로 손쉽게 프로세스를 모델링 할 수 있는 그래픽 기능 보완에 관한 연구가 추가되어야 한다. 그리고 XPDL 파일을 사용자 개입 없이 검증이 가능하도록 유효성 검증 도구와의 시스템 통합도 시급하게 요구된다. 마지막으로 생성된 프로세스 모델의 적합성을 판단할 수 있는 시뮬레이터 개발에 관한 연구도 필요하다.

참고 문헌

김상배, 배송용, 김광훈, 백수기. (2000), 실시간 협업 지원 그룹 워크플로 모델링 도구, 2000년 한국정보처리학회 추계 학술발표 논문집, 7(2), pp.125-128.

신동일, 신동규. (2000), 워크플로 관리 시스템의 설계 및 구현, 한국정보처리학회 논문지, 7(5), pp.1609-1619.

이창수, 최혁승, 김한중, 김정수, 김선호, 조학래. (2001), WfMC 표준 기반의 Web-Based Process Designer 개발, 제9회 첨단 생산시스템 Workshop 논문집, 한국 생산 기술원.

임동순, 김철환, 우훈식, 김중인. (1998), 비즈니스 프로세스 모델링 연계 방법론: IDEF0, IDEF3, Petri Net, 한국 CALS/EC 학회지, 3(2), pp.141-160.

한관희. (2002), 원격 분산 환경에서의 소프트웨어 개발을 위한 통합 정보 객체 관리, 정보처리학회 논문지 D, 9-D(3), pp.427-434, 2002.

한관희, 김강용, 경량 컴포넌트 구조의 XPDL 기반 워크플로 관리 시스템 개발, Technical Report SMI-WF-TC-02-02, 경상대학교 산업 시스템 공학부.

Altova. (2002), Introduction to XMLSPY 5 Enterprise Edition, www.altova.com/manual, .

Arkin, A. (2002), Business Process Modeling Language Version 1.0, BPMI.org.

Bastos, R. M. and Ruiz, D. D. A. (2002), Extending UML Activity Diagram for Workflow Modeling in Production Systems, Proceedings of the 35th Annual Hawaii International Conference on System Science (HICSS-35), Big Island, Hawaii.

Bosilj-Vuksic, V., Giaglis, G.M. and Hlupic, V. (2000), IDEF Diagrams and Petri Nets for Business Process Modeling: Suitability, Efficacy and Complementary Use, Proceedings of the International Conference on Enterprise Information Systems (ICEIS 2000), pp.242-247, Stafford, UK.

Boumphrey, F. (1999), Professional XML Applications, WROX Press, UK.

Carlson, D. (2001), Modeling XML Applications with UML, Addison-Wesley, Upper Saddle River, NJ.

Dumas, M. and ter Hofstede. A. H. T. (2001), UML Activity Diagram as a Workflow Specification Language, Proceedings of the International Conference on the Unified Modeling Language(UML), Toronto, Canada.

Ellis, C. A. (1979), Information Control Nets: A Mathematical Model of Office Information Flow, Proceedings of the Conference on Simulation, Measurement and Modeling of Computer Systems. pp.225-239, ACM Press, New York.

Eriksson, H.-E. and Penker, M. (2000), Business Modeling with UML, John Wiley and Sons, New York.

Hammer, M. and Champy, J. (1993), Reengineering the Corporation: a Manifesto for Business Revolution, Harper Business, New York.

Hollingsworth, D. (1995), The Workflow Reference Model Version 1.1, Document Number TC00-1003, Hampshire, UK.

Hruby, P. (1998), Structuring Specification of Business Systems with UML, OOPSLA-98 Business Object Workshop Proceedings, Vancouver, Canada.

Marshall, C. (2000), Enterprise Modeling with UML, Addison Wesley, Reading, MA.

Mayer, R. J., Menzel, C. P., Painter, M. K., deWitte, P. S., Blinn, T. and Perakath, B. (1995), Information Integration for Concurrent Engineering IDEF3 Process Description Capture Method Report, KBSI Systems Inc., Texas.

OMG. (2001), OMG Unified Modeling Language Specification Version 1.4, Object Management Group, www.omg.org.

OMG. (2002), OMG XML Metadata Interchange(XMI) Specification Version 1.2, Object Management Group, www.omg.org, 2002.

Salimifard, K. and Wright, M. (2001), Petri Net-Based Modeling of Workflow Systems: an Overview, European Journal of Operational Research, Vol. 134(3),

pp.664-676.
 Sheer, A. -W. (1999), *ARIS Business Process Modeling*, Springer-Verlag, Berlin.
 van der Aalst, W. M. P. (1998), The Application of Petri Nets to Workflow Management, *the Journal of Circuits, Systems and Computers*, Vol. 8(1), pp.21-66.
 WfMC. (2002), Workflow Process Definition Interface - XML Process Definition Language Version 1.0, Document Number WfMC-TC-1025, www.wfmc.org.

Wirtz, G., Weske, M. and Giese, H. (2000), Extending UML with Workflow Modeling Capabilities, *Lecture Notes in Computer Science v.1901*, pp.30-41, Springer-Verlag, Berlin.
 zur Muehien, M. and Becker. J. (1999), Workflow Process Definition Language Development and Directions of a Meta-Language for Workflow Processes, *Proceedings of the 1st KnowTech Forum, Potsdam*.



한관희

아주대학교 산업공학과 학사
 한국과학기술원 산업공학과 석사
 한국과학기술원 자동화및설계공학과 박사
 현재: 경상대학교 산업시스템공학부 조교수
 관심분야: Business Process Modeling, Workflow, PDM



황태일

경상대학교 산업공학과 학사
 경상대학교 산업시스템공학과 석사
 현재: 대우정보시스템(주)
 관심분야: ERP, Workflow