

e-Business와 같이 규모가 크고 복잡한 프로세스가 수행되는 동안 문서는 생성이 되고, 수정과정을 반복적으로 거쳐 프로세스의 종료와 함께 완성된다. 따라서 프로세스의 진행과정은 곧 문서의 완성과정으로 볼 수 있으며, 문서자원의 변경과정과 프로세스의 진행을 연계하여 체계적으로 관리할 필요가 있다. 이를 위해서는 공유자원으로서의 문서를 프로세스 중심으로 바라보고 이에 대한 변경을 관리하여야 한다. 본 연구에서는 프로세스가 사용하는 자원 중 가장 XML 문서에 대하여 그 변경을 프로세스의 진행과 연계하여 관리할 수 있는 새로운 방법론을 제시한다.

2. 연구의 배경

2.1 비즈니스 문서와 XML

비즈니스 환경에서는 많은 데이터와 정보와 문서의 형태로 처리가 되며, 이러한 문서는 정

보처리의 효율을 높이기 위하여 정형화된 형태로 구성된다. 이를 양식문서(form document)라고 하며, 1980년대부터 이의 처리에 대한 연구가 꾸준히 이루어져 왔다[14][15]. 양식문서의 완성과정은 비즈니스 프로세스와 밀접한 관련을 가지게 되며, 프로세스의 진행과정에 따라 변경을 거듭하게 된다. 따라서, 문서가 중심이 된 프로세스에서는 문서의 완성과정을 프로세스의 수행과정으로 볼 수 있다.

본 논문에서는 XML(eXtensible Markup Language)을 이용하여 비즈니스 양식 문서를 구현한다. XML은 플랫폼에 독립적인 문서 표준으로 문서의 내용과 표현을 분리함으로써, 양식문서로 사용하기에 적합하다. 즉, 비즈니스 문서의 내용부분의 변경에만 집중하고 표현 부분은 XSL(eXtensible Styling Language)과 같은 언어를 사용하여 자유롭게 표현할 수 있다. 또한, XML은 구조적 문서를 지원함으로써 정형화된 비즈니스 문서를 작성할 수 있다. (그림 1)은

```

<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet href="order.xml" type="text/xml"?>
<ORDER>
  <ORDERSPEC>
    <PROCESSNAME>Order Process</PROCESSNAME>
    <FORMCREATOR>W.C. HUR</FORMCREATOR>
    <CREATEDATE>2001-03-06</CREATEDATE>
  </ORDERSPEC>
  <PRODUCTSTATE Workunit="WS_P001_01">
    <ORDERQTY>34</ORDERQTY>
    <PRODUCTIONQTY>724</PRODUCTIONQTY>
    <INVENTORYQTY>135</INVENTORYQTY>
  </PRODUCTSTATE>
  <PRODUCTSTATE Workunit="WS_P001_02">
    <ORDERQTY>13</ORDERQTY>
    <PRODUCTIONQTY>465</PRODUCTIONQTY>
    <INVENTORYQTY>113</INVENTORYQTY>
  </PRODUCTSTATE>
  <PRODUCTSTATE Workunit="WS_P001_03">
    <ORDERQTY>45</ORDERQTY>
    <PRODUCTIONQTY>204</PRODUCTIONQTY>
    <INVENTORYQTY>350</INVENTORYQTY>
  </PRODUCTSTATE>
</ORDER>
    
```

The screenshot shows a web browser window titled "Order Specification - Microsoft Internet Explorer". The address bar shows "http://22opphy.stu.ac.kr/workflow/form/order.html". The form content includes:

- Order Specification** header
- Process Name: Order Process
- Process Instance Name: M001 Order Process
- Form Creator: wchur
- Form Create Date: 2000-05-20
- Participant in charge of: Hyerim, Bae
- Form fields: Client Name (text), E-mail Address (text), Due Date (date: 04-11-2000), Product Name (text), Product Code (dropdown), Order Quantity (text), Factory Location (text).
- Description: "This form is involved to a order process R001. Fill in your data.."
- Product State table:

	Jan	Feb	March	April	May	June
Order_Quantity(unit)	300	500	300	100	0	0
Production_Quantity(unit)	50	20	50	200	0	0
Inventory(unit)	80	50	50	35	0	0
Material_A(unit)	400	200	170	70	0	0
Material_B(unit)	400	400	400	0	0	0
- Buttons: Submit, Reset

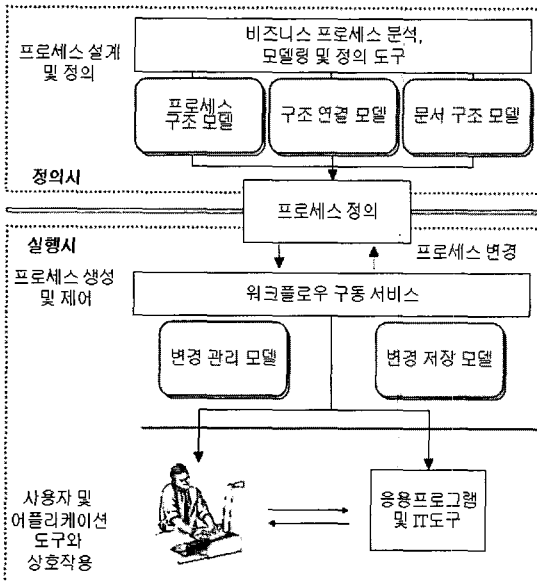
(그림 1) XML을 이용한 양식문서 구성의 예

XML을 이용하여 작성한 비즈니스 양식 문서의 예이다.

2.2 워크플로우관리시스템을 이용한 연구의 프레임워크

비즈니스 프로세스를 체계적으로 관리하는 통합관리 방법론인 BPM에서 핵심적인 정보시스템 역할을 수행하는 것은 워크플로우관리시스템(Workflow Management System, WfMS)으로 알려져 있다. 워크플로우관리시스템은 비즈니스 프로세스를 모델링하고 이의 체계적인 수행 및 관리를 통합적으로 담당하는 소프트웨어 시스템이다[9]. 본 논문에서는 이러한 워크플로우관리시스템 기능 위에 XML문서의 수명주기를 관리하기 위한 추가적 모듈을 제공함으로써 연구의 목적을 달성하고자 한다.

워크플로우 관리시스템은 정의 시 기능과 실행 시 기능으로 나누어 이해하는 것이 일반적인데[9], 본 연구에서는 이러한 WfMC(Workflow Management Coalition)의 표준 틀 위에서 다음 (그림 2)와 같은 연구의 프레임워크를 설정한다.



(그림 2) XML 수명주기 관리를 위한 연구의 프레임워크

먼저, 정의 시 기능은 프로세스 모델을 정의하여 프로세스가 수행될 수 있는 준비를 하는 단계의 기능이다. 다음으로 실행 시 기능은 프로세스 모델을 기반으로 프로세스를 시작시키고, 제어, 관리하는 데 관련된 기능이다. 본 연구에서는 프로세스를 관리하고 실행해주는 워크플로우 관리시스템이 문서의 변경을 체계적으로 관리하기 위한 추가적 모델들을 제안한다. (그림 2)에서 표현한 다섯 개의 세부 모델에 대하여 그 기능을 소개하면 다음과 같다.

- 프로세스 구조모델 : 프로세스의 구조를 표현하는 모델
- 문서 구조모델 : 문서 작업을 수행하는 데 필요한 문서의 구조를 정의하는 모델
- 구조 연결모델 : 분할된 각 문서의 영역을 프로세스의 단위업무에 할당하는 모델
- 변경 관리모델 : 프로세스의 구조 및 유형을 반영하여 버전이 자동으로 생성되도록 해주는 모델
- 변경 저장모델 : 생성된 문서를 효과적으로 저장하는 모델

위에서 기술한 세부 모델중 프로세스 구조모델, 문서 구조모델, 구조 연결모델은 정의 시에 이루어지는 모델에 해당하며, 변경 관리모델, 변경 저장모델은 실행 시 기능에 해당한다. 각 모델에 대해서 이어지는 두 장에서 상세히 설명한다.

3. XML문서관리를 위한 프로세스 정의 시 기능

일반적으로 워크플로우 관리시스템은 프로세스를 표현하는 모델을 가지는데[3], [4] 본 논문에서는 프로세스 기반으로 문서를 관리하기 위하여 프로세스뿐 아니라 문서를 구조적으로 표현하는 것이 필요하다. 먼저 프로세스와 작업단위에 대한 정의를 고려할 필요가 있으며 본 논문에서는 [5]에서 제시한 프로세스 구조모델을

사용한다. 이에 대한 더 자세한 사항은 [5]과 [6]을 참고하기 바란다.

■ 정의 1(프로세스 구조)

프로세스는 다음과 같은 유방향 비순환 그래프인 $P=(A, L)$ 로 구성된다.

- $A = \{a_i \mid i = 1, \dots, N\}$ 는 단위업무 집합이다. (a_i 는 i -번째 단위업무, N 은 G 에 속한 단위업무의 전체 개수)
- $L \subseteq \{(a_i, a_j) \mid a_i, a_j \in A \text{ and } i \neq j\}$ 는 링크의 집합이다. ((a_i, a_j) 는 a_i 가 a_j 에 대하여 바로 선행하는 링크)
- 분기 단위업무 a_i ($S = \{a_k \mid (a_i, a_k) \in L\}$, $|S| > 1$)에 대하여, 만일 모든 a_k 가 수행되어야 한다면 $f(a_i) = \text{'AND'}$ 이고 그렇지 않으면 'OR'이다.
- 병합 단위업무 a_j ($P = \{a_k \mid (a_k, a_j) \in L\}$, $|P| > 1$)에 대하여, 만일 모든 a_k 가 수행되어야 한다면 $f(a_j) = \text{'AND'}$ 이고 그렇지 않으면 'OR'이다.

다음으로 문서의 구조는 다음과 같이 정의된다.

■ 정의 2 (문서구조)

문서 d 는 다음과 같은 작업단위(Workunit) 들

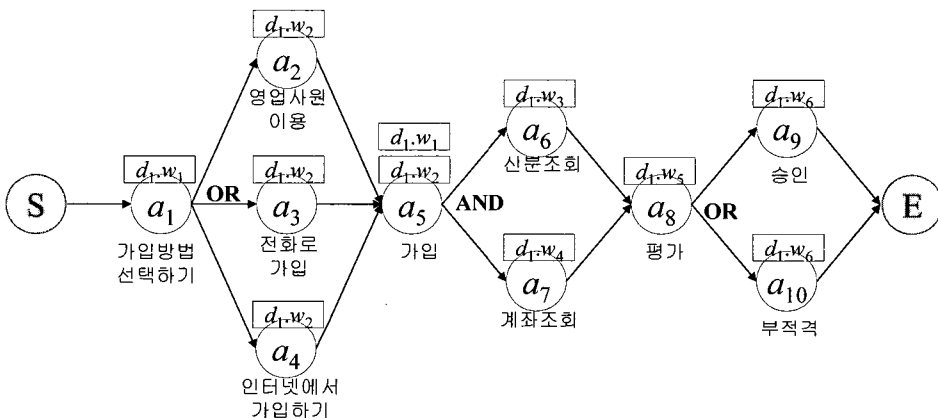
의 집합으로 정의되며, 각 작업단위는 다음과 같은 데이터 필드들로 구성된다.

- $d = \{w_m \mid m = 1, \dots, M\}$, (w_m 은 m -번째 작업단위, M 은 d 에 포함된 작업단위의 개수)
- $w_m = \{f_l \mid l = 1, \dots, L_m\}$, (f_l 은 l -번째 필드, L_m 은 w_m 에 포함된 필드의 수)

위의 정의에서 $w_m \cap w_n = \emptyset$ ($m \neq n$)이고, 하나의 데이터필드 f 는 이름, 값의 쌍인 (field name, value)로 정의된다.

본 연구에서 문서나 프로세스에서 사용되는 객체는 흔히 사용되는 .을 이용하여 표현한다. 예를 들면 프로세스 p 의 j 번째 단위업무 a_j 는 $p.a_j$ 그리고 문서 d_i 의 작업영역 w_j 의 필드 f_k 는 $d_i.w_j.f_k$ 와 같이 표기한다. 그리고, 프로세스와 문서를 서로 연계 시키는 모델로서 [5]에서 제시한 프로세스-문서 연계모델을 사용하기로 한다. 작업영역 $d_i.w_j$ 가 $p.a_j$ 에 할당되었을 때, 이를 ' $d_i.w_j \rightarrow p.a_j$ '과 같이 표현한다.

본 논문에서 고려하고 있는 프로세스-문서 연계모델을 이용하여 문서작업 중심의 워크플로우 프로세스를 정의한 예가 다음 (그림 3)에 나타나있다. 예를 들면 a_1 에서는 작업단위 w_1 을 처리하도록 할당되었다($d_1.w_1 \rightarrow p.a_1$).

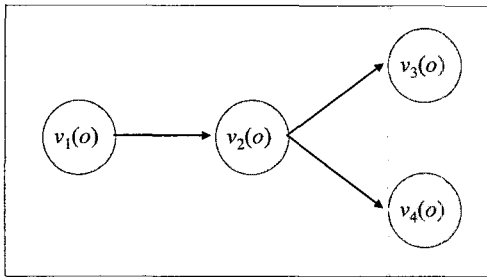


(그림 3) 보험 처리 예제 프로세스

4. 프로세스 실행과 XML 문서의 변경관리

4.1 문서 변경관리 모델

본 논문에서 XML의 수명주기 관리를 위해서 실행시에는 비즈니스 프로세스에서 XML 양식 문서의 사용에 따른 변경을 관리하기 위하여 버전관리 방법론[7]을 사용하기로 한다. 버전 관리 방법론은 객체의 변경을 다루어야 하는 다양한 분야에서 활용되어 왔으며([8], [10], [11], [12], [13]), 이러한 변경관리 방법론은 버전 그래프와 몇 가지 연계 모델을 개발함으로써 간단하게 적용해볼 수 있다. 먼저 버전 그래프는 다음 (그림 4)와 같이 표현되는데, 이 때, $v_1(o)$ 과 $v_2(o)$ 의 관계는 리비전 관계이며, $v_3(o)$ 와 $v_4(o)$ 의 관계는 배리언트의 관계이다.



(그림 4) 버전 그래프의 기본

다음으로 변경을 다루기 위한 각 객체 간의 연관관계를 표현하는 방법론으로, 문서 내에 존재하는 작업단위를 변경시키는 것은 해당 작업단위의 버전을 새로이 생성하고 이를 통해 그 작업단위를 포함한 문서의 버전을 새로이 생성하게 된다. ‘문서-작업단위 간 관계(Document-Workunit Relation, DWR)’는 문서의 버전과 작업단위버전간의 관계를 표현하기 위해 본 논문에서 고안한 모델이며 다음과 같이 정의된다.

■ 정의 3 (문서-작업단위간 관계)

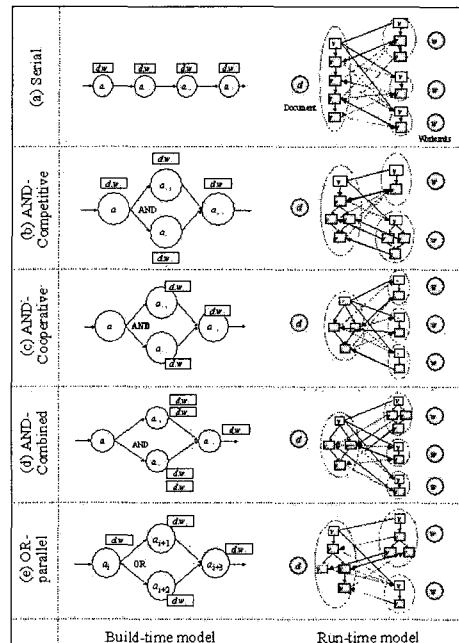
문서 d 와 작업단위 w 에서 DWR 는 d 의 p -번째 버전과 w 의 q -번째 버전, 즉 $v_p(d)$ 와 $v_q(w)$ 사

이의 관계는 다음의 세가지 중 하나이다.

- $v_p(d) \leftrightarrow^I v_q(w)$ 는 초기화 관계(initialization relation) DWR^I 이며, $v_p(d)$ 와 $v_q(w)$ 는 모두 초기 버전임을 나타낸다. 즉, $p = q = 0$ 인 경우이다.
- $v_p(d) \leftarrow^G v_q(w)$ 는 유도 관계(generation relation) DWR^G 이며, $v_q(w)$ 가 $v_p(d)$ 의 생성을 유도했음을 의미한다.
- $v_p(d) \rightarrow^C v_q(w)$ 는 조합 관계(composition relation) DWR^C 이며, $v_q(w)$ 가 $v_p(d)$ 의 요소임을 나타낸다.

4.2 버전의 유형

앞서 살펴본 정의 시 및 실행 시 모델에 기반하여 프로세스가 실행되는 동안 버전을 관리할 수 있게 된다. 생성되는 버전 및 그래프의 유형은 프로세스의 구조적인 유형에 따라 다르므로 각 프로세스의 구조 유형에 따른 버전 생성의 과정을 살펴보기로 한다. 더욱 자세한 사항에 대해서는 [3]을 참고할 수 있다.



(그림 5) 프로세스 유형에 따른 버전의 생성

4.2.1 직렬 프로세스(Serial Process)

직렬 프로세스에서의 단위업무들은 선형적으로 연결된다. 따라서, 직렬 프로세스는 항상 리비전만을 생성한다. (그림 5) (a)와 같은 직렬 프로세스를 고려하면 프로세스는 문서 d 에 대한 작업을 수행한다. 그리고, d 에 속한 작업단위들이 필요한 단위업무에 할당이 되어 있다. 그리고, 작업이 수행되는동안 작업을 위해 체크아웃되는 문서의 버전은 항상 최후 버전이다. 그리고, 체크인 과정에서는 항상 리비전을 생성하게 되므로 (그림 5) (a)의 오른쪽과 같은 버전 그래프가 생성된다.

4.2.2 And-병렬 프로세스(AND-parallel Process)

AND-병렬 프로세스는 여러 개의 단위업무가 동시에 수행되는 것을 지원한다. 분기 단위업무는 여러 단위업무가 하나의 작업을 수행하도록 허용한다. 분기된 단위업무들은 독립적으로 작업을 수행하고 그 결과로 역시 독립적인 버전을 생성한다. 본 논문에서는 AND-병렬 프로세스를 다시 경쟁분기(competitive split)와 협동분기(cooperative split)로 나누어 설명한다.

1) 경쟁분기(Competitive split)

경쟁분기는 분기된 각 브랜치의 작업 결과에 대하여 병합되는 시점에 평가하여 하나의 버전으로 합치게 되는 경우이다. 즉, 각 브랜치가 동일한 작업단위에 대한 작업을 수행하게 되므로 후에 이에 대한 버전을 합쳐야 되는 경우 각 결과에 대한 판단을 통하여 새로운 버전을 생성하게 된다. 이러한 경쟁분기 프로세스의 예제와 결과로 생성되는 버전 그래프를 (그림 5) (b)에 제시하였다.

2) 협동분기(Cooperative split)

협동분기는 분기된 브랜치가 서로 다른 작업

단위에 대한 작업을 수행하는 경우이다. 이 경우는 병합되는 시점에 자동으로 문서버전을 생성할 수 있다. 즉, 각 브랜치에서 작업한 결과가 다르므로 병합되는 시점의 새로운 버전의 문서는 각 브랜치에서 작업한 결과를 병합한 문서버전이 된다. 협동 분기의 버전 생성과정은 (그림 5) (c)에 제시하였다.

3) 혼합분기(Combined split)

혼합분기는 경쟁분기와 협동분기가 서로 혼재되어 있는 형태이다. 프로세스에 따라서 하나의 분기형태로 표현될 수 없는 경우가 있으며, 이 경우는 문서의 작업단위중 몇은 모든 브랜치에서 작업하고 몇은 서로 다른 브랜치에서 작업해야 하는 복잡한 경우가 발생한다. 이러한 경우에도 본 연구에서 고려하는 방법론으로 지원이 가능하며, 이에 대한 결과는 (그림 5) (d)에 제시하였다.

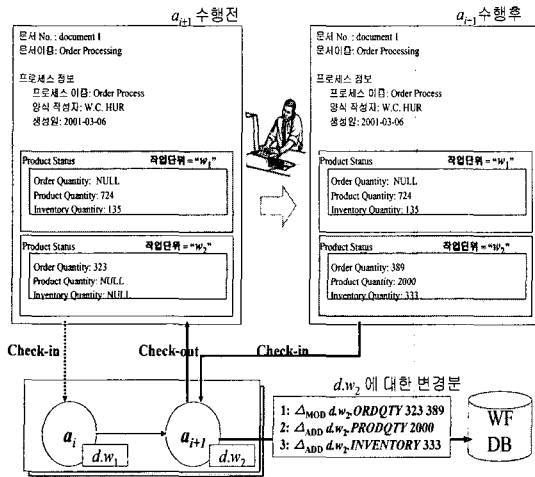
4.2.3 OR-병렬 프로세스(OR-parallel Process)

OR-병렬 프로세스는 분기와 병합이 있다는 점에서 구조적으로 AND-병렬 프로세스와 유사하지만, 의미적으로 차이가 있다. 즉, AND-병렬 프로세스와는 달리 분기된 각 브랜치 중 일부만이 의미를 가지는 경우이다. 이러한 경우는 버전 그래프에서 실제 수행된 브랜치에 대한 리비전만을 생성해주는 것으로 처리할 수 있다. 이에 대한 결과는 (그림 5) (e)에 표현하였다.

5. 변경의 저장

생성된 버전은 본 연구에서 고안하는 방식으로 저장이 되는데, 사용자의 작업으로 인한 변경은 크게 추가(add), 수정(mod), 삭제(del)의 세 가지 유형으로 저장된다. (그림 6)에 이러한 저장과정을 표현하였으며, 저장된 버전은 초기문서와 변경을 통하여 적절한 시점에 새로운 문서

로 구성하기가 용이한 장점을 지니고 있다. 이에 대한 자세한 사항은 [1]을 참고하기 바란다.



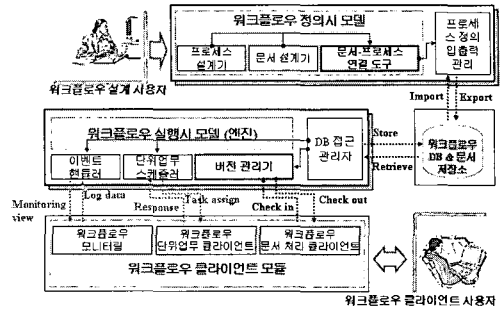
(그림 6) 생성된 새로운 버전의 문서 저장

6. 시스템 구현

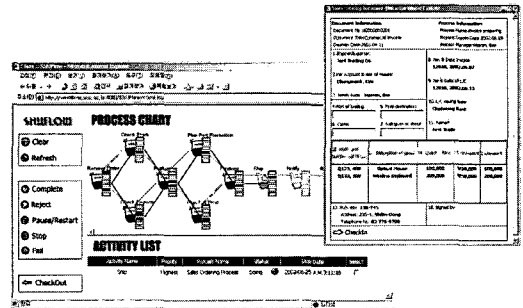
본 연구의 본문에서 제안한 바와 같은, 프로세스의 구조를 반영하여 자동으로 버전을 관리해주고 이를 저장 및 추출해주는 모델로 구성된 프레임워크에 대하여 프로토타입 시스템을 구현하였다. 이러한 시스템의 전체 구조는 (그림 7)과 같다. 그림에서 보는 바와 같이 전체 시스템은 크게 정의 시 모듈집합과 실행시 모듈집합, 그리고 클라이언트 모듈집합으로 구성된다. 먼저 정의 시 모듈집합은 프로세스와 문서의 설계를 각각 담당하는 프로세스 설계기와 문서설계기가 있으며 문서-프로세스 연결도구를 이용하여 둘 사이의 연관을 맺은 후 저장한다.

사용자는 주로 클라이언트 화면을 통하여 문서작업을 수행하는데 이에 대한 화면은 (그림 8)에 나타내었다. 그림에서 보는 바와 같이 사용자는 각 단위업무의 상태를 확인하면서 작업을 수행하도록 프로세스 차트를 제공한다. 어두운 아이콘은 이미 작업이 완료된 단위업무를 표현하며, 내부에 화살표 원을 포함한 아이콘은

현재 진행 중인 단위업무를 표현한다. 사용자는 아래쪽의 리스트를 확인하여서 자신에게 주어진 문서 관련 업무를 수행할 수 있다.



(그림 7) 전체 시스템 구조



(그림 8) 문서 변경관리를 위한 클라이언트 화면

7. 결론

본 논문에서는 프로세스가 수행되는 과정과 연계하여 XML양식 문서의 수명주기를 관리하는 통합적 방법론을 제시하였다. 이를 위하여 프로세스의 정의 시에 문서를 프로세스 수행에 적합한 형태로 구조화하여 이를 프로세스 단위 업무와 연계시키는 모델을 제안하였고, 이를 근거로 프로세스의 다양한 구조에 따라 문서의 버전이 자동으로 생성될 수 있음을 보였다. 또한, 각 버전이 효율적으로 저장 및 추출될 수 있도록 하는 모델로 제시하였다. 제안된 방법론은 기업에서 정형화된 양식문서를 기반으로 진행되는 프로세스에 적용 가능할 것으로 기대된다. 프로세스 정보를 바탕으로 한 문서 버전의

생성 및 관리는 여러 가지 장점을 제공한다. 먼저, 워크플로우 사용자가 작업해야 할 문서를 프로세스의 구조와 패턴에 맞게 자동으로 전달하여 줌으로써 사용자가 오류 없이 문서작업을 할 수 있도록 도와준다. 또한, 문서 버전을 프로세스와 연계함으로써 문서의 변경과정에 대한 사용자의 이해를 도와준다. 그리고, 프로세스의 전과정에서 문서를 추적하게 함으로써 프로세스가 종료되고 난 후에도 관리자가 특정 프로세스 위치에서 작업된 문서의 상태를 검색하는 것을 가능하게 한다. 특히, 여러 부서 혹은 기업이 관여된 복잡한 e-Business 프로세스에서 문서를 포함한 다양한 자원의 변경관리에 효과적으로 적용될 것으로 기대한다.

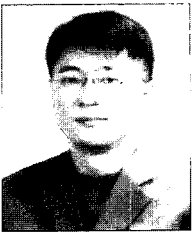
참고문헌

- [1] 배혜림, "e-Business 문서관리: 워크플로우 프로세스를 위한 비즈니스 문서 변경관리", 공학박사학위논문, 서울대학교, 2002.08
- [2] W.M. P. van der Aalst, "Process-oriented architecture for electronic commerce and interorganizational workflow," *Information Systems*, Vol.24, No.8, pp.639-671, 1999.
- [3] W. M. P. van der Aalst and A. H. M. ter Hofstede, "Verification of workflow task structures: A petri-net-based approach," *Information Systems* Vol.25, No.1, pp.43-69, 2000.
- [4] I. B. Arpinar, U. Halici, S. Arpinar and A. Dogac, "Formalization of Workflows and Correctness Issues in the Presence of Concurrency," *Distributed and Parallel Database*, Vol.7, pp.199-248, 1999.
- [5] H. Bae and Y. Kim, "A document-process association model for workflow management," *Computers in Industry*, Vol.47, No.2, pp.139-154, 2002.
- [6] H. Bae, W. Hur, W. S. Yoo, and Y. Kim, "Document Configuration Control Process Captured in in a Workflow," *Computers in Industry*, Vol.53, No.2, 2004
- [7] R. Conradi and B. Westfechtel, "Version Models for Software Configuration Management," *ACM Computing Survey*, Vol.30, No.2, June, 1998.
- [8] K. R. Dittrich and R. A. Lori, "Version Support for Engineering Database Systems," *IEEE Transactions on Software Engineering*, Vol.14, No.4, pp.429-437, Apr., 1988.
- [9] D. Hollingsworth, "Workflow management coalition specification: the workflow reference model", *WfMC specification*, 1994.
- [10] Randy H. Katz, "Toward a Unified Framework for Version Modeling in Engineering Database," *ACM Computing Surveys*, Vol.22, No.4, pp.375-408, Dec., 1990.
- [11] J. C. Miles, W. A. Gray, T. W. Carnduff, I. Santoyridis and A. Faulconbridge, "Versioning and configuration management in design using CAD and complex wrapped objects," *Artificial Intelligence Engineering*, Vol.14, No.3, pp.249-260, July, 2000.
- [12] E. Sciore, "Versioning and Configuration Management in an Object-Oriented Data Model," *VLDB Journal*, Vol.3, pp.77-106, 1994.
- [13] S. Y. W. Su, S. J. Hyun and H. M. Chen, "Temporal Association Algebra: A Mathematical Foundation for Processing Object-Oriented Temporal Database," *IEEE Transactions on Knowledge and Data Engineering*, Vol.10, No.3, pp.389-408, May/June, 1998.
- [14] D. Tschritzis, "Form Management", *Commu-*

nications of the ACM, Vol.25, No.7, pp.453-478, 1982.

- [15] S. B. Yao, A. R. Hevner, Z. Shi, and D. Luo, "FORMANAGER: An Office Forms Management System", ACM Transactions on Office Information Systems, Vol.2, No.3, pp.235-262, 1984.

저자약력



배 헤 립

1996년 서울대학교 산업공학과 학사
1998년 서울대학교 산업공학과 석사
2002년 서울대학교 산업공학과 박사
2002년~2003년 삼성카드 정보기획팀
2003년~2004년 동의대학교 인터넷비즈니스학과 전임강사
2004년~현재 부산대학교 산업공학과 조교수
관심분야: 워크플로우 관리시스템, 다조직 프로세스 통합 방법론, XML, eAI, 웹서비스
E-mail : hrbac@pusan.ac.kr



곽 병 권

1990년 숭실대학교 전자계산학과 학사
1995년 숭실대학교 전자계산학과 석사
1991년~현재 핸드소프트 BPM연구소장
관심분야: 워크플로우 관리시스템, BPM, XML, 웹서비스, 소프트웨어공학, 컴퓨터 아키텍처
E-mail : bkk@handysoft.co.kr