

워크플로우 시스템에서의 문서 변경 관리

배혜립*, 김영호*, 김규동**, 박병권**

* 서울대학교 산업공학과

** 핸디소프트 (주)

초록

본 논문에서는 문서를 중심으로 진행되는 비즈니스 프로세스에서의 문서변경을 워크플로우 시스템을 통해 지원하기 위한 새로운 프레임워크를 제안한다. 일반적인 워크플로우관리시스템은 크게 정의시 기능과 실행시 기능으로 나뉘어, 프로세스를 정의하고 이를 자동으로 실행할 것을 표준으로 정의하고 있다. 그러나, 워크플로우 시스템이 문서의 변경을 프로세스의 진행과정과 연계하여 효과적으로 관리하기 위해서는 이러한 기본적인 기능 외에 추가적인 기능을 요구하며, 본 연구에서는 이를 위하여 버전관리 기법을 도입한다. 제안된 방법론은 문서 및 프로세스의 구조, 문서와 프로세스간의 연계, 프로세스 수행에 따른 버전 자동생성, 버전의 저장 및 재구성, 그리고 버전에 대한 질의를 각각 표현해주는 다섯 가지 세부 모델을 포함한다. 제시된 세부모델의 유기적인 상호작용은 문서 변경과정에 대한 사용자의 이해를 돕고 문서작업의 편리를 도모한다.

1. 서론

워크플로우(Workflow)는 컴퓨터에 의해 자동으로 실행 및 관리되는 업무 프로세스를 말하며, 워크플로우관리시스템은 이 워크플로우를 정의하고 그에 따라 업무 프로세스를 정의, 실행, 관리해주는 소프트웨어 시스템이다[6].

최근에는 특히, e-Business의 환경에서 복잡하고 규모가 큰 프로세스가 실행되고 있는 점에 맞추어 워크플로우관리시스템이 많은 기업들의 근간시스템으로 자리잡고 있다. 이러한 환경에서 실행되는 프로세스들의

대부분은 문서를 기반으로 진행되거나, 문서가 프로세스의 진행에 중추적인 역할을 한다. 본 연구에서는 이러한 프로세스의 진행에 있어 발생하는 문서의 변경과정을 체계적으로 관리하는 워크플로우 프레임워크를 제안한다.

앞서 약속한 문서 기반 프로세스는 그 진행과정이 곧 문서의 변경 및 완성과정으로 인식되는 것이 일반적이다. 따라서, 워크플로우 관리시스템에서 문서작업을 효과적으로 수행하기 위해서는 문서에 대한 관리기능이 필요하다. 대부분의 상용 워크플로우관리시스템들은 부분적으로 문서관리 기능을 포함하는데 이들은 문서의 전달과 저장과 같은 단순한 기능만을 제공하여 프로세스 관점에서 변경을 관리하지 못하고 있다는 점에서 제한적이다.

본 연구에서는 문서의 변경을 프로세스의 진행과 연계하여 관리할 수 있는 새로운 프레임워크를 제시한다. 이러한 프레임 워크는 문서 및 프로세스의 구조, 문서와 프로세스간의 연계, 프로세스 수행에 따른 버전 생성, 문서의 저장 및 재구성, 그리고 문서에 대한 검색기능을 포괄적으로 제공한다.

본 논문의 문서 변경 관리 프레임워크는 프로세스의 관점에서 문서의 변경을 이해하도록 도와준다. 다시 말해서, 특정 프로세스의 상태 혹은, 위치에서의 문서 상태 및 변경 내용을 정확하게 확인할 수 있도록 함으로써 문서관리의 효율을 배가시킨다. 또한, 문서 버전을 저장하는데 있어 최소의 공간을 이용하여 변경의 내용만을 저장하는 것이 가능하고, 원하는 문서를 추출하는데 편리한 환경을 제공한다.

위에서 언급한 프레임워크는 여러 부서 혹은 회사가 함께 관여하여 프로세스를 진행하는 경우 더욱 효과적이다. 여러 부서가 관여하는 경우에는 다른 부서가 작업한 결과를 토대로 문서작업을 수행해야 하므로

각 부서가 작업한 내용의 이력을 기록하는 것이 문서의 효율적인 이용에 있어 중요하다. 즉, 프로세스가 종료된 후에도 전체 프로세스를 통해 발생한 문서의 변경이력을 추적할 수 있을 뿐 아니라, 프로세스의 이상종료에 의한 회복 요구가 발생한 경우에도 이를 토대로 문서의 상태를 복구할 수 있는 기반을 마련하게 된다.

2. 객체 변경과 워크플로우관리시스템

2.1 객체 변경 관리 기법

객체의 변경을 관리하는 기존 연구로 가장 대표적인 방법론은 버전관리 기법이다. 넓은 의미에서 버전관리는 시간에 따른 객체의 변화를 다루는 방법론이라 할 수 있으며, 버전이란 특정 시점에서 의미를 가지는 객체의 상태라고 정의할 수 있다[7].

버전 관리 기법에서 사용하는 모델은 대부분 객체의 변경 이력을 그래프로 표현하는 경우가 많으며[6][12] 다음 Figure 1에 간단한 버전 그래프를 표현하였다. 그림에서 각 버전간의 관계는 두 가지로 표현되는데 하나는 revision이며 다른 하나는 variant이다. 먼저 revision관계는 이전 버전의 수정에 의해서 새로운 버전이 생성되는 경우, 이들 두 버전간의 관계를 표현한다. 예를 들면, $v_2(o)$ 는 $v_1(o)$ 의 revision이며 $v_3(o)$ 는 $v_1(o)$ 의 revision이다. 한편, variant는 하나의 문서를 서로 다른 방식으로 수정하여 각기 다른 대안이 될 수 있는 두 버전간의 관계이다. 예를 들면 그림에서 $v_3(o)$ 는 $v_4(o)$ 의 variant이다.

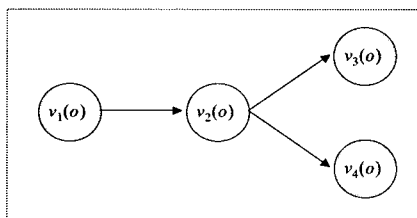


Figure 1. 기본 버전 모델

2.2 워크플로우 관리시스템과 문서관리

워크플로우 관리는 비즈니스 프로세스 자동화와 관련된 용어이며, 워크플로우관리시스템은 워크플로우 관리를 구현하는 소프트웨어 시스템으로 최근 그 적용영역을 넓혀가고 있다 [4]. 프로세스 자동화는 비즈니스 프로세스의 표현을 컴퓨터가 이해할 수 있는 형태로 정의한 프로세스

모델에 기반을 두고 있으며, 워크플로우 관리 시스템은 프로세스 모델을 이용하여 프로세스를 정의한 후 이를 실행하는 것을 통해 프로세스를 자동화한다.

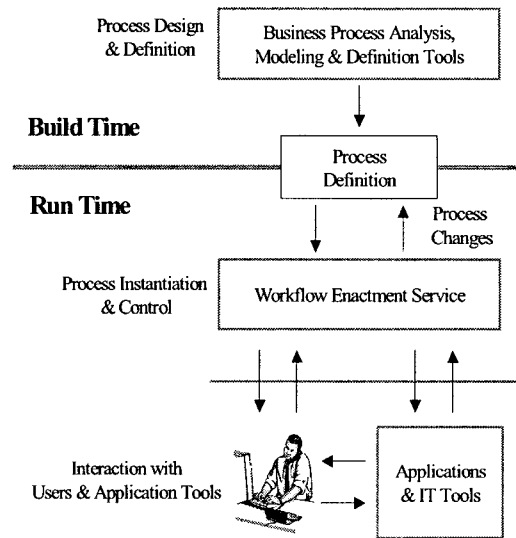


Figure 2. 워크플로우관리시스템의 기능특성[6]

이러한 워크플로우관리시스템의 특성은 워크플로우 표준을 관장하는 WfMC (Workflow Management Coalition) [6] 에서 제정한 표준명세에서 찾아볼 수 있는데, Figure 2에서 보는 바와 같이 두 가지 기능을 축으로 설명될 수 있다. 그 중 하나는 정의시 기능이며 다른 하나는 실행시 기능이다. 정의시 기능은 프로세스가 실행되는데 필요한 사항을 미리 정의하여 실행시에 곧바로 실행될 수 있도록 준비하는 단계의 기능이다. 이러한 정의의 내용은 프로세스를 이루는 단위업무와 이들간의 선후관계 그리고, 각종 속성(단위업무 이름, 처리예상시간, 담당자, 등)을 포함한다. 한편, 실행시 기능은 정의된 프로세스가 정확하게 수행되도록 스케줄을 하며, 단위업무를 구동시키고, 각종 자원을 할당하는 기능을 통해 프로세스를 실행, 조정, 제어한다.

워크플로우관리시스템이 관리하는 대부분의 비즈니스 프로세스는 문서작업을 포함하고 있으며, 상용 제품들을 이들 문서를 관리하기 위한 다양한 방법을 제공한다. 먼저 일부 상용 워크플로우관리시스템은 내장 함수를 가지고 있다. 이러한 내장함수를 통해서 프로세스가 진행되는 동안 문서의 전달 및 저장을 위한 기능을 제공한다. 워크플로우관리시스템에서 문서를 관리하는 다른 방법은 기존의 문서 관리 시스템 (EDMS,

Electronic Document Management System)과 인터페이스 하는 방식이다. 이러한 방식은 이미 문서관리 시스템을 구축하여 사용하고 있는 기업이 워크플로우를 도입하는 경우 유용하며 두 시스템이 상호 독립적일 수 있다는 점에서 효과적이다. 그러나, 이 두 가지 방식은 모두 문서의 최근 상태만을 유지한다는 데 한계가 있으며 프로세스의 관점에서 문서의 변경을 관리하지는 못한다.

3. 워크플로우 문서변경관리 프레임워크

앞서 언급한 바와 같이 워크플로우관리시스템은 정의시 기능과 실행시 기능으로 나누어 이해하는 것이 일반적이다. 이러한 틀 위에서 문서의 변경이 관리 되기 위해서는 워크플로우 프로세스의 정의시와 실행시에 이와 관련된 기능을 수행할 수 있는 모듈을 추가하여야 한다. 본 연구에서는 이러한 문서 변경 관리 모듈을 Figure 3과 같은 다섯 가지 모델을 통하여 제공한다.

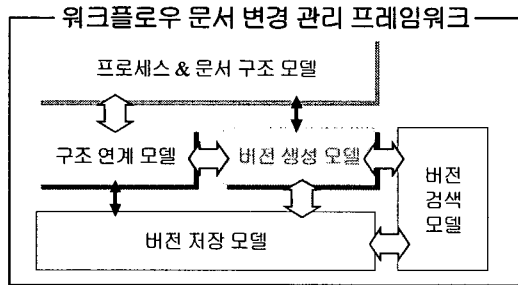


Figure 3. 문서 변경 관리 프레임워크

위의 그림에서 고려하고 있는 프레임워크 중 '프로세스 & 문서 구조 모델'은 기존 워크플로우 관리시스템의 표준 기능인 프로세스 구조 모델에 문서 구조모델을 추가한 것으로 프로세스 구조 모델에 대해서는 기존의 연구결과와 표준을 따른다. 프로세스 구조 모델을 제외한 나머지 세부 모델의 역할을 설명하면 다음과 같다.

- 문서구조모델: 문서구조모델은 프로세스의 진행과 연계하여 문서 작업을 수행하는 데 필요한 문서의 구조를 정의한다. 본 연구에서는 비즈니스 프로세스에서 사용되는 문서를 고려하여 문서의 일부분이 작업의 단위가 될 수 있도록 분할한다.
- 구조연계모델: 구조연계모델은 프로세스와 문서의 구조모델을 통하여

만들어진 프로세스와 문서가 상호 연관을 가지게 함으로써 실행시에 원하는 문서작업이 프로세스와 연계되어 이루어질 수 있도록 한다. 즉, 분할된 각 문서의 영역을 프로세스의 단위업무에 적절히 할당한다.

- 버전생성모델: 버전생성모델은 정의시에 명세된 프로세스 및 문서구조모델과 구조연계모델을 바탕으로 실행시에 버전이 자동으로 생성되도록 하는 모델이다. 이러한 버전 자동 생성은 프로세스의 구조 및 유형을 반영하여 이루어진다.
- 버전저장모델: 버전저장모델은 하나의 문서에 대하여 여러 버전이 생성됨으로 인해 발생하는 저장공간의 낭비를 최소화하는 모델이다. 본 연구에서는 버전간의 변화분만을 저장하고 이를 통하여 문서가 효율적으로 저장되도록 한다.
- 버전검색모델: 버전검색모델은 실행시에 생성된 문서 버전에 대하여 사용자가 원하는 정보를 정확하게 추출할 수 있도록 하는 방법을 제공한다.

위에서 기술한 세부 모델중 프로세스 및 문서구조모델, 구조연계모델은 워크플로우 표준 구조하에서 정의시 기능에 해당하며, 버전생성모델은 실행시 기능에 해당한다. 나머지 두모델인 버전저장모델, 버전검색모델은 문서 관련 프로세스의 실행에 직접적인 영향을 주지는 않지만 프로세스 실행을 측면에서 지원한다.

4. 구현 및 시스템 구조

본 논문의 본문에서 제안한 바와 같은, 프로세스의 구조를 반영하여 자동으로 버전을 관리해주고 이를 저장 및 추출해주는 프레임워크에 대하여 프로토타입 시스템을 구현하였다.

이러한 시스템의 전체 구조는 다음 Figure 4와 같다. 그림에서 보는 바와 같이 전체 시스템은 크게 정의시 모듈집합과 실행시 모듈집합, 그리고 클라이언트 모듈집합으로 구성된다. 먼저 정의시 모듈 집합은 본 연구의 프레임워크하에서 프로세스구조모델과 문서구조모델을 기반으로 프로세스와 문서의 설계를 각각 담당하는 Process designer와 Document designer가 있다. 이들이 설계가 되면 둘 사이에는 구조연계모델을 이용하여 할당에 의한 연관이 맺어

진다. 즉, 적절한 문서의 부분을 프로세스에 할당한다.

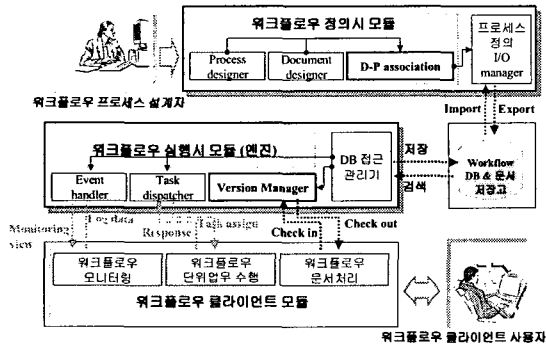


Figure 4. 전체 시스템 구조

실행시 모듈들은 정의시 모듈에서 제공하는 모델을 바탕으로 프로세스를 실행하는데, 먼저 엔진은 프로세스의 진행을 조정, 제어, 통제한다. 그리고, Event handler와 Task Dispatcher는 엔진과의 상호작용을 통해 각종 이벤트 로그를 기록하고 업무를 할당한다. 한편 Version manager는 버전생성모델을 이용하여 check in/out을 처리하여 자동으로 문서가 생성되도록 한다. 이러한 과정에서 문서 버전은 버전저장모델에 의하여 변경분만이 저장된다.

클라이언트 모듈들은 프로세스의 진행에 필요한 사용자 인터페이스를 제공하며 이에 모니터링, 단위 업무 수행, 문서작업이 포함된다. 특히, 모니터링 클라이언트를 통해서 문서의 상태를 확인할 수 있으며, 이는 버전검색모델에 의하여 이루어진다.

5. 결론

본 논문에서는 프로세스 정보를 바탕으로 프로세스가 수행되는 동안 프로세스의 의미를 반영하여 자동으로 버전을 생성하고 이를 바탕으로 문서의 변경과정을 관리하는 워크플로우관리시스템의 새로운 프레임워크를 제시하였다. 이를 위하여 프로세스의 정의시에 문서를 프로세스 수행에 적합한 형태로 구조화하여 이를 프로세스 단위업무와 연계시키는 모델을 제안하였고, 이를 근거로 프로세스의 다양한 구조에 따라 문서의 버전이 자동으로 생성될 수 있음을 보였다. 또한, 각 버전이 효율적으로 저장 및 검색될 수 있도록 하는 모델도 제시하였다.

프로세스정보를 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.

본 연구에 대한 추후 연구과제로는 먼저, 문서 버전을 바탕으로 컨피규레이션[9]을 구성하고 이에 대한 버전관리를 수행하는 것을 들 수 있다. 이러한 컨피규레이션은 사용자가 원하는 조건의 문서 집합을 반환함으로써 효과적인 검색을 가능하게 할 것이다.

참고 문헌

- [1] H. Bae and Y. Kim, "A document-process association model for workflow management", *Computers in Industry*, vol. 47, no. 2, pp 139-154, 2002.
- [2] R. Conradi and B. Westfechtel, "Version Models for Software Configuration Management," *ACM Computing Survey*, vol. 30, no. 2, June 1998.
- [3] 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.
- [4] A. Elmagarmid and W. Du, "Workflow Management: State of the Art Versus State of the Products", *Proceedings of the NATO Advanced Study Institute on Workflow Management Systems (WFMS)*, Istanbul, Turkey, August 12-21, 1997.
- [5] R. H. Güting, R. Zicari and D. M. Choy, "An Algebra for Structured Office Documents," *ACM Transactions on Office Information Systems*, vol. 7, no. 4, pp. 123-156, Apr. 1989.
- [6] D. Hollingsworth, "Workflow management coalition specification: the workflow reference model", WfMC specification, 1994.
- [7] 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.
- [8] 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.
- [9] E. Sciore, "Versioning and Configuration Management in an Object-Oriented Data Model," *VLDB Journal*, vol. 3, pp. 77-106, 1994.
- [10] 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.
- [11] L. Yu and D.J. Rosenkrantz, "A Linear-time Scheme for Version Reconstruction," *ACM Transactions on Programming Languages and Systems*, vol. 16, no. 3, pp 775-797, May 1994.
- [12] A. Zeller and G. Snelting, "Unified Versioning through Feature Logic," *ACM Transactions on Software Engineering and Methodology*, vol. 6, no. 4, pp. 398-441, Oct. 1997.