

협업관리와 형상관리의 통합

김대엽*, 윤 청**

* **충남대학교 컴퓨터공학과

e-mail : kdymn2@cnu.ac.kr

A Integration of Collaborative Work Management and Configuration Management

Dae-Yeob Kim*, Cheong Youn**

* **Dept. of Computer Engineering, Chung-Nam National University

요 약

본 논문은 워크스페이스(workspace) 기반의 협업개발 시스템과 소프트웨어 형상관리(Software Configuration Management, SCM) 시스템의 통합에 관하여 논한다. 협업개발 시스템은 개발 구성원들 간의 유기적인 의사소통 및 산출물에 대한 버전관리 기능 등을 제공한다. 여기에 프로젝트 및 액티비티를 중심으로 소프트웨어 개발 프로세스 전반에 걸친 문서의 체계적인 관리에 중점을 둔 형상관리 시스템을 결합하여 조직 내 협업 개발과 체계적인 형상관리를 수행할 수 있도록 하였다.

1. 서론

컴퓨터 기술의 발전과 인터넷의 확산으로 소프트웨어에 대한 사용자의 기대치가 높아지면서 새로운 소프트웨어 기능에 대한 요구가 끊임없이 이어지고 있다. 이러한 상황에서 소프트웨어의 크기 또한 급속히 확대되고 있으며 소프트웨어 개발 주기도 짧아지고 있다. 이런 변화는 소프트웨어 시스템의 복잡화, 프로젝트 팀의 거대화 및 분산화와 맞물려 소프트웨어 개발 프로세스 컨트롤, 변경관리, 의사소통, 지식 전달 등의 문제를 발생시킨다. 이에 최근에 규모가 큰 소프트웨어를 체계적으로 작성하고 관리하며 이를 바탕으로 새로운 버전의 소프트웨어를 보다 효율적으로 작성하는 것을 지원하는 기술들에 대한 연구가 활발히 진행되고 있다[1][2][3].

협업관리는 팀 단위의 개발환경에서 소프트웨어 협업개발을 원활히 수행할 수 있도록 하기 위해 수행된다. 협업환경에서는 구성원들 간의 자원 공유와 유기적인 의사소통이 매우 중요시된다. 특히 공유 자원(소프트웨어 개발 프로세스에서 발생하는 각종 산출물)에 대한 관리의 개발 효율성과 일관성 유지를 위해 핵심적으로 다뤄져야 하며, 이를 위해 자원에 대한 동시성 제어가 필요하다.

형상관리는 소프트웨어 개발 및 유지보수 과정에서 발생하는 각종 결과물들에 대한 계획, 개발, 운용 등을 종합하여 시스템의 형상(configuration)을 만들고 이에 대한 변경을 체계적으로 관리, 제어하기 위한 활동이다. 프로젝트의 프로세스 관리, 변경 프로세스 관리 기능 등을 통해 궁극적으로 소프트웨어의 품질 향상과 조직의 개발 성숙도를 향상시키기 위한 활동이

라고 볼 수 있다.

본 논문의 목적은 개발의 효율을 높이기 위한 협업관리와 개발 프로세스 전반에 걸친 형상관리의 기능을 통합하여 효과적인 소프트웨어 개발 환경을 제공하는데 있다.

본 논문의 구성은 다음과 같다. 1 장의 서론에 이어 2 장에서는 관련연구로 소프트웨어 협업개발 지원 도구와 소프트웨어 형상관리, 동시성 제어 기법에 대해 논한다. 3 장에서는 본 논문에서 제시하는 통합 시스템의 구조에 대해 설명하며, 4 장에서는 그 구현의 결과로 시스템 환경에 대한 모습을 소개한다. 마지막으로 5 장에서 본 논문의 결론을 맺도록 한다.

2. 관련 연구

2.1 소프트웨어 협업관리 지원도구

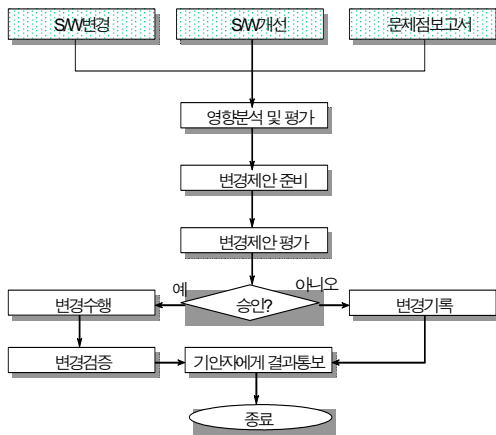
소프트웨어 협업개발 지원도구는 소스코드 관리, 문서 관리, 이슈트래킹, 풀 텍스트 검색, 패치 관리(patch management), 업무 관리(task management), SCM(Software Configuration Management) 도구와의 통합 기능 등과 같은 복합적인 기능들을 지원할 수 있어야 한다[4]. 특히 소스코드의 경우 일반적인 문서들과는 달리 복잡한 디렉토리 체계 내에 놓여 있으며 많은 파일 개수를 갖는 등의 특성을 지니고 있기 때문에 단순히 파일에 대한 버전 관리 차원을 넘어 그 특성에 맞는 별도의 추가 기능을 가져야 한다. 그 중 대표적인 것으로 효과적인 차이점 비교 및 병합 기능, 워크스페이스 관리 기능이 있다[5]. 이 중 워크스페이스 관리 기능은 로컬에서 수행되는 작업의 결과를 서버에 직접 반영할 수 있도록 하는 것이다. 워크스페이스

이스란 여러 개의 소스파일이나 프로젝트 파일을 포함하는 하나의 작업공간이다. 소스코드의 특성 상 파일들은 원래의 디렉토리 구조 그대로 저장되어야 하는데 이를 위해 특정 디렉토리를 소스코드 관리도구에서 관리하는 특수 디렉토리로 지정하고 그곳을 워크스페이스로 지정한다. 워크스페이스 내에서는 최소한의 명령어 수행만으로 그 내부 파일의 변경사항을 서버에 업로드 하거나 다운로드 할 수 있다. 이와 같은 기능들 외에도 조직 내 개발자 상호간의 유기적인 의사소통을 지원하는 기능 또한 필요하다.

2.2 소프트웨어 형상관리

소프트웨어 형상관리는 서론에서 설명한 바와 같이 소프트웨어 개발 및 유지보수 과정에서 산출되는 각종 결과물들에 대해 체계적으로 관리하고 제어하는 활동이다. 유지보수 활동이 고객에게 소프트웨어가 인도(delivered)된 이후의 변경 관리임에 비하여 형상관리 활동은 소프트웨어 라이프사이클 전체, 즉 인도 이전의 변경 관리와 인도 이후의 변경 관리를 모두 포함한다[6].

이 중 특히, 형상 제어는 형상항목에 대한 변경 요청을 분석하여 평가한 후 변경 요청에 대한 승인/반려 여부를 결정하고 이후 발생하는 소프트웨어에 대한 직접적인 변경을 관리하는 형상관리의 핵심 기능이다. 형상 제어에 사용되는 일반적인 변경 절차는 (그림 1)과 같다.



(그림 1) 일반적인 변경절차

형상 제어의 핵심은 변경 요소를 식별하고 그에 대한 요청을 수행하는 것으로부터 시작한다. 변경 요청에 대한 평가가 이루어지고 그에 대한 승인/반려 여부에 따라 대상 형상항목은 변경이 실행되거나 그대로 유지된다. 변경의 실행은 형상관리 저장소로부터 해당 형상항목을 체크아웃(checkout)해야 이루어질 수 있다. 체크아웃이란 형상항목 변경자가 정당한 절

차를 거쳐 저장소로부터 해당 형상항목을 획득하는 과정으로써 변경에 대한 권한을 획득하는 것이라 할 수 있다. 변경이 완료된 형상항목은 새로운 버전으로 새로운 기준선(Baseline)의 구성요소가 된다.

2.3 동시성 제어 기법

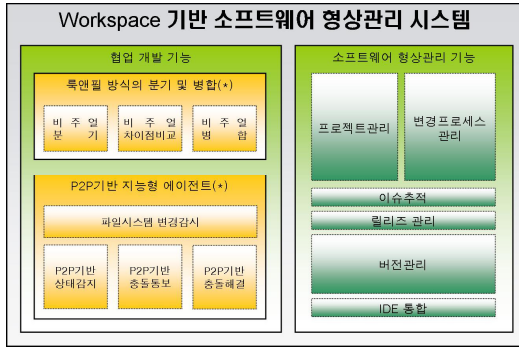
동시성 제어는 다수의 사용자가 동일한 자원에 접근하고자 할 때, 자원의 일관성 유지를 위해 적용하는 기법이다. 동시성 제어기법은 크게 낙관적(optimistic) 기법과 비관적(pessimistic) 기법으로 나뉜다[7]. 일반적으로 동시성 제어기법은 데이터베이스의 트랜잭션 처리 알고리즘에서 많이 다루어지는데, 데이터의 일관성 유지를 위해 비관적 기법은 각 트랜잭션 수행 시점에 검증을 수행하지만 낙관적 기법에서는 각 트랜잭션이 끝나는 시점에서만 검증을 수행한다.

데이터베이스에서 데이터의 일관성 유지를 위해 적용하는 동시성 제어 기법은 협업개발 환경에서의 공유 자원의 충돌 문제를 해결하기 위한 것과 그 목적이 같음을 알 수 있다[8]. 전통적으로 많은 협업개발 지원도구들이 비관적 기법을 채택했는데 이는 지정 체크아웃(reserved checkout)을 이용한 배타적 잠금(exclusive locking) 기법에 기반한다. 배타적 잠금 기능을 이용하는 경우 동일 자원에 대한 동시 작업을 원천적으로 봉쇄함으로써 충돌을 방지한다. 이는 동일 자원에 대한 일관성을 보장하기 위한 확실한 방법이 될 수 있다. 하지만 이 기법은 특정 시점에 단 한 명의 사용자만이 자원을 수정할 수 있기 때문에 불필요한 작업 지연이 초래될 수 있다는 단점이 있다.

이와 반대로 낙관적 기법에서는 비 지정 체크아웃(unreserved checkout)을 이용한다. 이는 여러 사용자들이 자신의 로컬 복사본을 자유롭게 수정한 후, 체크인 하는 시점에서 그 충돌 여부를 판별하여 충돌을 해결(병합)할 수 있게 해주는 것으로 위의 비관적 기법과는 달리 공통 자원에 대한 사용자들의 동시 수정을 가능하게 한다[10]. 따라서 불필요한 작업 지연이 초래되는 경우는 없다. 그러나 경우에 따라 사용자들의 충돌 인식 시점이 늦어지는 경우 충돌 해결에 대한 비용이 증가하고 동시 작업 중 중복된 작업이 진행될 수도 있다는 단점이 있다.

3. 시스템 구조

본 연구에서 제시하고자 하는 소프트웨어 형상관리 시스템은 조직 구성원들 간의 협업을 지원하는 팀 개발 기능과 소프트웨어 형상관리 기능으로 이루어져 있다. 이러한 구조를 통칭하여 워크스페이스 기반 소프트웨어 형상관리 시스템이라 부르며 (그림 2)는 그 모습을 보이고 있다.



(그림 2) 시스템 구성도

3.1 협업 개발 기능

협업 개발 기능은 워크스페이스 기반으로 관리되는 형상항목들에 대해 다수의 사용자가 접근하여 작업할 수 있는 협업개발 환경을 제공한다. 통합 워크스페이스 환경은 형상항목의 대상이 되는 자원에 대하여 구조적 형태와 변경 과정을 직관적으로 인식할 수 있도록 한다. 이러한 워크스페이스를 기반으로 하는 협업 개발 환경에서 P2P 기반 지능형 에이전트는 여러 명의 사용자가 동일 자원에 접근할 경우 발생할 수 있는 충돌문제를 해결하기 위해 설계되었다. 이는 형상항목이 아직 공식화되지 않은 상태에서의 병렬개발 환경을 지원하기 위한 목적으로 개발되었다. 공식화된 형상항목에 대한 동시 접근은 전통적인 비관적(Pessimistic) 기법을 적용하여 제한하지만, 공식화되지 않은 상태의 형상항목에 대해서는 낙관적(Optimistic) 기법을 적용하여 여러 사용자가 접근할 수 있도록 하였다[7][9][10].

다만 동시접근으로 인한 충돌문제를 해결하기 위해 각자 수정한 내용에 대하여 상태감시를 하고 충돌에 대한 사건을 통보하도록 한다. 충돌이 일어난 형상항목에 대해서는 차이점 비교와 병합 기능을 제공함으로써 향후 서버에 업로드 되는 시점에서 통합된 결과를 반영할 수 있게 된다. 일반적인 Optimistic 기법에서는 동일 자원에 대한 충돌의 여부를 서버에 업로드 하는 시점에서 인식할 수 있는 반면 본 연구의 P2P 기반 지능형 에이전트는 충돌 여부를 미리 감지하여 해결하도록 함으로써 병합에 드는 시간적, 인적 자원의 낭비를 줄일 수 있다.

3.2 소프트웨어 형상관리 기능

소프트웨어 형상관리 기능은 프로젝트 관리와 변경 프로세스 관리, 버전 관리 등의 기능을 포함한다. 소프트웨어 품질관리를 조직의 일반적인 관리 프로세스와 통합함으로써 조직 전체가 일관된 프로세스 표준에 맞춰 업무를 수행할 수 있도록 하는데 그 목적이 있다.

소프트웨어 형상관리 기능은 기업 내 문서 관리의 효율성과 체계적인 업무의 흐름을 관리하는데 중점을 두고 있다. 전체적인 소프트웨어 개발 업무의 흐름에 부합하면서 체계적으로 결과물을 관리하기 위하여 프로젝트를 중심으로 구성되어 있다. 프로젝트 관리는 소프트웨어 개발 기법에서 요구하는 프로세스를 중심으로 각 단계별로 이루어지는 활동 및 형상항목의 관리를 포함한다.

프로젝트 중심의 형상관리는 현재 기업의 업무 프로세스, 각 단계별 기능, 그리고 조직 체계의 성능 및 기업 업무의 특성이 반영되어 기업의 조직, 프로젝트 및 프로젝트와 관련된 형상항목이 통합적으로 관리, 운영될 수 있도록 지원한다.

소프트웨어 형상관리 기능에서는 사용자들 사이에 발생할 수 있는 개발 내용 및 개발 과정에서의 의견 불일치와 병렬식 개발에 의한 중복수정 등과 같은 문제점을 해결하기 위해 잠금기능(lock) 및 잠금 해제 기능(unlock)을 제공한다. 이는 전통적인 동시성 제어(Cocurrent access control) 기법 중 비관적(Pessimistic) 접근법에 해당하는 것으로, 지정 체크아웃(reserved checkout)을 이용한 배타적 잠금(exclusive locking) 기법에 기반한다. 따라서 이미 공식화된 형상항목에 대해서는 변경 허가를 받은 후 직접 파일서버에서 Check-out 을 한다. 특정 사용자에 의해 Check-out 된 형상항목에 대해서는 잠금 기능을 제공하여 다른 사용자에 대한 수정과 변경을 불허한다. 그리고 사용자가 Check-out 한 형상항목을 수정하여 다시 저장서버에 Check-in 하는 과정에서는 형상항목에 대한 잠금 해제 기능을 실행하여 다른 사용자들이 필요에 따라 수정을 할 수 있도록 한다.

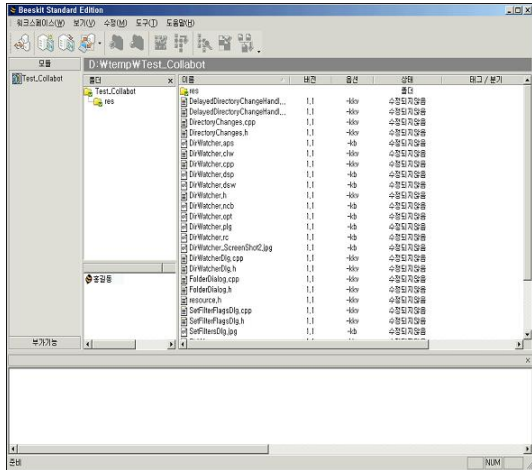
결론적으로 협업 개발 기능에서는 낙관적 기법을 적용하여 아직 공식화되지 않은 형상항목에 대한 병렬 개발을 지원하도록 하였으며, 소프트웨어 형상관리 기능에서는 비관적 기법을 적용하여 공식화된 형상항목에 대해 보다 엄격한 형상관리를 수행할 수 있도록 하였다.(그림 1)의 변경절차는 그러한 목적을 위해 수행되는 과정이라고 볼 수 있다. 다시 말해 협업 개발 기능의 목적은 공유 자원에 대한 소유권을 획득하기 위해 필요한 요청 및 응답 시간을 없애으로써 업무의 효율성을 높이고자 하는데 있으며, 소프트웨어 형상관리 기능의 목적은 프로젝트를 중심으로 하여 전체적인 업무 프로세스를 관리함과 동시에 각 단계에 해당하는 형상항목들을 보다 체계적으로 관리하고자 하는데 있다.

4. 시스템 환경

3 절의 시스템 구성도에서 나타나듯이 본 연구에서 개발한 시스템은 팀 개발 기능과 소프트웨어 형상관리 기능을 결합한 구조이다. 즉, 다수의 사용자가 동시에 자원에 접근할 수 있는 협업개발 환경과 프로젝트 및 프로세스, 변경 관리 등을 체계적으로 수행할

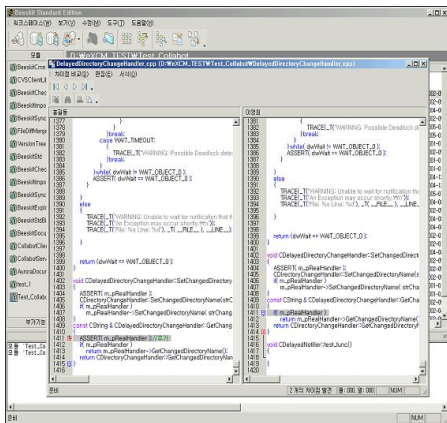
수 있는 소프트웨어 형상관리 기술을 통합하여 보다 효율적이고 신뢰성 있는 소프트웨어 개발을 가능하도록 구현하였다. 본 절에서는 시스템의 구현 결과를 통해 앞서 제시한 기능들을 설명하고자 한다.

(그림 3)은 협업 개발 기능에서 윈도우 탐색기 스타일의 통합 워크스페이스 환경을 보이고 있다. 사용자들은 이 공간을 통해 모든 변경작업을 수행하며, 다른 사용자와의 유기적인 의사소통을 통해 중복 작업에 대한 차이점 비교 및 병합 등의 기능을 수행할 수 있다.



(그림 3) 윈도우 탐색기 스타일의 통합 워크스페이스

(그림 4)는 동일 자원에 대한 다른 사용자의 변경내용에 대한 차이점 비교 화면이며, 사용자는 차이점 비교 후 자원의 통일성을 위한 병합 기능을 수행할 수 있다.



(그림 4) 공유 사용자와의 차이점 비교

소프트웨어 형상관리 기능은 사용자의 접근 편의성을 제공하기 위해 웹 어플리케이션으로 구현되었다. 프로젝트 관리, 변경 프로세스 관리를 중심으로 형상 항목에 대한 처리를 체계적으로 수행하기 위한 절차를 제공한다. 또한 협업 개발 기능의 일부를 웹 어플리케이션에 결합하여 사용자가 보다 편하게 자원에 접근할 수 있도록 하였다.

5. 결론

본 연구의 형상관리 시스템은 공간의 제약을 극복하고 효율적인 협업개발을 가능하도록 하는 워크스페이스 기반 협업 개발 기능과, 지식정보 인프라를 구축하고 있는 인터넷을 통한 웹 기반 소프트웨어 형상관리 기능의 통합을 제시하였다.

즉, 협업 개발 기능은 병렬 개발의 기회를 제공함과 동시에 그에 따른 문제를 적절히 해결할 수 있도록 지원하며, 소프트웨어 형상관리 기능은 협업 개발 기능을 바탕으로 소프트웨어 개발 전반에 걸친 프로세스의 체계적 관리를 가능하게 할 수 있다.

본 연구의 형상관리 시스템을 통해 소프트웨어 개발 조직의 업무 효율성 증가와 개발 성숙도 및 소프트웨어 품질의 향상을 기대한다.

참고문헌

- [1] Jacky Estublier, "Software Configuration Management: A Roadmap", ICSE-Future of SE Track, 2000
- [2] Reidar Conradi, "SPI frameworks: TQM, CMM, SPICE, ISO 9001, QIP Experience and trends", Norwegian SPIQ project
- [3] Magnus Larsson, "Applying Configuration Management Techniques to Component-Based Systems", MRTG Report 00/24, 2000
- [4] Melissa Webster, "An End-User View of the Collaborative Software Development Market", IDC, Dec. 2003
- [5] Ovum, "Ovum Evaluates: Configuration Management", Ovum, 2003.
- [6] 윤청, "소프트웨어 공학", 생능출판사, 1999
- [7] A. Makni, R. Bouaziz, F. Gargouri, "Formal Verification of an Optimistic Concurrency Control Algorithm using SPIN", Proceedings of the 13th International Symposium on Temporal Representation and Reasoning, pp.160-167, June. 2006
- [8] R. Strom, G. Banavar, K. Miller, A. Prakash, M. Ward, "Concurrency control and view notification algorithms for collaborative replicated objects", IEEE Transactions on computers, Vol.47, No.4, pp.458-471, April. 1998
- [9] 유재홍, 성미영, "공동저작 시스템에서의 동시성 제어와 쓰기권한 제어", 한국정보처리학회논문지, 제 7 권, 제 2 호, pp.347-354, 2000
- [10] Haifeng Shen, Suiping Zhou, Chengzheng Sun, "Flexible concurrency control for collaborative office systems", 3rd International Conference on Information Technology and Applications, Vol.2, pp.45-50, July. 2005