

논문 2004-41SP-6-16

멀티미디어 소프트웨어 개발환경을 위한 소프트웨어형상관리시스템

(The software configuration management system for Multimedia software development environment)

이 정 현*, 채 옥 삼**

(JeongHeon Lee and OkSam Chae)

요 약

일반적인 소프트웨어형상관리 시스템들은 멀티미디어 소프트웨어 개발 환경의 특성 및 특징들을 수용하기에 부족하다. 멀티미디어 소프트웨어 개발 환경은 기본적인 개발 도구 외에 컴포넌트(나 라이브러리)들을 인터랙티브하게 조합하고 테스트 할 수 있는 비주얼 프로그래밍 환경을 통한 시뮬레이션과 분석을 반복하면서 개발하는 특징을 가지고 있다. 뿐만 아니라, 시뮬레이션 과정 중에는 유사한 기능을 가진 컴포넌트 중에 적합한 컴포넌트를 빠르고 효과적으로 찾기 위한 방법이 필요하다. 본 논문에서는 이러한 멀티미디어 소프트웨어 개발 환경에서 사용하는 시뮬레이션 도구를 체계적으로 지원해줄 수 있는 비주얼 워크스페이스를 기반으로 한 소프트웨어형상관리 방법과 이러한 방법을 수용할 수 있는 소프트웨어형상관리 시스템을 개발하고자 한다. 또한, 개발된 시스템을 실제 시뮬레이션 도구와 연계하여 소프트웨어형상관리 시스템으로써의 적합성을 확인하고자 한다.

Abstract

The general software configuration management system shows lack of the properties and features of software development environment for multimedia due to its wide range to be covered. Multimedia software development environment has properties like repetitive analysis and simulation using visual programming environment where, beside support of elementary development functions, component or library can be combined and tested interactively. Moreover, the method to look fast and effective for component having similar function is required. In this paper, we present the system which supports the software configuration management method for a simulation tool and the system for the properties in the multimedia software development environment. And we relate our system to real simulation tool so as to check its ability as the software configuration management system for multimedia software development environment.

Keywords : SCM(Software Configuration Management) System, Multimedia Software Development Environment, Visual workspace based SCM, Integrated Development Environment

I. 서 론

컴퓨터 기술의 발전과 함께 소프트웨어에 대한 컴퓨터 사용자의 기대치가 커지면서 새로운 소프트웨어 기능에 대한 요구가 끊임없이 이어지고 있다. 이러한 상

황에서 소프트웨어 크기는 급속히 확대되고 있으며 소프트웨어 개발 주기도 짧아지고 있다. 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].

소프트웨어형상관리(SCM : Software Configuration Management, 이후 'SCM'으로 칭함.) 기술은 이 들 중 의 하나로서 소프트웨어의 전체 생명주기를 통해 형상

* 정희원, ** 정희원(교신저자),
경희대학교 컴퓨터공학과
(Dept. of Computer Eng., KyungHee University)
접수일자: 2004년8월6일, 수정완료일: 2004년8월24일

을 인식하여 체계적으로 변경 사항을 관리하고, 소프트웨어의 추적 용이성과 통일성을 유지하기 위한 활동을 가리킨다. 이러한 SCM 활동들을 시스템적으로 지원해 주기 위해 개발된 것이 바로 SCM시스템이다^[4,5,6,7, 11,12].

SCM 시스템은 초기에는 규모가 큰 일반 업무용 소프트웨어 개발 분야를 지원하기 위해서 주로 연구되었으나, 최근 들어 디지털 캠코더, 디지털 카메라, 핸드폰, MP3P (MP3 Player)등과 같은 멀티미디어 장치가 널리 보급되고 이들 장치를 지원하는 멀티미디어 소프트웨어의 규모가 커지고 개발주기가 짧아지면서 이를 위한 SCM 시스템의 필요성이 대두되고 있다^[8,9,20,23]. 그러나 기존의 SCM 시스템들은 업무절차(Business Process)용 소프트웨어 개발 도구에 적합하게 개발되어 있어서, 영상처리 및 신호처리를 포함하는 멀티미디어 소프트웨어 개발 도구에는 적합하지 않다^[13,14,15].

멀티미디어 소프트웨어 개발 환경과 업무절차용 소프트웨어 개발 환경 사이에는 차이가 존재한다. 업무용 소프트웨어 개발에서는 소프트웨어를 구성하는 컴포넌트들의 기능을 정확하게 정의하기만 하면, 소프트웨어 개발에 필요한 컴포넌트 기능을 컴포넌트 저장소(Component Repository)에서 정확하고 빠르게 찾을 수 있다. 그러나 멀티미디어 소프트웨어 개발에서는 동일한 기능을 수행하는 컴포넌트라도 적용조건에 따라서 다른 성능을 나타내는 여러 종류의 컴포넌트가 존재하기 때문에 최적의 컴포넌트들을 찾아서 사용하는 것이 쉽지 않다^[16,17,18]. 이러한 멀티미디어 분야의 특성으로 인하여 멀티미디어 응용 소프트웨어 개발과정에서는 실제 데이터를 이용한 반복적인 시뮬레이션과 분석을 통해서 컴포넌트를 선택하여 사용하게 되며, 대부분 개발자들은 이러한 과정을 거쳐 응용 소프트웨어를 개발하게 된다. 그러므로 기존의 멀티미디어 데이터 프로세싱 알고리즘이나 응용 소프트웨어 개발환경들은 다양한 컴포넌트를 체계적으로 관리하고 주어진 문제 해결을 위해서 컴포넌트들을 인터랙티브(interactive)하게 조합하고 테스트 할 수 있는 시뮬레이션 환경을 기반으로 하고 있다^[20,21,22,23,24,25]. 이러한 개발환경들은 멀티미디어 데이터 프로세싱을 위한 기본적인 컴포넌트들과 컴포넌트를 이용하여 작성된 고기능의 컴포넌트를 포함하고 있으며, 이들 컴포넌트를 결합하여 새로운 기능을 수행하는 컴포넌트나 응용프로그램을 작성할 수 있는 비주얼 프로그래밍 환경을 갖추고 있다.

기존의 SCM 시스템들^[26,27,28,29]은 이러한 멀티미디어 개발 도구의 특성 및 개발 산출물의 특성을 효과적으로

지원해주고 있지 못하고 있으며, 이러한 멀티미디어 개발 환경을 지원하기 위한 SCM 시스템이 되기 위해서는 비주얼 워크스페이스(비주얼 프로그래밍 환경에서 발생하는 워크스페이스) 기반의 SCM 기능과 멀티미디어 분야의 특성인 개발과정 중에 발생하는 중간산출물이 다른 적용환경에서는 적합한 결과물로 사용할 수 있다는 점을 고려하여야 하고, 적합한 솔루션을 찾기 위하여 시뮬레이션 시간을 줄여줄 수 있는 기능이 지원되어야 한다.

본 논문에서는 이러한 멀티미디어 개발환경의 특성을 고려하여 비주얼 워크스페이스 기반의 하이퍼링크 검색이 가능한 SCM 시스템을 개발하고, 실제 사용 중인 시뮬레이션 개발 도구들과의 인터페이스를 통하여 멀티미디어 개발환경에 적합성을 확인하였다.

II. 멀티미디어 소프트웨어 개발 환경에 적합한 SCM 시스템의 설계

서론에서 언급한 멀티미디어 소프트웨어 개발 환경에 대하여 소프트웨어형상관리 측면에 관련된 특징들을 정리하면 다음과 같다.

- ① 멀티미디어 소프트웨어 개발 시에는 기본적인 컴파일러 외에 기존 컴포넌트를 빠르게 조합하고 테스트 할 수 있는 시뮬레이션 환경과 멀티미디어 자료 분석을 위한 분석시스템이 사용된다^[20,21,22,23,24,25].
- ② 멀티미디어 컴포넌트는 업무절차용 컴포넌트와는 달리 동일한 기능을 하지만 동작환경에 따라 성능이 달라지는 컴포넌트들이 계속적으로 추가 개발되어 단순한 키워드 검색기능 만으로는 적합한 컴포넌트를 찾는 것이 어렵다^[16,17,18].
- ③ 업무절차용 알고리즘 개발 시에는 입력과 출력 값이 처음 설계되었을 때와 일치해야만 개발이 종료되지만 멀티미디어 알고리즘 개발 시에는 입력 값에 대한 출력 값이 처음 설계되었을 때의 출력 값에 근사하게 되는 경우 개발이 종료되는 특징이 있어, 개발과정 중에 발생하는 알고리즘이 다른 환경에서는 최적의 알고리즘일 경우가 많다^[16,17,18].
- ④ 업무절차용 컴포넌트와는 달리 멀티미디어 컴포넌트는 사용되는 위치에 따른 성능 달라지고 용도가 추가적으로 발생할 수 있어 이러한 정보를 자동적으로 추가하고 사용자에게 알려줄 수 있는 방법이 필요하다^[21,22,24,25].

전술한 멀티미디어 소프트웨어 개발환경의 특징들을 지원하기 위한 SCM 시스템을 개발하기 위한 고려사항은 다음과 같다.

- ① 업무 절차용 소프트웨어 개발 환경을 위한 텍스트 소스 코드 기반의 SCM 기능을 포함하는 시뮬레이션 환경의 작업영역인 비주얼 워크스페이스를 기반으로 한 SCM 기능을 지원하여야 한다. 즉, 일반적인 소스코드 파일 중심의 SCM 기능인 버전관리, 리빌드관리, 프로젝트관리등을 기본적으로 지원해 줄 뿐만 아니라 비주얼 워크스페이스 중심으로도 같은 기능이 지원되어야 한다.
- ② 비주얼 워크스페이스를 위한 비주얼 디스플레이 기능이 지원되어야 한다. 이러한 비주얼 디스플레이 기능의 지원은 디스플레이 기능을 중심으로 한 정보의 제공 및 컴포넌트, 응용 소프트웨어에 대한 관리 및 이해력을 증가시킬 수 있어 효과적인 재사용을 유도할 수 있어야 한다.
- ③ 개발과정 중에 발생하는 중간산출물을 간단히 최종 솔루션으로 분기할 수 있어야 한다. 멀티미디어 분야의 특성상 동작환경에 따라 중간산출물로 나온 결과물들이 다른 환경에서는 최적의 솔루션으로 적용되는 경우가 다수 발생하므로 이러한 경우의 지원을 위해서 중간산출물들이 최종 솔루션으로 바로 이용할 수 있어야 한다.
- ④ 멀티미디어 소프트웨어 개발 시에는 최적의 솔루션을 얻기 위하여 실제 데이터를 이용한 반복적인 시뮬레이션과 분석과정을 통하게 되고 이를 통하여 적합한 컴포넌트를 선택 사용하거나 적합한 기능을 개발하게 된다. 이러한 반복되는 개발과정의 시간을 줄여주기 위하여 관련된 정보들을 빠르게 검색하고 이동할 수 있는 기능이 필요하다.

1. 워크스페이스 기반의 형상 데이터 저장 구조

본 논문에서는 전술한 4가지 사항들을 고려한 그림 1 같은 구조의 형상관리를 위한 파일 관리 구조를 설계하였다. 그림 1의 구조는 제안 시스템이 워크스페이스를 기반으로 형상관리가 이루어지도록 하였으며, 기본적인 소스코드의 형상관리를 위한 기능도 만족하도록 설계되었다.

그림 1의 제일 상단에는 비주얼 워크스페이스를 기준으로 비주얼 워크스페이스의 히스토리 정보와 히스토리 정보에 보관되어 있는 각 비주얼 워크스페이스의 내재된 컴포넌트 세부정보들이 기록되도록 설계하였으며,

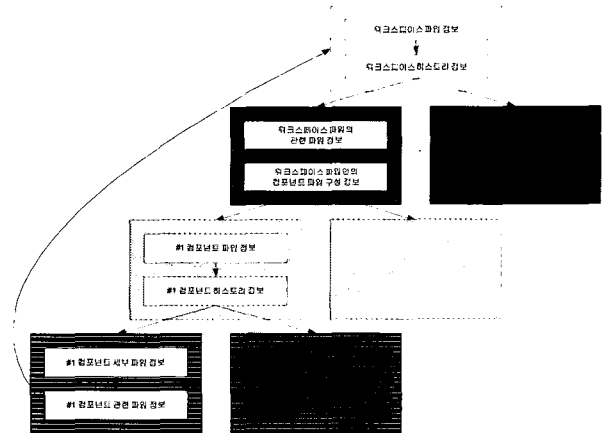


그림 1. 워크스페이스 기반의 형상 데이터 저장 구조
Fig. 1. Visual workspace based configuration data repository structure.

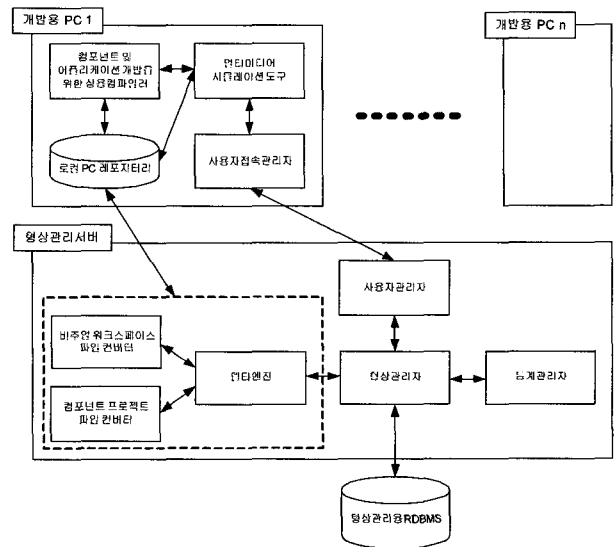


그림 2. 제안 시스템의 전체 구성
Fig. 2. The composition overview of the proposed system.

일반적인 SCM 시스템들이 가지는 컴포넌트 각각의 세부적인 소스코드 파일의 히스토리 정보를 유지하도록 구성하였다. 제안 시스템은 그림 1과 같은 관리 구조를 기반으로 그림 2와 같은 전체적인 구성을 이루도록 하였다.

2. 제안 시스템 전체 구조

제안 시스템은 형상데이터의 저장 및 복원을 위한 형상관리서버와 멀티미디어 개발 환경 지원을 위한 클라이언트 부분으로 구성되어 있다. 그림 2의 하단에 위치하고 있는 형상관리서버 부분은 개발 및 유지보수 과정에서 생기는 형상을 체계적으로 저장하거나 관리하고, Check In/Out 과정을 통하여 사용자에게 형상 데이터

의 분배 및 저장을 담당하며, 개발 및 유지보수 과정에 참여한 사용자들을 관리하기 위한 기능들로 구성된다. 상단에 위치하고 있는 클라이언트 부분은 개발자들이 자신의 PC에서 개발과정 중에 생성되는 모든 형상 데이터를 서버로 다운로드나 업로드 하는 기능을 수행하며, 시뮬레이션 도구의 인터페이스를 제공한다.

그림 2의 하단에 있는 비주얼 워크스페이스 파일 컨버터는 시뮬레이션 도구에서 만들어진 비주얼 워크스페이스 파일을 형상관리서버로 임포트(importing)하는 역할을 수행하며, 소프트웨어 개발에 사용하는 상용 컴파일러의 메이크파일(Makefile)이나 프로젝트 파일(Project File)을 임포트하는 것은 프로젝트 파일 컨버터가 담당한다. 형상관리서버에 데이터를 효과적으로 저장하기 위한 압축 엔진인 델타엔진(delta engine)과, 형상관리 서버에 접속할 수 있는 사용자들의 리스트 관리 및 접근 권한 관리는 사용자 관리자가 담당한다. 형상관리자는 개발용 PC로부터 Check In되는 모든 파일 및 이벤트 정보를 분석하여 하이퍼링크(hyper link)정보를 생성하며, 사용자 및 날짜 정보와 같이 형상관리 RDBMS에 저장, 복원, 검색하는 역할을 수행한다. 그리고, 형상관리서버에 누적된 형상관리 정보에 대한 통계 데이터 생성, 분석 및 관리를 위한 통계관리자와 형상관리 데이터들을 저장하기 위한 형상관리 DBMS가 있다.

그림 2의 상단에 있는 시뮬레이션 도구는 멀티미디어 소프트웨어를 개발할 때 사용하는 도구로 보통 컴포넌트 혹은 라이브러리를 조합하여 시뮬레이션 할 수 있는 비주얼 프로그래밍 환경을 가지고 있으며, 비주얼 프로그램으로 구성된 비주얼 워크스페이스 파일을 생성한다. 프로그래머가 개발용 PC에 형상관리 서버로부터 Check In/Out 절차를 통하여 다운로드 받은 관련 파일들을 임시로 저장하여 시뮬레이션 도구가 사용할 수 있도록 해주는 저장 장소인 로컬 PC 레포지터리(repository)가 있으며, 사용자 접속 관리자는 형상관리 서버로부터 데이터 파일을 다운로드/업로드 하기 위해서 형상관리 서버로 접속할 수 있는 인증을 받기 위한 관리자 기능 및 클라이언트 PC에서 접속하는 사용자들의 정보를 관리하는 역할을 수행한다.

3. 워크스페이스 기반의 하이퍼링크생성 구조

멀티미디어 개발 환경과 같이 시뮬레이션 도구를 사용하는 개발에서는 소스코드 외에 시뮬레이션의 비주얼 워크스페이스 또한, 체계적인 관리가 필요한 중요한 형상 관리의 대상이 된다. 뿐만 아니라 고려사항에서 언

급한 바와 같이 개발 단계상에 발생하는 시뮬레이션과 분석중에 발생하는 개발기간을 단축시켜주기 위하여 컴포넌트를 선택하면 바로 이동하여 관련 정보를 확인할 수 있고 관련 정보들을 통하여 다른 정보로 빠르게 이동할 수 있도록 하기 위하여 컴포넌트에 하이퍼링크(hyperlink) 정보를 지원하도록 하였다.

이러한 하이퍼링크정보의 생성을 위해서 시뮬레이션 도구의 비주얼 워크스페이스의 파일에서 사용하고 있는 컴포넌트(혹은 라이브러리)의 위치 정보를 그림 2의 검색해 형상관리자로부터 얻은 후 하이퍼링크 정보를 구성한 후 그림 1과 같은 형상데이터 저장 구조에 저장하도록 하였다. 그림 3은 그림 2의 형상관리자 안의 하이퍼링크 정보 생성 모듈을 보여주고 있다. 하이퍼링크생성 모듈은 비주얼 워크스페이스 파일을 분석하여 형상관리서버에 등록되어 있는 컴포넌트(혹은 라이브러리)를 검색한 후 비주얼 워크스페이스에 사용된 컴포넌트들의 위치 정보를 기반으로 링크정보 생성기를 통해 하이퍼 링크 정보를 생성, 저장하게 된다. 그리고 이 기능의 편리한 사용을 위하여 그림 4와 같이 제안된 SCM 시스템의 클라이언트 부분은 시뮬레이션 도구의 사용자 인터페이스와 유사한 사용자 인터페이스를 가지도록 설계하였으며, 우측 상단에는 등록된 비주얼 워크스페이스 파일의 내용이 시뮬레이션 도구에서처럼 보일 수 있도록 워크스페이스 디스플레이영역을 넣었다.

그림 4를 보면 좌측 리스트 영역은 SCM 시스템에 등록된 컴포넌트(혹은 라이브러리) 리스트와 비주얼 워크스페이스 리스트를 보여주는 영역으로 컴포넌트 리스트는 유사기능의 그룹으로 분리되어 트리 구조로 표현

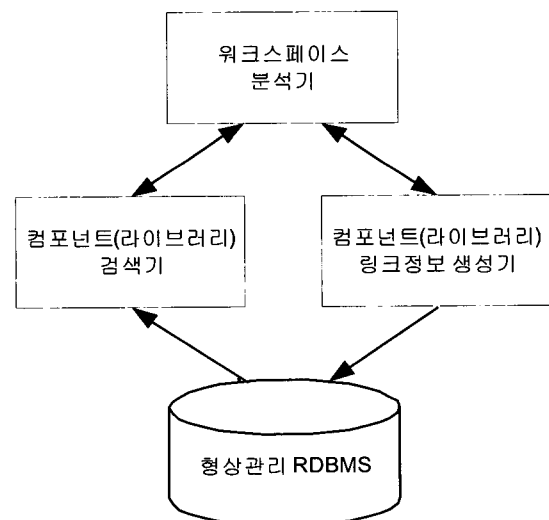


그림 3. 하이퍼링크 정보 생성 모듈
Fig. 3. Hyperlink information generator module.

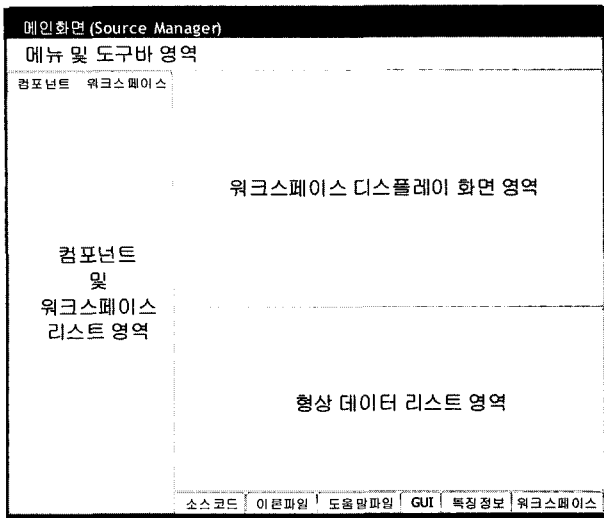
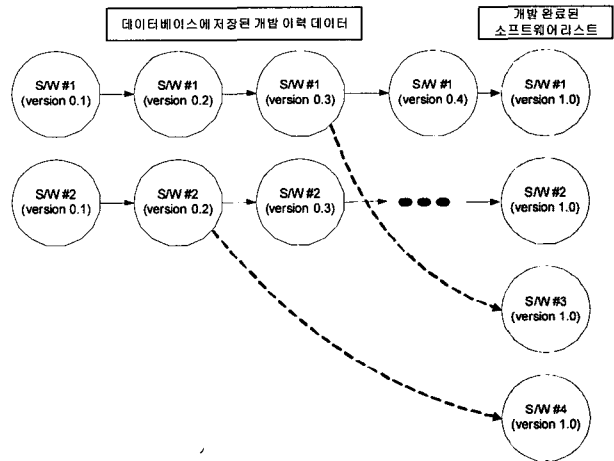


그림 4. 클라이언트 프로그램의 사용자인터페이스 구조
Fig. 4. Design of the User Interface for Client Program.

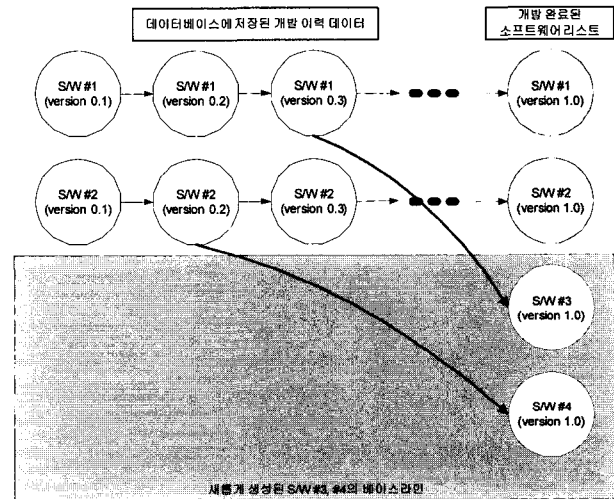
되며, 등록된 전체 컴포넌트 보기 기능을 통하여 형상 관리서버에 등록된 전체 컴포넌트 리스트를 한눈에 확인할 수 있도록 구성하였다. 컴포넌트를 선택하게 되면, 우측상단에는 선택된 컴포넌트 응용을 대표하는 비주얼 워크스페이스 내용이 디스플레이 되며, 우측하단에는 컴포넌트의 세부적인 형상관리 정보가 나타난다. 비주얼 워크스페이스 파일리스트를 선택한 경우도 동일한 형태의 내용이 나타난다. 우측 하단에는 선택된 컴포넌트와 관련된 세부 정보들로 구성되며, 워크스페이스 관련정보리스트에는 최초에는 하나의 비주얼 워크스페이스 파일만 연결되게 되지만, 현재 형상관리 서버에 등록되어 있는 비주얼 워크스페이스 파일들을 검색하여 선택된 컴포넌트를 이용하는 비주얼 워크스페이스 파일들을 리스트에 추가할 수 있도록 하였다. 이 기능과 하이퍼링크 기능을 통하여 선택된 컴포넌트와 관련된 다양한 예제 및 용도를 쉽게 알 수 있으며, 소프트웨어의 재사용성 및 활용성을 증가시키는 효과를 얻을 수 있다.

4. 중간산출물의 분기구조

전술한 바와 같이 멀티미디어 소프트웨어들은 개발 도중에 생기는 중간산출물들이 다른 환경에서는 최종결과물로 사용되는 경우들이 많다^{16, 17, 18}. 이러한 특징 때문에 제안된 SCM 시스템은 개발과정 중에 발생하는 중간산출물을 분기하여 새로운 소프트웨어인 최종출루션으로 등록할 수 있는 구조를 갖도록 하였다. 뿐만 아니라 분기된 소프트웨어가 개발 이력(history)을 가질 수 있도록 기존 개발이력 정보의 위치 정보를 연결정보



a. 분기 전 (before separation)



b. 분기 후(after separation)

그림 5. 중간산출물이 새로운 최종산출물로 등록된 구조

Fig. 5. The structure which is register with the last product where the intermediate product is new.

로 저장하도록 하였다. 기존의 SCM 시스템들은 새로운 베이스라인(baseline)이나 브랜치(branch)로 관리하는 기능들을 지원하고 있지만 이러한 기능들은 개발의 종료기점이 아닌 개발시작기점(베이스라인 : baseline)으로써 관리하기 때문에 이 같은 경우에는 적용할 수 없다.

개발과정 중에 발생하는 중간산출물을 선택적으로 분기하기 위해선 개발에 사용되고 있는 소스코드 파일 및 워크스페이스 파일의 각 리비전(revision)을 선택적으로 분기시킬 수 있어야 할 뿐만 아니라 특정 리비전에 해당되는 관련 파일(매뉴얼, 이론문서, GUI 정보, 비주얼 워크스페이스, 등...)을 동시에 분기해야 한다. 만약, 같은 리비전에 속하는 파일들이 존재하지 않는다면, 한 단계 낮은 리비전을 가진 파일들을 분기하여 분기된

관련파일들과 같이 새로운 정보로 등록된다. 그림 5는 중간산출물의 최종솔루션으로 분기되어 저장된 과정을 보여준다.

III. 구현 및 결과

제안된 시스템의 구현 환경은 다음과 같다.

- 서버 사양
 - : Windows 2000 Server / MS-SQL / TCP/IP
- 클라이언트 사양
 - : Windows 2000 이상 / Pentium III 이상
- 개발언어
 - : MS Visual C++ (MFC) / ASP
- 멀티미디어 시뮬레이션 환경
 - : Hello-Vision^[22] / Khoros^[20]

아래의 그림 6은 개발된 시스템의 클라이언트 메인 화면으로 좌측에는 영상처리 컴포넌트들의 트리 구조가 나와 있으며, 유사한 기능의 컴포넌트들이 묶여서 여러 개의 그룹으로 되어 있다. 좌측의 컴포넌트 트리에서 그룹을 선택하면 그룹에 속해있는 컴포넌트들이 나타나고, 그 컴포넌트 중 하나를 선택하게 되면, 우측에 컴포넌트와 관련된 정보들이 나타나게 된다. 우측상단에는 컴포넌트를 사용하는 대표적인 비주얼 워크스페이스의 내용이 디스플레이 되며, 우측하단에는 선택된 컴포넌트의 소스 파일을 포함한 관련정보들이 나타난다. 컴포넌트와 관련된 각 정보들은 우측하단의 위에 있는 탭바를 통해서 볼 수 있으며, 우측 상단의 비주얼 워크스페이스 디스플레이 화면에 표시된 비주얼 워크스페이스 내용에서 다른 컴포넌트를 더블 클릭하게 되면 하이퍼링크 정보에 따라 선택된 컴포넌트 정보로 이동하게 된다. 그림 7은 워크스페이

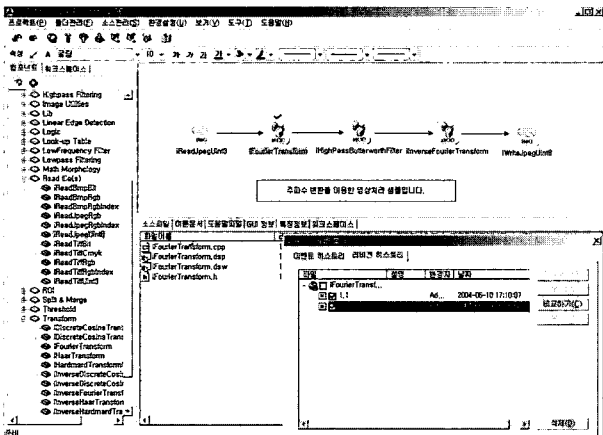


그림 6. 제안된 시스템의 메인 화면
Fig. 6. Main screen of the proposed system.

스의 하이퍼링크 정보를 사용하여 관련 정보를 검색하는 예제 화면을 보여주고 있다. 그림 6의 우측하단의 대화상자는 선택된 소스코드 파일의 개발 이력을 볼 수 있는 리비전 정보를 보여준다.

그림 8은 선택한 워크스페이스의 과거에 개발된 중간산출물중 특정 리비전에 해당되는 워크스페이스와 관련된 파일을 분기하는 예제를 보여주고 있다.

ImageEnhancementFilter.wks 파일의 이력 정보 중 0.7 버전에 해당되는 파일들을 새로운 이름인 ImageIdealEnhancementFilter.wks 파일로 분기 등록하는 과정을 보여주고 있다. 분기가 끝난 파일의 이력 정보는 분기이전의 컴포넌트의 이력 정보와 연결된다. 그림 9의 a는 워크스페이스 파일의 리비전이 다른 두 리비전 파일 사이의 차이점을 보여주는 기능이다. 0.2 버전과 0.7버전의 워크스페이스 파일에 달라진 부분을 보여주고 있으며, 달라진 부분은 아이콘의 위에 노란색의 바가 그려지게 되고, 같은 부분의 아이콘에는 아무 표시도 안된다. 빨간색 바가 그려진 아이콘 부분은 차이가 있는 아이콘 중에서 현재 마우스로 선택된 아이콘을 표시한다. 그림 9의 b는 워크스페이스 하부 구조에 있는 소스코드 파일들간의 비교 기능을 보여준다. 그림 9의 a와 마찬가지로 노란 부분은 다른 부분이고 빨간 부분은 다른 부분 중 마우스로 현재 선택한 부분을 의미한다.

제안된 시스템과 같은 목적으로 개발된 시스템이 없어 정확한 비교가 어렵지만 기존에 알려져 있는 시스템들의 기능을 조사, 비교함으로써 제안 시스템의 특징을 제시하였다.

제안 시스템은 표 1에서 보는 바와 같이 기존 시스템과 같은 텍스트 소스 코드 기반의 SCM 체계 및 기

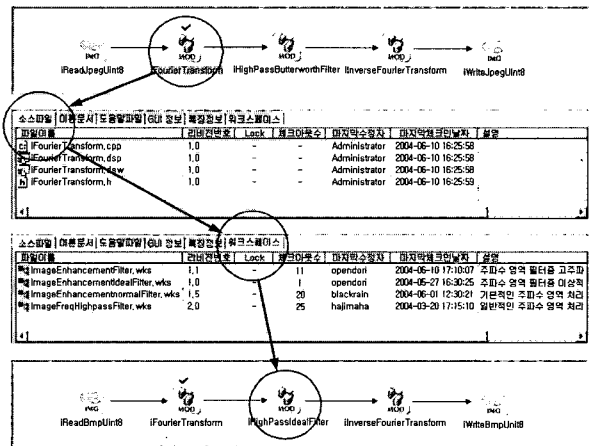


그림 7. 하이퍼링크 예제
Fig. 7. The example of the hyperlink.

표 1. 기존 시스템과의 특징 비교

Table 1. Feature comparison of existing system and proposed system.

- ① 제안된 시스템(EDCM) ② Clearcase^[26]
- ③ CCC/Harvest^[27] ④ PVCS^[28]
- ⑤ SourceSafe^[29]

항목	시스템				
	①	②	③	④	⑤
프로젝트관리	○	○	○	○	○
텍스트 소스코드 기반 SCM 체계	○	○	○	○	○
버전관리 및 변경관리	○	○	○	○	△
프로세스 관리	○	○	○	○	○
비주얼 워크스페이스 기반 SCM 체계	○	×	×	×	×
비주얼 워크스페이스 가시화	○	×	×	×	×
컴포넌트의 하이퍼링크 검색	○	×	×	×	×
컴포넌트의 하이퍼링크 자동 생성	○	×	×	×	×
중간산출물의 최종결과물로의 분기	○	△	△	△	△

능을 동일하게 지원하고 있으며, 키워드 입력을 통한 검색기능도 동일하게 지원한다. 그러나, 멀티미디어 소프트웨어 개발 시에는 업무절차용 소프트웨어 개발 시와는 달리 동일한 기능으로 개발되었지만 동작환경에 따라 성능이 달라지는 컴포넌트들이 계속적으로 추가 개발되어 등록되는 특징을 가지고 있어 기존 SCM 시스템들처럼 키워드 검색 기능을 통한 컴포넌트 이름만으로는 소프트웨어 개발에 적합한 컴포넌트를 찾기 어렵다. 뿐만 아니라 같은 컴포넌트라도 컴포넌트가 놓이는 위치에 따라서 다른 용도로 사용되는 경우도 많기 때문에 사용자들에게 정확한 사용을 유도해주기 위한 비주얼 워크스페이스와 같은 예시 제시방법이 필요하다. 그러나, 기존 시스템들은 비주얼 워크스페이스의 비주얼 디스플레이가 불가능하여 기존 시스템을 이용하여 형상관리를 하는 경우 선택한 컴포넌트를 사용하기 위해서 필요한 컴포넌트가 어떤 것들이 있는지 매뉴얼이나 도움말을 모두 읽어보아야 가능하고, 해당 컴포넌트를 다운로드 받아보아야만 선택한 컴포넌트를 사용하기 위해 필요한 컴포넌트들이 무엇인지 알 수 있어 선택한 컴포넌트가 적합한 것인지, 사용에 필요한 다른 컴포넌트들이 있는지 확인하는데 기본적인 시간이 소요된다. 만일 기존 SCM 시스템을 이용하여 제안된 시스템과 같은 정보를 직접 입력하여 등록한다고 해도 제안된 시스템과 같이 하이퍼링크가 자동으로 생성되지 않아 추가로 등록되는 컴포넌트들과의 관계는 설정될 수가 없다. 즉, 제안된 시스템은 비주얼 워크스페이스 등록 시마다 컴포넌트 사이의 관계가 재설정됨으로써 기존 시스템에서 찾지 못했던 관계 컴포넌트를 찾는 것이 가

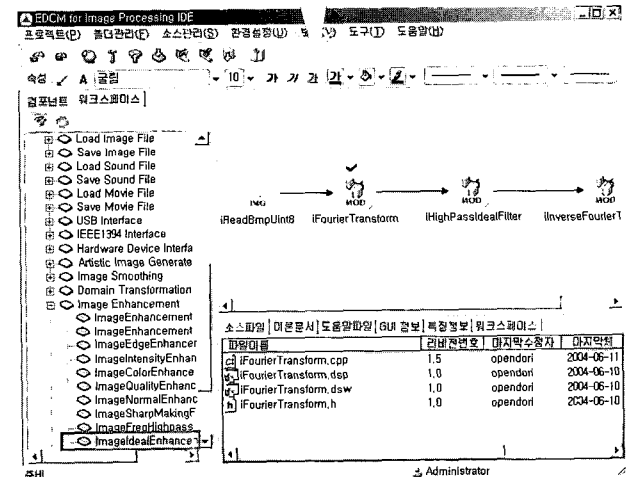
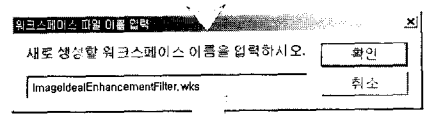
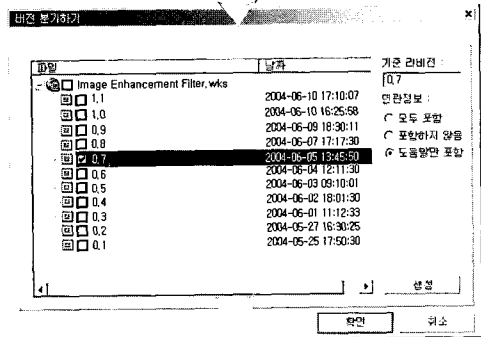
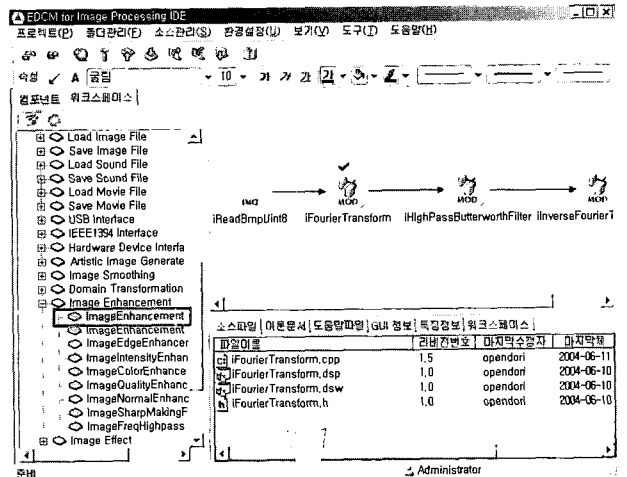
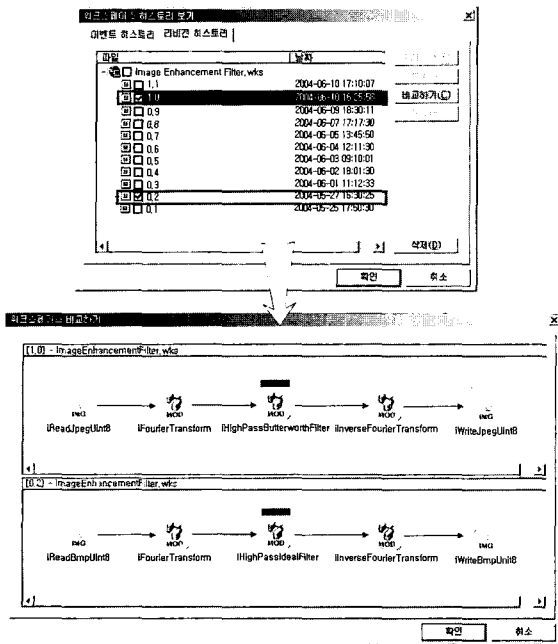
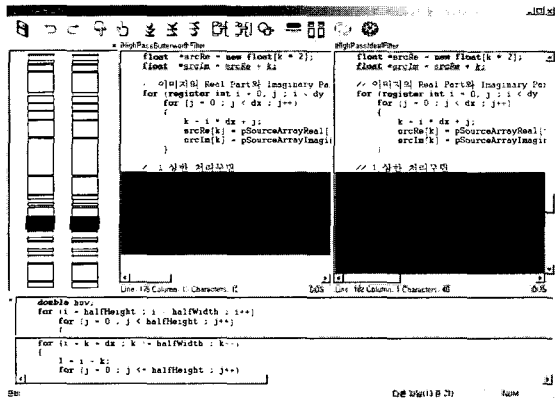


그림 8. 기존의 중간 산출물의 새로운 소프트웨어로의 분기 예제
 Fig. 8. The example which is register with the last product where the intermediate product is new.

능하다는 장점을 가진다. 그리고 하이퍼링크에 사용되는 정보는 외부에 오픈되어 있는 정보가 아니라 SCM 시스템 내부에 존재하는 DB의 인덱스 정보이기 때문에 기존 SCM 시스템을 사용할 때 외부에서 하이퍼링크 정보를 생성하는 것이 불가능하다.



a. 비주얼 워크스페이스용(for visual workspace)



b. 텍스트 소스코드용 (for text source code)

그림 9. 두 개의 파일간의 차이 비교하기
Fig. 9. Comparison function between two files.

제안 시스템은 멀티미디어 기술 개발환경에 대해서 기존 SCM 시스템에 비해 멀티미디어 소프트웨어 개발 환경에 특징들을 잘 지원해줄 수 있도록 개발되었다. 뿐만 아니라 비주얼 프로그래밍 환경을 기반으로 하는 시뮬레이션 환경을 체계적으로 지원해주는 비주얼한 관리 방법은 기존의 시스템들의 관리 방법에 비해 사용자들로 하여금 기 개발된 소프트웨어를 보다 쉽게 이해하고 사용할 수 있도록 유도해 줄 수 있어 재사용성을 극대화할 수 있다는 장점을 가진다.

본 논문에서 제안한 시스템은 현재 'EDCM^[30]'이라는 이름으로 2003년 행망용 적합성 인증을 획득하여 여러 공공기관 전산실에 설치되어 소프트웨어 개발과 유지보수의 생산성을 높여주고 있으며, 본 논문에서는 EDCM

을 멀티미디어 기술 개발 환경에 적합하도록 수정 개발한 프로토타입 시스템으로 멀티미디어 개발 환경을 위한 최적의 SCM 환경을 지원한다.

IV. 결론 및 향후 연구방향

본 논문에서는 빠르게 성장하고 있는 멀티미디어 분야의 개발과 유지보수에 적합한 SCM 시스템을 제안하였다. 제안 시스템은 멀티미디어 개발 환경의 특징을 지원하고 문제점을 해결하기 위하여 개발되었으며, 실제적인 구현을 통하여 멀티미디어 개발 환경에 적합한 SCM 시스템임을 확인할 수 있었다.

무엇보다도 일반적인 SCM 시스템과는 다른 비주얼 프로그래밍 도구를 기반으로 한 시뮬레이션 환경의 비주얼 워크스페이스와의 연계를 통하여 소프트웨어의 사용 및 다른 소프트웨어와의 관계에 대한 가시성을 극대화 하였으며, 이를 통한 소프트웨어에 대한 이해력을 한층 증가시켜 컴포넌트의 사용자, 개발자, 관리자로 하여금 보다 체계적인 관리 및 재사용성을 극대화할 수 있었다.

제안 시스템의 비주얼 워크스페이스에 기반한 형상 관리 기능을 일반 업무절차용 소프트웨어 개발용으로도 확장할 수 있다면 일반적인 SCM 시스템보다는 가시성 및 재사용성을 개선할 수 있으리라 여겨진다.

참고 문헌

- [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 Experiences and trends Norwegian SPIQ project.
- [3] Magnus Larsson, Applying Configuration Management Techniques to Component-Based Systems, MRTC Report 00/24, 2000.
- [4] Axel Mahler, Andreas Lampen, A toolkit for software configuration management,1988.
- [5] R. Conradi and B. Westfechtel, Configuring Versioned Software Product, SCM-6 Workshop. pp. 88-109. Springer LNCS 1167. Berlin, March 1996.
- [6] S. Dart., Concepts in Configuration Management Systems, Proc. of the 3rd. Intl. Workshop on Software Configuration Management. Trondheim, Norway, June, 1991.
- [7] J. Estublier, Workspace Management in Software

Engineering Environments, SCM-6 Workshop. Springer LNCS, 1167. Berlin, Germany, March 1996.

[8] J. Estublier and S. Dami and M. Amiour, High Level Process Modeling for SCM Systems, SCM 7, LNCS 1235. pages 81~98, May, Boston, USA, 1997.

[9] P. Feiler, Configuration management models in commercial environments, Technical report CMU /SEI-91- TR-7. SEI 1991.

[10] D. B. Leblang, The CM Challenge: Configuration Management that Works, Configuration Management, Edited by W. Tichy; J. Wiley and Sons. Trends in software, 1994.

[11] Walter F. Tichy, Tools for software configuration management, In Proc. of the Int. Workshop on Software Version and Configuration Control, pp. 1.20, Grassau, January 1988.

[12] R. C. Water, Automated software management based on structural models, Software Practice and Experience, 1989.

[13] Peter Herzum, Oliver Sims, Business Component Factory : A Comprehensive Overview of CBD for the Enterprise, OMG press, December, 1999.

[14] Darts, S., Concepts in Configuration Management Systems, Proceedings of Third International Conference on SCM, Trondheim, Norway, June, 12-14, pp.18, 1991.

[15] 양해술, 이하용, 소프트웨어 통합 환경구축을 위한 도구의 분석과 설계 방안, 한국정보처리학회, pp834~845, 1996년 7월

[16] William K. Pratt, Digital Image Processing : Luminance edge detector performance, A Wiley -Interscience publication, John Wiley & Sons, inc., pp 532~547, 1991.

[17] P.K.Sahoo, S.Soltani, A.K.C.Wong, and Y.C.Chen, A Survey of Thresholding Technique, Computer Vision, Graphics and Image Processing 41, pp 233~260, 1988.

[18] I.Pitas and A. N. Venetsannopoulos, Nonlinear Digital Filter: Principles and Applications, Kluwer Academic Publishers, pp 302~306, 1990.

[19] 이정현, 채옥삼, 컴퓨터비전과 영상처리 알고리즘의 유지관리와 재사용을 위한 통합개발환경, 정보과학회 논문지(B), 24권 3호, 1997.

[20] JeongHun Lee, Oksam Chae, Multimedia data processing algorithm development environment (MADE), Visual Data Exploration and Analysis VI, Proc. Of SPIE, pp 193~203, 1999.

[21] Khoros, <http://www.khoral.com>

[22] Wits, <http://www.logicalvision.com>

[23] Hello-Vision, <http://www.mtes.co.kr>

[24] DirectX graphedit, <http://msdn.microsoft.com>

[25] Matlab, <http://www.mathworks.com>

[26] ClearCase, <http://www-306.ibm.com/software>

[27] CCC/Harvest, <http://www3.ca.com>

[28] PVCS, <http://www.merant.com>

[29] SourceSafe, <http://msdn.microsoft.com>

[30] EDCM, <http://www.mg.co.kr>

저 자 소 개



이 정 현(정희원)
 1992년 경희대학교 전자계산공학과 학사 졸업.
 1994년 경희대학교 전자계산공학과 석사 졸업.
 1999년~2004년 MG Systems 부설 연구소 연구소장
 1994년~현재 경희대학교 전자계산공학과 박사과정
 <주관심분야: 신호처리, 영상처리, 컴퓨터비전, 미디어센터, 소프트웨어 공학>



채 옥 삼(정희원)-교신저자
 1977년 인하대학교 전자공학과 학사 졸업.
 1982년 오클라호마주립대학교 전기 및 컴퓨터공학과 석사 졸업.
 1986년 오클라호마주립대학교 전기 및 컴퓨터공학과 박사 졸업.
 1986년~1988년 Texas Instrument Image Processing Lab. 선임연구원
 1988년~현재 경희대학교 컴퓨터공학과 교수
 <주관심분야: 신호처리, 영상처리, 컴퓨터비전>

