

## 분산 소프트웨어 개발관리를 위한 공동작업장 시스템 A Shared Workspace System for the Distributed Software Development Management

한관희, 황태일, 김강용  
경남 진주시 가좌동 900 경상대학교 산업시스템공학부

### Abstract

분산 환경에서의 소프트웨어 개발관리를 효과적으로 지원하기 위해 필요한 기능 요건으로는 원격지 개발팀원들 간의 협동작업 지원기능, 개발 과정에서 산출되는 각종 정보객체들의 저장 및 공유 지원기능, 개발에 필요한 각종 프로세스 관리기능 및 프로젝트 관리기능 등을 들 수 있는데, 본 논문에서는 이를 위해 정보객체 관리를 중심으로 하는 공동작업장 시스템을 설계하고, 이를 구현하였다. 즉, 설계 문서, 소스 코드 등과 같은 개발 과정에서 산출되는 각종 정보객체들의 구조 및 관계를 통합관리하는 틀로서 BOC(Bill Of Class)를 제안하며 BOC를 이용한 정보객체 관리 기능의 구조와 기능을 제시한다. 그리고, 제시한 정보객체 관리 구조를 기반으로 분산환경에서의 효과적인 소프트웨어 개발을 지원하는 공동작업장 시스템을 설계 및 구현하고 이의 유용성을 보인다.

### 1. 서론

최근의 S/W(Software)제품의 개발환경은 기업환경의 글로벌화에 따라 개발규모는 보다 커지고 개발조직은 원격 분산화 되어 정해진 납기 안에 고품질의 S/W를 인도하기 위한 소프트웨어 개발관리와 개발팀원간 의사소통은 점점 더 복잡성을 더해가고 있다. 실제로 국내 소프트웨어 업체의 경우에도-특히 SI(System Integration) 회사의 경우- 개발 팀은 주로 고객사에 상주하여 개발을 하고, 프로젝트 매니저나 데이터베이스 엔지니어와 같은 전문 엔지니어들은 본사에서 작업을 수행하여 S/W를 개발하는 '협동작업에 의한 S/W개발'이 급증하여 개발팀원간의 업무할당 및 조정, 의사소통, 협동작업 등을 효율적으로 관리하여야 할 필요성이 과거보다 매우 커지고 있다.

이와 같이, 최근 들어 분산개발팀에 의한 S/W개발이 가속화되면서 기존의 CASE(Computer Aided Software Engineering) 도구에 의한 단위 개발업무 자동화를 넘어선 통합 개발관리 시스템에 대한 요구가 높아지고, 또 이에 대한 연구도 활발히 이루어지고 있다. 소프트웨어 개발은 개발에 관계하는 구성원들이 하나의 작업단위인 프로젝트를 구성하여 적합한 개발 프로세스를 수행하여 목표로 하는 S/W제품을 생산해내는 일련의 과정이라고 정의할 수 있는데, 분산환경에서의 통합 S/W

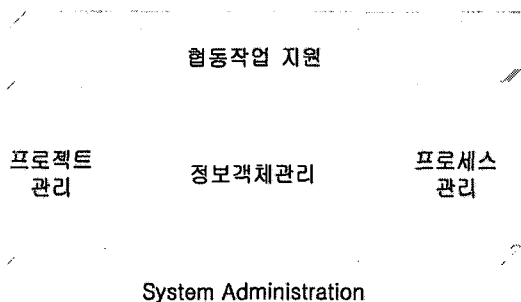
개발관리를 효과적으로 지원하기 위해 필요한 기능요건으로는 1)원격지 개발팀원들 간의 협동작업 지원기능, 2)설계문서, 소스코드 등과 같은 개발과정에서 산출되는 각종 정보객체(Information Object)들의 저장 및 공유를 지원하는 정보객체 관리기능과, 3)개발에 필요한 절차 및 단위업무를 정의하고 이를 자동 실행하는 프로세스 관리기능 및 4)개발 일정계획, 자원 할당 및 상태 모니터링 등과 같은 프로젝트 관리기능 등을 들 수 있다[1,2]. 즉, 요약하자면 분산 개발팀원들은 팀원간 협동작업을 가능케 하는 공동작업장(shared workspace) 구조하에서 S/W제품 생산에 필요한 각종 정보객체들 생성하거나 공유 및 분배해가면서 개발작업을 수행해나가게 되는데, 이 과정에서 협동작업 지원기능이나 프로젝트 관리기능 및 프로세스 관리기능들을 활용한다.

여기서 현 시점에서 손쉽게 활용 가능한 기술들을 살펴보면, 첫째로 마이크로소프트사의 MS Project[3]와 같은 프로젝트 관리도구를 들 수 있고 이를 이용해서 일정관리나 자원할당 및 균형화 등이 가능하나, 이러한 도구들은 개발과정에서 발생하는 각종 정보의 저장 및 유통 과정을 지원하지 못하고 팀원간 협동작업을 지원하는 기능도 미약하다. 둘째로, IBM사의 MQ Series Workflow[4], Fujitsu사의 Teamware Flow[5], Staffware사의 Staffware[6]와 같은 일반적인 워크플로우

제품들을 들 수 있는데, 이러한 제품들은 정해진 프로세스의 흐름관리 기능에 치중되어 있고 일반적으로 프로젝트 관리 도구와 통합되어 있지 않으며 S/W 개발시 필요한 문서의 저장/공유, 실시간 채팅, 게시판, 전자 칠판 기능 등 팀원 사이의 협동작업을 지원 할 수 있는 기능이 취약하다. S/W 엔지니어링 분야에서 연구되어 온 S/W 실행엔진(software enactment engine)은 일반 워크플로우 제품보다 S/W 개발 프로세스 관리에 유연성을 가지고 있고 여타 S/W 개발도구와의 용이한 통합 기능을 제공하는 장점이 있으나 다른 접근 방법에 비해 사용하기가 어렵고 여러 가지 협동작업을 지원하는 기능들이 제공되고 있지 않다. 그리고, S/W 형상관리 도구들을 들 수 있는데 형상관리 도구들은 버전관리 등과 같이 단위 정보객체의 변경관리에 주된 초점을 맞추고 있다.

이처럼 원격 분산환경에서의 S/W 개발관리를 위해서는 제조업의 PDM(Product Data Management) 시스템과 같은 개념으로 S/W 개발 프로세스와 그 과정에서 생성되는 각종 정보를 통합관리하고 개발팀간 협동작업을 지원할 수 있는 웹 기반의 통합 개발 관리 환경이 필요하다. 즉, 공동작업장을 기반기구조로 하고 각종 정보객체들의 저장 및 공유기능, 프로세스 관리기능, 협동작업 지원기능 및 프로젝트 관리기능을 갖춘 통합된 공동작업장 환경의 개발이 요구되는데, 이를 위한 시스템 구성도는 (그림1)과 같다. (그림1)에서 시스템 운영 기능은 시스템에서 공통적으로 사용되는 사용자 관리, 접근 및 보안관리 기능을 총칭한다. (표1)은 공동작업장 시스템이 갖추어야 하는 일반적인 요구 기능들을 나타낸다. (그림1)에서 보듯이 분산 S/W 개발관리를 위해 필요한 기능 중에서, 개발 과정에서 산출되는 각종 정보객체들을 체계적으로 분류하고 정보객체들 간의 연관 관계를 구조적으로 정립하여 통합관리하는 기능은 가장 기본이 되는 기능이라 할 수 있다.

본 논문은 분산개발팀에 의한 객체지향 S/W개발시의 효과적인 개발관리를 위해 정보객체 관리를 중심으로 하는 관리 틀을 제시하



(그림 1) 공동작업장 시스템의 기능구조

(표 1) 공동작업장시스템의 일반적 요구기능

| 서브 시스템   | 상세 기능               | 기능 설명   |
|----------|---------------------|---|
| 프로젝트관리   | WBS 정의              | -개발 활동의 생성 및 수정.                                    |
|          | 프로젝트일정 계획           | -프로젝트 계획수립,자원할당.<br>-간트차트에 의한 가시화.                  |
|          | 프로젝트진도 관리           | -개발활동의 상태/산출물의 상태 관리.<br>-프로젝트 진도 관리.               |
| 협동작업지원   | 게시판/포럼              | -비동기 text conferencing                              |
|          | 원격보고/회의             | -동기 의사소통<br>-전자칠판, 영상회의 기능.                         |
|          | 팀 일정 관리             | -팀 단위의 일정 관리 기능.                                    |
|          | 실시간 채팅              | -동기식 text conferencing.                             |
| 프로세스관리   | build-time workflow | -workflow 정의.                                       |
|          | run-time workflow   | -workflow 실행.<br>-사용자 work item 관리.<br>-event 발생통지. |
| 정보객체관리   | 소프트웨어 구조 관리         | -단위 컴포넌트 관리<br>-컴포넌트관계 관리<br>-파트 표준화 관리             |
|          | 변경영향 분석             | -구성,참조,종속,파생관계분석                                    |
|          | check-in/check-out  | - check-in<br>- check-out                           |
| 시스템 운영관리 | 보안/접근관리             | - 사용자 승인, - 접근 권한.                                  |
|          | 저장소 관리              | - DBMS/file system 관리.                              |
|          | Directory관리         | - 사용자/자원  |

고 이를 시스템으로 구현하는 것을 그 범위로 하며 논문의 목적은 다음과 같다. 첫째, 일반 제조업에서 제품의 구조를 표현하는 BOM (Bill Of Material)이 제품의 설계 및 생산 과정에서 관련정보들을 통합하는 기본 틀로 사용되는 것[7]과 유사하게 S/W개발에 있어서 필요한 각종 정보객체들의 구조 및 관계를 통합관리하는 틀로서 BOC(Bill Of Class)를 제안하며, BOC를 이용한 정보객체 관리 시스템의 구조와 기능을 제시한다. 즉, S/W 개발에 관련된 각종 정보객체들을 BOC에 의해 구조 및 관계를 일관된 시각으로 제공함으로써 각 정보객체들의 종속 관계, 참조 관계 및 구성 관계를 쉽게 파악할 수 있도록 한다. 둘째, BOC 구조내의 부품을 구성하고 있는 데이터들을 표준화하여 부품 데이터사전을 구성하여 프로그램 작성시 표준부품(데이터)만을 사용하여 함으로써 개발생산성과 유지보수 생산성을 향상시키는 방안을 제시한다. 셋째, 제시한 정보객체 관리 구조를 기반으로 분산환경에서의 효과적인 소프트웨어 개발을 지원하는 공동작업장 시스템을 설계 및 구현하고 이의 유용성을 보인다.

본 논문의 구성은, 2장에서 관련 연구를 정리하고 3장에서 공동작업장 시스템의 기본 관리 틀인 정보객체 관리 프레임워크를 제시하고 4장에서 공동작업장 시스템 구현 내용을 설명하며, 마지막으로 5장에서 결론 및 향후

방향을 서술한다.

## 2. 관련 연구

분산환경에서의 통합 S/W 개발관리를 위한 연구로서, 미국의 ARCADIA 프로젝트[8]의 목표는 대형 복합 S/W의 분석, 개발, 유지보수를 위한 S/W엔지니어링 환경을 구축하는 것으로 프로젝트 결과물 중의 하나인 Endeavors 시스템은 개발팀원들 사이의 프로세스 관리 및 조정 행위를 지원하기 위한 '개방형 분산 프로세스 실행환경'으로 동적인 프로세스 변경과 사건 모니터링 및 외부 개발도구와의 통합기능 등을 그 특징으로 하고 있다. Ozweb[9]은 프로세스 정의 및 실행을 위해 규칙 기반의 고유한 프로세스 모델링 언어를 사용하고 있으며 전후진 연쇄추론을 특징으로 하는 프로세스 엔진을 가지고 있다. Ozweb은 S/W개발 과정에서 생성되는 각종 산출물들의 저장소로서의 웹 기능을 제공하는 것에 초점을 맞추었다. MILOS 프로젝트[1]는 '분산 S/W개발을 위한 지원시스템'을 목표로 하고 있으며 개발 프로세스 계획, 프로세스 실행 엔진, 프로세스 실행 조정, 변경사항에 대한 통지 메카니즘 등을 제공하고 있으며 프로젝트 관리기능의 강화를 특징으로 들 수 있고, 최근에는 개발팀원의 경험이나 지식을 개발 조직 내에서 효과적으로 활용하기 위한 지식 관리 시스템으로 확장하고 있다[10].

분산환경에서의 정보객체 관리는 S/W형상관리 분야에서 활발히 연구되고 있으며[11], 최근에는 기존의 변경관리 기능에 부가해서 S/W개발 프로세스 및 프로젝트 관리 기능을 통합하려는 연구들이 이루어지고 있다[12,13]. [14]에서는 하나의 정보객체에 대한 동시적 변경이 발생했을 때 이를 처리하는 방법에 대한 연구가 수행되었는데 단위 정보객체의 변경과 그 처리 프로세스에 연구 대상을 제한하였다.

문서관리를 중심으로 하는 공동작업장 시스템인 BSCW (Basic Support for Cooperative Work)[15]에서는 정보객체들의 상태 변화를 초래하는 사건감지 및 팀원들 사이의 인지(awareness) 기능의 효과에 대한 연구가 이루어졌다. 즉, 협동작업을 위해서는 공유 대상이 되는 문서에 대해 누구에 의해 언제 무엇(문서의 생성, 삭제, 수정, 이동 등)이 이루어졌는지를 팀원들이 인지하는 인지 기능의 중요성을 강조하였다. DHT (Distributed semantic HyperText) 접근 방법[16]에서는 의미론적 하이퍼텍스트(semantic hypertext)모델에 기반하여 정보객체들의 관계를 모델링하고 이를 분산환경에서의 가상저장소로 구현하였으나 단위 클래스의 형상관리에 중점을 두었다. [17]에서는 S/W의 재사용성을 향상시키고 다양한 산출물을 효율적으로 관리하기 위해 소스 코드로부터 구문 분석에

의해 각 클래스의 클래스 부품정보(클래스명, 멤버함수, 속성)를 추출하고 관련 클래스간 관계를 형성함으로써 효율적인 객체관리 저장소를 설계하였으나 관리대상을 클래스로만 한정하였다. [18]에서는 S/W 부품관리 측면에서 S/W제품 개발에 사용되는 부품이라고 할 수 있는 데이터들을 표준화하여 다중 프로젝트간에 표준화된 부품(데이터)만을 사용함으로써 개발생산성 제고와 유지보수를 용이하게 하려는 연구가 수행되었는데, 이 연구에서는 관리대상을 S/W를 구성하는 부품 단위인 데이터와 데이터를 구성하는 약어로만 국한하였다.

위에서 살펴본 바와 같이 기존 연구의 대부분은 분산환경에서의 S/W 개발관리를 위해 필요한 기능 중 일부분만을 다루고 있어서 통합된 개발관리에 대한 전체적인 요구사항을 만족시키지 못하고 있는 실정이다.

## 3. 정보객체 관리 프레임워크

효과적인 S/W제품 개발을 위해서는 개발팀원들이 각종 정보객체들을 손쉽게 생성, 저장 및 공유할 수 있도록 지원해야 하는데, S/W 개발과정에서 관리대상이 되는 정보객체들은 각종 문서와 클래스로 구현된 소스 코드 및 클래스내의 속성명/메서드명이나 데이터베이스 테이블명/컬럼명 등을 정의하는 데이터들로서, 이들 정보객체들은 상호 여러 가지 형태로 복합적인 연관관계를 맺고 있기 때문에 개발 단계가 진행될수록 정보객체들 간의 상호 관계를 파악하기 어렵게 되어 효율적인 S/W개발을 저해하는 요인이 된다. 그러므로, 이러한 복잡한 연관관계를 구조화하여 관련된 정보객체들을 일관되게 관리할 수 있는 통합적인 관리 틀을 제공할 필요가 있다.

또, 분산개발팀에 의한 S/W 개발시에는 개발팀원간 협동작업이 필수적인데, 이를 위해서는 정보객체 자체의 콘텐츠(contents) 정보뿐만 아니라 컨텍스트(context) 정보에 대한 관리가 필요하며 컨텍스트 정보는 포함 관계, 참조 관계, 종속 관계 등 주로 관계 정보에 의해 표현되므로[19] 분산 S/W 개발관리 시스템에서는 정보객체들 사이의 연관관계를 일관된 방법으로 관리하는 것이 중요하다.

그리고, 특히 분산개발팀에 의한 S/W 개발시 S/W를 구성하는 부품이라고 할 수 있는 데이터를 개발팀원들 사이에 제약없이 사용함으로써 동일한 데이터가 서로 다른 용도로 사용되거나, 동일한 엔티티(entity)에 대해 서로 상이한 데이터로 정의하는 등의 사례가 빈번하여 개발 생산성이나 유지보수 생산성에 악영향을 주고 있어서 데이터 공용화 및 표준화 방안이 마련되어야 한다.

### 3.1 BOC(Bill Of Class) 관리

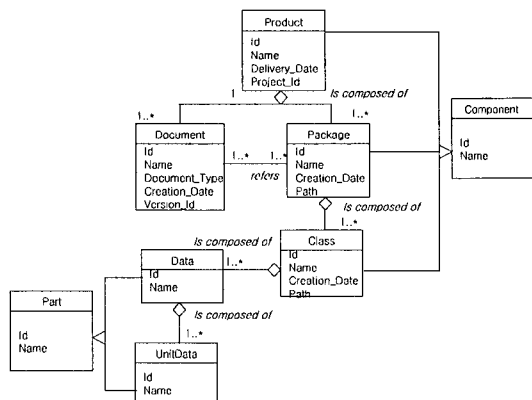
하나의 객체지향 S/W제품은 하나 이상의

패키지(Package)와 하나 이상의 문서로 이루어져 있다. 여기서 패키지는 상호관련성이 높은 클래스들의 묶음을 의미한다. 그리고, 하나의 패키지는 하나 이상의 클래스로 구성되어 있고, 하나의 클래스는 클래스명, 속성명, 메서드명 등 복수 개의 데이터로 구성된다. 여기서 하나의 데이터는 데이터를 구성하는 기본단위인 단위데이터(Unit Data)의 조합으로 구성된다(예를 들면, 재고수준을 나타내는 데이터인 InventoryLevel은 Inventory와 Level이라는 단위데이터들의 조합으로 구성됨). 단위데이터와 데이터는 S/W를 구성하는 최하위 부품으로서 이는 부품(Part) 클래스로 추상화될 수 있고 식별자와 이름을 공통 속성으로 갖는다. 또, 클래스, 패키지 및 제품은 소프트웨어를 구성하는 중간품으로서 이는 조립품(Component) 클래스로 추상화될 수 있고 식별자와 이름을 공통 속성으로 갖는다.

본 연구에서는 이와 같은 S/W 제품의 구성 관계를 'BOC(Bill Of Class)'라 칭하고 소프트웨어 개발사의 각종 산출물을 BOC의 관점에서 관리한다. BOC의 논리적 구조를 UML(Unified Modeling Language) 클래스 다이어그램으로 표시하면 (그림2)과 같다.

각 정보객체간의 관계는 하나의 제품 내에 어떤 패키지들이 소속되어 있고 또 하나의 패키지 내에 어떤 클래스가 소속되어 있는가를 표현하는 계층(hierarchy) 관계, 특정한 클래스에 관한 설계정보를 담고 있는 설계 문서는 어느 것인가를 표현하는 참조(reference) 관계, 클래스간의 슈퍼-서브 관계를 표현하는 종속(dependency) 관계 및 하나의 객체내에서의 변경사항을 표현하는 파생(derivation) 관계를 정의하고 관리하게 된다. 즉, S/W 개발 과정이 진행되면서 발생하는 산출물을 저장소에 저장하거나 변경할 때는 BOC 구조에 따라 등록/수정이 될 수 있도록 함으로써 단위 산출물과 산출물 사이의 관계를 일관성 있게 관리할 수 있다.

정보객체간 관계는 크게 문서-클래스 관



(그림 2) BOC의 논리적 구조

계, 패키지-문서 관계, 클래스-패키지 관계 및 슈퍼-서브클래스 관계로 분류된다. 문서-클래스 관계에서는 해당 클래스 정보가 어느 문서에 포함되어 있는지(클래스-문서)와 해당 문서는 어떤 클래스 정보를 포함하고 있는지(문서-패키지-클래스)를 표현한다. 패키지-문서 관계에서는 해당 패키지 정보가 어느 문서와 관련이 있는지를 나타내며(패키지-문서), 클래스-패키지 관계에서는 해당 클래스가 어느 패키지에 포함되어 있는지를 나타낸다(클래스-패키지-제품). 슈퍼-서브클래스 관계에서는 하나의 슈퍼클래스에 종속된 서브클래스들을 표시하며 동시에 각 서브클래스가 어느 패키지에 소속되어 있는지를 나타낸다(슈퍼클래스-서브클래스-패키지). 그리고, 하나의 정보객체 내에서의 변경 사항에 대한 파생 관계는 버전 트리(version tree)로 표현한다.

단위 정보객체 중에서 문서는 그 성격에 따라 관리용과 기술용으로 구분하는데, 관리 문서는 프로젝트 착수보고서, 진도보고서, 일정계획서, 예산 계획서 등과 같은 프로젝트 관리에 필요한 문서를 의미하며 기술 문서는 실제 제품 개발과정에서 발생한 요구사항 분석서, 시스템 분석/설계 문서, 시험 문서 등을 의미한다. 주요 기술 문서는 프로젝트 착수시 프로젝트 스태프 등록 과정에서 각 스태프에서 산출되어야 하는 베이스라인(baseline) 문서로 정의함으로써 프로젝트의 진도관리를 위한 마일스톤으로 사용된다. 개발과정에서 소스 코드의 형태로 산출되는 클래스는 그 용도에 따라 크게 세 가지로 나누어 사용자 인터페이스 부분인 프리젠테이션(presentation) 부분과 응용 로직(application logic) 부분 및 데이터베이스 연결이나 보안 관리 등과 같은 서비스(service) 부분으로 분류하여 관리한다.

### 3.2 파트사전(Part Dictionary) 관리

효율적인 정보객체 관리를 위해 (그림2)의 논리적인 BOC구조를 두 부분으로 나누어 제품-패키지-클래스 관계는 BOC 기능에서 관리하고, 데이터-단위데이터 관계는 파트사전 기능에서 분리하여 관리한다. 파트사전 관리에서는 S/W개발에 사용되는 각종 데이터를 파트사전에 미리 정의된 단위데이터의 조합만으로 구성하게 하고, 단위데이터의 조합으로 이루어지는 데이터도 파트사전에 등록하여 관리함으로써 데이터 표준을 꾀할 수 있다. 즉, 클래스 이름이나 데이터베이스 테이블 이름 등에서 사용되는 각종 데이터명을 표준화된 단위데이터들의 조합으로 구성하는 것만을 허용함으로써 동일한 데이터가 서로 다른 용도로 사용되거나 동일한 사물을 칭하는 데이터가 개발자들간에 서로 상이한 데이터 이름으로 사용되는 것을 방지하여 개발과 유지보수 생산성을 향상시킬 수 있다.

그러므로, 개발자는 개발과정에서 데이터를 정의하여 사용하고자 할 때에는 우선 파트사전에 원하는 데이터가 있는지를 찾아보고, 있으면 등록된 데이터를 사용하며 없는 경우에는 파트사전에 등록된 단위데이터의 조합으로 구성할 수 있는지를 살펴보고, 가능한 경우에는 단위데이터의 조합으로 데이터를 구성하여 승인과정을 거쳐서 파트사전에 등록하고, 원하는 단위데이터가 존재하지 않을 때는 먼저 단위데이터를 생성하여 승인과정을 거쳐 승인 후 단위데이터들을 조합하여 원하는 데이터를 생성한다. 이와 같이 단위데이터나 단위데이터의 조합으로 구성된 데이터를 파트사전에 등록할 경우에는 항상 정해진 승인 과정을 거치도록 함으로써 데이터 남발을 방지할 수 있다. 데이터에 관한 신규 등록뿐만 아니라 모든 정보객체에 대한 변경(등록/수정/삭제) 사항은 항상 정해진 승인 절차를 필요로 하며 이는 워크플로우 관리의 일부인 정보객체 변경관리에서 관리한다.

#### 4. 시스템 구현

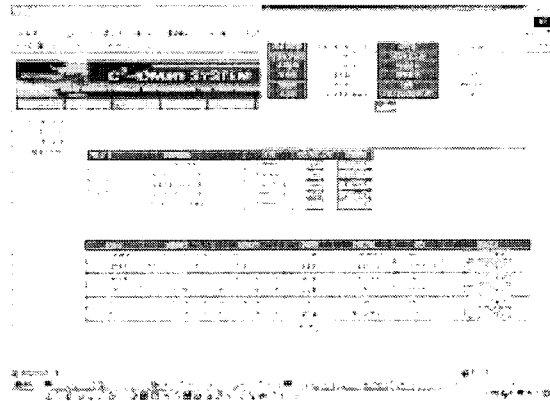
공동작업장 시스템은 (그림1)의 구성도를 기준으로 하여 (표1)에서 나타난 기능들을 중요 기능 중심으로 구현하였다.

시스템의 구현은 원격지에서의 용이한 접근이 보장되도록 웹 기반 시스템으로 구성하였고 시스템 구조의 개방성과 재사용성을 향상시키기 위해 3계층 CS(Client-Server) 구조로 설계하였다. 이를 위해 사용자 인터페이스 부분은 JSP(Java Server Page)와 자바 애플릿(applet)을 사용하였고, 응용 로직 부분은 자바 클래스와 자바빈즈(Java Beans)를 이용하였다. 웹서버로는 Apache 1.3.19과 서브릿(Servlet) 엔진으로 Tomcat3.2를 사용하였으며 정보객체들의 영속적인 저장관리를 위해서는 MS SQL2000을 사용하였다.

##### 4.1 프로젝트 관리

프로젝트 관리는 분산환경에서의 S/W 개발 프로젝트 수행에 필요한 세부활동들(WBS: Work Breakdown Structure)과 그 순서들을 정의하는 작업 템플릿 등록 기능과 등록된 프로젝트 작업 템플릿을 이용하여 해당 프로젝트 세부 스텝과 일정을 등록하여 진도 관리를 하는 기능으로 나누어져 있다.

프로젝트 스텝을 등록할 때는 해당 스텝에서 요구되는 베이스라인 문서를 등록할 수 있게 하여 이를 이용해 진도관리를 용이하게 하였다. 하나의 프로젝트는 5가지 상태(not ready, ready, running, completed, aborted)를 가질 수 있으며 시스템 내외부 사건에 의해 상태 전이가 발생한다. 프로젝트 내의 각 세부 활동들인 프로젝트 스텝도 프로젝트 상태



(그림 3) 프로젝트 정보

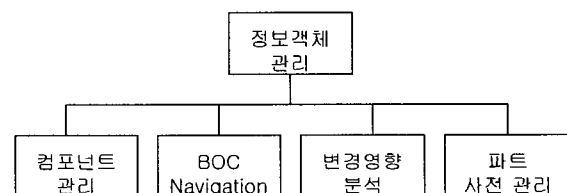
와 동일한 5 가지 상태를 갖는다. 프로젝트 스텝 등록시 베이스라인 문서가 등록된 작업에 대해서는 해당 베이스라인 문서가 승인 절차를 거쳐 시스템에 등록되면 자동적으로 해당 스텝의 상태가 'Running'에서 'Completed'로 전이된다. 현재 정의되어 있는 베이스라인 문서는 요구사항 분석서, 시스템 분석서, 시스템 설계서, 테스트 결과서, 사용자 설명서 등이 있다.

(그림3)은 프로젝트 정보를 보여주는 화면으로 중앙 프레임에는 현재 진행중이거나 기완료된 프로젝트 리스트를 보이며, 상단 프레임에는 특정 프로젝트의 구성원, 일정, 현재 상태 등과 같은 일반적인 정보를 보여준다. 하단 프레임에서는 특정 프로젝트를 구성하는 상세 스텝의 일정 및 상태 정보를 나타낸다.

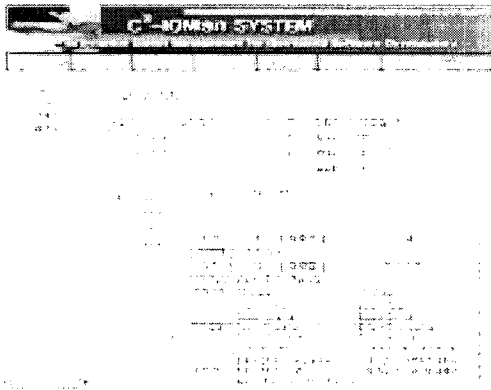
##### 4.2 정보객체 관리

정보객체 관리는 (그림 4)에서 보는 바와 같이 크게 네 가지 기능으로 분류되며, 이 중에서 컴포넌트 관리는 제품, 문서, 패키지, 클래스와 같은 단위 컴포넌트들의 상세 정보를 유지 관리하는 '단위 컴포넌트 관리'와 컴포넌트간 관계를 등록하는 '컴포넌트 관계 관리' 기능을 수행한다. (그림 5)에 정보객체 유형 중 하나인 '패키지'에 대해 객체저장소에 저장되어 있는 패키지에 대한 전체 목록과 특정 패키지 정보객체에 대한 상세 정보와 관계 정보를 예로 보여준다.

BOC 네비게이션에서는 하나의 제품에 대



(그림 4) 정보객체 관리 기능구조도



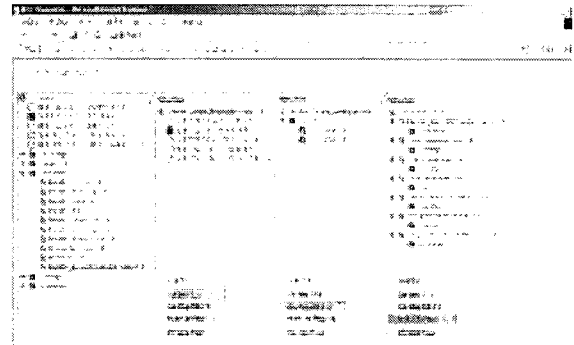
(그림 5) 컴포넌트 상세 정보

한 구조 정보를 트리 형태로 그래픽하게 보여주며, 변경영향 분석에서는 특정한 정보객체를 변경하고자 했을 경우 해당 정보객체의 변경이 다른 정보객체에 어떤 영향을 주는지를 분석하기 위해 정보객체간의 계층, 참조, 종속, 파생관계를 그래픽하게 보여준다. (그림 6)에는 특정한 정보객체(이 예에서는 C3-Space)에 관한 여러 가지 관계를 분석한 화면을 나타내는데, 이 예에서는 하나의 제품의 구성정보를 BOC 형태로 가장 좌측 프레임에 나타내며, 두 번째 및 세 번째 화면에서는 하이라이트된 해당 정보객체의 두 가지 관계(클래스-문서 관계, 클래스-패키지-제품 관계)를 별도의 프레임으로 구분하여 나타내고 있고, 마지막 화면은 슈퍼클래스-서브클래스 관계를 나타내고 있다.

파트사전 관리에서는 데이터의 표준화 사용을 위해 단위데이터와 데이터의 정보를 일관되게 유지 관리하여 프로그램 작성시 객체 저장소에 등록된 데이터만을 사용하도록 제한함으로써 개발 및 유지보수 생산성 향상을 꾀한다. 데이터의 유형으로는 클래스명, 클래스 속성명, 클래스 메서드명, 패키지명, DB 테이블명, DB 테이블 컬럼명이 있다. (그림7)에 단위데이터와 데이터 목록을 나타내는데, 데이터의 경우에는 이름, 한글 명칭, 영문 명칭, 데이터 유형, 길이, 설명, 작성자, 등록일 등과 같은 정보를 가지고 있다.

모든 정보객체가 객체저장소에 체크인(check-in)될 경우에는 사전에 정해진 승인 절차를 거치도록 하며, 체크아웃(check-out)될 경우에는 단순 사용인지 변경인지를 구분하여 변경을 위한 체크아웃일 경우에는 해당 정보객체에 잠금 기능을 설정함으로써 하나의 정보객체가 동시에 여러 곳에서 변경될 수 없게 제어한다. 그리고, 현재 잠겨있는 정보객체를 체크아웃하고자 할 경우에는 현재 소유자와 체크아웃 일자를 보여줌으로써 개발자간 상호 조정을 할 수 있는 정보를 제공한다.

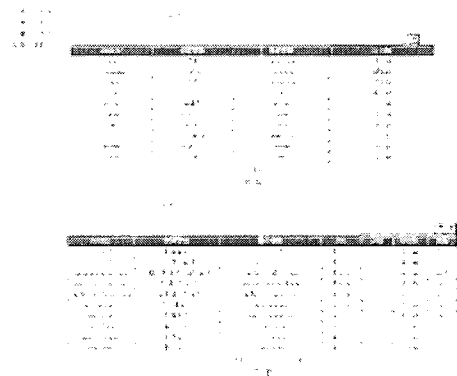
4.3 프로세스 관리



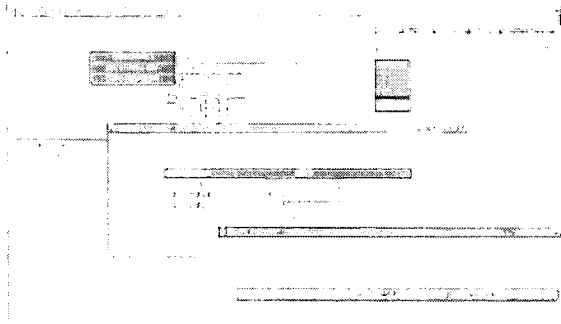
(그림 6) 정보객체 관계 분석

구현된 프로세스 관리 기능은 정보객체의 변경사항을 관리하는 정보객체 변경관리로서, 각 정보객체에 대한 변경사항은 버전 번호에 의해 관리되며 변경이력은 버전트리에 의해 관리된다. 정보객체 변경관리는 정보객체에 대한 등록, 수정, 삭제 등 각종 변경 사항이 발생했을 때 정해진 승인 절차를 관리하는 워크플로우(workflow) 관리와 버전 관리로 구성되어 있다. 워크플로우는 “전체적인 조직의 목표를 달성하기 위해 정해진 규칙들에 의거하여 참여자들 사이의 정보 및 업무가 전달되는 절차와 과정들을 자동화하는 것”[20]이라고 정의되며, 본 연구에서는 정보객체들의 변경사항에 대한 승인 절차를 위해 워크플로우 관리를 적용한다. 예를 들면, 신규 데이터 등록에 관한 승인절차는 승인요청->검토->승인의 과정을 거치게되며 각 단계에서 구체적으로 수행해야 하는 업무내용 및 담당자 등의 할당이 사전에 정의되고 실행시에는 이를 기반으로 승인절차가 자동적으로 실행된다.

(그림 8)에서는 데이터 신규등록 승인 절차를 처리하기 위해 사전에 정의된 프로세스 중에서 하나를 선택해서 실행 프로세스 이름을 기입하고 참여자를 재정의함으로써 실행 프로세스를 구동시키는 과정을 나타내고 있다. 이와 같이 워크플로우 관리는 필요한 변경 승



(그림 7) 단위데이터/데이터 목록



(그림 8) 워크플로우 실행

인절차를 사전에 정의하는 '프로세스 정의' 기능과 정의된 프로세스를 기반으로 실제 작업 흐름을 관리하는 '프로세스 실행' 기능 및 각 개발팀원들이 수행해야 할 작업 내용과 상태 등을 관리하는 '작업목록(worklist) 관리' 기능으로 이루어져 있다. 작업목록에는 수행해야 할 작업내용, 해당 작업의 현재 상태, 도착일 및 작업 수행에 사용되는 응용 시스템 등의 정보가 나타나 있다.

#### 4.4 협동작업 지원

협동작업 지원 기능에서는 가장 중요하고 많이 사용되는 게시판 기능을 구현하였는데, 게시판은 원격지에 떨어져 있는 프로젝트 팀원들에게 알려야 할 사항을 전체적으로 공지하거나 특정한 안전에 대해 상호의견을 교환할 때 사용된다.

#### 4.5 시스템 관리

시스템 관리 기능 중, 사용자 관리는 분산 개발팀원들의 소속팀과 역할 등을 정의하고 접속 승인이나 접근 권한 여부 등을 관리하는데, 구성원들은 하나 이상의 프로젝트팀에 소속될 수 있으며 하나 이상의 역할을 가질 수 있도록 설계하여 조직이나 역할 변화에 유연성을 갖도록 하였다. 각 팀원들의 역할정의는 워크플로우 관리에서 적합한 업무 담당자를 선택하는 기준의 하나로 사용된다.

### 5. 결론 및 향후 과제

현재 급속히 확산되고 있는 원격 분산환경에서의 S/W개발을 지원하기 위해서는 효과적인 개발관리가 매우 중요하나 기존의 연구에서는 1장에서 제시한 통합 개발관리 요구사항의 일부분만을 충족하고 있는 실정이다. 본 연구에서는 효과적인 통합 S/W 개발관리 시스템 정립을 위해 BOC라는 관리 틀을 제안하여 S/W 개발에 필요한 각종 정보객체들을 하나의 틀로 구조화하였고, 정보객체간의 관계

를 구성관계, 참조관계, 종속관계, 파생관계 등으로 정의하여 각종 정보객체들의 관계를 일관되게 관리할 수 있는 방법을 제시하였다.

그리고, S/W를 구성하는 부품이라고 할 수 있는 데이터들에 대해 파트사전 관리를 통해 데이터의 표준화 사용을 꾀함으로써 개발 및 유지보수 생산성을 향상시키는 방안을 제시하였다. 마지막으로, 제안한 정보객체 관리를 기반으로 하는 공동작업장 시스템을 구현하여 그 기능성과 구조를 보였다. 본 연구에서 구현한 공동작업장 시스템은 현재 급속히 확산되고 있는 분산개발팀에 의한 S/W개발과 컴포넌트 기반 S/W개발을 효과적으로 지원하는 유용한 개발관리 도구가 되리라 판단된다.

본 연구의 향후 과제로는 첫째, 정보객체 관리 측면에서 현재는 각 정보객체들의 메타정보나 정보객체간의 관계정보를 사용자가 수동으로 지정해주어야 하기 때문에 시간이 많이 걸리고 번거로운데 이 중의 일부를 시스템에서 소스코드 등을 읽어서 자동으로 정보를 추출하는 방법에 대한 연구가 필요하고, 둘째로 구현된 시스템과 외부 상업용 개발도구들과의 통합이 필요하며, 마지막으로 현재 본 연구는 정보객체 관리에 중점을 두고 개발되었는데 1장에서 언급한 분산환경에서의 통합 S/W개발 관리 시스템에 대한 네 가지 요구조건을 모두 충족시키기 위한 기능구조 설계 및 시스템 구현에 관한 연구가 있다.

#### 참고문헌

- [1] Frank Maurer, Barbara Dellen *et al.*, "Merging Project Planning and Web-Enabled Dynamic Workflow Technologies", IEEE Internet Computing, Vol.4, No.3, pp65-74, 2000.
- [2] Jarir Char, Santanu Paul and Ram Chillarege, "Virtual Project Management for Software", NSF Workshop on Workflow & Process Automation, University of Georgia, Athens, GA, May 1996.
- [3] [www.microsoft.com/office/Project](http://www.microsoft.com/office/Project)
- [4] [www.ibm.com](http://www.ibm.com)
- [5] [www.teamware.net](http://www.teamware.net)
- [6] [www.staffware.com](http://www.staffware.com)
- [7] 한관희, 박찬우, "제품정보관리시스템 개발을 위한 기능 분석에 관한 연구", 한국 CAD/CAM 학회 논문집, Vol.7, No.1, pp. 42-56, March 2002.
- [8] Gregory Alan Bolcer and Richard N. Taylor, "Endeavors: A Process System Integration Infrastructure", Proceedings of the 4th International Conference on the

- Software Process, Brington, England, December 1996.
- [9] G. E. Kaiser *et al.*, "WWW-based Collaboration Environment with Distributed Tool Services", World Wide Web Journal, Vol.1, No.1, pp3-25, 1998.
- [10] Frank Maurer and Harald Holz, "Process Oriented Knowledge Management for Learning Software Organizations", Proceedings of 12th Knowledge Acquisition Workshop (KAW '99), Banff, Canada, 1999.
- [11] Stephen A. MacKay, "The State of the Art in Concurrent, Distributed Configuration Management", Proceedings of the 5th International Workshop on Software Configuration Management, Seattle, WA, April 24-25, 1995.
- [12] Andre van der Hoek, Dennis Heimbigner and Alexander L. Wolf, "Does Configuration Management Research Have a Future? ", Proceedings of the 5th International Software Configuration Management Workshop, Seattle, WA, April 24-25, 1995.
- [13] Rational Software, "Simplifying the Process of Change- Rational ClearCase", www.rational.com, Cupertino, CA, 2001.
- [14] James J. Hunt, Frank Lamers, Jurgen Reuter, and Walter F. Tichy, "Distributed Configuration Management via Java and the World Wide Web", Proceedings of the 7th International Workshop on Software Configuration Management, pp161-174, Boston, MA, May 18-19, 1997.
- [15] Wolfgang Appelt, "WWW Based Collaboration with the BSCW System", Springer Lecture Notes in Computer Science 1725, pp66-78, Berlin 1999.
- [16] John Noll and Walt Scacchi, "Supporting Software Development in Virtual Enterprise," Journal of Digital Information, Vol.1, No.4, December 1998. <http://www.usc.edu/dept/ATRIUM/Papers/DHT-VE98.html>
- [17] 선수균, 송영재, "통합 관리 모델을 이용한 효율적인 객체 관리 저장소 설계", 한국정보처리학회논문지, Vol.8-D, No.2, pp166-174, 2001.
- [18] 이병엽, 박준기, 양성모, 김현정, "다중 프로젝트간의 데이터 표준화 지원을 위한 도구의 설계 및 구현", 한국정보과학회 '98 추계학술대회 논문집, pp535-537, 1998.
- [19] James A. Highsmith III, Adaptive Software Development, pp261-293, Dorset House Publishing, 2000.
- [20] Object Management Group, "Workflow Management Facility Specification: Version 1.2", <http://www.omg.org>, April 2000.