

실시간 컴포넌트 형상관리를 위한 프로세스 요구사항 Process Requirements for Real-Time Component Configuration Management

정대성*, 채은주*, 한정수*, 백순화**
천안대학교*, 백석대학교 컴퓨터정보과**

Jung Dae-Seong*, Chae Eun-ju*, Han Jung-Soo*,
Baek Soon-Wha**

Cheonan University*, Baekseok College**

요약

CBD 개발 방법론의 발전으로 많은 컴포넌트가 개발되면서 소프트웨어의 빠른 기술변화는 시스템 개발 생명주기가 갈수록 짧아지고 있으며, 최근 들어 시스템 개발에 있어서 오프라인보다 온라인으로 시스템 개발이 늘고 있다. 때문에 실시간으로 컴포넌트 형상관리의 필요성이 나타나고 있으며, 보다 향상된 기술의 개발방법이 요구되고 있다.

본 연구에서는 이를 위한 하나의 해결방법인 형상관리(Configuration Management)를 제안하고, 실시간 컴포넌트 형상관리 작업에 필요한 요구사항에 대해 기술하였다.

I. 서론

최근 들어 소프트웨어 개발방법이 상용컴포넌트, 표준컴포넌트 등을 사용하면서 새로운 패러다임으로 진행되고 있다. 컴포넌트 개발 방법론은 기존보다 빠른 개발을 할 수 있는 장점을 가진다. 이것은 시스템 개발 시 기존처럼 하나에 시스템을 구성할 때 컴포넌트가 집약되어 새로운 시스템이 만들어짐을 의미한다. 최근엔 소프트웨어 형상관리(SCM : Software Configuration Management)가 부각되는데, 이것은 불안정한 구성요소를 가지는 시스템에서 변경이 일어나는 관리기술을 말한다[1][3].

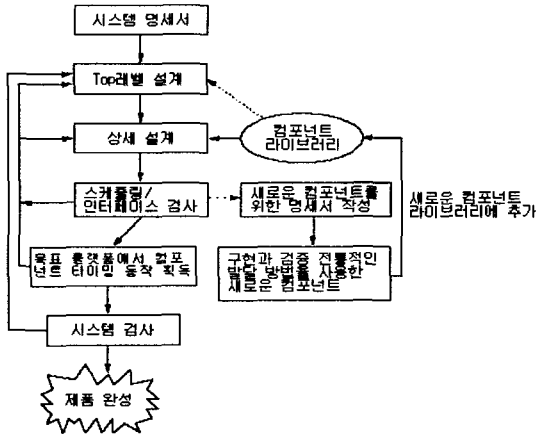
본 논문에서는 실시간 시스템의 개발로 CBSE(Component Based Software Engineering)를 적용하여 가능성을 분석하였다. 그 목적은 실시간 컴포넌트 속성을 식별하고, 재사용하기 위하여 적용할 수 있는 설계방법에 대해서 기술하였다.

II. 실시간 컴포넌트 재사용

재사용을 위한 설계는 현재의 프로젝트로부터 컴포넌트 사용을 위해 최소한의 변경을 요구한다. 추상화는 재사용을 위해 매우 유익하다. 컴포넌트를 설계할 때는 많은 애플리케이션을 생각하여 시도한다. 현재의 사용보다는 오히려 추상적으로 생각하면 재사용은 더 성공적이다. 컴포넌트사이의 차이를 최소화하는 것을 선택하고 추상화 이상 단계로부터 편차를 더 작게 하여, 보다 자주 컴포넌트를 재사용 할 수 있게 한다. 재사용 컴포넌트에 대한 중요한 요소에는 다음과 같다.

- 사용자가 무엇을 필요하고 불필요한지는 재사용 설계를 통해 알 수 있다. 또는 관련 방법을 강조하고 부적절한 정보를 숨긴다.
- 사용자로부터 무엇을 기대하고, 재사용 할 수 있는 설계에 관해 목표가 무엇인가.
- 이진 컴포넌트를 위한 실행이나, 컴포넌트의 적절한 부분을 사용자에게 설명되어야 한다.

실시간 컴포넌트를 위한 설계는 (그림 1)에서 묘사하고 있다.



▶▶ 그림 1. 실시간 컴포넌트를 위한 설계 모델

1. 컴포넌트 라이브러리

실시간 컴포넌트 라이브러리는 다음 요소들을 제공해야 한다.

- (1) 컴포넌트 식별 : 컴포넌트 버전에는 새로운 함수나 속성들이 추가될 수 있다. 만약 다른 컴포넌트 버전들이 같은 이름으로 식별된다면, 시스템에 테스트되지 않은 컴포넌트 버전을 포함할 위험이 있다. 이 때문에 컴포넌트들은 버전들 사이의 변경에 대하여 버전과 정보에 의한 식별이 요구된다.
- (2) 인터페이스 : 컴포넌트(이름, 형, 설명)의 입력포트와 출력포트를 정의한다.
- (3) 테스트 사례 : 테스트 사례들의 설정은 주로 입력 그리고 출력 값들에 회귀 검사를 위해서 한다. 테스트 사례들은 컴포넌트가 특정한 기능성을 넘겨주도록 사용된다. 또한 입출력포트들에 대하여 전제 조건들을 사용할 수 있다.
- (4) 컴포넌트 바이너리 : 특정 처리장치 계열에 대한 컴포넌트에 대한 바이너리 코드이다.
- (5) WCET 테스트 사례들 : 특별한 처리장치 계열에

서 컴포넌트 WCET를 나타낸 테스트 사례들이다. 이전에 사용되었던 대상들을 위한 WCET에 대한 정보는 컴포넌트들 처리장치 필요 조건을 위해 저장되어야 한다.

- (6) 종속성 : 다른 컴포넌트들에 종속성들을 기술한다.
- (7) 환경 : 컴포넌트 동작을 위한 환경을 의미한다.

2. 컴포넌트 온라인 업그레이드

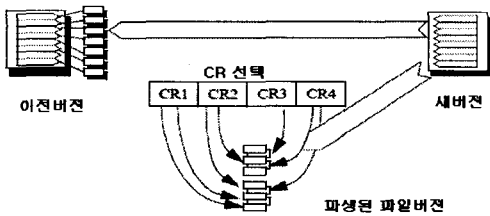
안전한 시스템에서 소프트웨어의 온라인 업그레이드를 위한 방법은 존재해야 하며, 컴포넌트 기반 시스템에서 컴포넌트를 교환할 때 적용할 수 있어야 한다. 시스템에서 컴포넌트 교환 시 새로운 컴포넌트가 불안정하다면 잘못된 결과를 가져올 수 있다. 새로운 컴포넌트를 포함하는 시스템은 시뮬레이션 하기는 어렵다. 실제 시스템에서의 테스트는 새로운 컴포넌트가 시스템을 위태롭게 할 수 있는 위험 가능성 또한 생각해야 하며, 이 문제를 해결하기 위해서 출력이 타당한 범위 내에 있다면, 새로운 컴포넌트는 모니터링 되어야 한다. 만약 그렇지 않을 경우 이전의 컴포넌트는 시스템을 재구성하여 제어하는 것으로 대신해야 한다. 또한 시스템은 컴포넌트의 버전 유지 보수를 지원해야 한다. 실시간 시스템의 온라인 업그레이드는 새로운 컴포넌트 WCET (Worst-Case Execution Time)가 컴포넌트 교환 시간이 짧거나 동시에 발생해야하며, 같은 인터페이스와 일시적인 특성들을 가지고 있는 것 또한 요구된다. 만약 이것이 실행 불가능하면, 새로운 스케줄을 생성하고 컴포넌트를 업그레이드한 시스템을 재구성한다. 이는 결합 허용 방법을 사용하여 짧은 휴지 시간으로 안전하게 할 수 있다[5].

III. 컴포넌트 버전 변경 관리

논리적인 아이템 정의는 변경요청(Change Request

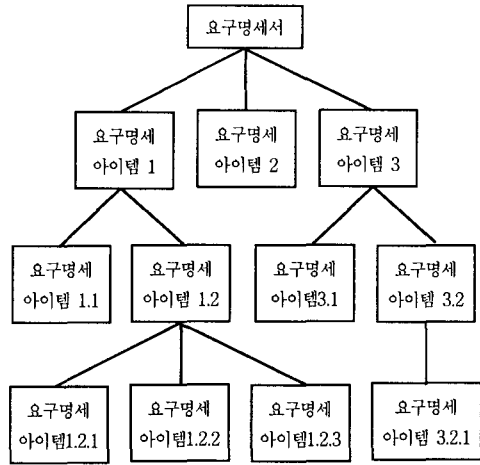
: CR)으로 한다. 사용자가 파일을 가져올 때 CR을 참고하게 되며, 파일이 사용자 작업 공간으로부터 되 돌아갔을 때, 파일버전은 관련 있는 CR에 등록된다. 이렇게 하여 CR은 모든 파일 이름과 변경된 버전들을 모은다.

새로운 컴포넌트 버전을 만들기 위해서는 기존 컴포넌트 버전에서 변경할 파일을 선택하여 만든다. 변경할 파일은 직접 선택되지 않는다. 그러나 변경요청(CR)은 새로운 컴포넌트 버전을 만들기 이전에 선택된다. 이 방식의 이점은 새로운 소프트웨어 버전과 논리적 변화사이의 맵핑이 계획된 논리적 교환과 직접적으로 이루어진다. 그리고 물리적 파일들은 프로그래머가 아닌 도구에 의해서 수행된다[2][4]. (그림 2)는 CR을 선택하여 새 버전으로 변경하는 방법을 묘사한 것이다.



▶▶ 그림 2. CR을 선택하여 나온 새로운 버전

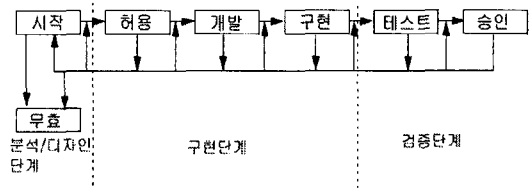
구는 하위 층에 담고 있으며, RS 아이템은 자연어로 기술해야 한다.



▶▶ 그림 3. 요구사항 명세서

1. 프로세싱 요구사항

요구사항 변경 프로세서를 관리하기 위하여 다른 형상관리의 특징들을 사용한다. 형상관리를 이용한 보고서는 추적을 용이하게 할 수 있게 한다. (그림 4)에서는 RS 아이템의 상태를 묘사하고 있다. RS 아이템은 계층적 구조로 구성되어 있다. 기능은 RS 아이템과 서브 트리 아이템의 개별적인 관계로 적용시킬 수 있다. 이와 같은 기능은 CM에 의해서 제공된다.



▶▶ 그림 4. RS 아이템 상태

IV. 형상관리 요구명세 구조

형상관리에서 요구사항은 구현 단계에서 사용된 형상관리(CM)와 도구는 버전 요구사항을 위해서 사용될 수 있다.

만약 요구명세(Requirements Specification : RS)가 한 장의 문서로 작성된다면, 버전이 다른 것은 관리할 수 있지만, 개별적인 요구명세의 항목은 제어할 수 없다. 이유는 요구명세 정의는 가상문서로 이루어져 있기 때문이다. 요구명세의 구조는 (그림3)에 나타나 있다. RS 아이템은 특정한 요구를 기술한다. 최상위층의 구조는 기본적인 요구를 담고 있고 상세 요

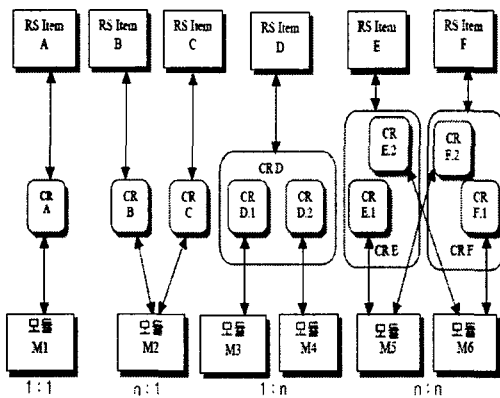
2. 버전 요구명세

RS 아이템은 각각 객체들이지만, 버전관리에 들 수 있다. 모든 RS 아이템은 CM 아이템으로써 새로

운 항목을 만들고, 기존 아이템을 수정하거나 삭제하고, 변경된 그 아이템을 통하여 새로운 버전을 얻는다. CM에서의 각 항목 버전식별은, 각 항목의 일치, 각 버전의 특징적인 관계, 같은 버전의 일치, 프로그래머가 개발한 날짜, 다른 속성 등을 CM에 의존하여 이용한다.

3. 컴포넌트 변경요청

요구사항 명세서와 아이템의 버전관리는 같은 의미로 볼 수 있다. 기존 명세서에 CR을 받아들여 새로운 모듈을 생성하면 새로운 버전이 창출되는 것이다. (그림 5)는 요구명세에서 아이템과 시스템 모듈 사이의 변경요청 관계를 나타내고 있다.



▶▶ 그림 5. 요구명세 아이템과 시스템모듈 사이의 변경요청 관계

이 모델의 목적 중 하나는 구현 프로세스 동안에 요구사항 필요조건을 시각적으로 표시되게 함으로써 개발자들이 인식하기 편하게 하는 것이다. 모듈의 생성은 RS 아이템에 CR을 접목시켜 새로운 모듈을 생성하는데 1:1, n:1, 1:n, n:n의 연관을 가지고 있고 그에 따른 여러 가지 모듈이 생성된다. CR 처리는 CR의 체크인/아웃을 시행하여 개발자는 RS항목에 접근할 수 있는데, CM은 이런 실행 가능한 수단을 제공해야 한다. 또한 특정 RS 아이템에서의 CR은 유사한 다른 아이템(소스코드, 문서)으로 취급할 수 있는데

이렇게 하면 CR에 관련된 항목의 모든 버전들은 용이하게 접근 가능하게 된다.

V. 결론

본 논문에서는 컴포넌트기반의 실시간 형상관리 시스템에 대하여 기술하였다. 기존의 형상관리 시스템의 룰을 그대로 적용시켜, 재사용성을 그대로 두었으며, 실시간 업그레이드 시 타이밍에 중점을 두어 시스템의 문제점을 야기시키는 부분을 제시하였다. 이의 해결방안으로는 타이밍을 통해 업그레이드를 수행한다. 또한 컴포넌트 형상관리에서 컴포넌트 변경요청이 발생할 때 새로운 요구사항을 적용시켜 기존의 컴포넌트 버전보다 진보된 버전으로 생성하여 새로운 컴포넌트 버전으로 접근할 수 있는 방법을 제시하였다.

■ 참고문헌 ■

- [1] Ivica Crnkovic, Peter Funk, Magnus Larsson, "Processing Requirements by Software Configuration Management", In euromicro 99, proceedings of the 25th EUROMICRO conference Milano, Italy, Sep 1999.
- [2] Ivica Crnkovic, "Experience with Change-oriented SCM Tools", In Proceedings Software Configuration Management SCM-7 Boston, MA, USA, May, 1997.
- [3] Magnus Larsson, Ivica Crnkovic, "Component Configuration Management", In ECOOP Conference, Workshop on Component Oriented Programming Nice, France, June 2000.
- [4] Ivica Crnkovic, "A Change Process Model in an SCM Tool", In Euromicro 98, proceedings of the 25th EUROMICRO conference, Sweden, Sep, IEEE, Computer society 1998.
- [5] Damir Isovich, Markus Lindgren, Ivica Crnkovic "System Development with Real-Time Components", In oc. of ECOOP2000 Workshop 22 - Pervasive Component-based systems Sophia Antipolis and Cannes, France, June 2000.