시스템 변형

HÕkan Dyrhage

Rational Software 백서

TP 155

Rational®
the software development company

목차

소개

이 문서에서는 시스템의 변형 유형을 정의하고 관리하는 방법을 논의합니다. 이 문서는 RUP 에 대한 이해를 돕기 위해 작성된 것이 아닌 RUP 의 확장으로 이해해야 합니다. 마지막 섹션에서는 변형 및 변동 개념에 대한 도입으로 RUP 에 미치는 영향에 대해 설명합니다.

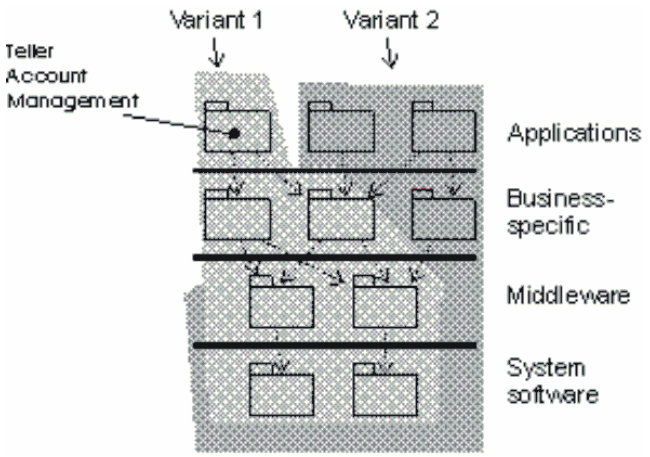
이 부분은 향후 RUP 의 개선 및 확장을 위한 주요 영역으로서 이 문서를 통해 기본 개념에 대해 설명합니다.

시스템 변형

많은 시스템은 여러 가지 변형으로 전달됩니다. 즉, 시스템은 각 고객(고객군)에 따라 다르게 구성, 패키지화 및 설치됩니다. 경우에 따라 같은 시스템을 다르게 설치 및 개조하여 다른 변형을 나타내기도 합니다. 또한 고객마다 다른 시스템 파트를 전달함으로써 변동을 나타내기도 합니다. 다음 섹션은 몇 가지 변형 예제에 대해 설명합니다.

다른 시스템 파트

각기 다른 고객군에 전체 시스템의 다른 파트를 전달합니다. 예를 들어, 은행 업무 시스템이 두 가지 다른 제품으로 전달됩니다. 이러한 경우 시스템 변형 1 에는 텔레 뱅킹과 관련된 모든 내용이 포함되며 변형 2 에는 은행원 계정 관리에 대한 모든 내용이 포함됩니다. 실행 파일은 응용프로그램 계층의 서브시스템에 정의됩니다. 이는 예를 들어, 은행원 계정 관리 서브시스템과 이 서브시스템이 컴파일하고 실행해야 하는 모든 서브시스템(즉, 이 서브시스템이 직접 또는 간접적으로 가져오는 모든 서브시스템)으로 하나의 변형(다음 그림의 경우 변형 1)이 빌드됨을 의미합니다.



하나의 은행 업무 시스템이 두 가지 변형으로 개발됩니다.

다른 언어

예를 들어, 영어, 불어, 일본어 등 여러 언어로 시스템을 개발하는 경우 각 언어에 대한 시스템 변형을 전달할 수 있습니다. 각 변형 간의 차이는 메뉴 및 도움말 텍스트와 같은 모든 텍스트가 해당 특정 언어로 작성된다는 것입니다.

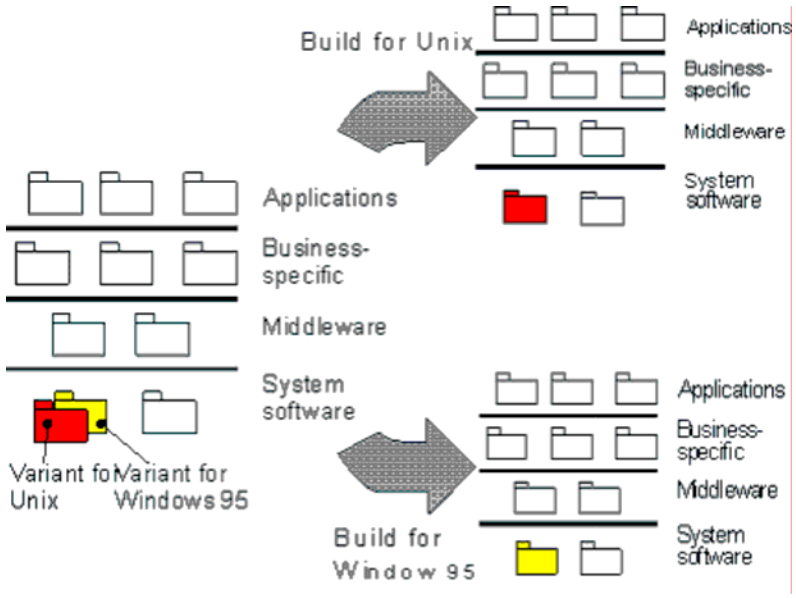
다른 언어를 처리하기 위한 한 가지 방법은 모든 텍스트를 하나의 파일에 모으고 각 언어마다 하나의 파일을 작성하는 것입니다. 특정 언어에 맞는 시스템을 전달한다는 것은 해당 언어의 텍스트가 들어 있는 파일을

포함하여 모든 것을 전달한다는 것을 의미합니다. 소프트웨어가 시작될 때 이 파일을 읽고 모든 관련 변수가 초기화됩니다.

복수 플랫폼

시스템이 호환되지 않는 복수 시스템을 지원하는 경우 각 플랫폼마다 하나의 시스템 변형이 필요합니다. 예를 들어, Windows NT 와 UNIX 에서 모두 시스템을 실행하는 경우 두 가지 시스템 변형이 생성됩니다.

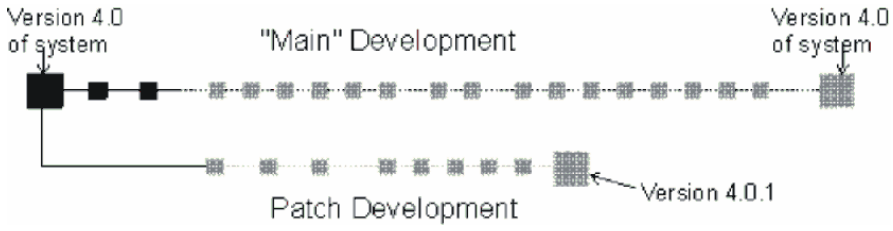
다음 예제의 경우 플랫폼 특정 코드는 하나의 서브시스템에 위치하며 두 가지 서브시스템 변형이 개발됩니다. 컴파일 파일(make 파일)은 함께 컴파일해야 하는 각 소스 코드 파일의 버전을 지정합니다. make 파일은 또한 빌드에 포함되어야 하는 각 서브시스템의 변형을 지정합니다. 따라서 각 플랫폼마다 하나의 make 파일이 필요합니다.



여러 플랫폼에서 실행될 시스템이 개발됩니다.

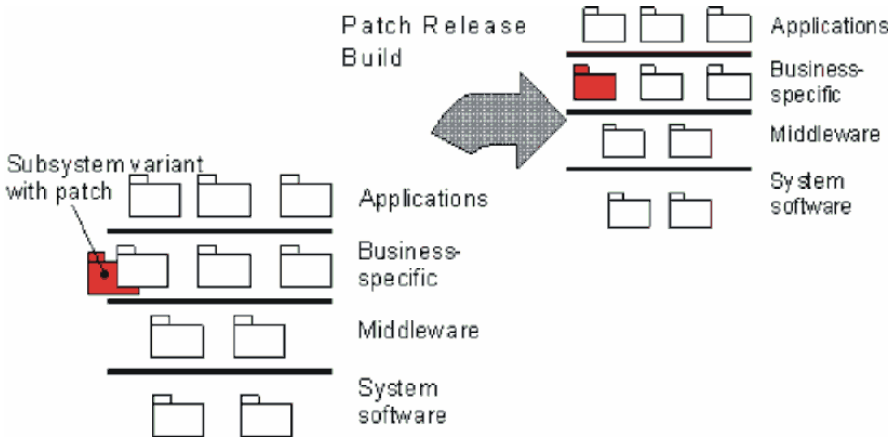
패치 릴리스

경우에 따라 시스템의 패치 릴리스를 개발해야 합니다. 이 작업은 일반적으로 개발 프로젝트와 동시에 수행됩니다. 즉, 패치 릴리스는 또 다른 시스템 변형이며 시스템의 "기본" 버전과 함께 존재합니다.



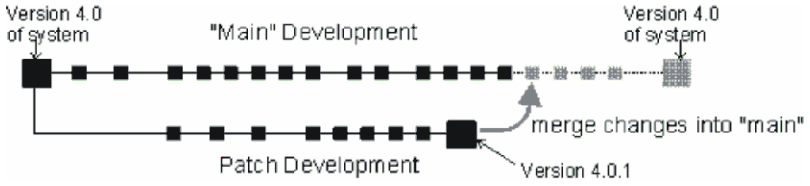
패치 릴리스 개발은 기본 개발 프로젝트와 동시에 수행됩니다. 회색 사각형은 향후 릴리스 및 기준선을 나타냅니다.

필요한 변경사항은 하나 이상의 구현 서브시스템에 위치합니다. 일반 개발 프로젝트는 병렬로 진행되므로 이 서브시스템의 변형은 기본 개발 노력과 병행하여 개발해야 합니다. 빌드를 작성하려면 빌드에 포함되어야 하는 각 서브시스템의 변형을 make 파일에 지정해야 합니다.



패치 릴리스 빌드가 해당 패치를 포함하는 서브시스템의 변형과 함께 작성됩니다.

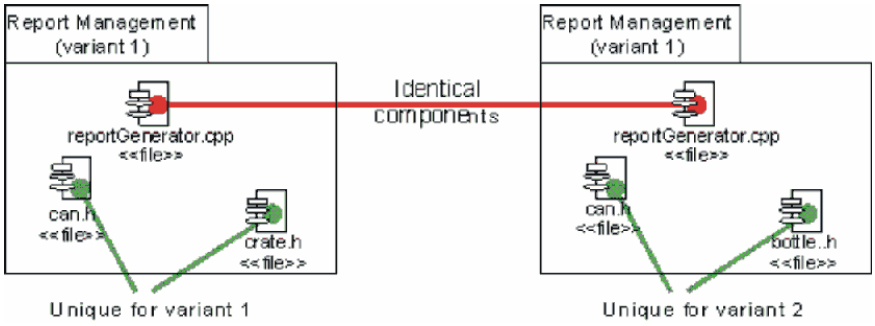
이러한 유형의 변형은 일반적으로 수명이 짧습니다. 패치가 릴리스된 후에는 이 변형이 더 이상 개발되지 않으며 값의 모든 코드 변경사항은 기본 개발 경로에 통합됩니다.



필요한 경우 패치 릴리스의 관련 변경사항은 기본 개발 경로에 병합됩니다.

서브시스템 변형

이전 예제에서는 특정 단일 서브시스템의 두 개 이상의 변형이 자주 필요한 것으로 가정합니다. 해당 변형은 몇 가지 컴포넌트를 공유하기도 하고 각 서브시스템 변형에 고유한 컴포넌트도 있습니다.



보고서 관리 구현 서브시스템의 두 변형(하나의 컴포넌트가 두 변형에서 동일한 경우). 나머지 컴포넌트는 각 변형에 고유한 컴포넌트입니다.

일반적으로 서브시스템의 각 변형은 단일 구현자에 의해 개발되며 서브시스템의 변형은 병렬로 개발됩니다. 특정 변형을 개발하는 단일 구현자가 컴포넌트를 변경하는 경우 해당 변경사항은 다른 변형에도 전달되어야 합니다. 이를 위해서는 **형상 관리 및 버전 제어 도구**(CMVC)에서 지원하는 도구를 사용하여 두 변형에서 동일한 컴포넌트를 관리해야 합니다.

변동 메커니즘

시스템 변형은 다양한 메커니즘으로 작성할 수 있습니다. 그 중 몇 가지는 이전 섹션에서 언급한 내용이며 각 메커니즘마다 다른 특성을 갖고 있습니다. 일반적으로 이러한 메커니즘을 함께 사용하여 시스템에 변동을 작성합니다.

- **컴파일(및 링크) 파일**. 컴파일 파일(Unix 환경의 경우 make 파일)에는 실행 파일에 함께 컴파일하여 링크해야 하는 각 소스 코드 파일의 변형을 지정합니다.
- **동적으로 로드된 컴포넌트**. 시스템 파트를 런타임에서 실행 프로그램에 링크할 수 있는 동적 링크 라이브러리, 애플릿 또는 Active X 컴포넌트로 개발합니다. 이러한 컴포넌트는 CMVC 도구로 관리할 수 있어 고객에게 해당 서브세트를 전달할 수 있습니다.
- **시작 파일**. 시스템이 시작되어 초기화될 때 소프트웨어가 읽는 정보를 포함하는 파일을 사용합니다. 예를 들어, 자원 파일(Windows), 시작 파일 또는 초기화 파일은 시스템이 시작되고 다른 설정을 적용할 때 읽는 파일입니다. 즉, 모든 텍스트를 시스템이 시작될 때 읽는 텍스트 파일에 보관하는 경우 시스템을 다른 언어에 맞게 사용자 정의하려면 이 파일을 사용합니다.
- **시스템을 여러 실행 파일로 나눕니다**. 시스템을 여러 실행 파일(예: .exe 파일)로 개발하며 고객에게는 이 실행 파일의 조합을 전달할 수 있습니다. 다양한 실행 파일 조합은 CMVC 도구로 관리합니다.

RUP 에 대한 영향

이 섹션은 변형 개념에 대한 도입으로 RUP 에 미치는 영향에 대해 간단히 설명합니다.

구현 원칙에 대한 영향

구현 원칙을 확장해야 하는 영역은 다음과 같습니다.

- 다음 단계를 **구현 모델 구조화** 타스크에 추가하십시오.:
  - 개발해야 하는 시스템 변형을 정의하는 방법
  - 변형이 필요한 서브시스템을 결정하는 방법
  - 변동을 달성하기 위해 사용해야 하는 변동 메커니즘을 결정하는 방법
- **시스템 통합 계획** 타스크에서는 변형을 고려하고 다른 시스템 변형의 통합 방법에 대한 계획을 수립해야 합니다.
- 서브시스템 변형 간의 병렬 개발에 대해 자세히 설명해야 합니다. 예를 들어, 동일한 컴포넌트를 분할해야 하는 경우 해당 결과 또는 컴포넌트 병합 방법에 대해 설명해야 합니다. 이는 또한 여러 구현자 관련 타스크에 영향을 줍니다.

기타 구현 타스크 및/또는 활동에도 영향을 줄 수 있습니다.

기타 원칙에 대한 영향

변형 또는 시스템 제품군을 개발하는 경우 모든 원칙에 영향을 줍니다. 이전 섹션에서는 구현 원칙에 대한 일부 영향에 대해 설명했습니다. 다음은 영향을 받는 나머지 원칙에 대한 간단한 설명입니다.

- **요구사항**—시스템 변형을 식별하는 방법과 각 고객군이 원하는 사항에 대해 설명해야 합니다.
- **분석 및 디자인**—디자인 모델에서 변형을 모델링하는 방법, 서브시스템 변형을 디자인하는 방법 및 변형 정의 방법에 대해 설명해야 합니다.
- **테스트**—시스템(변형) 제품군을 테스트하는 방법에 대해 설명해야 합니다. 각 시스템 변형을 별도 시스템으로 테스트합니다. 변형은 또한 통합 테스트에도 영향을 줍니다.
- **관리**—일반적으로 각 서브시스템 변형마다 별도 팀에 프로젝트를 구성해야 합니다. 대규모 프로젝트의 경우 각 시스템 변형마다 별도 프로젝트 관리자를 지정할 수 있습니다.
- **환경**—시스템 변형 관리를 지원하는 도구가 필요합니다.
- **배치**—최종 제품을 전달할 고객군이 여러 개이므로 훨씬 복잡한 작업입니다.

Rational
the software development company

본사 안내:

Rational Software
18880 Homestead Road
Cupertino, CA 95014
전화번호: (408) 863-9900

Rational Software
20 Maguire Road
Lexington, MA 02421
전화번호: (781) 676-2400

수신자 부담 전화번호: (800) 728-1212
전자 우편: info@rational.com
웹: www.rational.com
전세계 지사 안내: www.rational.com/worldwide