



RQA를 이용한 컴포넌트 테스트
 김성철(sckim@rational.com)

Agenda

- RQA를 이용한 컴포넌트 테스트
- 설계를 통한 품질 검증 (Quality by Design)
- 컴포넌트 테스트 방법
- RQA를 이용한 테스트
- 데모

Rational

초기 테스트의 필요성

- 모든 컴포넌트가 개발된 후에 테스트를 시행하는 것은 프로젝트의 위험을 증대시킨다.
- 프로젝트 개발 주기상 나중에 파악된 애러에 대한 조치는 매우 어렵고, 전체 프로젝트 일정을 연기하는 주원인이 된다.
- 특히, 아키텍처에 문제가 있는 경우에는 전체 시스템에 대한 재설계가 요구되기도 한다.

시스템 통합전에 테스트를 수행

Rational

설계를 통한 품질 검증(Quality by Design)

- Boeing 777
 - 설계된 모델을 통해 보잉 777에 대한 조립작업을 시뮬레이션 할 수 있었음.

Rational

Agenda

- RQA를 이용한 컴포넌트 테스트
- 설계를 통한 품질 검증 (Quality by Design)
- 컴포넌트 테스트 방법
- RQA를 이용한 테스트
- 데모

Rational

컴포넌트 테스트 방법

- 두 단계 전략(Two stage strategy)
 - 개별 컴포넌트의 각 인터페이스를 테스트

Rational

컴포넌트 테스트 방법 계속

- 두 단계 전략(Two stage strategy)
 - 개별 컴포넌트의 각 오퍼레이션을 테스트
 - 여러 컴포넌트들간의 오퍼레이션 호출순서에 따라서 테스트 수행

Rational

초기 테스트 수행의 장애점

- 테스트에 필요한 GUI가 없음
- 호출되는 컴포넌트가 아직 개발이 완료되지 않음
- 수작업으로 테스트 Driver와 Stub를 개발하는 것은 고통스러운 일
 - 개발 시간을 빼앗음.
 - 모든 컴포넌트의 모든 오퍼레이션에 대한 테스트는 불가능
 - 비용을 증대

Rational

초기 테스트 수행의 장애점 계속

현재 각 컴포넌트는 다음과 같은 신뢰성을 가지고 있다.

개별 컴포넌트의 신뢰성은 높지만

$$\begin{aligned}
 &.86 \times .90 \times \\
 &.92 \times .95 \times .90 \times \\
 &.91 \times .85 \\
 &= 47\%
 \end{aligned}$$

Rational

Agenda

- RQA를 이용한 컴포넌트 테스트
 - 설계를 통한 품질 검증 (Quality by Design)
 - 컴포넌트 테스트 방법
 - RQA를 이용한 테스트
 - 데모

Rational

COM Components에 대한 테스트

- 개별 컴포넌트에 대한 테스트
 - 메소드들을 호출하고 검증할 수 있는 테스트 코드
- 컴포넌트들간의 상호 호출 테스트
 - 시나리오를 구동하고 검증할 수 있는 테스트 코드
- 시뮬레이션 환경 구축
 - 테스트 드라이버와 스터브(driver and stubs)
- 배치되어 있는 컴포넌트에 대한 테스트
 - 세션 기록(Session recording)

Rational

Model로부터 테스트 코드 자동 생성

개발자

UML Model

테스트 데이터 제공

UML Model

- 테스트 코드 작성

개발자

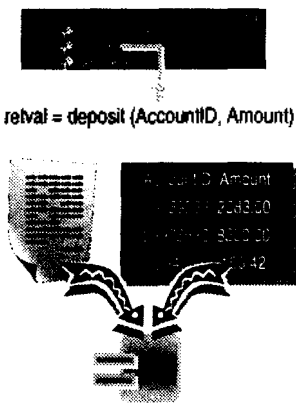
- 테스트 데이터 제공

Rational

모델을 이용한 단위 테스트(Unit Testing)

단위 테스트

- 개별 컴포넌트들의 오퍼레이션 테스트
- 컴포넌트들 간 상호작용을 통해 테스트 데이터 생성
- Rose와 RQA를 이용한 방법
 - JAR 파일 Import, COM Component 또는 model을 Import
 - Template 선택
 - 테스트할 오퍼레이션 선택
 - Dataprof 생성 또는 Import
- IDE
 - 테스트 수행



reval = deposit (AccountID, Amount)

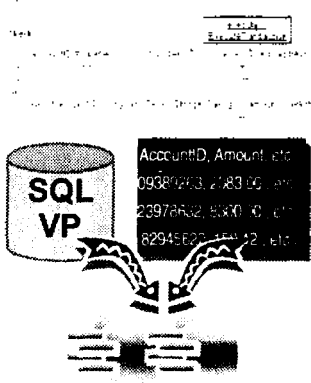
AccountID Amount
 09381213 2343.00
 23978632 4500.00
 82945E23 144.42

Rational

모델을 이용한 시나리오 테스트

Scenario 테스트

- 연결된 컴포넌트들 간의 트랜잭션을 테스트
- Interaction 다이어그램
- Rose 와 RQA 이용 방법
 - Template 선택
 - Sequence diagram 작성
 - Message 정의
 - Verification points 설정
- IDE
 - 테스트 수행



AccountID, Amount, etc
 09381213, 2343.00, etc
 23978632, 4500.00, etc
 82945E23, 144.42, etc

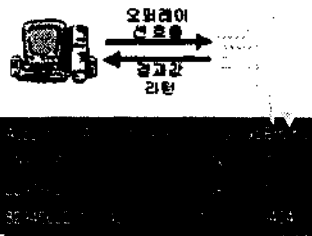
SQL VP

Rational

모델을 이용한 Stub 생성

Stub Generator

- Rose 모델을 위한 Stub을 컴포넌트 생성
- 컴포넌트별 주어진 오퍼레이션에 대해 Exception 또는 리턴값을 제공
- Rose 와 RQA 이용 방법
 - Template 선택
 - Stub로 만들 컴포넌트 선택
 - Lookup table 생성
 - Stub를 Deploy
- IDE
 - 테스트 수행



Customer (Customer)
 CustomerHome (CustomerHome)
 CustomerKey (CustomerKey)
 ExecuteTransaction (ExecuteTransaction)
 ExecuteTransactionHome (ExecuteTransactionHome)

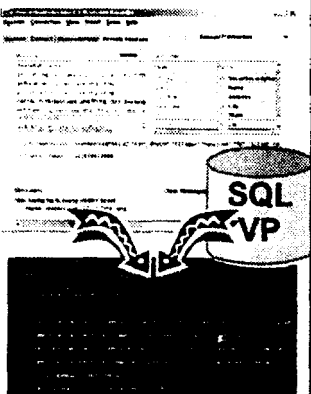
오퍼레이션 호출
 결과값 리턴

Rational

Deployed EJBs 테스트: Session Recording

EJB Session Recorder

- Enterprise JavaBeans에 대한 GUI 인터페이스 제공
 - Container에 접속
 - Beans를 인스턴스화
 - 오퍼레이션 테스트
 - Verification Points 삽입
 - Java Code 생성
 - 모델없이 테스트 가능




SQL VP

Rational

Agenda

- RQA를 이용한 컴포넌트 테스트
- 설계를 통한 품질 검증 (Quality by Design)
- 컴포넌트 테스트 방법
- RQA를 이용한 테스트
- 대모



Rational

