

웹 서비스 기반 시스템의 프로젝트 관리 기법

이중국⁰, 김동규, 백용규, 백종현
대우정보시스템
{jkleee690⁰, dkkim, ykbaek, baegih}@disc.co.kr

Project-Management Technics of System based on Web Service

JongKook Lee⁰, DongKu Kim, YonguKook Baek, JongHeon Baek
Daewoo Information Systems Co., Ltd.

요 약

본 논문에서는 웹 서비스 기반 시스템에 대한 프로젝트 관리 기법을 소개한다. 먼저 웹 서비스 기반 시스템에 서 발생할 수 있는 개발 복잡성이 무엇인지 설명한다. 발생 가능한 복잡성은 클라이언트와 서버 사이의 통신 프로토콜인 XML에서 발생한다. 또한 웹서비스의 갯수가 증가하면 WSDL의 관리도 중요해진다. 본 논문에서는 개발 복잡성을 줄이기 위한 웹 서비스 기반 시스템의 아키텍처는 무엇인지 소개한다. 또한 프로젝트 관리 도구를 사용하여 XML을 관리하는 방법을 소개한다. 마지막으로 이 관리 기법을 프로젝트에 도입하였을 때 어떤 생산성 증가 효과가 있었는지 소개한다.

1. 서 론

웹 서비스는 시스템 내부의 통합과 시스템 외부의 통합에 대한 표준으로 국내외 개발 프로젝트에서 계속 확산되고 있다. 웹 서비스를 도입하면 표준화된 시스템을 사용할 수 있으므로 향후 외부 시스템과의 통합이 쉽다. 또한 프로젝트 내부에서도 웹 서비스를 지원하는 다양한 기술들을 활용할 수 있다.

그러나 웹 서비스를 사용할 경우 웹 서비스의 특성으로 인한 새로운 개발 복잡성이 발생한다.

첫째, 웹 서비스는 XML을 기반으로 하고 있기 때문에 XML을 잘 관리해야 한다. 프로젝트의 규모가 커지면 관리해야 하는 XML의 갯수도 증가한다. 프로젝트 개발이나 유지보수시 요구사항의 변화는 XML의 변경을 발생시킨다. XML의 변경은 기존의 클라이언트/서버나 웹 기반 시스템처럼 컴포넌트나 함수만 변경하면 완료되는 것이 아니다. XML을 사용하는 클라이언트와 서버 프로그램을 모두 변경해야 한다.

둘째, 웹서비스로 제공하는 기능이 많아지면 WSDL의 갯수도 많아지며 요구사항의 변화로 인해 WSDL도 변경해야 한다. WSDL의 변경도 클라이언트 프로그램과 서버 프로그램의 변경을 요구한다.

본 논문에서는 웹 서비스 기반 프로젝트에서 이러한 개발 복잡성을 줄이기 위해 아키텍처 측면에서 해결 방법과 프로젝트 관리 측면에서 해결 방법을 제시한다.

2. 개발 복잡성을 줄이기 위한 웹 서비스 아키텍처

다음 그림은 본 논문에서 제시하는 웹 서비스 아키텍처이다.

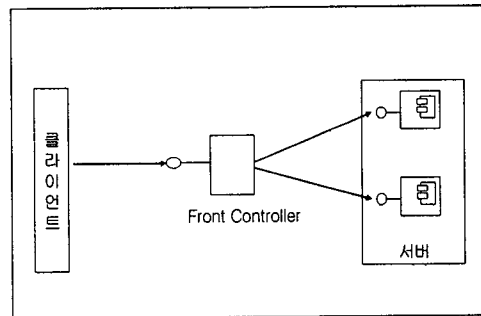


그림 2. 웹 서비스 아키텍처

프로젝트 규모가 커짐에 따라 발생하는 WSDL의 복잡성을 줄이기 위해 본 논문에서는 Front Controller라는 단 한 개의 웹서비스만을 제공할 것을 제안한다.[1] Front Controller는 클라이언트의 요청을 받는 Set(String xml) : String이라는 함수만을 가지고 있다. 클라이언트에서는 Set 함수의 입력 매개변수를 xml로 전달한다.

Front Controller는 서버 컴포넌트를 제어한다. Front Controller가 어떤 컴포넌트를 호출해야 하는지는 클라이언트는 알지 못한다. 클라이언트는 XML의 header에 자신이 원하는 기능을 대표하는 유스케이스 ID와 클라이언트 화면에서 어떤 이벤트를 발생시켰는지를 표시하는 이벤트 ID를 넣어서 Front Controller에 전달한다. Front Controller는 유스케이스 ID와 이벤트 ID를 사용하여

적절한 서버 컴포넌트를 호출한다. 어떤 서버 컴포넌트를 호출해야 하는지 판단하기 위해 Front Controller는 유스케이스 ID와 이벤트 ID, 서버 컴포넌트에 대한 매핑 파일을 가지고 있다. 매핑 파일의 구조는 다음 표와 같다.

유스케이스 ID	이벤트 ID	서버 컴포넌트	오퍼레이션
UC-LCAR-001	EV-LCAR-001AC-01	Uclcar001ac	searchCar

Front Controller는 이 매핑 파일을 사용하여 동적으로 서버 컴포넌트를 찾아서 호출한다.

Front Controller를 사용하는 이점은 웹 서비스의 구조가 간단해진다는 것과 클라이언트와 서버가 완전히 분리된다는 것이다. 클라이언트의 변경이나 서버 컴포넌트의 변경은 Front Controller의 매핑 파일만을 변경시키며 클라이언트나 서버를 다시 구현할 필요성을 줄인다.

3. 웹서비스 기반 프로젝트의 관리 방법

Front Controller를 사용하여 WSDL의 복잡성은 줄었지만 웹서비스로 전달되는 XML의 복잡성은 줄어들지 않는다. 클라이언트 개발자와 서버 개발자는 동일한 XML과 동일한 매핑 파일을 필요로 한다. 따라서 클라이언트 변경에 따른 XML의 변경은 서버 개발자에게도 알려져야 하며 프로젝트 관리자는 요구사항의 변경이 XML을 어떻게 변경시키는지 판단하고 클라이언트 개발자와 서버 개발자에게 모두 통보해야 한다.

따라서 본 프로젝트의 아키텍처에 따라 시스템을 개발하기 위해서는 설계자와 클라이언트 개발자, 서버 개발자가 반드시 분리되어야 한다. 설계자는 업무 로직뿐 아니라 XML과 매핑 파일까지 정확하게 설계하여 클라이언트 개발자와 서버 개발자에게 전달해야 한다. 또한 모든 요구사항 변경은 개발자에게 직접 전달되서는 안되며 설계자의 설계 변경을 거친 후 개발자에게 전달되어야 한다. 또한 클라이언트 개발과 서버 개발이 완료된 후에는 매핑 파일을 사용한 통합 작업이 정기적으로 이루어져야 하며 설계자는 자신이 요구한 대로 통합이 되었는지 테스트를 실시해야 한다.

설계자가 해야 할 일들이 많아지며 설계자와 개발자 사이의 커뮤니케이션이 원활해야 하기 때문에 본 프로젝트에서는 다음과 같은 프로젝트 관리 도구를 개발하였다.

첫째, 매핑 파일을 확대하여 요구사항부터 테이블까지 호출 관계를 관리할 수 있도록 요구사항 추적 기능을 제공한다.

둘째, 설계자의 설계 의도가 클라이언트와 서버 개발자에게 정확히 전달될 수 있도록 매핑 파일과 소스 코드 생성 기능을 제공한다.

3.1 요구사항 추적 기능

매핑 파일을 작성하기 위해 사용자는 어떤 요구사항에서 어떤 유스케이스가 생성되고 어떤 유스케이스에서 어떤 화면이 생성되며 그 화면의 어떤 이벤트를 발생시키면 어떤 컴포넌트를 호출하는지 알아야 한다.

클라이언트에서 서버까지 호출 관계는 시퀀스 다이어그램을 작성하면 알 수 있다. 그러나 그 관계를 관리하기 위해서는 요구사항 추적 기능이 필요하다. 설계자들은 각 화면에서 어떤 서버 컴포넌트를 호출하며 어떤 파라미터가 전달되는지에 대해 정확하게 정의해야 한다.

본 프로젝트 관리툴에서는 요구사항 추적 기능을 상세 설계까지 확대하여 설계자가 어떤 화면에서 어떤 매개변수를 전달하여 서버 컴포넌트를 호출하는지를 표현하도록 한다. 요구사항 추적 프로그램은 다음과 같다.

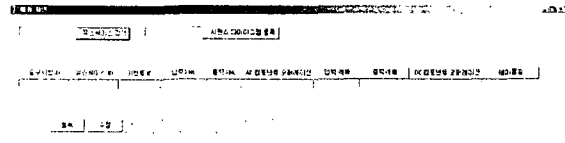


그림 2. 요구사항 추적 프로그램

요구사항 추적의 대상은 다음 표와 같다.

추적 순서	추적 대상
1	요구사항
2	유스케이스
3	화면
4	화면 이벤트
5	화면 이벤트에서 서버에 전달되는 파라미터
6	서버 컴포넌트 오퍼레이션
7	테이블

3.3 상세 설계 및 소스 코드 생성

요구사항 추적 프로그램을 통해 상세 설계를 완료한 후에는 컴포넌트 명세서 작성 프로그램을 사용하여 컴포넌트 명세서를 설계하고 소스 코드를 생성한다. 컴포넌트 명세서 작성 프로그램은 다음 그림과 같다.

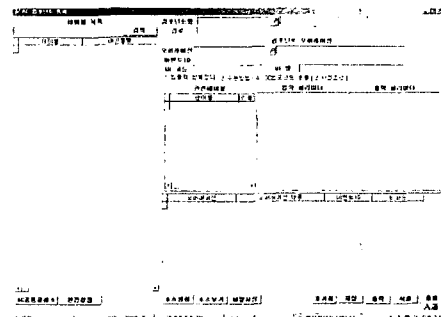


그림 3. 컴포넌트 명세서 작성 프로그램

명세서에서는 컴포넌트의 오퍼레이션 이름, 매개변수, 내부 로직을 기술한다. 컴포넌트 명세서 작성 프로그램은 명세서를 기반으로 컴포넌트에 대한 소스를 출력한다.

3.4 산출물 관리 도구

상세 설계를 통한 소스 코드 생성이 가능하도록 하기 위해서는 개념 단계의 산출물도 참고해야 한다. 본 프로젝트에서는 요구사항, 유스케이스 명세서, 화면 정의서를 데이터베이스에서 관리하고 설계자가 상세설계를 할 때 참고하도록 한다. 예를 들어 유스케이스 명세서는 다음과 같은 데이터베이스를 사용하여 관리한다.

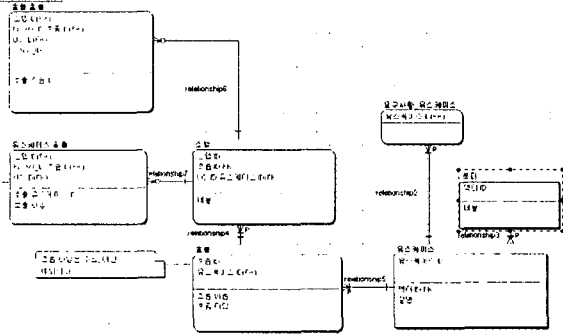


그림 1. 유스케이스 명세서 관리 ERD

명세서 관리 ERD에서는 유스케이스의 개요, 스텝, 주요 흐름, 대안 흐름, 예외흐름, 각 스텝의 흐름 호출, 각 스텝에서 유스케이스 호출을 관리한다.

3.5 일정 및 진척 관리 및 통제

일정 관리 및 진척 관리는 별도의 문서를 작성하지 않고 프로젝트 관리 도구에서 작업을 완료하면 작성자의 이름, 산출물 번호, 진행 상태가 산출물과 함께 레퍼지토리에 저장된다. 즉 다음 그림과 같이 진척 관리에 대한 모니터링이 이루어진다.

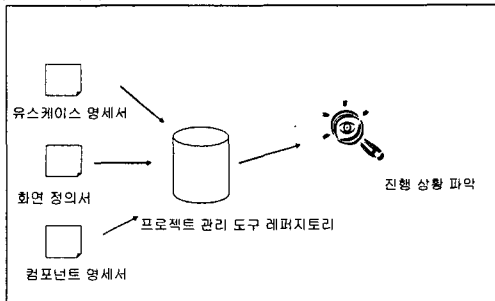


그림 4. 진척 관리 모니터링 방법

사용자가 자신의 정보를 입력하고 툴을 사용하면 툴은 사용자가 산출물을 작성할 때마다 자동적으로 사용자의 이름과 작성일자를 등록한다. 따라서 프로젝트의 진행상황을 실시간으로 모니터링할 수 있다.

3.6 기타 관리 도구

본 프로젝트 관리 도구에는 기타 품질 평가, 오류 추적 특색그래프가 포함되어 있다.

4. 사례

본 논문에서 제시한 프로젝트 관리 도구의 장점은 다음과 같다.

첫째, 이 프로젝트 관리 도구는 설계자들이 어떤 항목을 설계해야 하는지에 대한 가이드를 제시한다.

WebService를 사용할 경우 클라이언트와 서버사이에는 XML로 통신하고 컴포넌트를 끼리는 오퍼레이션과 객체를 사용하여 통신한다. 설계자는 XML과 객체기반 파라미터를 정확하게 정의해야 한다. 상세 설계 프로그램은 XML과 컴포넌트 명세서를 어떻게 입력해야 하는지에 대해 가이드한다. 또한 설계시 참조해야 하는 화면 정의서, 유스케이스 명세서, 요구사항 정의서, 테이블 정의서를 틀에서 쉽게 참조할 수 있기 때문에 유용하다.

둘째, 설계 변경이나 요구사항이 추가될 경우 어떤 산출물이 추가되어야 하는지를 쉽게 알 수 있다.

프로젝트 관리 도구에 들어 있는 요구사항 추적 기능을 사용하여 관련 산출물을 추적할 수 있기 때문에 빠지거나 변경되어야 할 산출물이 무엇인지 알 수 있다.

현재 제시된 아키텍처와 툴 사용 결과, 툴을 사용하지 않고 웹서비스 프로젝트를 진행할 경우에 대비하여 상세 설계시 설계시간이 약 2/3 정도 줄어든다.

6. 결론

본 논문에서는 웹 서비스 기반 시스템의 개발 복잡성을 줄이기 위한 아키텍처와 프로젝트 관리 기법을 소개하였다. 향후 이 관리 도구는 각종 관리 기능을 추가하고 기존의 프로젝트 일정 관리 도구, 모델링 도구, 개발도구와 연결할 예정이다.

참고문헌

[1] Fowler, M. Patterns of Enterprise Application Architecture. Addison-Wesley, 2002