The Journal of Korean
Institute of CALS/EC
Vol 6, No. 3, Dec. 2001

소규모의 웹 응용 개발을 위한 역할 분담

이우진*, 조용선*, 정기원**

Individual Roles for Small-sized Web Application Development

Woo-Jin Lee, Yong-Sun Cho, Ki-Won Chong

Abstract

This paper proposes the individual roles for developing small web application systems based on the Client/Server architecture with the activities and artifacts of each role and cooperation. The roles of Web Server part (i.e. User Interface Designer, Web Designer, HTML Writer), the roles of Application Server part (i.e. Domain Expert, Application Developer, Tester) and the roles of DB Server part (i.e. Database Administrator, Data Designer) are described. Furthermore, the role of the Development Leader that participates in development and manages all works in project and finds the solutions of problems in project, is also discussed. The Domain Expert analyzes the domain of the application in order to send the artifacts to the Application Developer. Then the Application Developer analyzes, designs and implements the application based on the artifacts of the Domain Expert and integrates the implemented program modules. Roles are related each other in this way, and cooperate until the application development is completed. Finally, we analyzed and compared these roles with the roles of RUP (Rational Unified Process) and Web Wave. Suggested roles in this paper turned out to be efficient compared to the roles of the existing large-scale methodology.

Key Word : role, small web application, architecture

* 숭실대학교 대학원 컴퓨터학과
** 숭실대학교 컴퓨터학부 교수

1. 서론

정보기술과 인터넷의 발전에 따라 웹 기반 시스템의 구축 수요가 증가하고 있고, 이에 따라 중소, 대기업들이 웹 기반의 응용을 개발하는 사례가 증가하고 있다. 이런 상황에서 중,대규모의 웹 응용을 효율적으로 개발하기위한 방법들은 꾸준히 나오고 있으나 소규모의 웹 응용 개발을 위한 방법들은 제시된 것들이 거의 없어서 필요성이 크다. 대규모의 개발을 위한 내용을 모두 포함하고 있는 기존의 방법론들은 소규모의 개발에 적용하기에 불필요한 내용들도 있고, 적용하기 위해 커스터마이징을 하는 데에 많은 노력이 필요하다. 따라서 본 논문에서는 소규모의 웹 응용 개발을 효율적으로 수행하기 위한 방법으로 개발자간의 역할 분담을 제시한다. 응용을 개발하는데 있어서 역할 분담은 필수적이므로, 역할 분담을 적절하게 한다면 응용을 보다 효율적으로 개발할 수 있을 것이다.

2. 관련 연구

본 논문에서 제시하는 역할 분담이 소규모의 웹 응용 개발에 효율적인지를 평가하기 위해서는 기존의 방법론에서 제시하는 역할 분담과의 비교가 필요하다. 따라서 2장에서는 본 논문에서 제시하는 역할 분담과의 비교 대상이 되는 RUP에서의 역할 분담과 Web Wave에서의 역할 분담에 대해서 알아본다. 또한 본 논문을 작성하는데 많은 지침을 제공해 준 'Role of Developers as Part of a Software Process Model'이라는 논문에서 Kari Kivisto가 제시한 역할분담에 대해서도 알아본다.

2.1. RUP에서의 역할 분담[9]

RUP에서는 역할들을 Analysts, Developers, Testers, Managers의 네 가지 타입으로 구분한다. 또한 네 가지 타입에는 속하지 않지만 필요한 역할들을 Others 그룹으로 분류한다.

Analysts 그룹에서는 비즈니스에 대한 분석, 설계, 모델링을 위한 역할과 시스템의 요구사항을 분석, 설계하기 위한 역할, 그리고 사용자 인터페이스의 설계를 위한 역할을 제시한다. Developers 그룹에서는 아키텍처 설계를 위한 역할, 시스템을 설계, 구현하기 위한 역할 및 데이터베이스의 설계를 위한 역할 등을 제시한다. Testers 그룹에서는 구현된 시스템을 테스트하기 위해 필요한 역할들을 제시한다. Managers 그룹에서는 변경, 형상관리, 배포 및 프로젝트 전반에 걸친 관리를 위한 역할들을 제시한다.

이 밖에 네 가지 그룹에는 속하지 않지만 프로젝트를 수행하는데 있어서 필요하다고 여겨지는 역할들을 Others 그룹으로 분류하여 제시하는데, 이 그룹에는 제품 교육자, 사용자 지침서 작성자, 그래픽 디자이너, 툴 전문가 등의 프로젝트 지원을 위한 역할들이 제시되어 있다.

다음의 목록은 RUP에서의 역할을 정리한 것이다.

Analysts

- Business-Process Analyst
- Business Designer
- Business-Model Reviewer
- Requirements Reviewer
- System Analyst
- Use-Case Specifier
- User-Interface Designer

Developers

- Architect
- Architecture Reviewer
- Capsule Designer
- Code Reviewer
- Database Designer
- Design Reviewer
- Designer
- Implementer
- Integrator

Testers

- Test Designer
- Tester

Managers

- Change Control Manager
- Configuration Manager
- Deployment Manager
- Process Engineer
- Project Manager
- Project Reviewer

Others

- Any Worker
- Course Developer
- Graphic Artist
- Stakeholder
- System Administrator
- Technical Writer
- Tool Specialist

RUP에서의 역할 분담은 어떤 응용의 개발에서도 적용할 수 있도록 범용적이며, 대규모의 프로젝트에 적용하기에 적합하다. 또한 프로젝트의 전반에 걸친 모든 역할이 제시되어 있는 것을 알 수 있다. 그러나 RUP에서의 역할 분담은 소규모의 웹 응용 개발에 적용하기에는 수가 너무 많고, 웹 응용만이 아닌 다른 모든 응용에서도 적용할 수 있도록 범용적이기에 웹 응용에 적용하기 위해서는 커스터마이징을 해야 할 필요가 있다.

RUP에서의 역할 분담은 본 논문에서 응용을 개발하는데 있어서 꼭 필요한 역할과 작업을 추출해내기 위한 지침이 되었다. 또한 RUP는 응용 개발에 많이 사용되는 프로세스라는 점에서, RUP에서의 역할 분담은 본 논문에서의 역할 분담과의 비교대상이 되었다.

2.2. Web Wave에서의 역할 분담[3]

WebWave에서 정의한 역할은 Project Manager, Principals, Technical Manager, Creative Team, Technical Team, Content Management Team이 있다. Web Wave에서의 각 역할들은 관리자 역할을 제외하고는 RUP에서와 같이 개개인이 아니라 모두 팀으로 구성되어 있다. 따라서 프로세스에서 특정한 작업을 수행하기 위한 역할이 구체적으로 제시되어 있는 것이 아니라 전체 프로세스를 통해서 각 역할 팀에 대한 작업을 제시하고, 각각의 팀에서는 그 작업에 알맞은 사람을 뽑아서 수행하도록 제시한다. Web Wave에서 제시하는 역할과 각 역할에 대한 설명을 정리하면 <표1>과 같다.

<표 1> Web Wave에서의 역할 및 각 역할에 대한 설명

[illegible]	[illegible]
Project Manager	프로젝트 전반에 걸쳐 프로젝트를 관리하는 역할
Principals	official group과 interested parties로 구성 official group은 2-10명 정도의 business managers, departmental managers, technical managers로 구성. interested parties는 인원수에 제한 없음 Management 결정사항을 만들고 프로젝트에 대한 지원을 통해서 사용자의 견해에 영향을 줌 Private Web이 성공적으로 수행될 수 있도록 만들 책임이 있음
Technical Manager	Technical Team을 관리하는 역할
Creative Team	아키텍처 설계, Home Page 설계 등의 creative한 일들을 처리 인원수에는 제한 없음
Technical Team	프로젝트를 수행하는데 있어서 기술적인 부분들을 처리 인원수에는 제한 없음
Content Management Team	웹의 컨텐츠와 관련된 일을 처리 인원수에 제한 없음

Web Wave에서의 역할 분담은 웹 응용 개발을 위한 것이며 RUP와 마찬가지로 대규모의 개발에 적용하기에 적합하다. 그러나 역할이 팀 단위로 되어 있어서 개발 인원이 적은 소규모의 개발에는 적합하지 않다.

Web Wave에서의 역할 분담은 본 논문에서의 웹 응용 개발을 위한 작업에 지침이 되었고, Web Wave는 웹 응용 개발을 위한 프로세스라는 점에서, Web Wave에서의 역할 분담은 본 논문에서의 역할 분담과의 비교대상이 되었다.

2.3. Kari Kivisto가 제시한 역할 분담[6]

Kari Kivisto는 "Roles of Developers as Part of a Software Process Model" 이라는 논문에서 IT 기업들이 객체 지향 client/server 응용 개발 프로세스 모델을 적용하여 응용을 개발하고자 할 경우의 역할 분담을 제시하고, 각 역할의 프로세스 상에서의 작업 및 산출물 등을 제시하였다.

Kari Kivisto가 제시하는 역할로는 Application Developer, Business Object Developer, Database Developer, User /End-User, Team Leader, Project Manager, Quality Assurancer/Tester, Expert가 있다. 각 역할에 대한 설명을 간단히 정리하면 <표 2>와 같다.

<표 2> Kari Kivisto가 제시하는 역할 및 각 역할에 대한 설명

[illegible]	[illegible]
Project Manager	Team Leader와 관계를 갖고 프로젝트 전반에 걸쳐 프로젝트를 관리하기 위한 역할
Team Leader	Team을 관리하고, 응용의 아키텍처를 설계하기 위한 역할
Application Developer	응용을 분석, 설계, 구현하고, 인터페이스를 설계 및 구현하며 보고서를 작성하기 위한 역할

Business Object Developer	재사용 가능한 비즈니스 객체의 설계, 구현, 유지보수를 위한 역할
Database Developer	데이터 모델링, 데이터베이스 설계 등의 데이터베이스와 관련된 모든 작업의 수행을 위한 역할
User/End-User	사용자를 말함. 시스템 정의, 분석, 테스트에 참여하고, Application Developers와 협력하여 사용자 인터페이스를 설계하기 위한 역할
Expert	도메인 전문가와 기술 전문가로 구성. 도메인 전문가는 Business Object Developers와 함께 작업을 수행. 기술 전문가는 아키텍처와 관련된 작업을 수행. installation 명령을 검사하고 installation을 테스트하며 시스템 테스트를 수행하기 위한 역할
Quality Assurancer	프로젝트의 품질과 관련된 작업을 수행하기 위한 역할
Tester	테스트를 수행하기 위한 역할

Kari Kivisto가 제시한 역할은 소규모의 웹 응용 개발에 적용하기에 적합하며 프로젝트의 전반에 걸친 모든 역할이 제시되어 있다. 그러나 개발에 직접 참여하는 역할을 살펴보면, 하나의 역할에 할당되어 있는 작업이 너무 많다. 또한 Kari Kivisto는 객체지향 client/server 개발 프로세스 모델에 초점을 맞추어 역할을 제시하였다. 반면, 본 논문에서 제시한 역할은 객체지향 뿐만이 아니라 여러 개발 방법에 적용할 수 있다.

Kari Kivisto는 소규모의 웹 응용 개발에 적용하기에 적합한 역할 분담을 제시하여 본 논문에 가장 큰 지침이 되었다.

3. 개발자 간의 역할 분담

이 장에서는 본 논문에서 제시하는 역할을 적용하기 위한 소규모의 개발에 대한 특징과 웹 응용의 개발에 있어서 가장 많이 사용되는 Client/Server 아키텍처에 대하여 살펴보고, 이를 바탕으로 개발자들의 역할을 Web Server 부분, Application Server 부분, DB Server 부분으로 나누어서 정의했다. 본 논문은 소규모 개발을 위한 역할 분담을 제시하는 것이므로, 개발에 필요한 역할의 종류를 간소화 하였다. 또한 제시하는 역할들은 한 사람이 하나의 역할만을 수행해야 하는 것이 아니라 여러 역할을 수행할 수 있고, 여러 사람이 하나의 역할을 함께 수행할 수도 있다. 이는 프로젝트와 조직의 특성에 따라 달라질 것이다.

본 논문에서는 제시한 각 역할들에 대하여, 서로 산출물을 주고 받는 협력 관계에 있는 역할들은 서로의 작업에 있어서 필요시마다 함께 산출물을 검토하는 것을 기본적인 작업으로 제시하였다.

3.1 소규모 개발의 특징

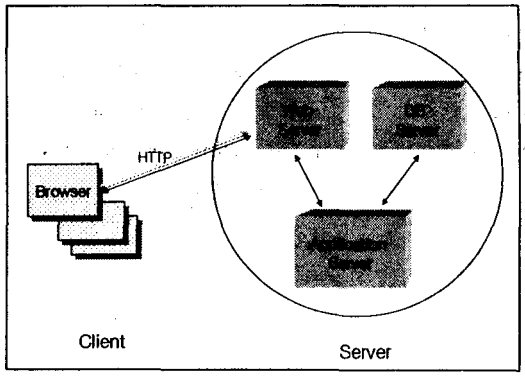
본 논문은 소규모의 개발에 초점을 두고 웹 응용의 역할 분담을 제시한다. 따라서 소규모의 개발이 어떠한 특징을 가지고 있는가를 잘 알아야 본 논문에서의 역할 분담을 제대로 적용할 수 있을 것이다. 본 논문에서 말하고 있는 소규모 개발의 특징은 다음과 같다.

- ◆ 비교적 덜 엄격한 요구사항을 기반으로 한다. [7][10]
- ◆ 상대적으로 덜 복잡하다. [7][10]
- ◆ 개발자들이 도메인에 대하여 잘 이해하고 있다. [7]
- ◆ 모든 개발자들이 동등하게 개발에 필요한 기술을 충분히 갖추고 있다. [7]
- ◆ 피드백 시간이 짧다. [7]
- ◆ 개발에 참여하는 인원의 수가 적다. [10]
- ◆ 비용곡선의 증가율이 큰 복잡한 기술을 도입하지 않는다. [7]

3.2. 웹 응용의 아키텍처

웹 응용은 웹 사이트로부터 생겨났다. 웹 응용은 사용자가 비즈니스 로직을 포함하고 서버에 있는 비즈니스의 상태를 변경할 수 있도록 함으로써 웹 사이트를 확장한다[1]. 이러한 웹 응용의 정의는 웹 응용을 구성하는 세 가지의 아키텍처 컴포넌트를 포함한다. 세 가지 컴포넌트는 바로 클라이언트 브라우저, 웹 서버, 응용 서버이다. 그리고 대부분의 웹 응용은 DB 서버를 사용한다. 이러한 이유로 우리는 웹 응용을 client/ server 소프트웨어 시스템으로 정의할 수 있다[4][5].

따라서 본 논문에서도 client/server 아키텍처[4][8]를 기반으로 웹 응용을 개발하고자 할 때의 역할 분담에 대해서 이야기하고자 한다. 본 논문에서 적용하고자 하는 client/server 아키텍처는 <그림 1>과 같은 구조로 되어 있다.



<그림 1> 웹 응용을 위한 client/server 아키텍처

클라이언트측을 살펴보면, 사용자들은 Microsoft Internet Explorer, Netscape Navigator와 같은 브라우저만 있으면 언제, 어디서나 인터넷을 통하여 서버에 접속하여 응용을 실행할 수 있다.

단순히 보여주는 것 이외의 모든 일은 서버측에서 수행한다. 서버측을 살펴보면, 사용자들이 브라우저를 통해 볼 수 있도록 HTML, XML 등으로 작성한 인터페이스 부분을 실행할 수 있도록 하는 Web Server, 그리고 JSP, EJB, Java Servlet 등과 같이 응용에서 비즈니스 로직을 담당하는 부분을 실행할 수 있도록 해주는 Application Server, 그리고 응용을 수행하는데 필요한 데이터를 저장하기 위한 DB Server 등이 있다. 대부분의 웹 응용은 데이터 중심의 응용이어서 DB가 중요한 부분을 차지한다.

각 서버들의 역할을 정리해 보면 Web Server는 사용자가 브라우저를 통해서 응용과 상호작용을 할 수 있도록 HTML, XML 등과 같은 웹 페이지를 보여주는 역할을

하므로 Interface 부분으로 볼 수 있고, Application Server는 실제로 응용이 수행되도록 하는 로직을 실행할 수 있도록 하므로 Business Logic 부분으로 볼 수 있다. 그리고 DB Server는 응용에서 필요한 모든 데이터를 저장하고 관리하는 부분이므로 Data Management 부분으로 볼 수 있다.

3.3. Web Server 부분을 위해 필요한 역할

Web Server 부분은 응용과 사용자 간의 interface를 담당하는 부분이기 때문에 사용자 인터페이스 부분을 개발하기 위한 역할들이 필요하다. 또한 작업을 감독하고 문제를 해결하기 위한 역할이 필요하다.

개발 책임자(Development Leader DL) - 프로젝트의 수행에 있어서 모든 역할 수행자들의 작업을 감독하고 관리하며, 프로젝트를 수행하는 과정에서 발견된 문제점에 대한 해결 방안을 찾는다.

사용자 인터페이스 설계자(User Interface Designer - UI) - 사용자 인터페이스를 설계한다. 산출물로는 사용자 인터페이스 설계서가 있다.

웹 디자이너(Web Designer - WD) - 사용자 인터페이스 설계자로부터 받은 인터페이스 설계서를 바탕으로 웹 사이트의 구성에 맞도록 사용자 인터페이스를 디자인하여 이미지 파일의 형태로 만든다. 또한 웹 페이지에서 사용하는 이미지 및 멀티미디어 등을 작성한다. 산출물로는 사용자 인터페이스 디자인과 이미지 및 멀티미디어 파일 등이 있다.

웹 페이지 작성자(HTML Writer - HW) - 웹 디자이너가 디자인한 인터페이스를 바탕으로 사용자 인터페이스를 HTML 페이지로 작성한다. 또한 프로그램에 필요한 추가적인 HTML 페이지를 작성한다. 산출물로는 HTML 페이지가 있다.

3.4. Application Server 부분을 위해 필요한 역할

Application Server 부분은 business logic을 담당하는 부분이기 때문에 응용을 위한 분석, 설계, 구현, 테스트를 수행하는 역할들이 필요하다. 또한 작업을 감독하고 문제를 해결하기 위한 역할이 필요하다.

개발 책임자(Development Leader DL) - 프로젝트의 수행에 있어서 모든 역할 수행자들의 작업을 감독하고 관리하며, 프로젝트를 수행하는 과정에서 발견된 문제점에 대한 해결 방안을 찾는다. 또한 응용의 분석, 설계 및 구현에 참여한다.

도메인 전문가(Domain Expert - DE) - 도메인을 분석하여 도메인 분석서를 만든다. 응용의 분석, 설계 시 응용 개발자가 작성한 응용의 분석서를 함께 검토한다. 산출물로는 도메인 분석서(비즈니스 요구사항 명세서, 응용 요구사항 명세서)가 있다.

응용 개발자(Application Developer -

AD) - 응용을 개발하기 위한 분석, 설계를 수행[6]하고, 이를 바탕으로 로직을 작성하여 웹 페이지 작성자가 만든 HTML 페이지와 연결한다. 구현한 것들은 단위 테스트를 통해서 결점을 고치고, 통합한다. 또한 도메인 전문가가 작성한 도메인 분석서를 함께 검토한다. 산출물로는 응용의 분석, 설계서 및 소스 코드가 있다.

테스터(Tester - TE) - 응용 개발자가 구현한 모듈을 넘겨받아 통합 테스트를 수행한다. 또한, 개발 시 응용 개발자와 함께 소스코드를 검토한다. 결과물로는 테스트 보고서를 작성한다.

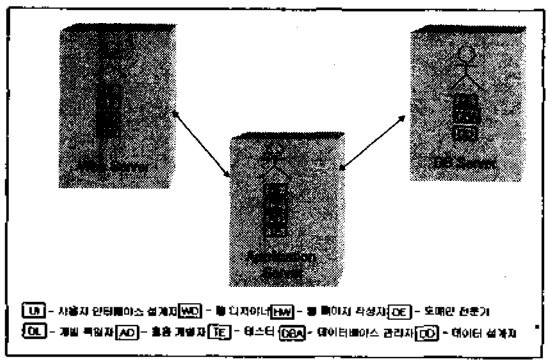
3.5. DB Server 부분을 위해 필요한 역할

DB Server 부분은 데이터를 관리하기 위한 부분이므로, 데이터 및 데이터베이스와 관련된 작업을 수행하는 역할들이 필요하다. 또한 작업을 감독하고 문제를 해결하기 위한 역할이 필요하다.

개발 책임자(Development Leader DL) - 프로젝트의 수행에 있어서 모든 역할 수행자들의 작업을 감독하고 관리하며, 프로젝트를 수행하는 과정에서 발견된 문제점에 대한 해결 방안을 찾는다.

데이터베이스 관리자(Database Administrator - DBA) - 데이터베이스의 성능을 최적화하는 일을 수행하며, 데이터베이스의 물리적인 설계 및 생성, 데이터베이스의 보안, 권한부여 등을 담당한다. 산출물로는 물리적인 데이터베이스 모델 설계서, 데이터베이스와 테이블 생성코드(DDL)가 있다.

데이터 설계자(Data Designer - DD) - 데이터베이스의 논리적인 설계를 담당하는데, 응용에 필요한 여러 개체들과, 그들 간의 관계 등을 정의한다. 또한, 프로그램에서 필요한 질의문(SQL)을 작성하여 테스트를 수행한 후 응용 개발자에게 넘겨준다. 응용의 분석, 설계 시 응용 개발자가 작성한 아키텍쳐와 설계서를 함께 검토한다. 산출물로는 논리적인 데이터베이스 모델 설계서(테이블 설계서), 데이터베이스 질의문이 있다.



<그림 2> client/server 아키텍처 기반의 역할 분담

<그림 2>는 3.1절에서 제시한 웹 응용을 위한 client/server 아키텍쳐에, 본 논문에서 제시한 역할을 적용시킨 것이다.

3.6. 각 개발 단계에서의 역할별 작업과 산출물

이 절에서는 개발의 분석, 설계, 구현 및 테스트 단계에서의 역할별 작업과 산출물에 대해서 말하고자 한다. 본 논문에서는 편의상 분석, 설계, 구현 및 테스트가 순서대로 이루어지는 것처럼 이야기 하지만, 실제로는 분석, 설계, 구현 및 테스트 각 단계의 많은 작업들이 개발 모듈별로 병행해서 이루어짐을 미리 밝혀둔다.

3.6.1. 분석 및 설계 단계

분석 및 설계 단계에서는 개발 책임자, 도메인 전문가, 사용자 인터페이스 설계자, 웹 디자이너, 데이터베이스 관리자, 데이터 설계자, 응용 개발자 역할의 수행자들이 사용자 인터페이스 설계, 데이터베이스의 설계, 응용의 분석 및 설계와 관련된 작업들을 수행한다.

<표 3>은 분석 및 설계 단계에서의 각 역할별 작업과 산출물을 정리한 것이다.

<표 3> 분석 및 설계 단계

역할	작업	산출물
개발 책임자 (DL)	응용에 대한 분석, 설계를 감독한다.	작업 수행 보고서
	프로젝트 계획서를 작성 및 수정한다.	프로젝트 계획서
	발견된 문제점에 대한 해결방안을 찾는다.	문제점 해결 보고서
	응용 개발자와 함께 응용에 대한 분석 및 설계를 수행한다.	응용 분석서, 응용 설계서
도메인 전문가 (DE)	도메인을 분석하여 비즈니스 요구사항 명세와 응용 요구사항 명세를 작성한다.	도메인 분석서 (비즈니스 요구사항 명세서, 응용 요구사항 명세서 등)
	응용 개발자가 작성한 분석서를 함께 검토한다.	
사용자 인터페이스 설계자 (UI)	사용자 인터페이스를 정의하여 사용자 인터페이스 설계서를 만든다.	사용자 인터페이스 설계서
웹 디자이너 (WD)	사용자 인터페이스, 이미지 및 멀티미디어를 디자인한다.	이미지 및 멀티미디어 파일, 사용자 인터페이스 디자인(이미지 파일 형태)
데이터 설계자 (DD)	응용에 필요한 테이블과 그들간의 관계 등을 설계(데이터베이스의 논리적인 설계)한다.	테이블 설계서 (ER 다이어그램), 각 테이블과 데이터에 대한 정의서
	응용 개발자가 작성한 아키텍처와 설계서를 함께 검토한다.	
데이터베이스 관리자 (DBA)	데이터베이스의 물리적인 설계를 한다.	데이터베이스의 물리적인 설계서, DDL
응용 개발자 (AD)	응용 요구사항을 분석하고, 응용을 설계한다.	응용 분석서(아키텍처 설계서 등), 응용 설계서
	도메인 전문가가 작성한 요구사항 명세를 함께 검토한다.	

3.6.2. 구현 단계

구현 단계에서는 개발 책임자, 웹 페이지 작성자, 응용 개발자, 데이터 설계자, 테스터 역할의 수행자들이 응용의 구현과 관련된 작업들을 수행한다.

구현 단계에서 각 역할별 작업과 산출물은 <표 4>와 같다.

<표 4> 구현 단계

역할	작업	산출물
개발 책임자 (DL)	응용의 구현에 대한 작업을 관리한다.	작업 수행 보고서
	발견된 문제점에 대한 해결방안을 찾는다.	문제점 해결 보고서
	응용 개발자와 함께 응용을 구현한다.	소스코드
웹 페이지 작성자 (HW)	웹 디자이너가 작성한 인터페이스 디자인을 바탕으로 사용자 인터페이스를 HTML 페이지로 작성한다.	HTML 페이지
	프로그램에 필요한 추가적인 HTML 페이지를 작성한다.	
	응용 개발자와 함께 응용을 구현한다.	소스코드
응용 개발자 (AD)	비즈니스 로직을 작성하여 각 모듈별 단위 테스트를 수행하고 통합한다.	소스코드
데이터베이스 관리자 (DBA)	데이타베이스 및 테이블을 생성한다.	
데이터 설계자 (DD)	응용의 구현에 필요한 질의문을 작성하여 테스트한다.	질의문
테스터 (TE)	응용 개발자가 작성한 소스코드를 검토한다.	

3.6.3. 테스트 단계

테스트 단계에서는 개발 책임자, 테스터 역할의 수행자가 개발된 응용을 테스트하는 작업을 수행한다.

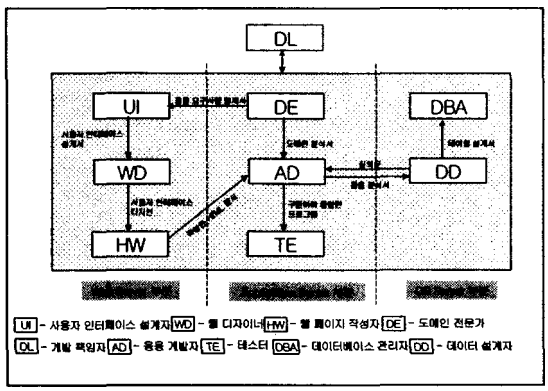
<표 5>는 테스트 단계에서 각 역할별 작업과 산출물을 정리한 것이다.

<표 5> 테스트 단계

역할	작업	산출물
개발 책임자 (DL)	응용에 대한 테스트를 감독한다.	작업 수행 보고서
테스터 (TE)	응용에 대한 통합 테스트를 수행한다.	테스트 보고서

3.7. 역할 간의 협력 관계

<그림 3>은 본 논문에서 제시한 역할 간의 협력 관계를 보여준다. 그림에서 박스는 각 역할들을 나타내며 실선은 서로 협력 관계에 있는 역할들을 연결한 것이다. 또한 화살표는 산출물의 이동을 나타낸다.



<그림 3> 역할 간의 협력 관계

먼저 Web Server 부분을 살펴보면, 사용자 인터페이스 설계자(UI)가 도메인 전문가(DE)로부터 받은 응용에 대한 요구사항 명세서를 바탕으로 인터페이스 설계서를 작성하여 웹 디자이너(WD)에게 넘겨주면 웹 디자이너(WD)는 인터페이스 설계서를 바탕으로 웹 사이트의 전체적인 구성에 맞도록 인터페이스를 디자인하여 이미지 파일의 형태로 웹 페이지 작성자(HW)에게 넘겨준다. 웹 페이지 작성자(HW)는 웹 디자이너(WD)가 디자인한 인터페이스를 바탕으로 HTML 페이지를 작성하고, 추가적으로 필요한 HTML 페이지를 작성하여 응용 개발자(AD)에게 넘겨준다.

응용의 로직 작성과 관련된 모든 일은 Application Server 부분의 역할 수행자들의 몫이다. 도메인 전문가(DE)는 만들고자 하는 응용에 대한 도메인을 분석하여 도메인 분석서를 작성하고, 이를 바탕으로 응용 요구사항 명세서를 작성하여 응용 개발자(AD)에게 넘겨준다. 응용 개발자(AD)는 도메인 전문가(DE)로부터 받은 응용 요구사항 명세서를 바탕으로 응용에 대한 분석 및 설계를 수행하고, 웹 페이지 작성자(HW)로부터 받은 HTML 페이지와 데이터 설계자(DD)로부터 받은 데이터베이스 질의문을 바탕으로 응용을 구현하고, 구현한 응용을 단위 테스트 하여 통합한 후 테스터(TE)에게 넘긴다. 또한 사용자 인터페이스 설계를 위하여 응용에 대한 분석서를 사용자 인터페이스 설계자(UI)에게 넘겨준다. 구현된 응용을 넘겨 받은 테스터(TE)는 통합 테스트를 수행한다.

DB Server 부분에서는, 데이터 설계자(DD)가 응용에서 사용되는 테이블들에 대한 설계서를 넘겨주면 데이터베이스 관리자(DBA)는 데이터베이스에 이를 생성시키고, 데이터베이스와 관련된 일들을 처리한다. 데이터 설계자(DD)는 또한 응용에서 사용하는 데이터베이스 질의문들을 작성하여 이를 테스트 한 후 응용 개발자(AD)에게 넘겨준다.

개발 책임자는 이러한 모든 역할 수행자들의 작업을 감독하고, 프로젝트를 수행하면서 발생하는 문제점에 대한 해결방안을 찾고, 응용 개발자와 함께 응용의 분석, 설계 및 개발에 참여한다. 각 역할 수행자들은 응용의 개발이 완료될 때까지 끊임없이 서로 의사소통을 하고 산출물을 피드백하며 협력한다.

4. RUP, Web Wave의 개발 역할과의 비교 분석

4장에서는 본 논문에서 제시한 역할이 소규모의 웹 응용 개발에 적합한 것인지를 평가하기 위하여 RUP, Web Wave에서의 역할 분담과 비교하여 분석한다. 또한 비교 분석을 통하여 본 논문에서 제시하는 역할과 RUP, Web Wave에서의 역할 분담의 장단점을 도출하여 본 논문에서 제시하는 역할이 소규모의 웹 응용 개발에 적합한 것임을 보일 것이다.

<표 6>은 본 논문에서 제시한 아키텍쳐 기반의 역할 분담과 RUP, Web Wave에서의 개발 부분의 역할 분담을 비교한 것이다.

<표 6> RUP, Web Wave의 개발 역할과의 비교

역할	작업	RUP 에서의 유사한 역할	Web Wave 에서의 유사한 역할
UI	•사용자 인터페이스 설계 •응용 개발자가 작성한 아키텍처와 설계서 검토	User-Interface Designer, Architecture Reviewer, Design Reviewer	Creative Team
WD	•사용자 인터페이스 다자인 •이미지 및 멀티미디어 작성	Graphic Artist	Creative Team
HW	•웹 디자이너가 작성한 인터페이스 다자인을 바탕으로 HTML 페이지 작성 •응용에 필요한 웹 페이지 작성		Technical Team
DE	•도메인을 분석하여 도메인 분석서 및 응용 요구사항 명세서 작성 •응용 개발자가 작성한 분석서 검토	Business-Process Analyst, Business Designer, System Analyst, Requirements specifier	Principals, Project Manager
AD	•도메인 전문가가 작성한 요구사항 명세 검토 •응용의 분석, 설계 및 구현	Business-Model Reviewer, Requirements Reviewer,	Technical Team, Technical Manager
	수행 •구현한 것들을 단위 테스트 •구현 모듈 통합	Software Architect, Designer, Implementer, Integrator	
TE	•개발 시 응용 개발자와 함께 소스코드 검토 •응용 개발자가 구현한 모듈을 넘겨받아 통합 테스트	Code Reviewer, Test Designer, Tester	Technical / Creative / Content Management/ Team, PM
DBA	• DB의 성능을 최적화 , 물리적 설계 • DB의 보안, 권한부여 등을 담당	Database Designer	
DD	• DB의 논리적인 설계 담당 •개체들과, 관계 등을 정의 •데이터베이스 질의문 작성 및 테스트		

RUP는 대규모 프로젝트를 지원할 수 있도록 만들어졌기 때문에 프로젝트 전반에 걸친 모든 역할들이 포함되어 있으며, 개발 부분만을 보아도 17개의 역할이 존재한다. 반면, 본 논문에서 제시한 역할은 소규모의 웹 응용 개발을 위한 역할이며, 개발 활동을 중심으로 구성한 역할이기때문에 관리자 부분 및 지원 부분은 포함되지 않았다. 또한 RUP에서는 검토를 수행하기 위한 역할을 따로 두었으나, Web Wave나 본 논문에

서는 검토를 수행하기는 하나 역할을 따로 나누지는 않았으며, 본 논문에서 제시하는 방법에서는 pair-review 기법을 사용하여, 산출물을 직접 주고 받는 역할들에게 서로 상호 검토를 위한 작업을 할당하였다.

본 논문에서 제시한 역할은 아키텍처를 바탕으로 구성하였기 때문에 역할이 시스템의 구조와 자연스럽게 조화되며, 역할 수행자 간의 의사소통의 라인을 쉽게 알 수가 있다. 또한 아키텍처가 변경된다면 그에 따라 역할 분담도 유연하게 조절할 수 있다.

본 논문에서 제시한 역할 중에는 질의문을 작성하는 역할이 따로 있다. 웹 응용은 대부분이 데이터 중심이기 때문에 빈번히 데이터베이스와 데이터를 주고 받는다. 따라서 질의문이 프로그램에서 주요한 부분이 되므로 질의문을 작성하는 역할을 따로 두어 프로그램의 작성 및 수정이 효율적이 되도록 하였다.

RUP는 모든 개발경로에 적용할 수 있는 범용적인 방법론이다. 그러나 본 논문에서 제시한 역할은 웹 응용의 개발을 위한 역할이다. Web Wave는 웹 기반의 개발 프로세스이기때문에 웹 응용을 개발하기에 적합한 역할을 제시하나 대규모의 응용을 개발하기 위한 내용들이 모두 포함되어 있으며 관리자를 제외하고는 역할이 팀 단위로 되어 있어서 소규모의 개발에 적용하기에는 너무 포괄적이고, DB와 관련된 작업의 수행을 위한 역할이 뚜렷하지 않다.

본 논문에서 제시한 역할은 소규모의 개발의 위해서 축소시켰기 때문에 응용 개발자가 응용에 대한 분석, 설계, 구현을 모두 수행한다. 이것은 응용 개발자에게 너무 의존하는 면이 있지만, 의사소통의 라인을 줄일 수 있어서 오히려 효율적이다. 또한, 최근에는 개발 속도의 향상을 위하여 분석, 설계와 구현이 병렬적으로 수행되는 경우가 많이 발생하므로 이러한 접근 방법이 자연스럽게 사용될 수 있다. 2명 이상의 응용 개발자가 서로 검토를 하면서 함께 분석, 설계, 구현을 수행한다면 개발을 효과적으로 할 수 있다.

<표 7>은 지금까지의 분석 결과를 바탕으로 RUP의 역할 분담, Web Wave의 역할 분담, 본 논문에서의 역할 분담에 대한 장단점을 정리한 것이고, <표 8>은 분석 결과를 도식화하여 비교한 것이다.

<표 7> RUP, Web Wave의 역할과의 장단점 비교

	RUP의 역할 분담	WebWave의 역할 분담	본 논문에서 제시한 역할 분담
장점	• 개발 위주가 아닌 프로젝트 전반에 걸친 모든 역할이 있다. • 대규모의 프로젝트에 적용 가능하다. • 검토를 수행하기 위한 역할을 따로 두었다.	• 웹 응용 개발을 위한 역할이다. • 프로젝트 관리자와 모든 팀이 함께 테스트를 수행하므로 품질 보증 면이 강화된다. • 대규모의 프로젝트에 적용 가능하다.	• 웹 응용 개발을 위한 역할이다. • 소규모의 응용 개발을 위한 것으로 개발 활동을 중심으로 구성한 역할이다. • 아키텍처를 바탕으로 구성하였기 때문에 역할이 시스템의 구조[2]와 자연스럽게 조화되며, 각 역할 수행자 간의 의사소통의 라인[2]을 쉽게 알

	•웹 응용 이외의 응용 개발에도 적용 가능하다.		수가 있으며, 아키텍처에 따라서 유연하게 조절할 수 있다. •질의문을 따로 작성하는 역할이 있어 데이터 처리 중심인 일반적인 웹 응용에 효율적이다. •응용 개발자가 응용에 대한 분석, 설계, 구현 및 통합을 모두 수행하므로 의사소통의 라인을 줄일 수 있어서 효율적이다.
	•역할의 수가 너무 많다(전체 31개, 개발 부분 17개). •웹 기반의 역할 분담이 아니라, 범용적인 역할 분담이다.	•역할이 팀 단위 이므로 개발 인원이 소규모에 적합하지 않다. •DB와 관련된 작업의 수행을 위한 역할이 뚜렷치 않다.	•응용 개발자가 응용에 대한 분석, 설계, 구현 및 통합을 모두 수행하여 응용 개발자 역할에 대한 의존도가 높다.

<표 8> RUP, Web Wave의 역할과의 비교표

	RUP의 역할 분담	Web Wave의 역할 분담	본 논문의 역할 분담
프로젝트 전반에 걸친 모든 역할이 있다.	○	○	×
모든 응용 개발에 적용할 수 있는 범용적인 역할 분담이다.	○	×	×
웹 응용 개발을 위한 역할 분담이다.	×	○	○
대규모 응용 개발에 적용 가능하다.	○	○	×
소규모의 응용 개발을 위한 역할 분담이다.	×	×	○
검토를 위한 역할이 있다.	○	×	△
질의문 작성을 위한 역할이 있다.	×	×	○
데이터베이스와 관련된 역할이 뚜렷하다.	○	×	○
아키텍처 기반의 역할 분담이다.	×	×	○
하나의 역할이 응용의 분석, 설계, 구현 및 통합을 수행한다.	×	×	○
모든 개발자가 테스트에 참여한다.	×	○	×
역할의 수가 많다. (개발 부분만 15개 이상)	○	×	×
역할이 팀 단위이다.	×	○	×
의사소통 라인이 간단하다.	×	△	○

5. 결론 및 향후 연구 방향

지금까지 본 논문에서는 소규모의 웹 응용 개발을 효율적으로 수행하기 위한 방법으로 역할 분담을 제시하였다.

Client/Server 아키텍쳐를 바탕으로 Web Server 부분의 사용자 인터페이스 설계자, 웹 디자이너, 웹 페이지 작성자, Application Server 부분의 도메인 전문가, 응용 개발자, 테스터, DB Server 부분의 데이터베이스 관리자, 데이터 설계자, 그리고 모든 작업을 감독하고, 문제 해결 방안을 찾기 위한 개발 책임자 역할들을 제시하고, 이들이 수행하는 작업을 제시하였으며, 이러한 역할들을 기존의 방법론인 RUP, Web Wave에서 제시하는 역할들과 비교 분석하였다.

본 논문에서 제시한 역할 분담을 적용하였을 경우에 얻을 수 있는 장점은 다음과 같다.

첫째, 역할 간의 의사소통 라인이 줄어들기 때문에 개발을 효율적으로 수행할 수 있어 개발 시간을 단축시킬 수 있다. 본 논문에서 제시한 역할은 소규모의 웹 응용 개발에 초점을 맞추었으므로 기존의 방법론에 비하여 역할이 간소화되었기 때문이다. 둘째, 필요이상의 산출물을 만들지 않기 때문에 개발 시간을 단축시킬 수 있다. 이는 소규모의 개발에 맞도록 각 역할들의 작업을 간소화하였기 때문이다. 셋째, 기존 방법론의 역할들을 웹 기반의 응용에 맞도록 커스터마이징을 하는 노력을 줄일 수 있다. 이는 본 논문이 웹 기반의 응용 개발을 위한 필수적인 작업들을 중심으로 역할들을 제시하였기 때문이다. 넷째, 웹 응용을 위한 아키텍처와 쉽게 조화될 수 있다. 이는 본 논문의 역할들이 웹 기반의 응용을 위해 가장 많이 사용되는 Client/Server 아키텍쳐를 바탕으로 한 것이기 때문이다.

본 논문은 소규모의 응용 개발에 초점을 맞추었기 때문에 개발 중심으로 역할을 분담하였다. 앞으로는 관리와 지원 부분의 역할을 추가하여 소규모의 웹 응용 개발을 위한 프로세스로 발전시키고자 한다.

참고 문헌

[1] Eun Sook Cho, et al., Object-Oriented Web Application Architectures and Development Strategies, APSEC '97 and ICSC '97. Proceedings, pp. 322-331, IEEE, 1997

[2] IEEE Recommended Practice for Architectural Description of Software-Intensive Systems, IEEE Std 1471-2000

[3] James martin + co., webWAVE Blueprints release 8.0, 1998

[4] Jim Conallen, Building Web Application with UML, Addison Wesley Longman, 2000

[5] Jim Conallen, Modeling Web Application Architectures with UML, Communications Of The ACM October 1999/Vol.42,No.10, ACM, 1999

[6] Kari Kivisto, Role of Developers as Part of a Software Process Model, Proceedings of the 32nd Hawaii International Conference on System Sciences, pp.13, IEEE, 1999

[7] Kent Beck, Extreme Programming Explained: Embrace Change, Addison Wesley, 1999

[8] Len Bass, Paul Clements and Rick Kazman, Software Architecture in Practice, Addison Wesley, 1998

[9] Rational Software , Rational Unified Process 2001.03.00, 2001

[10] Roger S. Pressman, Software Engineering - A Practitioner's Approach Fourth Edition, McGraw-Hill, 1997.

저자소개

이우진 (bluewjl@dreamx.net)
2000 숭실대학교 소프트웨어공학과 학사
2000-현재 숭실대학교 컴퓨터학과 석사과정
관심분야 : 개발 프로세스, 컴포넌트, 웹 개발, 모바일 솔루션

조용선 (yongsuns@shinbiro.com)
1997 숭실대학교 소프트웨어공학과 학사
1999 숭실대학교 컴퓨터학과 석사
1999-현재 숭실대학교 컴퓨터학과 박사과정
관심분야 : 개발 프로세스, 컴포넌트, 웹 개발

정기원 (chong@computing.soongsil.ac.kr)
1975.8-1990.4 국방과학연구소(부장, 책임연구원)
1990.5-1998.7 한국전산원 [소프트웨어 표준화연구회 위원장(90.5-91.12),
응용기술표준화연구 위원회 의장(95.7-97.9), 위촉연구위원(98.3-98.7)]
1991.4-현재 한국도로공사 자문위원(전산분야)
1992.8-1996.8 금융전산망추진위원회 자문회의 위원
1994.1-현재 한국정보시스템감리인협회[부회장(94.1-99.12), 회장(00.1-01.2)]
1996.2-현재 한국전자거래(CALS/EC)학회 부회장
1990.3-현재 숭실대학교 교수
관심분야 : 소프트웨어 프로세스, 방법론, 모델링, 실시간 응용, 전자거래,
정보시스템 개발 및 평가