

## 닷넷 스마트 클라이언트의 배포 및 관리를 위한 콜트리뷰 프로세스의 설계

### Design of Call Tree View Process for .Net Smart Client Solution Deployment and Management

강윤성, 이준환, 조한진  
극동대학교

Kang Yun-Sung, Lee Jun-Hwan, Cho Han-Jin  
Far East Univ.

#### 요약

프로그램의 개발과 운영 방식이 C/S 시스템에서 N-티어 방식 등으로 다양해짐에 따라, 프로세스의 설계 및 디자인 방식 또한 다양성을 띄게 되었다. 닷넷 스마트 클라이언트는 특정 기술을 지칭하는 것이 아니며, 닷넷의 Click Once, NTD, .NET Remoting 등의 여러 기술로 구성되어 N-티어 환경에서 사용되어진다. 본 논문에서는 3-티어 환경에서, 닷넷 스마트 클라이언트 프로젝트를 진행하면서 이슈화된 배포와, 부가적으로 효율적인 메뉴 관리를 위한 콜트리뷰 프로세스를 설계하였다.

## I. 서론

스마트 클라이언트는 C/S 시스템의 무거운 크기, 보안 등의 단점을 개선한다. 사용자는 자신이 사용하는 프로세스만 부분적으로 다운받으며 항상 최신버전을 유지할 수 있으며 사용자에게 UI만 제공하여 비즈니스 환경, 데이터베이스의 환경을 감추어 보안을 향상한다.

닷넷 스마트 클라이언트는 Click Once, NTD(No Touch Deploy) 방식으로 프로그램을 배포하며 .NET Remoting 기술로 통신한다. 이것은 마치 사용자가 웹 페이지를 요청하고 웹서비스 또는 Ajax의 방식으로 데이터를 전송받는 것과 유사하여 솔루션의 구축을 웹사이트로만 요구하던 고객에게 가볍고 더욱 강력한 UI를 제공하는 닷넷 스마트 클라이언트는 설득력 있는 대안점이 될 수 있을 것이다[1].

3-티어 환경에서는 서로 참조하는 컴포넌트들의 역할, 물리적 위치가 계층별로 분리되어 약한 참조관계를 갖게 되며 참조의 유효성 여부는 런타임에만 알 수 있다. 개발이 완료된 업무의 프로세스를 배포하려면 컴포

넌트들의 위치를 파악하여 적절한 대상의 수집이 필요하며 변경된 내용에 대해서도 즉시 반영하여 항상 완전한 상태를 유지하여 배포에 문제가 없도록 해야 한다 [2]. 본 논문에서는 이러한 처리를 수행하기 위한 프로세스를 콜트리뷰라 명명하고 설계한다.



▶▶ 그림 1. 일반적인 3-티어 모델

## II. 콜트리뷰 프로세스 설계

### 1. 어플리케이션 아키텍처 설계

어플리케이션 아키텍처 설계는 일반적인 3-티어 모델을 참조하였다. 어플리케이션 아키텍처가 3단계로 분리됨에 따라 실행 컴포넌트 또한 분리되어 역할이 단계별로 나뉘게 된다.

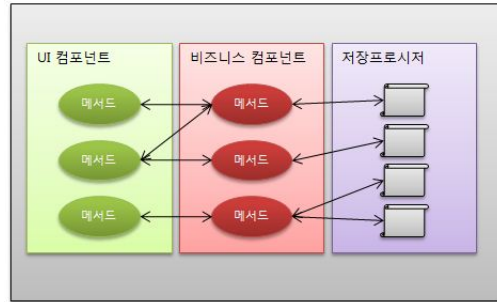
프리젠테이션 티어는 사용자 인터페이스로서 닷넷 스마트 클라이언트 UI 컴포넌트가 해당된다.

미들웨어가 되는 비즈니스 티어는 비즈니스 로직을 담당하는 비즈니스 컴포넌트가 위치하며 비즈니스 컴포넌트는 사용자 요청에 따른 데이터 작업의 중계 역할을 담당한다.

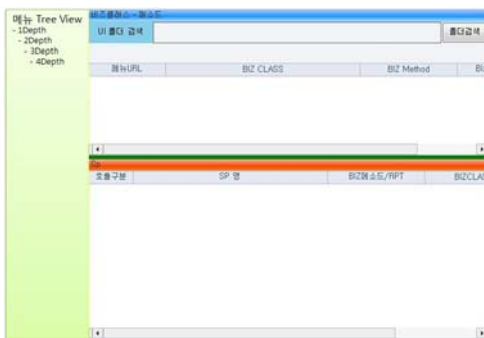
데이터 티어는 데이터베이스 액세스만을 담당하는 데이터 액세스 컴포넌트가 위치한다. 데이터 액세스 컴포넌트는 비즈니스 컴포넌트의 요청을 처리하여 반환한다.

닷넷 프레임워크가 프로젝트의 기반이 되며 자주 사용되는 메서드는 공통 프레임워크로 관리하여 사용한다[3].

콜트리뷰에서 수집한 배포목록의 데이터를 그림 1의 어플리케이션 아키텍처 구성도에 맞추어 도식화 하여 도메인 뷰로 표현함으로써 업무 메뉴의 처리 흐름을 한 눈에 파악할 수 있게 한다[4].



▶▶ 그림 3. 도메인뷰 화면 설계



▶▶ 그림 2. 콜트리뷰 화면 설계

### 2. 콜트리뷰 프로세스 설계

UI 컴포넌트와 비즈니스 컴포넌트를 참조하여 배포목록을 생성하고 관리하는 콜트리뷰와 더불어 저장된 배포목록을 도식화 하는 도메인뷰를 설계한다.

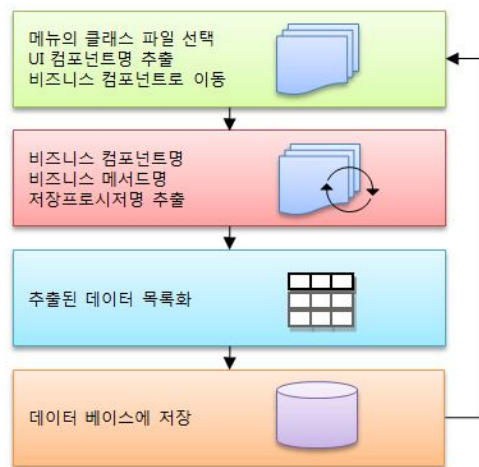
#### 2.1 콜트리뷰

개발자는 담당 메뉴의 개발이 완료되면 콜트리뷰를 통하여 메뉴와 배포 목록을 연결한 뒤 배포 담당자에게 요청한다. 읽기, 저장, 수정, 삭제가 모두 가능하다.

#### 2.2 도메인뷰

### 2.3 동작

최하위 메뉴가 사용하는 UI 컴포넌트의 실제 클래스 파일을 선택하면 파일을 읽어 들여 UI 클래스명과 비즈니스 클래스명, 위치 정보를 추출한다. 다음 UI에서 추출한 비즈니스 클래스 위치 정보로 해당 파일을 읽어 들여 비즈니스 클래스에서 사용하는 메서드명, 저장프로시저명을 추출한다. 끝으로 추출한 정보와 메뉴의 정보를 연결하여 데이터베이스에 저장한다. 컴포넌트 또는 코드의 수정으로 배포목록이 변경되었을 경우 UI 클래스 파일을 다시 선택하여 처음과 같은 과정을 거친다.



▶▶ 그림 4. 콜트리뷰 내부 동작

## 2.4 명명규칙

콜트리는 클래스 파일에서 정보를 추출하므로 일관된 프로젝트 명명 규칙이 필요하며 개발자들은 반드시 프로젝트 명명 규칙을 따른다.

## Ⅲ. 결론

콜트리는 일관된 자동화 처리 프로세스를 거쳐 결과물을 얻고 검증함으로써 사용자 임의의 생성에서 발생하는 오류를 감소하여 더욱 정확한 배포를 할 수 있다. 프로젝트에 참여하는 모든 개발자들이 콜트리를 통해 배포목록을 생성하고 관리하도록 한다.

콜트리 기능과 배포를 하나의 작업으로 보았을 때 배포를 두 단계로 분리함으로써 권한을 부여하여 업무의 통제가 가능하며 배포 프로세스의 구현이 한층 수월해 질 것이다.

향후 콜트리 설계를 토대로 콜트리를 구현해보고자 한다.

## ■ 참고 문헌 ■

- [1] <http://www.simpleisbest.net/archive/2006/01/02/360.aspx>
- [2] 허혁, PC 기반 그리드 환경에서 협업지원을 위한 어플리케이션 콘텐츠 배포 시스템, 석사학위논문, 2006.
- [3] 전병선, .NET Enterprise System 객체지향 CBD 개발 방법론, pp.221-225, 영진닷컴, 2004.
- [4] Steven Sanderson, PRO ASP.NET MVC Framework, pp63-68, BJ퍼블릭, 2009.