

스마트 클라이언트를 적용한 프로그래밍 과제평가 시스템 설계 및 구현

장정일*, 지정훈, 우균
부산대학교 컴퓨터공학과
e-mail : {jjjang, jhji, woogyun}@pusan.ac.kr

Design and Implementation an Evaluating System for Programming Assignment Applying Smart Client

Jung-II Jang*, Jeong-Hoon Ji, Gyun Woo
*Dept. of Computer Engineering, Pusan National University

요 약

인터넷의 발달로 온라인 원격강의나 E-Learning 형태의 교육이 많이 확산되었다. 온라인 교육의 확산으로 웹을 기반으로 하는 클라이언트 소프트웨어들이 많이 개발되어 사용되고 있다. 특히 교과목의 특성상 실습위주로 진행되는 프로그래밍 관련 교과목에서는 자동화 된 과제평가 시스템이 필요하다. 프로그래밍 과제평가 시스템에서는 과제제출, 소스코드 업로드, 컴파일 및 실행, 채점, 표절검사 등의 과제평가를 위한 기본적인 작업들을 수행된다. 위와 같은 작업들이 모두 평가시스템 서버에서 진행되기 때문에 서버에 많은 부하가 걸린다. 또한 공정한 평가를 위해 보안이 최우선적으로 고려되어야 한다. 본 논문에서는 스마트 클라이언트 기술을 이용한 과제평가 시스템을 설계하고 구현하였다. 본 시스템에서는 과제평가 시스템의 서비스를 분산시켜 시스템의 부하를 줄이고, 보안을 향상시킴으로써 학생들에게 공정한 평가를 가능하게 하였다.

1. 서론

최근 정보통신 기술의 발달로 인터넷을 통한 원격 강의나 E-Learning 형태의 교육이 많이 확산되었다. 이로 인해 교육을 위한 다양한 형태의 웹 클라이언트 기반 소프트웨어들이 개발되었고 사용되고 있다. 원격 강의나 E-Learning 과 같은 온라인강의뿐만 아니라 대학에서 이루어지는 오프라인 강의에서도 웹을 통한 강의 또는 평가 시스템은 많이 사용되고 있다.

특히 컴퓨터 프로그래밍 관련 강의에서는 자동화된 과제평가시스템이 필요하다[1]. 어문학이나 상공계열의 교과목과 달리 컴퓨터 프로그래밍 관련 교과목에서는 이론강의뿐만 아니라 소프트웨어를 만드는 실습 위주로 강의가 진행된다. 학생들을 평가하는 과제와 시험 또한 프로그램 코드를 작성하고 결과를 제출하는 방식으로 진행된다. 교과목의 특성상 교육자(instructor)가 문서를 통하여 학생들을 평가하기가 힘들다.

위와 같은 이유 때문에 컴퓨터 프로그래밍 강의에서는 자동으로 학생들을 평가할 수 있는 온라인 프로그래밍 과제평가 시스템이 필요하다. 프로그래밍 과제평가 시스템에서는 학생들을 관리하고, 온라인으로 과제를 제출할 수 있어야 하고, 학생들의 프로그램 소스코드를 원격의 서버(server)에서 컴파일하고 실행할 수 있어야 한다. 또한 컴파일 된 프로그램을 실행하여 결

과를 비교할 수 있어야 하며 모든 과제의 제출과 평가는 실시간으로 이루어져야 한다.

프로그래밍 과제평가 시스템에서는 보안이 상당히 중요하다. 학생들의 평가에 사용되는 데이터 파일이나 점수, 시도횟수와 같은 정보들은 공정하게 이루어져야 하며 절대로 유출되거나 수정되어서는 안 된다. 또한 프로그래밍 과제평가 시스템에서는 학생들의 공정한 평가를 위해서 표절검사가 이루어져야 한다. 교수 또는 조교가 과제를 등록하고 시스템에 업로드 된 학생들의 프로그램을 컴파일하고 실행하여 점수를 계산하는 모든 작업들은 과제평가 시스템 서버에서 이루어진다. 동시에 많은 학생들이 과제를 제출할 경우, 서버에서 모든 요청 작업을 처리해야 하기 때문에 많은 부하(overload)로 인해 문제가 발생할 수 있다.

본 논문에서는 스마트 클라이언트 기술을 이용한 과제평가 시스템을 설계하고 구현하였다. 스마트 클라이언트 기술은 분산서비스에 뛰어난 환경을 제공한다[2,3,4]. 이를 과제평가 시스템에 적용함으로써 시스템의 보안을 향상시키면서 과제평가를 위한 작업을 분산시킬 수 있다. 이는 평가 서비스의 향상과 공정한 평가를 통한 교육의 질적 향상을 가져올 수 있다.

본 논문의 구성은 2 장에서는 스마트 클라이언트의 정의, 필요성, 그리고 XML 웹 서비스에 대해 알아본다. 3 장에서는 스마트 클라이언트를 적용한 프로그래밍 과제관리 시스템을 설계하였으며, 4 장에서는 스마

트 클라이언트를 적용한 프로그래밍 과제관리 시스템을 구현하였다. 마지막으로 5 장에서는 결론 및 향후 연구를 제시한다.

2. 관련연구

2.1 스마트 클라이언트의 정의

스마트 클라이언트는 저장, 처리 또는 데이터 캡처를 위해 하드웨어와 같은 로컬 리소스를 활용할 수 있고, 웹 브라우저 기반 응용프로그램과 달리 스마트 클라이언트 응용 프로그램은 PC, 랩톱, Tablet PC 또는 스마트 장치에 사용이 가능하며, 온라인 및 오프라인 모두에서 사용이 가능하다. 또한 인텔리전트 배포 및 업데이트가 가능하다.

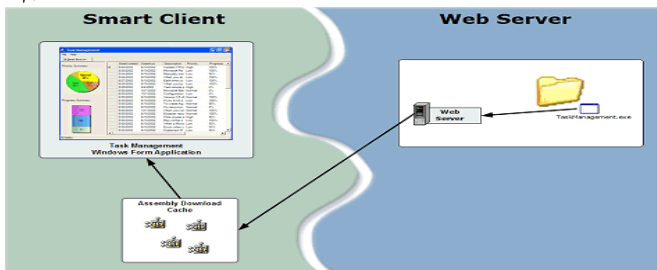


(그림 1) 스마트 클라이언트의 정의

(그림 1)은 리치 클라이언트, 썬 클라이언트, 스마트 클라이언트의 특성을 정의한 것으로 스마트 클라이언트는 리치 클라이언트와 썬 클라이언트의 장점을 혼합한 개념으로서, 비교적 손쉽게 배포가 가능하고 실행 시 로컬의 리소스를 사용하고 데이터를 원격에서 끌어와서 동작하는 보다 진보된 아키텍처를 가진 어플리케이션이다.

2.2 스마트 클라이언트의 필요성

스마트 클라이언트는 제작 언어를 자유롭게 선택할 수 있으며, Code Access Security 에 기반한 보안으로 인해 사용자 입장에서 안전하게 사용할 수가 있다. 스마트 클라이언트 코드의 실행은 클라이언트에서 이루어지기 때문이다. 스마트 클라이언트를 적용함으로써 개발생산성 및 보안 요구사항을 현저하게 높일 수 있다.



(그림 2) No-Touch 배포 시나리오

또한 (그림 2)와 같이 닷넷 프레임워크의 No-Touch 배포방식을 이용하면 스마트 클라이언트는 HTML 페이지와 동일한 방식으로 설치 및 유지 관리가 가능해 응용 프로그램 설치 시 공유되는 DLL 이 기존에 있던 DLL 을 덮어 씌어질 경우 버전 정보에서 충돌이 발생

하거나 새로운 버전의 DLL 이 기존의 DLL 을 완벽히 지원하지 못해 사용하는 응용 프로그램 에서 오류가 발생하는 “DLL Hell”이라는 일반 시나리오 문제를 해결할 수 있다.

2.3 XML 웹 서비스

닷넷 프레임워크 스마트 클라이언트는 HTTP 프로토콜로 접근되는 일종의 인터넷 어플리케이션이지만 서버와 데이터를 주고받기 위해 별도의 포트에 연결되는 다른 프로토콜을 사용하지 않는다[5]. 스마트 클라이언트가 MISL(Microsoft Intermediate Language)코드로 작성되어 있으므로, 만약 악의적인 목적을 가진 사용자가 해당 스마트 클라이언트의 코드를 역 어셈블리를 통해 본다면 코드 안에 썬어있는 데이터베이스 문자열등을 발견할 수 있고, 그것은 심각한 사태를 초래할 수 있기 때문에 스마트 클라이언트에는 보안이 요구되는 코드나 로직을 직접 삽입해서는 안되며 XML 웹 서비스를 이용한 코드보안이 필요하다.

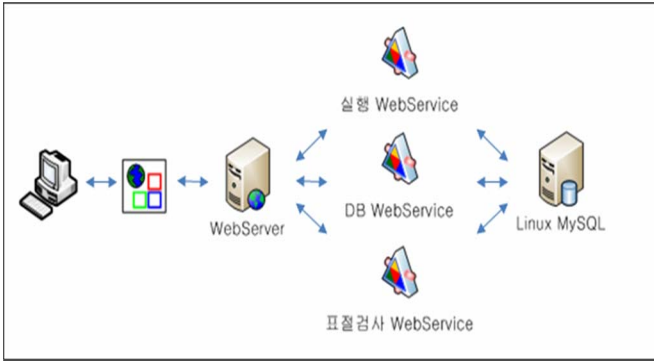


(그림 3) XML 웹 서비스의 구성 요소간의 관계

(그림 3)은 XML 웹 서비스의 구성 요소간의 관계를 보여주고 있다. XML 웹 서비스는 SOAP 및 HTTP 와 같은 표준 프로토콜을 기반으로 하기 때문에 플랫폼에 관계 없이 응용 프로그램 간에 상호 작용할 수 있으며 새 응용 프로그램과 레거시 코드를 완벽하게 통합할 수 있게 해준다[5].

3. 스마트 클라이언트를 적용한 프로그래밍 과제관리 시스템 설계

스마트 클라이언트의 가장 큰 장점은 닷넷을 이용하여 이 기종의 다양한 플랫폼에 원격접속 가능한 웹 서비스를 동시에 구현할 수 있다. 본 논문은 웹에 임베디드 된 형태가 아닌 독립 실행형 스마트 클라이언트로 설계하였다. (그림 4)는 스마트 클라이언트를 이용한 프로그래밍 과제관리 시스템의 구조도이다. 사용자가 인터넷을 통해 프로그램 과제관리 시스템에 접속하면 사용자 권한에 따라 관리자와 사용자로 구분된 다른 형태의 클라이언트 프로그램이 실행된다. 관리자는 사용자정보, 강의정보 및 과제를 등록할 수 있고, 사용자는 등록된 과제를 확인, 등록된 요구사항, 라이브러리 및 Sample 데이터 등의 정보를 확인 후 과제를 제출하면 Service Control 웹 서비스가 제출자료에 대한 정보를 DB 웹 서비스를 통해 데이터베이스에 저장하고, 실행 웹 서비스를 통해 컴파일 및 오류검사를 실시한 후 결과를 데이터베이스에 저장 후 실행 결과를 사용자에게 전달해 준다.



(그림 4) 프로그래밍 과제평가 시스템 구조도

또한 표절검사항목이 선택된 경우 실행 결과 저장 후 표절검사 웹 서비스를 통해 제출된 과제에 대한 표절검사를 수행하고 그 결과를 데이터베이스에 저장한다.

4. 스마트 클라이언트를 적용한 프로그래밍 과제관리 시스템 구현

스마트 클라이언트를 적용한 프로그래밍 과제관리 시스템을 구현하기 위해서는 Visual Studio 2005, 닷넷 프레임워크 2.0 그리고 데이터베이스는 MySQL 5.0 을 사용한다. XML 웹 서비스를 사용하기 위해 XML 웹 서비스 클래스 라이브러리인 ESPACapture, ESPAQuery, ESPACrib 을 참조한다. 또한, 각 실행 모듈 별 참조 코드 및 XML 웹 서비스 참조를 위해 작성된 공용 클래스 라이브러리인 ESPAClass 라이브러리를 참조한다.

스마트 클라이언트를 적용한 프로그래밍 과제관리 시스템을 구현한 결과는 (그림 5), (그림 6)이다. (그림 5)는 관리자가 과제를 등록하는 화면이다. 관리자는 실행횟수, 실행시간, 제출기간, 표절검사 유무를 선택 입력한다. (그림 6)은 사용자 별 제출된 과제에 대한 성적관리를 위한 화면이다. 제출된 과제에 대한 각 과제별 성적통계, 분포 및 표절여부를 위한 유사도 검사의 실행 후 결과 등을 조회할 수 있다.

기존 웹 클라이언트 기반의 어플리케이션에서는 많은 동시 접속자에 대한 서버 부하, 과제 제출을 통한 웹 서버에서의 악의적 코드 실행으로 시스템 다운 및 해킹에 대한 보안문제가 있다. 이에 반해 스마트 클라이언트 어플리케이션은 비연결성(Disconnected)환경인 XML 웹 서비스를 통한 분산처리로 서버부하에 따른 문제를 해결하고, 악의적 과제코드가 실행되더라도 전체 시스템에 영향을 미치지 못하도록 과제 실행 웹 서비스를 분리함으로써 전체 시스템의 안정성 및 보안성을 높일 수 있다.

5. 결과 및 향후 연구

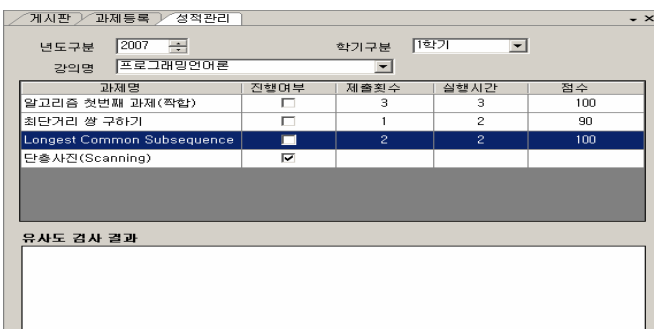
스마트 클라이언트를 적용하여 구현된 프로그래밍 과제관리 시스템은 웹에서 실행하는 프로그램이지만 사용자의 로컬 환경에서 실행되는 윈폼 형태의 프로그램과 동일해 풍부한 유저 인터페이스를 제공해 주고, 웹 브라우저의 스크립트 렌더링(Rendering)으로 인한 대기시간이 사라지고, 많은 동시 접속자의 요청에 의한 서비스 서버의 리소스 증가에 따른 부하도 XML 웹 서비스 분산환경을 구축함으로써 해결 할 수 있고 악의적 코드에 대한 전체 시스템의 보안성을 높일 수 있다. 따라서 사용자는 제출 과제에 대한 빠른 응답과 다양한 형태의 제출과제에 대한 정보제공으로 해당 프로그래밍 과목에 대한 전체적인 성취도를 높일 수 있다. 본 논문에서 구현한 클라이언트 프로그램은 닷넷 프레임워크 기반으로 구축되어 닷넷 프레임워크가 설치되는 환경에서만 실행될 수 플랫폼 종속적인 시스템이다. 이러한 문제는 크라이언트 프로그램을 다양한 플랫폼에서 실행 가능한 JAVA 로 개발한다면 보다 플랫폼 독립적인 시스템이 될 것이다. 따라서, 이러한 부분에 대한 연구가 필요할 것이다.

참고문헌

[1] Brenda Cheang, "On automated grading of programming assignments in an academic institution", Computer & Education, 2003.
 [2] 이영순, "스마트 클라이언트를 적용한 전자상거래 시스템 설계 및 구현", 한국정보과학회, Vol. 32 No. 2, pp.520~522, 2005.
 [3] 박종현, "스마트 클라이언트 응용을 이용한 문서 시스템의 설계 및 구현", 한국정보처리학회, 제 13 권 1 호, 593~596, 2006.
 [4] Narayan Veeramani, "Smart Clients versus Web Forms", IEEE Computer, Vol. 39 No. 08, pp.93~95, 2006.
 [5] Panagiotis Louridas, "SOAP and Web Services", IEEE Software, Vol. 10 No. 06, pp.69~75, 2006.



(그림 5) 과제등록



(그림 6) 학생성적관리