

플래시 컴포넌트와 서비스 API를 이용한 X-인터넷 개발 프레임워크 설계

고 대 식*

A Design of X-internet Development Framework by Using Flash Component and Service API

Dae-Sik Ko*

■ Abstract ■

In this paper, we designed a new type of X-internet framework by using Flash component and server service API and analyzed its performance. Proposed X-internet framework uses Flash MX Professional 2004, Java, and software which opened middleware and database. Since proposed framework use the server service API that we developed in this paper and Flash player, this framework does not need server module. Proposed framework enables to obtain design with dynamic user interface compare to Web application and enables to reduce development time.

In analytical results, it has been shown that proposed x-internet framework have efficient characteristics such as network traffic, low development cost and dynamic user interface implementation. Since proposed X-internet framework can operate in environment of current developer friendly, it is useful for development of various new application programs and we confirm it through Flash web mail implementation by using proposed x-internet framework.

Keyword : X-internet, Interaction Program, Development Framework, Web Application,
Client/server

* 목원대학교 전자정보보호공학부 교수

1. 서 론

메인프레임 컴퓨터시대에서 시작된 정보시스템은 Client/Server 시대를 거치면서 최근에는 거의 모든 어플리케이션 소프트웨어를 Web으로 변화시켰다. '웹 기반'의 시스템에서 클라이언트는 브라우저만으로 구성되는 장점이 있지만, 서버 측은 상당히 많은 컴퓨팅이 필요하게 되고 개발생산성도 낮아지는 단점이 있었다[1, 2]. 더군다나 최근 정부기관, 기업, 그리고 교육기관을 막론하고 정보시스템 사용자들은 다양한 상호작용(interaction)과 미학적이고 화려한 사용자 인터페이스를 요구하고 있다.

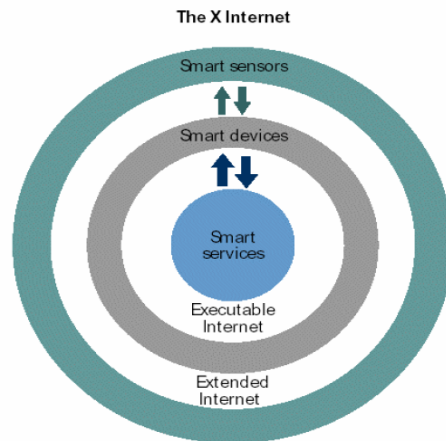
X-인터넷 어플리케이션은 기존의 Web과 C/S의 장점을 취하여 사용자 생산성을 높이고, 개발 소요 비용을 낮추는 스마트 클라이언트 개발 환경으로 주목받고 있으며, 사용자들의 화려한 사용자 인터페이스 요구에 대한 솔루션으로 제시되고 있다[3~5].

본 연구에서는 이러한 X-인터넷 기술동향을 분석하고 플래시컴포넌트와 API를 기반으로 하는 새로운 형태의 X-인터넷 개발프레임워크를 개발하였다. 본 논문에서 제안하는 X-인터넷 개발프레임워크는 Flash MX Professional 2004와 Java만을 이용하고 기타 요구되는 미들웨어나 데이터베이스는 오픈 소프트웨어를 이용하였다. 플래시 클라이언트는 XML을 이용하므로 서버 단에서 XML 파서를 이용하여 해당 내용을 이해하게 하며, 서버 모듈이 아닌 적절한 API를 개발 제공하여 손쉽게 DB관련 작업을 수행하도록 설계 하였으며 클라이언트는 크게 조회(Select) 작업과 상호작용(Update/Insert/Delete)을 수행할 수 있도록 이에 따른 서버 서비스 API를 개발하였다.

2. X-인터넷 분석

X-인터넷이란 용어는 2000년 10월 Forrester Research의 CEO인 George F. Colony가 자사

웹사이트에 'My View : X-Internet'이라는 단문을 기재한 것부터 시작되었다. X-인터넷은 eExecutable Internet과 eXtended Internet의 X를 인용한 것으로 Forrester Research는 [그림 1]을 이용하여 X-인터넷을 정의하고 있다[2].



[그림 1] X-인터넷 개요(Forrester research)

[그림 1]에서, Executable Internet은 인터넷 기반의 풍부하고 상호작용이 가능한 응용프로그램으로서 클라이언트 쪽에 실행코드를 두고 실행하는 인텔리전트 프로그램 즉 웹 상에서 수행되는 클라이언트 어플리케이션을 통하여 사용자들의 온라인 작업을 향상시킬 것이고 Extended Internet은 인터넷이 가능한 칩이 장착된 모든 기기들이 유기적으로 연결됨으로써 기기와 기기, 사람과 사람, 기기와 사람 등의 상호관계의 연결성을 강화하여 유비쿼터스 환경을 구축하는 것으로 말할 수 있다.

X-인터넷은 결국 인터넷이기 때문에 아키텍처는 우리가 흔히 구축해온 인터넷과 동일하지만 사용자 인터페이스는 C/S의 형식을 따른다. X-인터넷의 개발방법을 살펴보면, 화면을 구성하고 이벤트를 처리하는 것은 C/S 방식을 따르고 여타 데이터베이스 접근방식 등은 모두 인터넷 개발 시 JSP/ASP 혹은 EJB를 개발했던 방식과 대동소이

하다. 즉, X-인터넷은 사용자 PC, 노트북, PDA, WebPad 등에 지역적으로 인스톨되어 기능적인 면에서 C/S처럼 풍부함을 제공하고 웹 시스템처럼 한 곳에 집중된 관리를 함으로써 전개(Deploy)를 용이하게 하는 Web과 C/S의 장점이 융합된 개발환경이다. 더군다나 X-인터넷은 C/S 프로그램처럼 필요한 정보만을 혹은 바뀐 정보만을 서버와 송수신 한다. 즉, 웹 시스템처럼 화면을 디스플레이 해주기 위한 태그 정보를 재송수신 할 필요가 없기 때문에 트래픽측면에서도 커다란 장점이 있다.

국내에 소개되고 있는 X-인터넷 제품들은 모두 서버 모듈과 재생기를 이용하는 개발환경이고 특히 Adobe의 Flex를 제외한 타 제품들은 Flash 재생기를 이용하지 않기 때문에 Flash의 장점을 활용하기 어려운 구조를 갖고 있다[6~13].

X-인터넷은 특히, 관리의 편리성, 개발 기간 단축, 유지 보수의 비용의 측면에서는 탁월한 장점을 가지며, 기능성, 응답속도, 양방향성, 보안성 등의 특징에 대해서는 C/S에 버금가는 능력을 가지고 있다. 국내은행이 2003년 8월 오픈한 X-Internet 기반의 인터넷 뱅킹에 대한 측정결과를 발표한 자료에 의하면, 기존 인터넷 뱅킹 대비 X-인터넷 기반 시스템의 처리속도는 15% 개선되고 송수신데이터는 9.5배나 축소되었으며, 인터넷 뱅킹 요구 시 사용자의 화면에 출력되는 소요시간은 평균 2~3초가 단축된 것으로 보고하고 있다[1].

3. 플래시컴포넌트와 API 기반 X-인터넷

3.1 개발환경

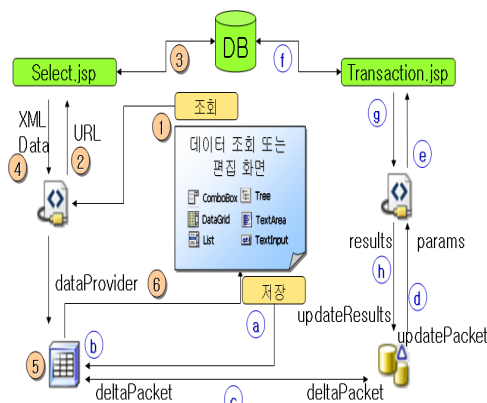
본 논문에서는 이미 90% 이상의 사용자 PC에 탑재되어 있는 Flash 재생기를 활용하고, 서버 모듈을 별도로 설치하지 않아도 되는 서버 API를 이용하는 구조로 비용효율성을 갖는 개발 프레임워크를 개발하였다. 본 논문에서 제안하는 개발프

레이프워크는 첫째로 프레젠테이션 로직과 비즈니스 로직을 명확히 구분하여 설계하였다. 즉 프레젠테이션 로직은 Flash MX 틀을 이용하고 비즈니스 로직은 Java를 이용하므로, 근본적으로 로직이 구분될 수 있도록 하였다. 둘째로, 개발 및 운영을 위한 별도의 서버를 두지 않도록 한다. 서버 서비스 개발은 특정 환경 및 서버에 의존하지 않고 비용면에서도 효율적이도록 하기 위하여 서버 모듈이 요구되지 않는 API 방식으로 설계 개발하였다.

3.2 플래시 어플리케이션의 개발

플래시 어플리케이션을 개발은 DB 설계 및 구축 →스키마 생성기를 통한 Schema XML 파일 제작 또는 직접 작성 →Flash MX 2004에서 Data Components를 이용한 어플리케이션 제작 →JSP를 이용한 서버 서비스 제작 →HTML과 SWF는 웹 서버 또는 서블릿 컨테이너(Tomcat)에 탑재 →JSP는 서블릿 컨테이너(Tomcat)에 탑재 순으로 이루어진다.

플래시 어플리케이션을 개발하기 위해서는 데이터 흐름이 무엇보다 중요하다. 데이터의 흐름은 조회의 경우와 저장의 경우로 나뉘지게 되며, 본문에서 제안하는 개발 프레임워크의 데이터 흐름은 [그림 2]와 같다.



[그림 2] 데이터의 흐름

[그림 2]에서 나타난 조회 데이터의 흐름은 아래와 같다.

- ❶ 화면상에서 폼이 로드되었거나 조회 버튼을 누르면 XMLConnector를 trigger한다.
- ❷ XMLConnector는 지정된 서비스 URL로 접속(Select.jsp) 한다.
- ❸ Select.jsp는 조회 조건으로부터 DB를 Query하여 데이터를 획득한다.
- ❹ 획득된 데이터는 서비스 API를 통해 XML 포맷으로 변경된다.
- ❺ XMLConnector가 수신 받은 데이터는 바인딩된 DataSet에 저장된다.
- ❻ DataSet에 바인딩된 화면 구성요소에 해당 데이터가 표현된다.

한편 저장데이터의 흐름은 아래와 같다.

- a) 화면상에서 저장 버튼을 누르면 DataSet의 ApplyUpdates() 함수를 실행한다.
- b) 최초 DataSet에 보관된 데이터에 변화가 일어났을 경우, deltaPacket 생성/수정된다.
- c) deltaPacket이 존재하면 DataSet과 바인딩된 RDBMSResolver가 deltaPacket을 수신한다.
- d) RDBMSResolver는 deltaPacket을 XMLConnector에 파라미터로 전달한다.
- e) XMLConnector는 파라미터를 XML포맷으로 변경하여 서비스 URL(Transaction.jsp)를 호출한다.
- f) Transaction.jsp는 서비스 API를 통하여 파라미터를 해석하고, 그에 따라 DB를 갱신한다.
- g) 갱신 결과는 서비스 API를 통하여 결과 패킷을 생성하고, XMLConnector의 results로 전달한다.
- h) 결과 값은 RDBMSResolver 인식하고, 성공 및 실패에 따른 해당 처리를 수행한다.

3.3 서버 서비스 API 개발

서버 서비스는 플래시 컴포넌트의 요청에 의해

조회와 역할과 데이터 저장의 역할을 수행한다. 이때, 플래시 컴포넌트는 URL(GET방식)이나 XML(POST방식) 형식으로 데이터 처리를 요청하게 된다. 따라서, 서버서비스는 플래시 컴포넌트의 XML에 의해 요청을 해석하고, 그 결과를 XML 형식으로 되돌려 주는 역할을 수행하게 된다.

서버서비스 API 역시 조회성 부분과 저장성 부분으로 구성되며, 플래시 컴포넌트와 XML 형식으로 통신하게 된다.

(1) 조회성 함수

조회성 함수는 클라이언트의 특정 요청에 대해

<표 1> 서버 서비스 조회성 함수 목록

| 함수명 | 설명 |
|--|-------------------------------|
| void makeRow() | 조회성 함수의 시작 |
| void startRow() | 새로운 ROW의 시작을 지정 |
| void addColumn (String column, String value) | 칼럼을 추가하고, String 타입의 값을 지정 |
| void addColumn (String column, boolean value) | 칼럼을 추가하고, boolean 타입의 값을 지정 |
| void addColumn (String column, int value) | 칼럼을 추가하고, int 타입의 값을 지정 |
| void addColumn (String column, long value) | 칼럼을 추가하고, long 타입의 값을 지정 |
| void addColumn (String column, float value) | 칼럼을 추가하고, long 타입의 값을 지정 |
| void addColumn (String column, Time value) | 칼럼을 추가하고, Time 타입의 값을 지정 |
| void addColumn (String column, Date value) | 칼럼을 추가하고, Date 타입의 값을 지정 |
| void addColumn (String column, Timestamp Timestamp value) | 칼럼을 추가하고, Timestamp 타입의 값을 지정 |
| void endRow() | ROW의 종료를 지정 |
| void commit() | 조회성 함수의 종료를 알리고, response로 출력 |

적절한 응답을 할 수 있도록 하는 데 사용하는 함수들로 구성된다. 통상 관계형 데이터베이스의 칼럼(column)과 로우(row)로 구성되는 테이블 형식의 데이터 XML 포맷으로 리턴하게 된다.

조회성 기능의 예제를 아래에 보인다.

Fria객체를 선언하고, makeRow()와 commit()로 데이터의 전체 영역을 감싸게 된다. 그리고, 데이터베이스로 부터 읽어 들인 값을 매로우마다 startRow()와 endRow()로 감싸게 되며, 실제 데이터의 칼럼과 포함하는 값은 addColumn()에 의해 수행된다.

```

1 conn = DriverManager.getConnection
  (dbhost, dbuser, dbpassword);
2 stmt = conn.createStatement();
3 rs = stmt.executeQuery("SELECT
  msgid, uname");
4 FRIA fs = new FRIA(request,
  response);
5 fs.makeRow();
6 while(rs.next())
7 {
8 fs.startRow();
9 fs.addColumn("msgid", rs.getInt
  ("msgid"));
10 fs.addColumn("uname", rs.
  getString("uname"));
11 fs.endRow();
12 }
13 fs.commit();
    
```

(2) 저장성 함수

저장성 함수는 클라이언트가 서버에 저장하는 기능을 처리하기 위한 함수의 모음으로써, 실제 INSERT, UPDATE, DELETE의 기능을 수행하기 위한 함수들로 구성되어 있다.

클라이언트는 서버에게 입력, 갱신, 삭제의 기능을 요청하게 되는데 이 요청을 구분하여 각각을 수행하게 된다. <표 2>는 저장성 기능을 수행하기 위한 함수목록으로 클라이언트의 요청을 해석하는 함수이다.

<표 2> 서버 서비스 저장성 기능의 함수 목록

| 함수명 | 설명 |
|-------------------|--|
| void doParse() | 플래시의 델타패킷 값을 파싱할 때 호출 |
| boolean nextRow() | 현재 파싱하고 있는 델타 패킷의 다음 ROW가 있으면 True 아니면 False |
| int getRowMode() | 현재 파싱하고 있는 델타 패킷의 현재 ROW의 상태를 나타내는 값. FRIA_INSERT, FRIA_UPDATE, FRIA_DELETE 중 하나를 리턴 |

클라이언트의 요청이 해석되면, 실제 클라이언트가 원하는 변경된 값, 즉 델타 값을 얻어와야 한다. <표 3>은 칼럼 명으로 클라이언트가 변경을 원하는 실제 값을 얻어오는 함수들의 목록이다.

<표 3> 서버 서비스 델타값 획득 함수 목록

| 함수명 | 설명 |
|------------------------------------|------------------------------------|
| String getString (String column) | 델타 패킷의 column 값을 String 타입으로 획득 |
| boolean getBoolean (String column) | 델타 패킷의 column 값을 boolean 타입으로 획득 |
| long getLong (String column) | 델타 패킷의 column 값을 long 타입으로 획득 |
| int getInt (String column) | 델타 패킷의 column 값을 int 타입으로 획득 |
| int getFloat (String column) | 델타 패킷의 column 값을 Float 타입으로 획득 |
| int getDate (String column) | 델타 패킷의 column 값을 Date 타입으로 획득 |
| int getTime (String column) | 델타 패킷의 column 값을 Time 타입으로 획득 |
| int getTimestamp (String column) | 델타 패킷의 column 값을 Timestamp 타입으로 획득 |

이상과 같은 함수들은 다음의 예제처럼 사용된다.

```

1 FRIA fs = new FRIA(request, response);
2 fs.doParse();
3 while(fs.nextRow())
4 {
5 if (fs.getRowMode() == fs.FRIA_INSERT)
6 {
7 PreparedStatement pstmt =
  conn.prepareStatement(
8 "INSERT INTO fb_master (wname, subject)
  VALUES (?,?) WHERE msgid=?");
9 pstmt.setString(1, fs.getString(
  "wname"));
10 pstmt.setString(2, fs.getString(
  "subject"));
11 pstmt.setInt(3, Integer.
  parseInt(msgid));
12 pstmt.executeUpdate();
13 } else if (fs.getRowMode() ==
  fs.FRIA_UPDATE)
14 {
15 PreparedStatement pstmt = conn.
  prepareStatement("UPDATE fb_master
  SET wname=?, subject=? WHERE
  msgid=?");
16 pstmt.setString(
  1, fs.getString("wname"));
17 pstmt.setString(
  2, fs.getString("subject"));
18 pstmt.setInt(3, Integer.
  parseInt(msgid));
19 pstmt.executeUpdate();
20 } else if (fs.getRowMode() ==
  fs.FRIA_DELETE)
21 {
22 PreparedStatement pstmt = conn.
  prepareStatement(
  "DELETE FROM fb_master WHERE
  msgid=?");
23 pstmt.setInt(1, Integer.parseInt(
  msgid));
24 pstmt.executeUpdate();
25 }
26 }

```

doParse() 함수로써, 저장성 기능임을 지정한 후, 클라이언트의 요청들을 모두 한 번에 처리하기 위해 nextRow() 함수로 루프를 수행한다.

각 로우에 대한 해당 요청을 getRowMode()를 이용하여 INSERT/UPDATE/DELETE를 구분한 후 각각의 기능을 수행하게 된다. 이 때 클라이언트의 변경 값들은 getXXX 함수들을 이용하

여 얻어오며, 그 값을 SQL 명령으로 처리한다.

끝으로, 제안된 개발 프레임워크의 유효성을 확인하기 위하여 Flash 웹 메일 시스템을 구현하였다. 웹 메일 시스템은 다양한 방법으로 구현가능하지만, 본 시스템은 기존의 메일 서버를 그대로 이용하고, 자바메일을 통한 POP3와 SMTP를 이용하여 이메일 처리 시스템을 구현하였다. 이메일의 수신/송신 정보는 데이터베이스에 저장되며, 내용의 경우 일반 텍스트(plain/text)는 데이터베이스에 저장하지만 기타 HTML 형식이나 첨부파일들은 로컬 디스크에 파일의 형태로 유지하도록 하였다.

구현된 웹메일 시스템은 하나의 플래시 파일로써, 서버 서비스와의 통신을 통해 기본적인 웹메일 기능을 수행한다. 플래시의 기본 컴포넌트만을 이용하였기 때문에, 단순한 인터페이스로 구성되어 있으나 플래시의 디자인 기능을 활용하면 화려하고 동적인 화면의 구성도 가능하게 된다.

앞서 언급한 것처럼, 하나의 어플리케이션처럼 동작하기 때문에 HTML로 구성되는 일반 웹메일과는 달리 화면 전환에 따른 지연이 없고, 사용자의 요청에 빠르게 응답하며, 디자인 측면만 좀 더 보완된다면 기존 웹메일의 한계를 넘는 가장 화려한 인터페이스도 가능한 구조이다.

4. 결 론

본 논문에서는 플래시컴포넌트와 서버서비스 API를 이용한 새로운 형태의 X-인터넷 개발 프레임워크를 개발하여 그 성능을 분석하였다.

X-인터넷 개발환경은, C/S와 웹 방식의 장점을 수용한 방식이기 때문에 기존 웹방식의 인프라는 그대로 사용되며, 여기에 클라이언트 측에서는 재생기 또는 컴포넌트가 서버 측에서는 X-인터넷 서버 솔루션이 설치되게 된다. X-인터넷 솔루션마다 차이는 있지만, 개발자는 X-인터넷 솔루션을 이용하여 어플리케이션을 개발하고, 기능 개선 및 유지보수를 행한다. 이러한 도구들은 통상 웹

방식의 어플리케이션 개발방법보다 개발이 용이하다. 인터페이스 측면으로는 웹 브라우저의 영역에서 제한되었던 것이 X-인터넷 솔루션이 제공해주는 영역으로 확대될 수 있으며 통상적으로 X-인터넷 재생기는 Active X 또는 Java 기술을 통하여 클라이언트에 손쉽게 설치한다.

본 논문에서 제안하는 X-인터넷 개발 프레임워크는 기존 웹 방식과 동일한 인프라 구조를 갖고 있으며, 클라이언트 측에서는 이미 대부분 설치되어있는 Flash 재생기를 이용하며, 서버 측에서는 Java/JSP 언어를 위한 API를 이용하므로 구조적으로 새로운 서버를 설치하지 않는 특징이 있다. 개발자는 Flash 개발 환경에서 Flash 파일을 제작하고, 서버 측에서는 API를 이용하여 서비스를 개발하는 식으로, 통상적인 X-인터넷 개발 방법과 유사하며 화면 인터페이스는 플래시가 제공해주는 영역까지 확대된다.

연구결과, 본 논문의 Fria는 시제품 단계이기 때문에 Flex와 같은 상용 제품에 비교하면 개발 및 유지보수의 효율성에 있어서는 다소 떨어질 것으로 판단된다. 그러나, 네트워크 트래픽 효율은 상용 수준에 근접하고, 별도의 서버 솔루션이 요구되지 않으므로 시스템 비용측면에 장점을 갖는다. 또한 사용자 인터페이스가 플래시 기반이 가장 효율적이다 라는 전제하에 Flex에 대비해서 무비클립을 직접 포함할 수 있는 구조인 Fria에게

조금 더 높은 점수를 부여하였다. AJAX 또한 네트워크 트래픽 측면과 개발 및 유지보수 측면에서는 장점이 있지만 사용자 인터페이스 측면에서는 낮은 수준에 위치하는 것으로 평가되었다.

참 고 문 헌

- [1] 김영현, "X Internet 시작과 발전에 관하여", 투비소프트 technical document.
- [2] Joshua Duhl, "The Business Impact of Rich Internet Applications", IDC white paper.
- [3] 이민주, "기술연재 플래시", Digital Contents, 애슬론 white paper
- [4] 김태식, "플래시를 기반으로한 리치 인터넷 어플리케이션 개발", 블루마인드 커뮤니케이션 white paper
- [5] <http://www.tobesoft.com>
- [6] <http://www.comsquare.co.kr>
- [7] <http://shift.co.kr>
- [8] <http://www.droplets.com>
- [9] <http://www.cokinetic.com>
- [10] <http://www.altio.com>
- [11] <http://www.curl.com>
- [12] <http://www.nexaweb.com>
- [13] <http://www.macromedia.com>

◆ 저 자 소 개 ◆

**고 대 식 (kds@mokwon.ac.kr)**

현재 목원대학교 전자정보보호공학부 교수로 재직하고 있으며, 경희대학교 전자공학과에서 학사, 석사, 박사를 취득하였다. 목원대학교 학술정보처장, 중앙공무원교육원 외래 강사, 대전광역시 및 산림청 자문위원, 한국통신학회 이사를 역임하였으며 현재 한국정보기술학회 부회장으로 활동하고 있다. IEEE, 한국통신학회, 대한전자공학회, 한국정보 기술학회 논문지등에 게재한바 있으며 주요 관심분야는 VoIP, 웹서비스, EA 등이다.