

# 아파치 POI를 이용한 웹기반 엑셀-DB연동 기법<sup>+</sup>

이상민, 김상우, 이상원  
성균관대학교 정보통신공학부

e-mail:nice2345@naver.com, spun@skku.edu swlee@skku.edu

## Web-based Excel-DB Integration using Apache POI

Sang-Min Lee, Sang-Woo Kim, Sang-Won Lee

School of Information and Communication Engineering, Sungkyunkwan University

### 요 약

정형 정보를 저장하는 대표적인 수단인 Microsoft Excel 과 Database 간 상호변환에 있어서 다양한 방법들 중 Web을 기반으로 하는 변환 기술은 활용도가 매우 높을 것이다. 특히 많은 자료가 엑셀로 관리되는 경우 데이터 중복의 문제가 발생하며 DB를 통한 통합관리는 사용자 편의성 측면에서 높은 필요성을 갖는다. 본 논문은 이런 문제를 해결하기 위한 방안으로 POI를 이용한 변환 기법을 소개한다.

### 1. 서론

우리가 다루는 정보는 비정형과 정형의 두 가지로 구분되어진다. 비정형 정보의 경우 비정형이라는 특성에 의거하여 여러 가지 다양한 형태의 정보들을 포함하며 이러한 정보들을 저장하는 방법 또한 다양하다. 비정형 정보를 저장하는 수단으로는 각 비정형 정보의 특성에 따른 파일 포맷으로 저장하는 것이 일반적이다. 예를 들면 시각적인 정보는 JPG나 GIF와 같은 그래픽 파일로 저장되며 일반적인 자료들은 문서형태로 구성되어 워드프로세서 가용한 파일 포맷을 이용하여 저장된다.

우리는 일반적으로 정형화된 정보에 더 많은 관심을 갖는다. 정형화된 정보는 대부분 그 양이 방대하여 데이터를 활용하여 여러 가지 작업들이 가능하다. 정보의 이용 측면에서 비정형 정보보다 높은 가치를 갖는다고 할 수 있다. 이러한 정형 정보를 저장하는 방식 중 가장 지배적으로 사용되는 두 가지 중 하나는 MS社의 Office 제품군 중 하나인 Excel 이며 다른 하나는 Database 이다.

Excel은 사용편의성 차원에서 누구나 쉽게 다룰 수 있고 필터링, 피벗, 목표 값 찾기 등의 다양한 기능을 갖추고 있다. Database의 경우는 방대한 자료의 처리와 응용 프로그램과의 연동 등이 용이하다는 장점을 갖고 있다.

따라서 자료의 사용 목적이나 규모에 따라 DB형태로 혹은 Excel파일의 형태도 상이하게 관리되고 있다. 그런데 엑셀을 이용한 데이터의 관리의 문제점은 정보의 중복이라는 문제점을 가지고 있으면 따라서 Excel 파일로 작성된 정보들은 DB를 통한 통합관리가 사용자 편의성 측면에서 필요하다. 여기서 문제점은 이 두 가지 지배적인 정형정보 저장수단 사이의 자료변환을 Excel 과 DB 어느 쪽도 제공

하지 않는다는 것이다.

본 연구는 이러한 문제점을 다룬다. 해결 방안으로서 Excel 파일을 핸들링 할 수 있는 POI API를 이용한 Web 기반의 엑셀-Oracle DB 변환 기법을 제안한다. POI API는 Apache 그룹의 Jakarta 프로젝트의 하나인 POI 프로젝트의 결과물로서 Microsoft社의 OLE 2 Compound Document Format에 기반한 파일들에 대한 접근을 가능하게 하여 Excel 뿐 아니라 Word, Powerpoint 등 MS Office 제품군의 어플리케이션에서 사용되는 모든 파일을 핸들링 할 수 있게 해주는 API 이다.

본 논문의 구성은 다음과 같다. 2장에서는 csv 파일을 이용한 변환 기법의 단점을 기술한다. 3장에서는 POI API에 대하여 설명한다. 4장과 5장에서는 Excel-Oracle DB 변환 프로그램의 설계와 구현에 대하여 설명하고, 6장에서 본 논문을 마무리 한다.

### 2. sqlloader 를 이용한 엑셀파일 변환의 한계

Oracle DBMS 에서는 sqlloader 를 이용하여 엑셀파일을 DB에 저장할 수 있는 기능을 제공한다. sqlloader 를 이용하는 방법은 다음과 같다.

1. Excel 의 '다른이름으로 저장' 기능을 이용하여 .xls 파일을 .csv 파일로 변환.
2. 파일의 각 컬럼 속성을 적용한 테이블을 DB 상에 생성.
3. .csv 파일을 .ctl 파일로 변환.
4. sqlldr 명령을 이용한 .ctl 파일 DB로 load.

위와 같은 4단계의 걸친 변환작업을 통해서 엑셀파일을 DB로 변환하는 방법에서 여러 문제점을 발견할 수 있다. 먼저 단계가 매우 복잡하며 엑셀과 DB를 오가며 일일이

<sup>+</sup> 본 연구는 지식경제부 및 정보통신연구진흥원의 대학 IT연구센터 지원사업의 연구결과로 수행되었음 (IITA-2008-(C1090-0801-0046))

변환과 명령 실행을 수행하여야 한다. 파일이 한 두 개가 아닌 이상 이런 작업은 매우 비효율적으로 보인다. 또한 .xls → .csv → .ctl → DB 의 여러 단계에 걸친 변환 과정에서 .csv, .ctl 파일을 생성해야 하는데 이 파일들은 특별한 경우를 제외하고는 삭제되어야 하는 파일들이다.

또한, 이 같은 작업은 독립적인 시스템에서는 가능 할지 몰라도 네트워크상에서 웹을 기반으로 구현하기는 거의 불가능 하다. 그리고 변환하고자 하는 시트의 칼럼들을 미리 알아야만 수행 가능하다는 것도 제약사항이다. 사용자가 엑셀파일의 칼럼정보를 이용해 테이블을 생성해야 하므로 결국 수동적인 방법으로 변환이 이루어진다는 것을 의미한다.

sqlloader 를 이용하는 방법 외에도 MS Access를 경유하여 DB로 보내는 방법도 가능하다. 혹은 DBA 를 위해 오라클에서 제공하는 툴을 이용하는 방법도 있다. 하지만 이 방법들 또한 각각 취약점들을 가지고 있으며 그러한 이유로 잘 사용되지 않고 있다.

### 3. Apache Jakarta POI API 의 활용

Apache 그룹의 Jakarta 프로젝트의 하나인 POI[1] 는 MS社의 OLE 2 Compound Document Format에 기반하는 파일들에 접근할 수 있게 하는 API를 제공하고 있다.

OLE 2 Compound Document Format은 Microsoft office군 어플리케이션 문서에 의해 사용되므로 POI API 를 통해 Excel, Powerpoint, Word 파일에 접근이 가능하다. 표 1은 POI API 를 구성하는 Sub-API들 이다.

<표 1> POI API의 Sub-API

API	Description
POIFS	the set of APIs for reading and writing OLE 2 Compound Document Formats using (only) Java.
HSSF	the set of APIs for reading and writing Microsoft Excel 97(-XP) spreadsheets using (only) Java.
HWPFP	the set of APIs for reading and writing Microsoft Word 97(-XP) documents using (only) Java.
HSLF	the set of APIs for reading and writing Microsoft PowerPoint 97(-XP) documents using (only) Java.
HPSF	the set of APIs for reading property sets using (only) Java.

HSSF 클래스에는 엑셀 파일에 접근하기 위한 메소드들이 정의되어 있다. POIFS 클래스의 POIFSFileSystem를 이용하면 파일 생성과 이미 생성되어 있는 파일에 대한 접근이 가능하다.

### 4. Web 기반의 Excel-Oracle DB 변환 설계

session을 이용하여 로그인과 회원가입이 가능한 웹페이지를 선구축하고 사용자가 신규회원이입을 하게 되면 사용자 정보를 DB 'MEMBER' 테이블에 추가한다. 회원

가입과 동시에 회원 ID와 동일한 이름을 갖는 폴더를 server에 생성한다. 이 폴더는 회원 개개인의 root 디렉토리 역할을 하게 된다. 이러한 폴더 필요한 이유는 사용자들이 각각 자신의 파일을 관리할 수 있게 하며 각자의 폴더에 저장되어 있는 파일을 선택해서 DB로 변환 할 수 있게 하기 위한 것이다. 사용자는 자신의 폴더에 파일을 업로드 할 수 있으며 삭제 또는 업데이트된 파일로 변경할 수 있다[2].

사용자 폴더에 저장된 파일을 DB로 변환하는 과정에서 3가지 선택사항을 제공한다. 엑셀 파일의 각 행을 DB에 저장하기 전 테이블 생성이 우선되어야 하며 테이블을 생성하는 방식은 3가지로 나누어진다.

첫 번째는 자동모드이다. 사용자는 자동모드를 통해서 편리하게 엑셀파일을 DB로 변환 할 수 있다.

두 번째는 머리글 행 모드이다. 엑셀 파일에 포함된 각 시트의 가장 첫 번째 행에 각 칼럼의 이름이 명시되어 있을 경우 사용할 수 있다.

세 번째는 사용자선택 모드이다. 각각의 칼럼 이름과 데이터 형을 사용자가 직접 선택하여 테이블을 생성할 수 있게 한다.

자동모드를 통해 생성되는 테이블은 내부 알고리즘에 의해 생성 쿼리를 작성한다. 먼저 첫 번째 행이 각 칼럼들의 이름을 나타내고 있는지 판단한다.

판단 기준은 다음과 같다.

1. 첫 번째 행에 포함된 모든 셀의 타입이 문자열이다.
2. 두 번째 행과 세 번째 행의 임의의 i번째 셀의 타입이 서로 같다.
3. 두 번째 행에 문자열 타입을 갖는 셀이 한 개 이상 존재한다.

위의 판단 기준을 모두 만족할 경우 첫 번째 행의 각 셀에 들어있는 문자열을 테이블 생성 시 칼럼 이름으로 사용하며 각 칼럼의 데이터 형은 두 번째 행의 셀 타입을 참조하여 결정한다.

첫 번째 행이 각 칼럼의 이름을 나타내고 있지 않다고 판단되면 각 칼럼명은 "COLUMN0, COLUMN1, ... COLUMNn "과 같이 칼럼 개수에 따라 번호만 부여하게 되며 칼럼 데이터 형은 일괄적으로 VARCHAR 로 설정하여 테이블을 생성시킨다.

두 번째 선택모드인 머리글 행 모드는 첫 번째 행을 읽어 들여 칼럼 이름을 정하고 두 번째 행의 각 셀 타입을 해당 칼럼의 데이터 형으로 하여 테이블을 생성한다.

세 번째 선택모드인 사용자선택 모드는 사용자가 각 칼럼의 이름을 기입하고 해당 칼럼의 데이터 형을 NUMERIC, STRING, DATE 중에서 선택하게 하여 테이블을 생성 한다.

### 5. Excel-Oracle DB 변환 Architecture 및 구현

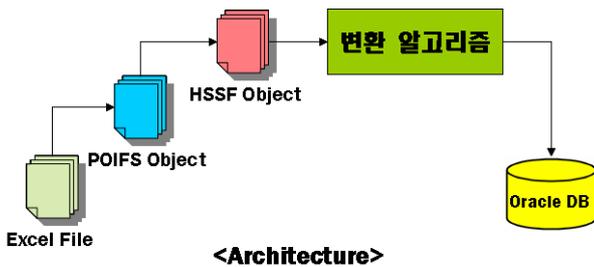
엑셀-DB 변환의 첫 번째 단계는 서버에 업로드 된 엑셀파일을 이용하여 POIFSFileSystem 객체를 생성하는 것

이다. 기존 파일을 이용하여 FileInputStream 객체를 생성하고 new POIFSFileSystem 의 생성자를 이용하여 새로운 POIFSFileSystem 객체를 생성한다.

두 번째 단계는 생성된 POIFSFileSystem 객체를 이용하여 HSSFWorkbook 객체를 생성하는 단계이다. 이때도 첫 번째 단계와 마찬가지로 생성자를 이용하여 생성한다.

세 번째 단계는 변환 알고리즘을 통해서 DB로 저장하는 단계로 두 번째 단계에서 생성된 HSSFWorkbook 객체의 getSheetAt() 메소드를 이용하여 파일에 포함된 각각의 시트에 대한 변환을 수행한다. 이번 단계에서는 DB와의 연결이 필요하며 JDBC를 통해서 Oracle DB와의 연결을 생성한다. DB로 저장에 앞서 DB상에 테이블을 생성하게 되는데 테이블의 생성에 변환 알고리즘이 사용된다. 변환 알고리즘의 가장 큰 역할을 파일정보와 사용자의 모드선택 정보를 이용하여 각 칼럼명을 설정하고 해당 칼럼에 대한 데이터형을 지정하는 것이다. 또한 지정된 데이터형을 이용하여 데이터 입력을 위한 쿼리를 작성한다.

3단계에 걸친 변환을 그림으로 나타내면 아래와 같다.



(그림 1) POI를 이용한 Excel-Oracle DB 변환기 구조

이 같은 엑셀-Oracle DB 변환을 위해 excelToDB 클래스를 정의하였다. excelToDB 클래스의 생성자는 다음과 같다.

```
excelToDB( Connection conn , String filePath, String fileName)
```

생성자는 변환한 파일이 위치한 경로와 파일명을 전달 인자로 하여 변환을 위한 POIFSFileSystem 객체 생성과 HSSFWorkbook 객체 생성을 담당한다.

실질적인 변환은 pushToDB() 메소드에서 처리된다. 생성자에서 생성한 HSSFWorkbook 객체의 getSheetAt() 메소드를 이용하여 각 시트에 접근한다.

pushToDB() 메소드는 사용자 선택모드에 따른 처리를 수행하기 위해 두 개의 메소드를 오버로딩 하였다. pushToDB(int , String, String[] ) 과 pushToDB(int, String) 두 개의 메소드가 변환을 수행하는데 첫 번째 pushToDB() 메소드의 경우는 사용자가 지정한 칼럼명과 해당 칼럼의 데이터형을 2차원 배열로 전달 받아 테이블 생성과 DB로의 데이터 저장을 수행한다. 두 번째 pushToDB() 메소드는 사용자 선택 모드가 아닌 경우의 변환에 사용된다. pushToDB() 메소드는 첫 번째 인자로

시트번호를 두 번째 인자로 테이블 이름정보를 전달 받는다.

## 6. 결론 및 향후 연구

이상의 연구에서 Excel-Oracle DB 변환 기법에 대한 설계와 구조에 대해서 알아보았다. Excel은 실제 가장 널리 사용되는 정형정보 저장 수단으로 기능적인 면에서 우수하지만 DB로의 변환을 통해 관리되어 사용자의 편의성을 확장 할 필요가 있다. POI API는 Excel-DB 간 변환을 구현하는데 유용한 수단이지만 변환의 과정에서 DB와의 데이터형 일치와 테이블 생성에 관련한 이슈들이 존재한다. 본 연구에서는 이러한 변환 상의 이슈에 대한 기본적인 방향과 기법을 알아보고 단순화된 아키텍처를 통해 구현하였다.

Excel-Oracle DB 간 변환 기법의 연구는 확장되어 복수의 DB와 연결을 통해 그 효과를 극대화 시킬 수 있다. 이와 관련하여 Oracle에서는 Jakarta API 를 이용하여 여러 개의 오라클 DB로부터 데이터를 읽어서 엑셀 리포트를 작성하는 방법에 대해 기술해 놓고 있다[3]. 자바 1.4부터 추가된 XML process 기능을 이용하여 접근 가능한 DB의 접속 configuration 을 정의하고 정의된 접속 설정을 이용하여 여러 DB에서 데이터를 읽어 하나의 엑셀 파일로 구성하게 한 것이다. 이를 적용하여 서비스에 응용하면 더욱 활용도가 높은 서비스의 제공이 가능 할 것이다.

## 참고문헌

- [1] Andrew C. Oliver, Glen Stampoultzis, Avik Sengupta, Rainer Klute, Apache POI project home page, <http://poi.apache.org> (Dec 1. 2007), Overview
- [2] 조행남 “Oracle and JSP programming” 컴윈미디어 서울 2007
- [3] Casimir Saternos, Technology Network of Oracle Home page, [http://www.oracle.com/technology/pub/articles/saternos\\_broadcast.html](http://www.oracle.com/technology/pub/articles/saternos_broadcast.html) (Dec 14, 2007)