

# SGML 문서를 XML 문서로 변환하는 사례 연구

## (A Case Study for Migration from SGML Document to XML Documents)

조민호<sup>†</sup> 류성열<sup>\*\*</sup> 박시형<sup>\*\*\*</sup>

(Min Ho Cho) (Sung Yul Rhew) (Si Hyoung Park)

**요약** 최근 인터넷을 기반으로 한 정보 환경의 범위는 단순한 정보 제공의 차원을 넘어서서 비즈니스를 구성하는 핵심 부분에 까지 확산되고 있다. 특히, 마크업 언어에 근거한 기술은 WWW기술의 확산과 함께 인터넷 기반의 비즈니스에서 매우 중요한 부분으로 부각되고 있다. 하지만, 대표적인 마크업 언어인 SGML로 구성되어 있는 자료는 SGML 브라우저를 이용하여 보아야 하기 때문에, 웹 브라우저를 기반으로 하는 인터넷 사용자에게 정보를 제공하기 어렵다. 더구나, SGML 브라우저 간의 호환성이 결여되어서, 상호 자료의 공유에도 많은 문제점을 안고 있었다. 이러한 점에 착안하여, 이번 연구에서는 기존의 SGML 자료를 SGML의 특성을 유지하면서 웹에 적용할 수 있도록 XML 자료로 변환하는 것에 대하여 사례 연구를 통해 상세한 기법을 제시 하였다. 이번 연구의 내용은 동일한 구조를 가지는 3Tera의 SGML 자료에서 600MB의 SGML 자료를 선별하고 이를 기반으로 수행한 Pilot Project의 결과이다. 이번 연구에서 개발된 기법은 SGML 자료를 XML 자료로 변환하는 것과, Migration을 위한 전체 시스템 구조 그리고, Migration된 SGML 문서에 대한 Web 환경에서의 Access 방법으로서, 기존의 SGML 방식보다 자료 표시 속도를 줄일 수 있고, Mobile Computing이 가능하며, 기존 자료의 변환에 따른 업무의 연속성 및 자료의 공유를 보장하는 기법이다. 그러므로 이러한 분야에 관심을 가지는 실무자들에게 매우 도움이 되리라 생각된다.

**Abstract** Recently, The range of Internet based information environment is spreading over core business area, as well as simple information provision area. Especially, with spreading WWW technology, markup language based technology is emerging as an important part in Internet based business. But, the data made by SGML can only see by using SGML Browser, so it has some problem in information providing at Internet, and compatibility of data between Data source. So, this study suggests essential architecture and technique for migrating from SGML to XML environment. In our study, we use 600MB SGML data that are selected from 3Tera DataBase of SGML as testing target for migration. We can reduce data displaying time after migration, can do mobile computing which is based on Internet as a result of this study. And the same technique and idea that is used in this study can apply to more large SGML Environment without changing. So, It will be very helpful to the reader who is interesting to migrate from SGML doc to XML doc.

### 1. 서론

최근 인터넷의 빠른 성장과 함께 W3C에 의해 표준으로 제정된 XML(Extensible Markup Language)의

중요성이 매우 커지고 다양해지고 있다[1][2][3][4]. SGML(Standard Generalized Markup Language)의 장점[5][6][7]과 HTML의 편리성을 결합한 XML의 등장은 현재 단순 정보 조회의 수준에 머물고 있는 웹 환경을 보다 넓고, 편리하고 다양하게 적용하는 핵심 기술로 자리잡고 있다[1][8]. 이러한 기술적인 흐름 속에서 기존의 SGML 사용자들도 자신의 환경을 XML 기반으로 이전하여 인터넷 환경을 이용하고자 하는 요구가 많이 발생하고 있다[9] 기존의 SGML은 출장과 이동 중에도 자료를 액세스할 수 있는 모바일 컴퓨팅 환경 구

<sup>†</sup> 비 회 원 : Openwave Korea Sales consultant  
cho.min-ho@openwave.com

<sup>\*\*</sup> 종신회원 : 숭실대학교 컴퓨터학부 교수  
syrhew@selab.soongsil.ac.kr

<sup>\*\*\*</sup> 비 회 원 : Datec 수석 컨설턴트  
shpark@datec.co.kr

논문접수 : 2001년 1월 22일

심사완료 : 2001년 9월 1일

현의 측면과 인터넷을 이용한 업체 독립적인 환경을 구현한다는 측면에서 실제 업무의 수행에 매우 큰 불편을 초래하고 있다. 이러한 어려움에도 불구하고, SGML을 XML로 변환하는 것이 실제적인 측면에서 매우 막연하다는 점이 SGML을 XML로 Migration 하고자 하는 사용자가 가지는 불만이다. 이러한 점에 착안하여, 실제적인 업무를 모델로 하여 SGML을 XML로 자연스럽게 변환하기 위한 여러 기술적인 측면을 제공하고자 하는 것이 이번 연구의 목적이다. 이번 연구에서 제공하는 내용을 요약하면 다음과 같다. 첫째, 변화에 따른 업무의 중단 없이, 운영 중인 SGML환경을 XML로 변환하기 위한 개념과 시스템 구성에 대한 방안을 제시한다.<sup>1)</sup> 둘째, SGML 데이터를 XML DataBase로 이동시키는 프로그램의 구성에 대하여 상세한 사항을 제공한다.<sup>2)</sup> 셋째, Web Browser를 이용하여 XML DataBase에 있는 자료를 Access하는 방법 및 구성에 대하여 자세한 정보를 제공한다.<sup>3)</sup> 이번 연구에서 제시한 기법은 SGML에 대한 폭 넓은 지식을 가지지 않아도 XML로의 자연스러운 변환이 가능하며, 사용된 기술과 개념은 보다 큰 SGML환경에서도 거의 그대로 사용할 수 있기 때문에, 실무적인 차원에서 환경 변화에 따른 업무의 연속성이 이어갈 수 있는 바람직한 방법이다.

## 2. 고객 환경 및 필요성

이번 연구의 대상이 되는 환경은 현재 SGML로 되어 있는 약 3Tera의 자료를 운영하는 환경이다. 현재 운영되고 있는 환경에 대한 요약된 구성은 그림 1에 제시되었다[5][6][10][11]. 고객 환경에서, 입력 받은 자료는 그림에 명시되지 않은 별도의 통합 서버에서 종합 관리하고 관리 중인 자료 중에서 그림 1에 명시된 클라이언트가 필요로 하는 자료는 선택적으로 SGML 서버로 복사된다. 이때, 통합 서버의 Relational데이터베이스에 저장되어 있던 자료는 SGML형태로 포맷이 변경된다. SGML로 변경되어 저장된 자료는 SGML 서버 세대가 공유하는 별도의 객체지향 데이터베이스에 저장된다[2][11]. 그리고 각 클라이언트는 자신이 속한 서버에 액세스하여 원하는 SGML 문서를 연다.

이때, 각 클라이언트는 저장된 SGML 문서를 보기

위하여, 별도로 개발된 SGML Browser를 사용한다.

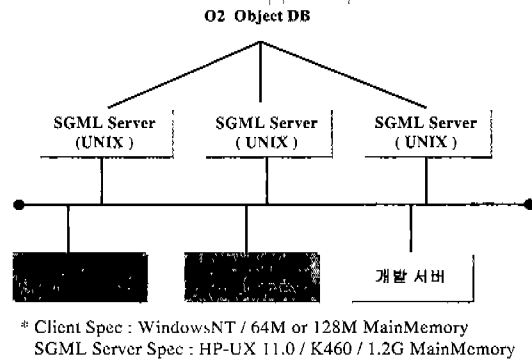


그림 1 SGML기반의 고객 환경

앞에서 언급한 현재의 사용 환경은 자체에서 개발한 SGML 브라우저의 사용자 인터페이스나, 관련된 SGML 파일의 구성 등이 사용자의 요구에 맞게 잘 설계되어 있으며, 현재 운영 중에 있다. 그러나, 기존 시스템에 저장된 자료를 액세스하는 기법이 SGML 전용 브라우저를 통해서만 가능하기 때문에, 출장이나 이동 중에 업무를 수행할 수가 없다. 즉, 인터넷 기반의 모바일 환경에 적합하지 않다는 점이 사용자의 불만 사항이다. 또한, SGML의 태그를 너무 복잡하게 구성하여서 전체적인 성능이 매우 떨어진다는 단점이 있다. 이러한 이유 때문에, 클라이언트 사용자들은 다른 환경 즉, 모바일을 지원하고 보다 빠른 속도로 액세스할 수 있는 환경을 원하게 되었다.

## 3. 운영 중인 SGML환경을 XML로 변환하기 위한 개념과 시스템 구성에 대한 방안

앞 절에서 언급한 것과 같이, 기존의 SGML환경은 좋은 사용자 인터페이스를 제공함에도 불구하고, 전용 브라우저와 복잡한 Tag를 사용해야 하는 단점 때문에, 모바일 환경과, 성능/유지보수에 문제점을 가지고 있었다[5][9][10]. 하지만, 이미 보유하고 있는 자료의 크기가 3Tera를 넘어서고 있는 상황에서 모든 자료를 한꺼번에 변환한다는 것은 매우 어렵고, 업무의 연속성 보장에 있어서 많은 위험성을 가지고 있었다. 이러한 이유로, Migration에 따른 업무의 진행 중단에 대한 위험성을 최소화 하면서 기존의 환경을 새로운 환경으로 이전하기 위한 변환 전략이 필요하게 되었다. 즉, 변환으로 인하여 업무의 공백이나, 자료의 분실이 발생하면 안 된다는 것

- 1) "3. 운영 중인 SGML환경을 XML로 변환하기 위한 개념과 시스템 구성에 대한 방안" 참조
- 2) "4. SGML 데이터를 XML DataBase로 이동시키는 프로그램의 구성" 참조
- 3) "5. Web Browser를 이용하여 XML DataBase에 있는 자료를 Access하는 방법 및 구성" 참조

이 전체 연구의 전체 사항이었고, 사용자들이 현재의 User Interface에 익숙해져 있으므로, 현재의 User Interface를 큰 변화 없이 그대로 유지하는 것이 또한 중요한 문제였다. 추가적으로, 현재 사용중인 O2 데이터베이스가 개발 회사의 문제로 인하여 더 이상 기술 지원을 받기 어렵기 때문에, 장기적으로 볼 때 O2 데이터베이스를 제거해야 한다는 것이 시스템 운영자의 입장에서 제시한 또 다른 기본 전제였다. 이러한 모든 점을 고려하여 본 연구에서 제시한 변환 기법은 그림 2에서 나타내고 있다.

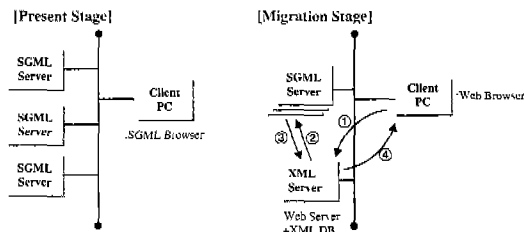


그림 2 SGML과 XML기반의 변환 개념 비교

그림 2에서 제시하는 변환기법은 현재의 SGML 서버와 객체지향 데이터베이스 시스템 환경은 그대로 유지하고, 사용자의 환경에서 SGML 브라우저를 웹 브라우저로 대체하는 것이다. 이때, 웹 브라우저에서 SGML 문서를 액세스하기 위하여, 중간에 XML 서버를 위치시켜 운영하는 개념이다. 이때, XML 서버에는 웹 서버와 XML 데이터베이스를 설치하여 운영 시킨다.

그림 2에서 제시한 변환 기법의 운영 절차를 분석하여 보면, 우선 클라이언트 컴퓨터는 웹 브라우저와 URI를 이용하여 XML 서버에 연결한다. 연결 후에 사용자는 원하는 SGML 문서에 대한 요청을 XML 서버에 하게 되고, 이러한 요청을 받은 XML 서버는 요청된 자료가 자체의 XML 데이터베이스에 있는 가를 검사하게 되고, 있는 경우에는 클라이언트 컴퓨터에 직접 서비스하고, 없는 경우에는 미리 정의된 프로그램을 수행하게 된다. 이때 수행되는 프로그램은 SGML 서버를 액세스하여, 클라이언트로부터 요청된 문서를 찾고, XML 서버로 가져와서 XML 데이터베이스에 저장한다. 저장이 완료된 후, XML 서버는 클라이언트 PC에 요청된 문서를 제공한다.

그림 2에서 제시한 기법은 기존의 자료에 대한 변화가 없이 단순히 클라이언트 컴퓨터에 대한 환경을 SGML 브라우저에서 웹 브라우저로 변화 시키고, XML 서버를 게이트웨이의 개념에서 제공하는 기법으로서 사용자나 관리자의 입장에서는 커다란 변화 없이 자연스럽게

변환하는 기법을 제공한다. 더구나, 사용자에 의해 한번 액세스된 SGML 파일은 XML 데이터베이스에 저장하게 되므로 3개월 정도의 사용기간이 경과한 후에는 SGML 서버의 중요한 SGML 파일이 모든 XML 데이터베이스로 자연스럽게 변환되어 일정 시간의 경과 후, 전체 SGML 문서의 변환 시에도 커다란 위험 부담이 없이 변환을 완료하게 된다. 이러한 환경의 성공적인 구현을 위하여는 두 가지 부분이 해결되어야 하는데, 그 중 첫째가 SGML DataBase에 있는 자료를 XML DataBase로 이전하는 부분이고, 두번째는 클라이언트에서 XML DataBase를 Access하는 부분이다. 다음 Chapter에서 이러한 부분에 대하여 상세히 기술한다.

#### 4. SGML 데이터를 XML DataBase로 이동시키는 프로그램의 구성

##### 4.1 현재 사용 중인 SGML 파일 구조와 문서 형태

이번 연구에 사용된 SGML 문서의 파일 구성 및 타입에 대한 내용은 그림 3에서 자세히 보여주고 있다. 특히, 그림 3에서 제시한 SGML 문서의 구조는 3 Tera 전체에 대하여 동일한 형태를 가지고 있다.

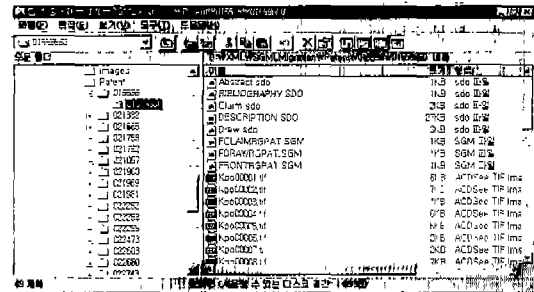


그림 3 SGML 파일 구조

그림 3의 정보는 이번 연구의 수행을 위해서, 기존의 SGML 서버에 저장되어 있는 SGML 문서를 Windows NT 환경으로 일부 이전한 화면이다. 문서의 일부는 NT 환경으로 이전하였지만, 기존의 SGML 데이터베이스에서도 동일한 구조를 가지고 저장되어 있으므로, 전체 자료의 구조에는 변화가 없다. 그림 3에 따르면 전체 SGML 문서는 Patent라는 디렉토리에 저장되어 있고, Patent 디렉토리는 각각의 SGML 문서마다 별도의 디렉토리를 가지고 있다. 즉, 각각의 SGML 문서는 위의 그림과 같이 각각의 디렉토리에 저장된다. (예 : patent\015596\10559660 디렉토리) 그리고, 표 1은 각각의 디

렉토리를 구성하는 파일에 대한 상세 정보를 보여주고 있다. 이번 연구의 대상이 된, SGML파일은 SDO연장자를 가지는 파일은 SGML문서 자체를 의미하고 있고, SGM연장자를 가지는 파일은 SGML에 대한 DTD를 의미하도록 구분하여 관리되고 있다.

앞에서 언급된 부분은 이번 연구에 관련된 고객의 특성이 반영된 SGML문서의 구성이며, SGML의 일반적인 개념은 아니다. 전체 내용의 이해를 돕기 위해 전체 파일 중에서 중요한 3개의 파일의 내용에 대한 것을 그림 4에 제시하였다.

표 1 디렉토리 구성

File name	Description
Abstract.sdo	전체 SGML 문서의 내용에 대한 요약 부분
bibliography.sdo	SGML문서를 구성하는 Header 부분
Claim.sdo	문서의 특정 부분 (예: 발명의 구성 및 작용 부분)
description.sdo	문서의 본문을 구성하는 부분
Draw.sdo	문서의 내용과 연계된 Image파일의 정보 관리 부분
*.SGM	위에 언급된 각종 자료의 연계관계를 보여 주는 부분

그림 4에 제시된 자료는 전체 SGML 파일을 구성하는 한 부분으로서, 한 개의 SGML 파일을 클라이언트에게 보여 주기 위해서는 그림 4에 제시된 3개의 파일을 주축으로 기타 다른 파일들이 연계되어야 한다. 이러한 사항은 XML로 변환한 경우에도 그대로 적용되는 것이어서 실제적인 변환을 위한 프로그래밍에서도 고려되어야 하는 부분이다.



그림 4 abstract.sdo, bibliography.sdo, description.sdo 파일 예

#### 4.2 SGML을 XML로 변환하는 프로그램

실제 SGML문서를 XML문서로 변환하기 위하여 제작된 folder.asp프로그램의 flow chart를 그림 5에 제공하였다. 그림 5에 제시된 folder.asp는 기존에 사용 중인 SGML 파일을 분석하여 본 결과를 바탕으로 하고 있다, 즉, 기존 SGML 파일을 XML로 Migration하는 데 있어서 문제가 되는 점은 첫째는, 기존의 SGML 문서 중에서 <EMI>, <B110> Tag에 Attribute가 있는데, Attribute를 둘러싸는 인용표시(“)가 빠져있다는 점이고, 둘째는, 기존의 SGML 문서 중에서 <EMI>, <DRAWREF> Tag에 close tag가 없다는 점이다. 이러한 점에 착안하여, folder.asp 프로그램은 SGML을 읽어서 위에 언급된 부분을 보강하고, 보강이 완료된 것을 XML DB에 기존의 Tag를 유지한 상태에서 저장하도록 제작하였다.

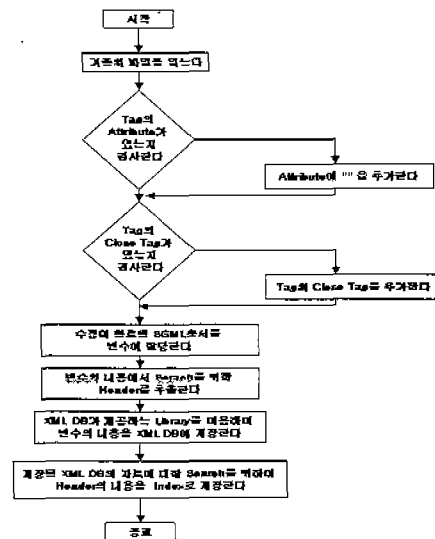


그림 5 folder.asp프로그램의 flow chart

실제 SGML문서가 여러 개의 파일로 나뉘어져 구성되어 있으므로, folder.asp 프로그램은 이들 각각의 부분을 나누어서 Tag와 Attribute를 보강하고, 보강이 완료된 부분을 실제로 하나의 XML문서로 병합하여 XML DataBase에 저장하게 된다. 이때, 보강이 완료된 부분을 실제로 하나의 XML문서로 병합하는 부분에 대한 Logic은 그림 6에 제시하였다. 그림 6의 내용은 XML 문서를 구성함에 있어서 필요한 부분은 임의로 보강하고, 기타 내용에 대한 부분은 BIBLIGRAPHY, ABSTRACT, DESCRIPTION, CLAIM, DRAW의 부

분으로 나누어서 별도의 보장을 거친 후에 update\_str 이라는 변수에 순서대로 할당하여, XML DataBase가 받아들일 수 있는 Validate XML 문서를 구성하는 실제적인 Logic을 담고 있다. 실제 환경에서 사용된 XML DataBase인 excelon의 경우에는 별도의 DTD가 없어도, XML문서를 직접 저장할 수 있으므로, 이러한 구조를 가지고 프로그램을 제작하였다. 만약, XML DataBase가 DTD를 요청하는 경우에는 기존의 DTD파일인 SGM연장자를 가지는 파일을 분석하는 별도의 Logic이 추가되어야 할 것이다.

```
set HResult = doc.selectNodes('//xlnml:real')
if Cint(HResult.item(0).text) = 0 Then

update_str = '<?xml version="1.0" standalone="yes"?' >'
update_str = update_str + '<!\[update version="1.0" \]' >'
update_str = update_str + '<update select="//Patents(1) \]' >'
update_str = update_str + '<element location="lastchild" \]' >'
update_str = update_str + '<PATENT \]' >'
update_str = update_str + '<B11D \]' >'sp_number' </B11D \]' >'
update_str = update_str + BIBLIOGRAPHY
update_str = update_str + ABSTRACT
update_str = update_str + DESCRIPTION
update_str = update_str + CLAIM
update_str = update_str + DRAW#
update_str = update_str + '</PATENT \]' >'

update_str = update_str + '</element></update></xlnupdate>'

```

그림 6 Folder.asp 중에서 변환이 끝난 부분은 한 개의 XML문서로 합병하는 Logic

4.3 SGML 변환을 위한 XML 데이터베이스 구조

본 연구를 위하여, 선택한 구조에서, SGML의 자료를 변환하여, 클라이언트에게 서비스하기 전에, XML 서버에 저장하는 부분은 전체 연구의 핵심을 이루는 부분이다.

XML 데이터의 저장과, 실제적인 자료 서비스를 하기 위하여, XML 데이터베이스의 사용은 필수적인 부분이고, 실제적인 구현에는 eXcelon 데이터베이스가 사용되었다. 실제로 4.2에서 제공한 folder.asp를 수행한 후에 생성된 XML DataBase에 대한 상세한 정보를 그림 7에 제시하였다. 그림 7의 화면은 excelon database에서 제공하는 탐색기 화면이다. 그림 7의 내용을 분석하면, 기존의 SGML문서는 모두 "PSMHCHO95/kipo/patent"에 저장되어 있고, 실제 저장된 형태는 화면의 오른쪽에 제시하였다. 추가적으로, 저장된 자료의 탐색을 위하여 생성된 Index정보가 "PSMHCHO95/kipo/header"에 저장되어 있다. Header에 저장된 정보는 나중에 일반 사용자가 Web Brower를 이용하여 내용을 조회하는 경우에 사용하게 될 것이다.

5. Web Browser를 이용하여 XML DataBase에 있는 자료를 Access하는 방법 및 구성

5.1 새롭게 개발된 웹 브라우저 화면 분석

SGML의 데이터가 XML 데이터베이스로 변환되면, 그 다음에 남은 과정은 웹 브라우저를 통하여 XML 데이터베이스에 있는 자료를 보여주는 부분이다. 그림 4에 제시된 화면과 같이, 실제적으로 한 개의 SGML 문서는 여러 개의 파일로 구성되어 있다. 이러한 구조에서 여러 개의 파일을 하나로 묶어서 사용자에게 보여주는 것은 성공적인 Migration을 위하여 매우 중요한 부분이다. 본 연구에서 수행한 결과에 대한 것을 그림 8에 제시하였다. 그림 8에 제시된 화면은 실제 웹 브라우저에서 eXcelon을 액세스하여 보여지는 화면이다. 그림 8의 화면에서 확인할 수 있는 바와 같이, 아무런 검색 조건이 없는 경우에는 전체의 문서가 검색되고, 검색에 대한 키워드를 제시하는 경우에는 이러한 검색에 맞는 SGML 문서가 검색이 되게 된다. 검색의 조건은 AND와 OR을 모두 사용할 수 있으며, 검색 엔진은 eXcelon 자체에서 제공하는 것을 사용할 수도 있고, 별도의 검색 엔진을 사용하여 검색을 수행할 수도 있다.

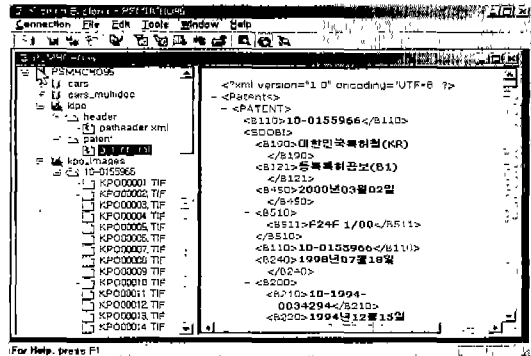


그림 7 Migration된 SGML문서의 형태

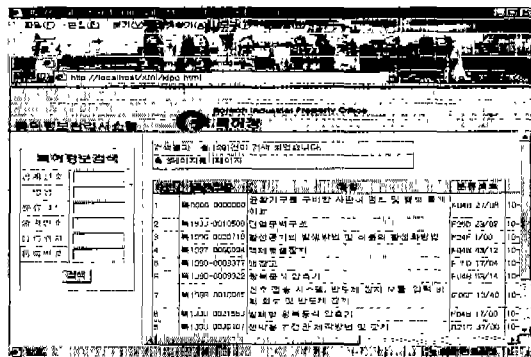


그림 8 클라이언트 브라우저의 초기 접속 화면

이러한 검색은 XML 데이터베이스에서 제공하는 기능을 이용하여 수행하도록 되어있으며, 실제로 eXcelon의 경우에는 자체에 검색엔진과 XQL의 기능을 가지고 있다[12]. 그리고, 위에서 제시된 화면은 기존의 SGML 브라우저를 사용하는 경우에 제시되는 화면을 근거로 거의 동일하게 Display 되도록 제작되었다. 이를 위하여, 기존의 SGML 문서에서 사용된 태그를 HTML로 매핑하는 작업이 필요하며, 이러한 모든 과정은 제공되는 ASP 프로그램에서 수행된다. 이를 위한 실제적인 ASP 프로그램의 일부는 그림 10에 제시되어 있다. 그림 8에서, 특정 문서를 선택한 경우에 화면에 문서가 Display 되는 것에 대한 사례는 그림 9에 제시하였다. 이때, 사용한 기법도 그림 8에서 사용한 것과 동일한 요령으로 구현되었고, 역시 ASP로 프로그램 되었다.

이 부분을 수행하는 프로그램에 대한 것을 그림 10에 제시한다.

```
<html>
<head>
</head>
<frameset rows="79,*" border=0>
  <frame name="topFrame" src="topmenu.html"
    scrolling="no">
  <frameset cols="175,*" border=0>
    <frame name="leftFrame" src="leftmenu.asp"
      scrolling="no">
    <frame name="rightFrame" src="patientlist.asp"
      scrolling="yes">
  </frameset>
</frameset></noframes></noframes>
</html>
```

그림 10 클라이언트 화면을 위한 소스 코드

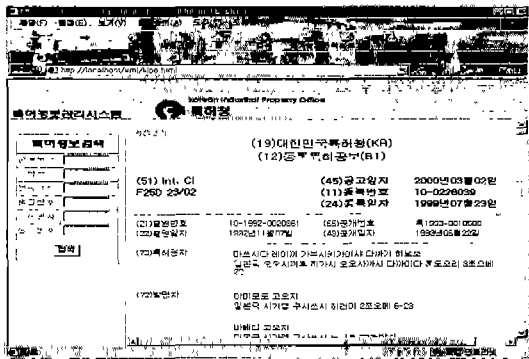


그림 9 선택된 문서의 내용보기 화면

그림 9에 제시된 결과에서 알 수 있는 것처럼, 여러 개의 파일로 구성되어 있는 XML파일을 모아서 사용자에게 친숙한 화면을 제공할 수 있다. 이러한 방식을 사용하는 이유는 기존의 SGML 문서에 대한 변경을 최소화하고, SGML의 변환을 지원하는 것이 가장 중요한 이유이다. 이와 같은 개념으로 시스템을 구성하면, SGML의 파일을 동시에 변환하지 않고, SGML 서버와 사용자간에 XML 데이터베이스 서버를 위치하고도 실시간으로 사용이 가능하며, 기존의 SGML 브라우저도 동시에 사용할 수 있는 장점이 있다. 그리고, 화면의 변화가 최소화되어서, 기존의 사용자들의 반발을 방지할 수 있는 장점이 있다

**5.2 웹 브라우저에 XML문서를 보여주는 프로그램 분석**

실제 SGML의 태그를 HTML의 태그구조에 맞게 매핑하여 Display하는 부분을 담당하는 ASP프로그램은 전체 시스템의 부분 중, 많은 부분을 담당하는 곳이다.

그림 10에서 제시한 HTML 파일의 내용을 보면 확인할 수 있는 것처럼 그림 8에서 제시된 처음의 화면은 3개의 부분으로 구성된다. 첫째가 topmenu.html, 둘째가 leftmenu.asp, 셋째가 patentlist.asp이다. 이 중에서 patentlist.asp는 해당되는 SGML파일을 찾아서 그림 8의 오른쪽 부분과 같이 list형태로 보여주는 ASP 프로그램이다. 그림 10에 제시된 프로그램의 핵심은 기존에 사용하는 SGML Browser의 화면을 보고, 이에 해당하는 SGML 태그를 분석하는 것이다. 즉, 기존에 사용했던 SGML 태그의 역할과 동일한 HTML 태그를 찾아내서 상호 매핑하는 것이다. 이러한 Mapping에 대한 상세한 정보는 그림 11에 제시하였다. 즉, XML로 Migration된 자료는 아직도 SGML에서 사용하던 Tag를 그대로 가지고 있다. 이러한 문서를 Web Browser에서 보여주기 위해서는 그림 11처럼, 해당되는 Tag를 특정 변수에 할당하고, 이러한 Tag가 실제 화면에서 어떻게 보여지는가를 분석하여, Web Server에서 제공하는 함수를 이용하여 그림 11과 같은 조작을 수행한다.

```
Set NDRDES = objSelectNode(objPATENT/SDDDE/DRDES/P)
If NDRDES.length > 0 Then '도면의 간단한 설명-
*
  Response.write "<font color=red><b>도면의 간단한 설명</b></font><br>"
  Response.write "<Table border=0><TR><TD width=600><font size=2>".
  for i=0 to NDRDES.length-1
    Response.write Replace(NDRDES.Item(i).text, "&nbsp;" & "&nbsp;")& "<BR>"
  next i
  Response.write "</font></TD></TR></Table>"
End If
```

그림 11 XML DB의 자료를 Web Browser에 보여주는 프로그램(일부)

그림 11의 경우에서 볼 수 있는 것처럼, NDRDES Tag는 기존의 SGML Browser에서 Red and Bold의 특성을 가지고 Table 형태로 Display되기 때문에, 이번 연구에서는 IIS Web Server의 Response 객체에 Red and Bold의 특성을 부여하고, Table 형태로 Display하도록 ASP로 Program 하였다

이러한 과정을 통하여, SGML에 대한 깊은 이해가 없어도, 기존 사용 환경을 기반으로 XML로의 변환을 수행할 수 있는 것이다. 실제 업무를 대상으로 수행하여 본 결과, 기존에 사용하는 SGML Browser의 화면을 거의 동일하게 Web Browser에서 구현할 수 있었으며, 화면이 동일하고, 사용법이 같기 때문에, 기존 사용자에 대한 재교육이 필요 없을 정도로 구현할 수 있었다.

## 6. 평가

이번 연구에서 시도한 것은 앞의 내용에서 확인할 수 있는 것처럼, SGML을 사용하는 사용자가 웹에서 동일한 업무를 할 수 있도록 하기 위하여 XML기술을 적용하고 구현하기 위한 상세한 방안을 제시하는 것이다. 그리고 이와 더불어 현재 사용자가 가지고 있는 문제점을 해결함으로써보다 나은 환경의 구현을 목적으로 하였다.<sup>4)</sup>

이를 위하여 수행한 Pilot 연구의 결과는 다음과 같다.

- 기존의 사용자는 업무의 정지 없이 SGML에서 XML로 자연스러운 환경의 변화가 가능한 시스템 구조를 가지게 되었다
- 기존에 사용하는 SGM Browser화면과 거의 동일한 화면을 제공 받게 됨으로써, 재교육이 필요하지 않은 환경을 가지게 되었다
- 기존 사용자의 컴퓨터에 Web Browser만을 설치하면 업무의 수행이 가능하게 되어서, 전체적인 시스템의 관리비용이 절감되었다
- Web을 기반으로 하는 환경을 구현함으로써, Mobile 환경을 지원하는 시스템 구성을 가지게 되었다
- 기존의 SGML Browser를 사용할 때 보다, 30% 이상 성능이 향상되었다. 이는 기존의 환경과 비교하여 Access하는 속도, 즉, 요청된 자료가 화면에 Display되는 속도를 비교한 것이다.

추가적으로, 이번 연구에서 제시된 방법은 SGML에 대한 깊은 지식을 가지지 않아도 적용될 수 있는 기법이며,<sup>5)</sup> 사용된 기술이 현재 산업계에서 널리 사용되고 있는 Windows NT기반의 IIS서버와 ASP 언어가 사용

됨으로써, 보다 많은 사람들이 이해하고 활용할 수 있는 특성을 가지고 있다

이번 연구의 수행 결과를 SGML에서의 대표적인 Migration Tool인 omnimark와 비교할 때, omnimark는 일반화된 Package로서, 다양한 분야에 적용될 수 있는 많은 기능을 가지고 있다. 예로서, 다양한 Script와 Middleware로서의 기능을 가지고 있다. 그래서, omni-mark를 적용한다고 하여도 거의 비슷한 결과를 얻을 수 있으리라 판단되었다. 하지만, 이번 연구의 전체 사항은, 현재 사용 중인 SGML자료를 존속시키는 것이 아니라, 완전히 Migration시키는 것이고, 또한, Migration대상이 되는 3 Tera의 SGML문서가 1개의 Format에 따라 만들어 졌고, 현재 사용 중인 User Interface의 모양을 Web Browser상에서 동일하게 구현하는 것이었기 때문에, OmniMark를 쓰는 것보다, 실제적인 Program을 작성하는 것이 예산이나 시간적인 측면에서 훨씬 효율적인 방법이다[13].

추가적으로, 기존의 excelon이나 다른 제품에서 제공하는 변환기능을 이용하는 방식도 고려되었지만, 기존의 SGML문서가 Tag의 구조 등이 완벽하지 않아서 자동변환이 되지 않고, 용량이 많아서 Migration에 소요되는 시간이 길고, 실패하는 경우에 업무의 공백이 우려되어서 이번 연구에서는 제외되었다. 하지만 다른 전제를 가지는 연구에서는 시도할 만한 가치가 있다고 판단된다.

## 7. 결론

SGML은 전세계적으로 중요 관공서를 중심으로 많이 활용되고 있다. 그리고 많은 SGML 사용자들이, 자신의 환경을 인터넷 기반의 환경으로 이전하고자 원하고 있다. 하지만 이를 수행하기 위한 보다 상세하고 실무적인 방법에 대한 가이드라인은 거의 전무한 것이 현재의 실정이다. 이러한 점에서, 본 연구에서 제시한 방법은 실제적으로 SGML을 XML로 변환하고자 하는 사용자에게 자료의 이전에 관한 사례로써 많은 도움이 되리라 생각된다.

이번 연구의 대상이 된 SGML데이터베이스는[10][11] 각각 단위 SGML문서의 구조가 모두 동일하기 때문에, 전체 3Tera의 자료를 대상으로 하지 않고, 고객이 제공한 600Mbyte에 해당하는 자료를 대상으로 수행한 것이다. 그래서, 전체 DataBase를 대상으로 하여도, 이번 연구에서 제시한 시스템 구성이나 개념은 거의 변동 없이 적용될 수 있다. 이러한 점에서 이번 연구에 대한 전체 시스템 구성도를 그림 12에 제시하였다. 그림 12는 이번 연구에서 수행한 3가지 분야 중에서, Architecture부분을 제외한 실제적 프로그램에 해당되는 2개 부분을 중

4) "2. 고객 환경 및 필요성" 참조

5) "5. Web Browser를 이용하여 XML DataBase에 있는 자료를 Access하는 방법 및 구성" 참조

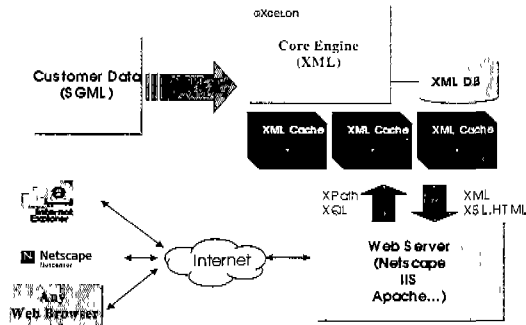


그림 12 전체 시스템 수행 개념도

함하여 요약 제시하고 있다.<sup>6)</sup>

향후, 전체 사용자를 대상으로 하는 연구의 수행을 하게 되는 경우에는 아키텍처의 변동이 없이 개발 환경을 현재의 NT기반에서 UNIX기반으로 바꾸어 적용하고자 한다. 그리고 UNIX 환경에서의 적용이므로 ASP보다는 자바를 기반으로 하는 JSP를 적용하여 동일한 알고리즘을 구현하고자 한다[2][3][12].

이번 연구의 특이한 점은 인터넷에서 사용되는 표준적인 기술을 그대로 적용했다는 점이다. 즉, ASP를 활용함으로써, Web 환경과 WAP환경을 동시에 지원할 수 있는 기반을 조성했다는 점이 특이할 만 한 점으로 강조될 수 있을 것이다. 이는 향후 JSP를 이용하는 경우에도 그대로 적용될 수 있다. 이러한 기술적 가능성을 기반으로, 향후 WAP을 지원하는 것에 대한 추가적인 연구가 필요하다고 판단되며, 이미 ASP/JSP가 WML을 지원하기 때문에 기술적인 어려움은 없을 것이라 판단된다[14][15]. 단, 핸드폰과 같이 조그만 화면에서 사용자의 화면을 구성하는 것에 대한 것은 GUI의 설계에 있어서 보다 많은 연구가 필요하리라 생각된다.

참 고 문 헌

[1] Rohit Khare and Adam Rifkin, XML:A door to automated Web application, IEEE 인터넷 computing, July-August 1997.  
 [2] Brett McLaughlin, Java and XML, O'Reilly, 2000.  
 [3] Light Richard, North Simon, Allen Charles, Presenting XML, Macmillan Publishing, 1997.  
 [4] W3C Homepage, xml. at http://www.w3c.org  
 [5] Bryan,Martin, SGML:An Author's Guide to the Standard Generalized Markup Language, Addison-Wisley 1988.

6) "1. 서론" 부분 참조

[6] Von Hagen, Williams, SGML for Dummies, IDG Books Worldwide Inc, 1997.  
 [7] Danny R. Vint, Daniel Vint, SGML at Work, Prentice Hall. 1988.  
 [8] Alex Cepenkus, Faraz H Object, Applied XML:A Toolkit for programmers, John Willey & Sons, 1999.  
 [9] 박기영, "SGML 문서의 XML 문서로의 변환에 관한 연구", 경상대 대학원 2000.  
 [10] 정연수, "O2기반의 XML 문서관리 시스템 설계 및 구현", 충북대 대학원 2000.  
 [11] Nextech HomePage, O2, at http://www.nextech.co.kr  
 [12] Datec Homepage, excelon, at http://www.datec.co.kr  
 [13] Omnimark HomePage,omnimark, at http://www.omnimark.com  
 [14] Charles Arehart 外 14인, Professional WAP, Wrox, 2000.  
 [15] Mark Baartsc 外 12인, Professional ASP XML, Wrox, 2000.  
 [16] O'REILLY XML Homepage, xml, at http://www.xml.com/xml  
 [17] SGML World, SGML. at http://user.chollian.net/~h5526366/index.html  
 [18] SoftwareAG HomePage, tamino, at http://www.softwareag.com  
 [19] Oracle HomePage,oracle 8i, at http://www.oracle.com



조 민 호  
 1963년생. 1989년 인하대 산업공학과 졸업. 1998년 숭실대 정보과학 대학원 졸업. 현재 숭실대 박사과정 재학중. 1989년 ~ 2000년 HP Korca Sales consultant. 2000년 ~ 현재 Openwave Korea Sales consultant. 관심분야는 XML, Semantic Web

류 성 렬  
 정보과학회논문지 : 컴퓨팅의 실제 제 7 권 제 4 호 참조



박 시 형  
 1966년생. 1991년 한양대 전자공학과 졸업. 1991년 ~ 1992년 삼성 종합 기술원 선임연구원. 1992년 ~ 1998년 인성 정보 Technical consultant. 1998년 ~ 현재 Datec 수석 컨설턴트. 관심분야는 XML, B2B, B2B Workflow