

SM-GUI: 그리드 기반의 XML 스키마 관리를 위한 그래픽 유저 인터페이스

강미란[○], 정갑주, 임상범, 김동광
건국대학교 신기술융합학과
{meela, jeongk, sblim, walhalla}@konkuk.ac.kr

SM-GUI: A Grid-based XML Schema Management GUI

Meilan Jiang[○], Karpjoo Jeong, Sang Boem Lim, Dongkwang Kim
Department of Advanced Technology Fusion, Konkuk University, Seoul, Korea

요 약

과학연구 분야에서는 부단히 새로운 연구 과제를 제기되고 있으며 이에 대한 연구조건 및 연구방법을 다각도로 변화시키면서 보다 효과적인 연구결과를 얻기 위한 노력을 거듭하고 있다. 이와 같은 연구활동이 활발히 진행됨에 따라 급증하는 데이터를 효율적으로 관리할 필요성이 증가하였으며 다양한 데이터 저장소에 저장되어 있는 다량의 데이터를 통합관리하기 위한 연구가 계속되어 왔다. 이런 연구를 진행함에 있어서 데이터를 효율적으로 관리하고 데이터 구조, 형태의 변화에 유연하게 대처하기 위해서는 데이터 모델 관리가 필요하다. 본 논문에서는 사용자 하여금 다양한 저장소의 데이터 모델을 관리하고 이로 부터 동적으로 데이터를 생성하여 통합 관리할 수 있는 XML Schema 관리 툴을 제공한다.

1. 서 론

다양한 응용연구 분야에서는 기존 연구에서 얻은 데이터를 분석 처리하고 더 좋은 결과를 얻기 위하여 연구 방법, 과정을 변화시켜 새로운 연구를 진행하고 있으며 끊임없이 진행되는 연구활동으로 인해 급증하는 과학기술 데이터에 대한 효과적인 관리방법이 매우 필요하다.

또한 기존에 생성된 데이터는 다양한 형태로 File System, RDB, XML DB 등 storage에 존재하는데 연구에 관련된 과학기술 데이터를 편집 관리하기 위해서는 여러 데이터 저장소에 대한 유연한 접근방법을 필요로 한다.

본 논문에서는 이런 문제점을 보완하고 해결하기 위하여 연구에서 정의된 데이터 모델을 XML Schema로 표현하고 GUI로 시각화하여 사용자에게 보다 편리한 관리 방법을 제시한다. XML Schema는 XML Data의 시멘틱과 구조와 내용을 정의하는 방법을 제공하고 있다. 우리는 이런 XML Schema의 수정이 용이한 장점을 이용하여 연구 데이터 모델의 변화에 따라 XML Schema를 변경, 추가하며 정의한 데이터 모델 기반으로 연구 데이터 입력 폼을 동적으로 생성, 관리한다.

본 논문은 다음과 같이 구성되어 있다. Section 2에서는 이런 문제들의 요구사항과 해결 방안을 다룰 것이다. Section 3에서는 그리드 기반의 스키마 관리 GUI를 설명하고, Section 4에서는 시스템 구조와 구현된 상황을 설명할 것이다. 마지막으로 Section 5는

관련연구, Section 6는 결론으로 구성되었다.

2. 관련연구

2.1 JAXFront

JAXFront[5]는 XML Schema에 기초하여 그래픽 유저 인터페이스를 생성하기 위한 기술이다. 동적으로 생성된 GUI는 사용자 하여금 XML 기술에 대한 지식이 없이 XML 데이터를 편집할 수 있게 한다.

이는 스마트 렌더링 메커니즘과 모델링, 구현에서의 XML 표준을 통하여 XML 데이터 모델을 그래픽 유저 인터페이스와 동기화하는 문제점을 해결하였다. GUI는 아무런 코드가 필요 없이 생성되며 XML 인스턴스 편집, 추가하는 기능을 제공한다. 또한, JAXFront UI에서 이루어진 모든 수정사항은 XML Schema로 유효성을 우선 체크하며 유효한 수정사항만이 XML 문서에 기록된다.

2.2 Oxygen XML Editor

Oxygen[6]은 자바 기반의 XML 편집기이며 XSD, DTD를 지원하는 XSLT 디버거이다. Oxygen XML 편집기는 스키마 파일에 대한 개발과 이해를 간단히 하게끔 W3C XML Schema와 Relax NG Schema에 대한 시각적인 XML Schema 편집기능을 제공한다.

Oxygen XML 편집기는 XML DB 서버에 대한 연결을 통하여 XML DB에 XQuery/XPath 쿼리를 실행할 수 있으며 데이터베이스 탐색창에 대한 전용 컬렉션은 데이터베이스 perspective 레이아웃에 그룹 지어져

있다. 데이터베이스는 범용적인 서버를 포함한 XML 저장소: Tamino, XHive, MarkLogic, TigerLogic, eXist, Berkeley or mixed 와 관계형 DB: DB2, SQLServer, Oracle 도 지원한다.

3. 구조적 디자인 이슈

SM-GUI 시스템의 디자인에 대해 서술하기 앞서 우선 X-SIGMA 시스템에 대해 간단하게 소개하고자 한다. SM-GUI의 기반 시스템인 X-SIGMA[4]는 사용자에게 편리한 데이터 관리기능을 제공하고 있다. SM-GUI 시스템은 보다 다양한 데이터베이스를 지원하면서 범용적인 Data management 시스템에서의 편리한 스키마 데이터 관리를 제공한다.

3.1. X-SIGMA

X-SIGMA는 과학 실험 데이터들을 관리하고 통합하고 접근할 수 있는 그리드 기반의 통합 시스템이다.

X-SIGMA 시스템은 글로벌 X-SIGMA라고 명명된 글로벌 관리 계층과 로컬 X-SIGMA라고 명명된 로컬 데이터 관리 시스템으로 구성되어 있다. 각 시스템은 다음과 같이 3개의 주요 컴포넌트들로 구성된다: (1)컨텍스트 데이터를 위한 스키마 관리 시스템, (2)검색 질의어 처리 시스템, (3)실험 결과 데이터의 접근 관리 시스템.

글로벌 스키마 관리 컴포넌트는 데이터 통합을 위해서 스키마 통합에 중점을 두고 있다. 검색 질의어 처리 컴포넌트는 XML Schema와 XML Database로 이루어져 있으며, 주어진 질의어에 맞는 데이터들을 찾을 수 있는 기능을 제공한다. 데이터 관리 컴포넌트는 컨텍스트 데이터의 위치 정보를 이용해서 실험 결과 데이터를 접근할 수 있는 기능을 제공한다.

3.2. 디자인 이슈

X-SIGMA는 OGSA-DAI 데이터베이스 관리 시스템을 사용하여 그리드 환경의 데이터 관리에 대처하고 있다. 이는 그리드 환경에서 진행되는 실험연구를 목표로 한 것으로서 모든 애플리케이션에 적용되게끔 디자인 되었다. 그러나 특정된 지역시스템(Local)에서 이루어지는 실험의 경우에는 X-SIGMA에 제한이 존재함을 발견하였다. 이런 경우 X-SIGMA는 불필요하게 무겁기 때문에 OGSA-DAI 외에 기타 단일 데이터베이스를 제공하여 범용적인 애플리케이션을 다룰 수 있게끔 하는 것이 바람직하다.

이 문제를 해결하기 위하여 우리는 SM-GUI 데이터 관리 시스템을 설계하였다. 이 시스템에서는 데이터베이스 시스템, 데이터 모델 기술과 유연성 있는 시스템 디자인, 세 부분에 초점을 맞추고 있다.

시스템에 적합한 데이터베이스를 제공하는 것이 매우

중요한데 데이터베이스의 대중성과 중요성이 충분히 고려되어야 한다. 너무 많은 데이터베이스를 제공하면 자원을 낭비하게 되고 너무 적은 데이터베이스를 제공하면 시스템이 취약해지는 부정적인 영향을 일으킬 수 있다. 관련되는 데이터의 사이즈가 크고 구조가 동적이기에 우리는 eXist, Berkeley DB, OGSA-DAI[14]와 같은 XML 데이터베이스를 선택하였다. 데이터베이스 선택 이외에 사용자와 각 사용자의 유효한 데이터베이스 연결을 관리하기 위해 사용자 정보(사용자 ID, 이름, 패스워드)와 DB 정보(DB URI, 사용자 ID, 파일 시스템의 지정된 위치에 저장된 jar 파일)에 대한 관리도 필요하다.

또한 실험연구 정보의 데이터타입을 표현하는 데이터 모델을 정의하여야 한다. 이 데이터 모델을 사용하여 보다 효율적으로 데이터를 검색할 수 있다. XML 스키마는 직관적이며 계층적인 구조를 허락하기에 우리는 XML 스키마를 사용하여 데이터 모델을 기술한다. 그리고 애플리케이션마다 복잡한 구조의 다양한 데이터 타입을 가질 수 있기에 XML과 XML 데이터베이스를 사용하여 스키마와 실험 데이터를 관리하도록 한다.

우리의 시스템이 가능한 유연하게 하기 위하여 SM-GUI 시스템을 두 개의 레이어아웃: 스키마 관리와 인스턴스 데이터 관리로 나뉜다. 스키마 관리는 데이터 모델의 변화와 확장에 유연하게 대처할 수 있게끔 하며 인스턴스 데이터를 추가하기 위한 간단하고 자동적인 인터페이스도 제공한다.

4. 시스템 아키텍처

SM-GUI 시스템(Figure 1)은 프레젠테이션 계층, 데이터 관리 계층, iXDBC(integrated XML Database Connectivity) 계층 와 물리적 계층, 4개 레이어를 포함한다.

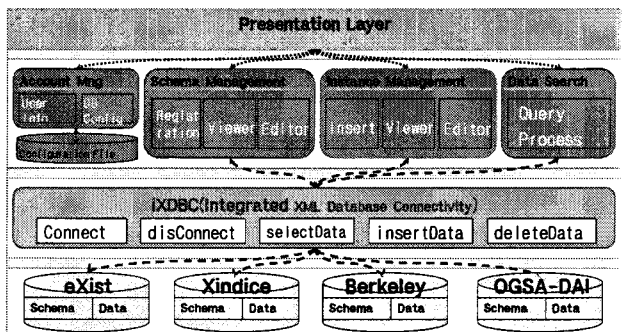


Figure 1. 시스템 아키텍처
물리적 계층은 시스템에서 지원하는 실제

데이터베이스를 포함한다. 시스템은 현재 eXist, Xindice[6], Berkeley[7]와 분산된 그리드 기반의 데이터베이스를 위한 OGSA-DAI를 제공한다. 앞으로 필요하다면 더 많은 데이터베이스를 추가할 것이다.

iXDBC 계층은 물리적 계층에서 데이터 관리 계층에 표준 인터페이스를 제공한다. 이 계층에서는 각 물리적 계층을 위한 어댑터를 구현하고 있는데 이런 어댑터는 특정 데이터베이스에 접근하는데 사용될 것이다. 이 계층의 흐름을 놓고 보면 iXDBC가 표준 데이터베이스 인터페이스를 통해 데이터 관리 계층으로부터 요청을 받아들인 후 대상 데이터베이스에 맞게 내부적으로 적합한 데이터 포맷으로 전환하여 요청사항을 대상 데이터베이스 시스템에 전송하여 준다.

프레젠테이션 계층은 사용자에게 데이터 관리를 위한 간단하고 편리한 인터페이스를 제공한다.

데이터 관리 계층은 계정관리, 스키마관리, 인스턴스 관리, 데이터 검색 4개 컴포넌트들로 구성되어 있다.

- *Account Management*: SM-GUI 를 필요한 XML DB 에 연결시키는 작용을 한다. 이는 사용자 정보와 데이터베이스 정보를 관리하는데 사용자들은 SM-GUI 시스템에 등록해야 하며 처음으로 데이터베이스에 접근할 경우 사용자가 시스템에서 필요로 하는 데이터베이스 정보를 등록하여야 한다. 사용자, 데이터베이스 정보는 파일시스템에 저장되며 등록된 사용자의 접근을 허락하고 자동으로 데이터베이스에 연결해준다.

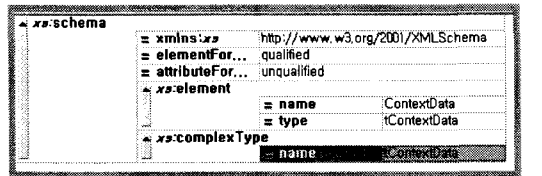
- *Schema Management*: 스키마 등록, 뷰, 편집 기능을 제공한다. 데이터 모델의 변화를 정확하게 파악하여 효율적인 데이터 검색을 제공하기 위하여 XML 스키마(Figure 2-a)로 데이터 모델을 하며 이를 변화시키고 확장하여 새로운 데이터 모델을 구성한다. 부가 메타데이터(스키마 이름, id, site 이름, 수정된 날짜)와 함께 XML 문서에 나열된 XML 스키마를 관리하여야 한다. Figure 2-b 는 XML 스키마 예제를 보여준다.

현재, 스키마 뷰어는 스키마 리스트 뷰어와 트리 뷰어를 포함하고 있다. 스키마 리스트 뷰어는 모든 스키마를 리스트 형태로 보여준다. 그 중 하나의 스키마를 선택하게 되면 컨텍스트 데이터와 데이터 타입에 대해 트리 뷰어로 보여준다.

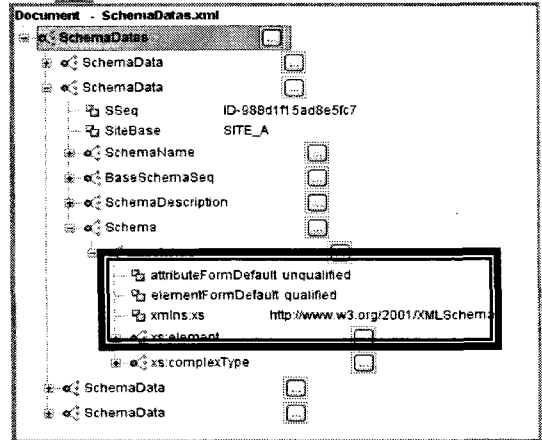
또한 XML 스키마 기반의 인스턴스 입력 폼을 자동적으로 정확하게 생성하는 기능도 갖고 있다.

- *Data Management*: 인스턴스 추가, 뷰, 편집 기능을 제공한다. 인스턴스 추가는 자동으로 생성된 입력 폼을 작성하여 완성할 수 있다. 인스턴스 데이터는 기타 메타데이터(인스턴스 이름, id, 스키마 id, site 이름, 입력날짜)와 함께 XML 문서에 저장된다. Figure 3 은 인스턴스 데이터 예제를 보여주고 있다.

인스턴스 리스트 뷰어와 트리 뷰어가 있는데 인스턴스 리스트 뷰어는 같은 XML 스키마 기반의 인스턴스 데이터들을 보여주고 트리 뷰어는 그 중 선택된 인스턴스 엘리먼트들의 이름과 값을 사용자에게 보여준다. 또한 인스턴스 편집기로 인스턴스 데이터를 등록, 삭제, 수정할 수 있다.



a) Standard XML Schema Format



b) An Example of XML Schema

Figure 2. XML 스키마

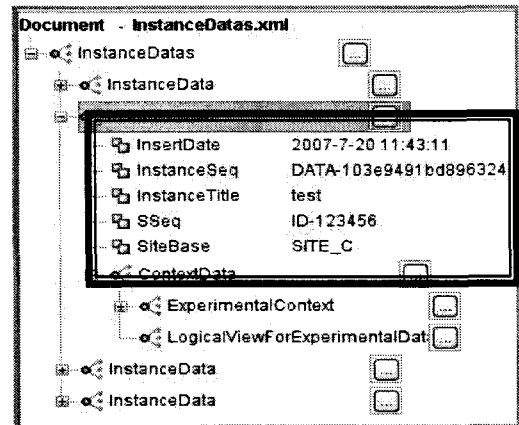


Figure 3. 인스턴트 데이터

5. Implementation

SM-GUI의 기본 시나리오는 Figure 4에서 설명하고 있다. 우선 사용자는 적합한 XML 데이터베이스에

연결되어야 한다. 사용자가 처음 XML 데이터베이스에 접근할 경우 사용자는 데이터베이스에 대한 위치, 타입 등 정보를 등록하여야 하는데 이런 정보는 사용자 정보와 함께 시스템에 의해 파일 시스템에 저장되어 해당 사용자에게는 자동으로 데이터베이스를 연결시켜 준다.

XML 데이터베이스에 연결한 다음, 사용자는 XML 스키마를 등록하거나 다른 XML 스키마를 기반으로 하는 새로운 스키마를 등록한다. XML 스키마 데이터는 기타 메타데이터(스키마 id, 이름, 시스템 이름과 수정날짜)와 함께 XML 데이터베이스에 XML 문서에 삽입되어 저장된다.

선택된 XML 스키마 데이터는 사용자를 위해 자동으로 인스턴스 데이터 입력 폼을 생성할 수 있다. 이 폼을 작성하여 제출하면 내부적으로 문서를 파싱하여 연결된 XML 데이터베이스에 저장하게 된다.

등록한 XML 스키마와 인스턴스 데이터를 관리하기 위하여 사용자는 삽입, 선택, 삭제 등 기능을 지원하는 인터페이스를 통해 XML DB에 쿼리를 실행시킬 수 있다.

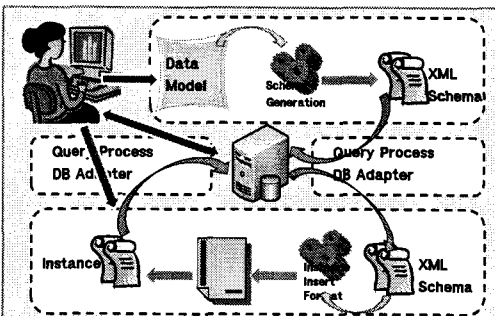


Figure 4. SM-GUI User Scenario

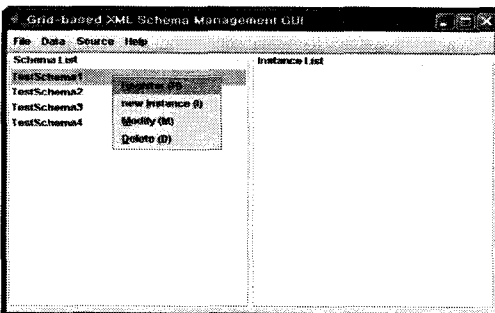


Figure 5. Main User Interface

5.1 현재 구현

상술한 시나리오와 아키텍처에 근거하여 우리는 SM-GUI의 프로토타입을 구현하였다. 앞으로 현재의 구조적 디자인을 발전시켜 새롭게 구현할 것이다.

현재 구현 사항은 다음의 3개의 중요한 메뉴로

이루어져 있다:

- Main UI(Figure 5): 스키마 뷰어와 Instance 뷰어로 구성된다. 스키마 뷰어는 스키마 리스트를 보여주고 있으며 Instance 뷰어는 스키마 뷰어에서 선택된 스키마를 기반으로 생성된 인스턴스들의 리스트를 보여주고 있다.
- Schema Management(Figure 6): 이는 스키마의 트리 구조 뷰어와 에디터를 구성된다. 뷰어에서는 에디터를 통해 선택한 트리 노드 하위에 데이터를 추가하거나 선택한 데이터를 삭제하는 기능을 제공한다.
- Data Management(Figure 7): 이 메뉴는 스키마 기반으로 XML 데이터를 입력하고 관리하는 기능을 제공한다. XML 데이터(Instance)의 입력 폼은 정의된 XML Schema 에 따라서 자동으로 정의되어 사용자가 입력할 수 있게 되어있다.

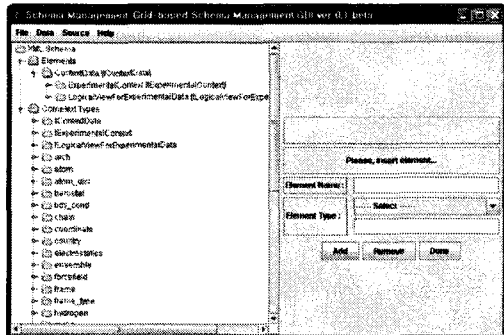


Figure 6. Schema Management

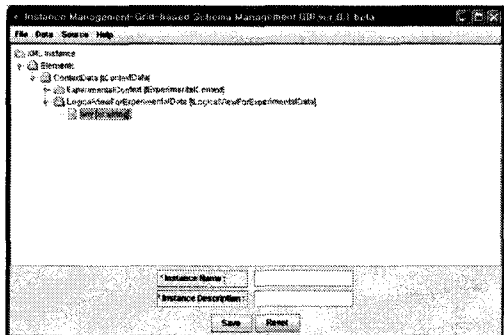


Figure 7. Instance Management

6. 결론과 향후 연구방향

과학연구 분야에서 보다 효율적인 검색을 위하여 다량의 급증하는 데이터에 대한 효과적인 관리가 필수이다. 우리는 이런 데이터를 데이터 모델에 따라 구분하여 데이터 공유, 검색 및 기타 용도를 위해 데이터를 효과적으로 관리한다.

전체 내용을 개괄해 보면 다양한 데이터 저장소를 지원하는 간단하고 편리한 스키마 관리 메커니즘을

보여주고 있다. 본 논문에서 우리는 SM-GUI라고 불리는 스키마 관리 시스템에 대해 설명하고 있는데 1) 확장성 - 데이터 모델 관리 2) 유연성 - 자동적으로 인스턴스 데이터 입력 폼 생성 3) 호환성 - 통합 XML DB Connectivity-이 강조되고 있다.

이 시스템은 아직 데이터 검색에 대한 구현이 완성되지 않은 상태이다. 가까운 장래에 현재의 SM-GUI 시스템을 로컬, 글로벌 환경에서 데이터 검색을 효율적으로 할 수 있게끔 확장할 것이다. 지금까지 iXDBC는 단지 eXist(로컬 XML 데이터베이스)와 OGSA-DAI를 지원하지만 앞으로 더 많은 저장소를 지원하도록 개선시킬 필요성이 존재한다.

7. 참조

- [1] Tim Bray and C.M. Sperberg-McQueen, "Extensible Markup Language (XML): Part I. Syntax", World Wide Web Consortium Recommendations, February 1998, Available at <http://www.w3.org/TR/REC-xml>
- [2] David C. Fallside, "XML Schema Part 0: Primer", World Wide Web Consortium Candidate Recommendation, October 2000, Available at <http://www.w3.org/TR/xmlschema-0/>.
- [3] Mario Antonioletti, Malcolm Atkinson, Rob Baxter, Andrew Borley: The design and implementation of Grid database services in OGSA-DAI(Database Access and Integration Services) . Concurrency and Computation: Practice & Experience archive Volume 17 , Issue 2-4 , Pages: 357 - 376 (2005)
- [4] Dongkwang Kim, X-SIGMA: XML based Simple data Integration system for Gathering, Managing, and Accessing Scientific Experimental Data in Grid Environments
- [5] mike_leber, JAXFront - an XML Rendering Technology
- [6] Oxygen, <http://www.oxygenxml.com/>
- [7] Oracle Berkeley DB XML Getting Started with Berkeley DB XML for Java. Available at <http://www.oracle.com/technology/documentation/berkeley-db/xml/>
- [8] Xindice 1.1 Developer Guide, Available at <http://xml.apache.org/xindice/guide-developer.html>
- [9] Avi Silberschatz, Henry F. Korth, S. Sudarshan: Database System Concepts Fifth Edition , ISBN 0-07-295886-3
- [10] AnHai Doan, Alon Y.Halevy, Semantic Integration Research in the Database Community: A Brief Survey.
- [11] S.Boag, D.Chamberlin, M.F.Fernandez : XQuery 1.0: An XML query Language, 30 April 2002, <http://www.w3.org/TR/xquery>