

XTrans: 변환 적합성을 고려하는 XML 스키마 매퍼

신철민⁰ 이경하 이규철
 충남대학교 컴퓨터공학과
 {feminie⁰, bart, kclee}@cnu.ac.kr

XTrans: The XML Schema Mapper considering transformation feasibility

Cheol-Min Shin⁰ Kyong-Ha Lee Kyu-Chul Lee
 Dep't. of Computer Engineering, Chungnam National University

요 약

웹 서비스(Web Services)는 WSDL로 정의되는 인터페이스와 XML 스키마로 정의되는 메시지를 이용하여 시스템 통합 및 분산 컴퓨팅이 가능하도록 하는 기술이다. 서비스 간 조합을 수행하기 위해서는 각기 다른 입출력을 가지는 웹 서비스 간에 메시지 변환을 위한 과정이 절대적으로 요구된다. 본 논문에서는 이러한 다대다 스키마 문서 간 매핑, 다대다 노드 간 매핑을 지원하고, 노드 간 타입 이질성을 해결할 수 있는 함수를 제공하며, 노드정보를 이용해 매핑 적합성 및 빈도의 적합성을 검사하여 XSLT를 스타일 시트를 생성하는 XML 스키마 매퍼, XTrans를 개발하였다.

1. 서론

웹 서비스는 XML, SOAP, WSDL 등 XML 표준 기술들을 이용하여 인터넷 상에 분산되어 있는 서비스 컴포넌트들을 통합하고 이용할 수 있도록 업계 주도도 고안된 표준 프레임워크이다[1]. 서비스 이용자는 서비스 제공자에 의해 제공되는 서비스 컴포넌트에 접근하기 위해서 해당 서비스 인터페이스가 기술된 WSDL(Web Service Description Language)[2] 문서를 참조함으로써 서비스의 구현 내용과는 상관없이 표준화된 XML 메시지를 통한 서비스의 제공이 가능하게 된다. 웹 서비스 프레임워크 상의 서비스 제공자와 이용자 사이의 직접적인 메시지 교환은 서비스 제공자가 작성한 WSDL 문서에 정의한 메시지 형식만을 이용하므로 문제가 없다. 하지만, 서비스 통합의 관점에서는 특정 서비스로부터 받은 출력 메시지를 다른 서비스의 입력으로 보내야 하는데, 이때 각각의 서비스는 서로 다른 메시지 구조를 가지는 경우가 대부분이다. 이로 인해 서비스 조합 시에는 각 서비스의 입출력 메시지의 구조를 고려하여 적절한 메시지 변환을 해주어야만 서비스를 통합하여 사용하는 것이 가능하게 된다.

본 연구에서는 XML 메시지의 각 요소들을 서로 대응시켜 변환 규칙을 정의하고 적절한 XSLT[3] 문서를 만들어 내기 위한 매퍼(이하 XTrans라 칭함)의 개발을 목표로 하였다. 기존의 매퍼 들은 노드 간 1:1 매핑만을 지원하여 다대다 노드 간의 매핑은 불가능하고, 매핑 시 각 노드 간 타입 호환성 및 적합성을 검사하지 않는다. 또한 일부 매퍼 들은 다대다 스키마[4] 문서간의 매핑을 지원하지 못한다. XTrans는 기존의 매퍼 들이 갖는 문제점을 해결하여, 다대다 노드 간 매핑, 다대정보인 데이터타입과 발생빈도를 이용한 변환적합성 검사, 데이터타입 변환을 지원하는 함수와 사용자정의 함수 사용가능, 다대다 스키마 문서 간 매핑을 지원하는 매퍼이다.

2. 관련연구

현재의 웹 서비스 분야를 보면 단일 웹 서비스들이 많이 만들어지고 사용되고 있는 상태이다. 이렇게 만들어진 개개의 웹 서비스들을 조합하여 더 큰 범위의 기능 내지는 목적으로 하는 결과를 얻으려 하는 연구들이 폭

넓게 진행되고 있다. 현재 개발되어 있는 매퍼로는 MS사의 BizTalk Mapper[5]와 Sonic software의 Stylus Studio[6], Altova의 Mapforce[7] 등이 있다. 이상의 기존 매퍼 들은 단말 노드와 단말 노드 사이의 매핑 시 추가적인 연산이 필요한 경우 특정 함수를 적용할 수 있으나 아직 지원 범위가 미흡하고, 사용자 정의 함수에 대한 지원이 없거나 특정 언어 및 플랫폼에 종속적이다.

또한 각 스키마 문서에서 정의하고 있는 각 요소들에 대한 타입의 호환성 및 적합성에 대한 검사를 하지 않거나, 정확하게 일치 되지 않는 타입에 대해서는 매핑이 불가능하도록 하고 있어 데이터의 손실 없이 타입 캐스팅할 수 있는 경우에도 애러 처리를 하고 있다. 본 연구에서는 스키마 기반 매핑규칙 정의, 다대다 매핑을 위한 구조를 제시하고, XSLT변환 시 노드 간 데이터 타입의 적합성 검사 방법을 제시하여 XML 스키마 기반 시스템의 통합 및 상호 운용을 가능하도록 하는 것을 목적으로 한다.

3. XML 스키마 매핑

XML 스키마는 최종 XML 인스턴스 문서를 만들기 위해 필요한 문서의 구조뿐만 아니라, 각 엘리먼트의 이름, 설명, 타입, 제약사항까지 포함하고 있다[8]. 따라서 스키마 문서는 XML 인스턴스 문서에서는 얻을 수 없는 부가적인 정보를 제공함으로써 매핑 단계에서 사용자가 각 엘리먼트나 애트리뷰트에 대한 더욱 정확하고 적절한 처리를 할 수 있는 정보를 제공 받을 수 있게 한다.

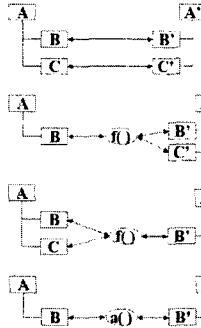
3.1 매핑의 유형

매핑은 다른 스키마 구조로부터 같은 의미를 지니는 요소 간에 연관 관계를 맺어 주는 것으로, 단순한 요소간의 관계를 나타낼 수도 있으나 연관 관계뿐만 아니라 부가적인 처리를 수반하는 연관 관계를 나타낼 수도 있다.

대응의 유형은 크게 문서간의 대응과 문서 내의 노드들 간의 대응으로 나뉜다. 문서간의 대응은 매핑에 참여하는 원본 측과 목적 측의 문서의 숫자에 의해 매핑의 유형은 1:1, 1:N, M:1, M:N의 형태를 가진다. 그 중에 일반적으로 나타나는 문서 간 연관형태인 M:N 관계는 1:N과 M:1이 섞여 있는 구조이다. 1:N의 형태는 원본 측 문서 한 개가 목적 측의 여러 개의 문서에 대응되는 형

태이며 한 개의 문서가 나뉜다는 뜻에서 분리(split)라고 할 수 있다. N:1의 형태는 원본 측 문서 여러 개가 목적 측 한 개의 문서에 대응되는 형태이며 이는 여러 문서가 합쳐진다는 뜻에서 병합(merge) 또는 조인(Join)이라고 할 수 있다.

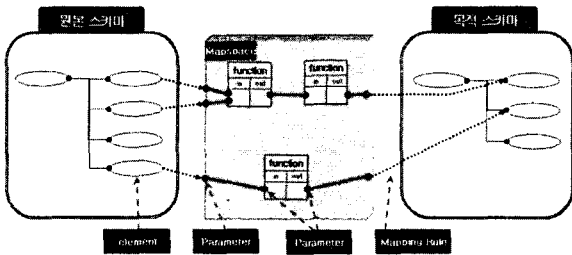
- 1:1 노드 간
- 1:N 노드 간
 - > generic function
- M:1 노드 간
 - > generic function
 - > aggregate function



<그림 1> 노드 간 대응의 종류

또 다른 대응의 유형으로는 <그림 1>에서 보여주는 문서내의 노드들 간의 대응 유형이 있다. 문서내의 노드는 스키마의 엘리먼트와 애트리뷰트가 되며 이들 사이에도 문서 간 대응 유형과 같이 1:1, 1:N, N:1, M:N의 네 가지의 대응 유형이 생긴다. 하지만, 노드 간 대응이 문서 간 대응과 다른 점은 함수가 사용되어야 한다는 것이다. 1:1 노드간의 대응에서는 함수가 사용될 수도 있고 사용되지 않을 수도 있다. 이에 반해 1:N이나 N:1노드 간 대응의 경우 노드의 내용을 합쳐주거나(문자열의 concatenation)이나 분리(문자열의 substring)의 기능을 하는 함수가 필요하다. 또한 M:1노드의 경우 스키마 트리 상에서는 하나의 노드로 표현되지만 실제의 XML인스턴스 문서상에서는 여러 개가 올 수 있는 발생빈도가 2이상인 경우에 이들을 묶어주는(aggregation)함수가 필요한 경우도 생긴다.

3.2 XML 스키마 단말 노드의 매핑

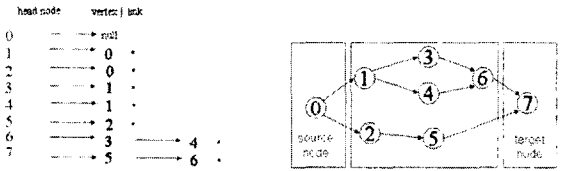


<그림 2> 1:1스키마간 매핑의 구조

<그림 2>는 기본적인 매핑 메커니즘을 위한 매핑 구조를 그린 것이다. <그림 2>에서 Mapspace는 목적 측 스키마와의 매핑 관계가 설정되면 만들어지는 가상의 컨테이너로 원본 측 스키마의 단말 노드를 재가공하고 타임 불일치를 해결하기 위한 함수들을 포함하고 있다. Map 안에서는 하나 이상의 함수가 토폴로지 형태로 함수 체인(chain)을 형성할 수 있다. Mapspace는 입력 측 인자(parameter)를 1개 이상 가지고, 출력도 하나 이상의 인자를 통해 내보내는 블랙박스 형태를 하고 있다. <그림 2>의 Parameter는 그런 Mapspace의 입출력 포트(port)이다.

그러나 이런 요소들만으로는 다대다 문서 또는 다대다 스키마 간 매핑을 처리할 수 없다. 출력 측 1개의 스키마 문서에 대해서 1개의 Mapspace가 정의되고 이런 Mapspace 들을 포함하는 전체의 매핑을 가지는 Workspace가 존재한다. Workspace는 매핑 작업을 위한 가상의 통합 작업 공간으로 사용되며 전체 매핑 작업 필요한 부가적인 정보들을 담고 있다.

3.2 매핑정보의 표현



<그림 3> 매핑정보의 역 인접리스트 표현

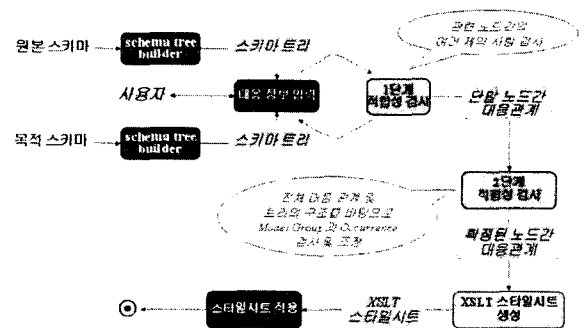
사용자에 의해 이루어지는 매핑은 역 인접리스트를 이용하여 관리된다. 역 인접리스트[9]는 방향그래프에서 각 정점에 대한 그 정점으로 돌아오는 연결선과 인접한 정점들로 구성된 연결리스트 구조이다. 매핑에 참여하는 스키마의 노드와 함수들을 정점으로 보고 매핑을 방향성을 가진 에지로 간주하여 매핑 된 상태를 처리하며 <그림 3>은 그 구조를 보여준다. 목적 스키마 측의 노드 7을 보면 노드5와 노드6에서 에지가 오는 것을 리스트를 따라가면 알 수 있고 이렇게 검색된 노드5는 노드2에서 오는 것을 알아 낼 수 있다. 이런 방식을 통해 매핑을 표현한다.

3.3 매핑 시 사용되는 함수

Function은 스키마의 노드를 변환하기 위한 실제 로직(logic)을 담고 있는 것으로 일반적인 프로그래밍 언어에서 사용되는 한 개 혹은 그 이상의 함수 전달 인자를 받아 처리 후, 출력으로 내보낸다. 출력 결과는 다른 함수의 입력 인자로 사용되거나 Mapspace의 출력 측 포트와 연결된다. Function에서는 수학 관련, 문자열관련, 데이터타입관련, 사용자정의관련 함수들로 크게 분류된다.

4. XTrans의 개발

4.1 처리 흐름



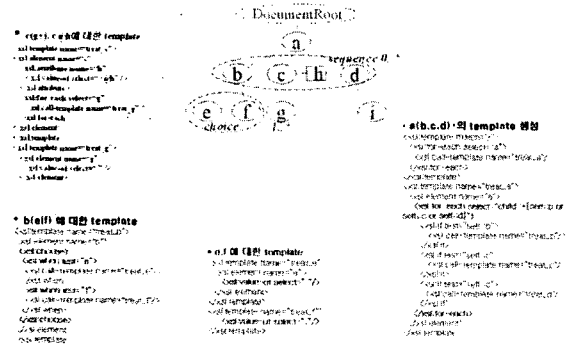
<그림 4> 전체 처리 흐름

<그림 4>과 같이 XTrans는 최종적으로 스키마 정보를 이용하여 문서 변환을 위한 XSLT문서를 만들기 위해 대응 정보 입력, 변환 적합성 검사, XSLT스타일시트 생성의 3단계를 수행하게 된다. 대응 정보 입력 단계는 사용자에게 의해

스키마의 각 요소 간의 관계를 맺어 주는 단계로 특별한 사용자 인터페이스가 제공된다. 변환 적합성 검사는 다시 1,2 단계로 나누어 순차적으로 진행된다. 1단계 적합성 검사에서는 사용자에게 의해 원본 축과 목적 축의 스키마상의 데이터 타입과 발생빈도의 적합성을 결정하도록 유도한다. 원본 축 스키마의 데이터 타입과 발생빈도가 목적 축의 데이터 타입과 발생빈도가 일치하거나 목적 축 데이터 타입과 발생빈도가 원본 축 데이터 타입과 발생빈도수를 포함할 수 있는 경우에는 적합한 경우이고, 이 밖의 경우는 사용자의 진행여부의 의사를 확인하고 그에 따라 진행 여부를 결정한다. 2단계 적합성 검사에서는 문서의 레벨과 순서정보 그리고 매핑에 관여하지 않은 여분의 노드들의 적합성 검사를 수행하게 된다. XSLT스타일시트 생성단계에서는 적합한 데이터 타입을 확인하고 매핑정보를 출력하기 위한 XSLT문서를 생성하게 된다. 이때 생성된 XSLT는 트리를 기반으로 하여 최상위 스키마 노드부터 부모노드에 포함된 자식 스키마 노드 순으로 템플릿 기반의 XSLT문서를 생성하게 된다. 출력단계에서 생성된 XSLT문서는 2단계 적합성 검사 단계를 거치게 되고 적합성 검사를 끝낸 XSLT문서는 정제된 권고 XSLT문서로서의 역할을 하게 된다. 이렇게 생성된 XSLT문서는 원본 축의 XML인스턴스 문서와 함께 XSLT프로세서를 통해 목적 축의 입력 XML인스턴스문서로 변환된다.

4.2 XSLT스타일시트의 생성

트리의 위치정보를 가지고 사용자가 매핑을 하면 그 출력물로 XSLT문서를 생성한다. XSLT를 만드는 유형에는 하나의 템플릿이 모든 변환규칙을 가지는 one template 방식, 템플릿 선언 시 mode를 지정하여 apply-template호출을 이용하는 applying by mode 방식, 템플릿 선언 시 name을 지정하고 call-template호출을 이용하는 call by name방식의 세 가지를 들 수 있다. 이 중에 매핑의 결과로서 생성되는 XSLT문서는 모듈화가 가능하고 목적스키마를 대상으로 한 변환순서 조정이 용이한 call by name방식을 사용한다. 그리고 템플릿 생성은 트리 구조에 기반 하여 부모노드의 템플릿에서 자식노드를 호출할 수 있게 하여 가독성을 향상시켰다. 또한 추가되거나 삭제된 노드가 있을 경우 그 노드를 호출한 부모노드의 템플릿을 수정함으로써 재사용성을 증대시켰다.

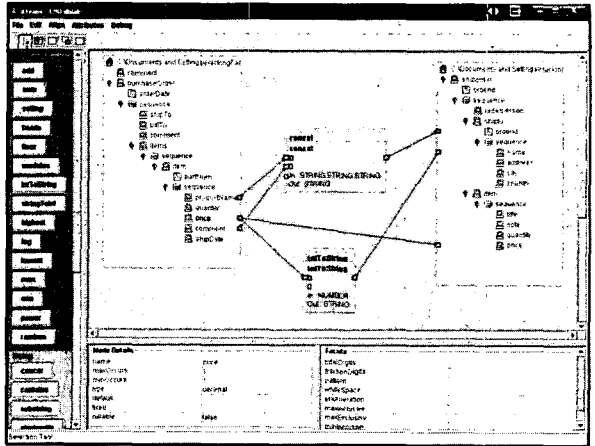


<그림 5> 생성되는 XSLT스타일시트

<그림 5>에서는 원본 축 스키마의 구조가 가운데 트리와 같은 경우에 매핑에 의한 XSLT결과를 만들어내는 템플릿들을 보여준다. 먼저 깊이 우선 탐색으로 노드 e,f를 방문하여 템플릿을 만들고 그 상위노드인 노드b에서는 모델그룹정보인 choice를 표현한 템플릿에 하위의 노드 e,f의 템플릿을 call-template하여 생성하고 최종적으로 노드 b,c,d를 가지는 노드a에 대한 템플릿을 완성하게

되면 전체 XSLT생성이 끝나게 된다.

4.3 XTrans 사용자 인터페이스



<그림 6> XTrans 사용자 인터페이스

<그림 6>은 개발된 XTrans의 실제 사용자 인터페이스이다. 왼편으로 매핑 시 사용되는 함수들의 목록이 있고 그 아래에 사용자가 함수를 추가할 수 있는 버튼이 있다. 오른쪽 위에는 실제 매핑을 할 수 있는 작업 영역이 있다. 함수를 사용할 경우에는 왼편의 함수목록에서 드래그 & 드롭 기능을 이용하여 Workspace에 끌어다 놓은 후, 스키마의 노드를 각각 연결하면 함수를 사용할 수 있다. 매핑 작업 영역 아래쪽에는 노드에 대한 정보와 패시(facets)정보를 보여주는 테이블이 있어 선택된 노드에 대한 정보를 한 눈에 확인할 수 있다.

5. 결론 및 향후 연구

지금까지의 매핑 도구들은 1:1 스키마문서 간 매핑만을 지원하거나, 노드 간의 1:1 매핑만을 지원했고, 매핑 시 노드의 타입과 속성에 따른 변환 적합성 검사를 실시하지 않아 데이터에 대한 파손 없이 변환할 수 있는 매핑 노드 간에도 매핑 할 수 없는 불편함이 있었으나, 본 연구에서는 다대다 스키마 및 노드 간 매핑이 가능하고 변환 적합성 검사 알고리즘을 통해 효율적인 매핑 작업이 가능하다. 또한 템플릿 기반의 XSLT문서 생성알고리즘을 이용하여 가독성과 템플릿의 재사용성이 뛰어난 XSLT스타일 시트를 생성할 수 있다. 향후연구로서 단일노드 간 매핑뿐 아니라 여러 단말 노드를 갖는 모델그룹단위의 매핑을 지원하고 원본 축 노드의 구조와 목적 축 노드의 정보를 이용한 복합타입 차원의 매핑 문제에 대한 연구가 필요하다.

참고문헌

[1] W3C, "Web Services", <http://www.w3.org/2002/ws>
 [2] W3C, "Web Services Description Language(WSDL) 1.1", <http://www.w3.org/TR/wsd/>
 [3] W3C, XSL Transformations (XSLT) Version 1.0, <http://www.w3.org/TR/xslt>, Nov. 16, 1999
 [4] W3C, XML Schema Part 0 : Primer, <http://www.w3.org/TR/xmlschema-0>, May, 2, 2001
 [5] MS Biztalk Server, <http://msdn.microsoft.com/biztalk/>
 [6] Stylus Stylusstudio, <http://www.stylusstudio.com>
 [7] Altova, MapForce, <http://www.altova.com>
 [8] N.Santoro and R.Khatib, "Labeling and implicit routing in networks", In The Computer J., 28:5-8, 1985
 [9] Horowitz, Sahni, Anderson-Freed, 이석호 역, "C로 쓴 자료구조론", p307-322, 사이텍미디어, 1993