Visitor 디자인 패턴을 이용한 XML Pull Parser Generator의 설계

고방원° 곽동규 유재우
숭실대학교 컴퓨터학과
{bwko°, coolman}@ss.ssu.ac.kr, cwyoo@computing.ssu.ac.kr

A Design of XML Pull Parser Generator Using Visitor Design Pattern

BangWon Ko°, DongKyu Kwak, ChaeWoo Yoo
Dept. Computer, Soongsil Univ.

요 약

기존에 사용되고 있는 XML 파싱 모델에는 Object 모델과 Push 모델이 있다. Object 모델은 문서 내의 컨텐츠를 트리 구조로 변화하기 때문에 메모리의 사용이 많아지고 Push 모델의 경우 문서 내의 컨텐츠를 이벤트 처리하는 각각의 메서드를 이용함으로써 이벤트 처리의 지연 및 지연을 위한 메모리를 추가로 사용하는 등의 단점이 있다. 이를 해결하기 위해 Pull 모델이 등장했으나 DTD를 지원하지 않음으로서 XML 문서의 유효성을 검증할 방법이 없으며 파서의 기능이 미약하다. 본 논문에서는 Visitor 패턴을 이용하여 기존의 Pull 파서들이 가지지 못하던 XML 문서의 유효성 검증 문제를 해결하고, 파서에 부가적인 기능을 하는 모듈을 추가하여 Interpreter 시스템으로 확장 가능한 파서를 생성해주는 XML Pull Parser Generator를 제안한다.

1. 서 론

현재 XML 파싱 모델은 여러 모델이 연구되고 있으며 현재 사용하는 모델에는 XML 문서의 정보를 트리 구조로 표현하는 Object 모델, 문서의 구성요소 하나하나를 Event로 처리하는 Push 모델이 있다[1]. 하지만 Object 모델인 DOM이나 Push 모델인 SAX는 실시간 데이터 처리뿐만 아니라 대용량의 XML 문서의 처리에 있어서 메모리의 비효율적인 사용 및 지연현상이라는 단점을 가지고 있다[1][2]. 이를 극복하기 위해 나온 모델이 바로 Pull 모델이다[3].

Pull 모델은 이벤트 기반 모델이라는 점에서 Push 모델과 비슷하지만 지연 시간의 감소와 메모리의 효율적인 사용이라는 점에서 뛰어나다는 특징을 가지고 있다. 그리고 파싱 속도에서도 다른 파싱 모델에 비해 상당히 빠른 장점을 가지고 있다[1].

본 논문에서는 Visitor 패턴을 이용한 XML Pull Parser를 생성하는 XML Pull Parser Generator를 제안한다. 제안된 XML Pull Parser Generator는 DTD를 입력받아 XML Pull Parser에서 사용되는 Visitor의 여러 오페러이션들을 생성하고 입력받은 DTD에 유효한 XML 문서만을 파싱하게 된다. 이는 기존의 XML Pull 파서들이 가지지 못했던 XML 문서의 유효성(Validation) 검증 문제를 XML 문서의 파싱 과정이 아닌 파서의 생성 과정에서 처리하였고 Visitor 디자인 패턴을 사용하여 생성된 파서의 구조 자체를 변경시키지 않고 새로운 기능을 쉽게 추가, 확장할 수 있는 장점들을 제공한다. 그리고 파서에 부가적인 기능을 하는 Action Module를 사용하여 Interpreter 시스템으로의 확장 역시 가능하다.

2. 관련연구

2.1 XML 문서의 파싱

일반적으로 XML 문서는 ELEMENT, ATTRIBUTE, ENTITY, DTD(Document Type Definition)의 4가지 주요 구성 요소가 있다. ELEMENT는 태그 데이터를 표현하는 것이고, ATTRIBUTE는 ELEMENT에 대한 정보를 더하기 위해 사용된다. ENTITY는 XML 문서에서 참조할 수 있는 데이터의 실질적인 값이다. 이중 DTD는 각각의 XML 문서의 구조를 정의하는 XML의 선택적인 부분이다.

XML 파싱은 문서로부터 태그의 의미와 구조에 기초하여 이러한 구성 요소들을 추출해내는 것을 의미한다.

2.2 XML Object 파싱 모델

XML 파싱 모델은 그 처리 방법에 따라 세가지의 모델이 사용되어 지고 있다. 첫번째 파싱 모델은 XML 문서의 요소를 트리화하여 처리하는 Object 파싱 모델, 두번째는

XML 문서의 컨텐츠들을 파서가 응용프로그램(Application)에 전달하여 이벤트로 처리하는 Push 파싱 모델, 마지막으로 응용프로그램이 파서에게 파싱을 요청하여 이를 기반으로 처리하는 Pull 파싱 모델이 있다[1].

Object 파싱 모델은 DOM(Document Object Model)이 있으며 XML 문서의 전체 구조를 메모리 내에서 트리화하여 구성함으로 효율성이 낮다는 단점이 있다[1].

2.2 XML Push 파싱 모델

Push 파싱 모델은 SAX(Simple API for XML)가 있으며 그림 1과 같이 Push Parser Interface를 통해서 문서의 요소들을 이벤트 처리한다[1]. 이 모델은 응용프로그램의 상황에 관계없이 정보를 전달함으로서 처리될 필요가 없는 정보들에 의해 지연현상이 발생할 수 있으며 또한 이벤트가 처리되지 않은 정보들을 저장해야 하므로 많은 메모리를 사용한다.

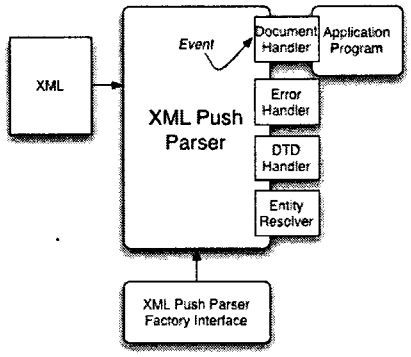

그림 1. Push 파서의 구조

2.3 XML Pull 파싱 모델

Pull 파싱 모델은 이벤트 기반이라는 점에서 Push 파싱 모델과 비슷하지만 그림 2에서 볼수 있듯이 XML 문서의 파싱을 응용 프로그램이 직접 파서에서 요청하므로 지연현상이 적고 지연시간을 처리하기 위한 추가적인 메모리를 사용하지 않으므로 메모리의 효율성도 뛰어나다.

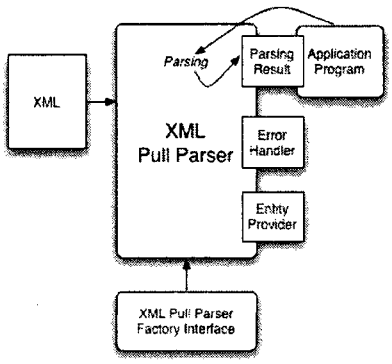

그림 2. Pull 파서의 구조

2.4 Visitor 디자인 패턴[4]

Visitor 디자인 패턴은 GoF(Gang of Four)의 디자인 패턴 중 행동에 관한 패턴이다. 이 패턴은 처리되어야 하는 요소에 대한 클래스를 변경하지 않고 새로운 오퍼레이션을 정의할 수 있게 한다. Visitor 객체에는 객체 구조에 속한 요소에 수행될 오퍼레이션을 정의할 수 있다.

Visitor 디자인 패턴은 첫째, 객체 구조가 다른 인터페이스를 가진 클래스들을 포함하고 있어서 구체적 클래스에 따라 이들 오퍼레이션을 수행하고자 할때, 둘째, 객체 구조를 정의한 클래스는 거의 변하지 않지만, 전체 구조에 걸쳐 새로운 오퍼레이션을 추가하고 싶을때, 셋째, 전체 구조에 걸쳐 새로운 오퍼레이션을 추가하고 싶을때 사용할 수 있다. 그리고 Visitor 디자인 패턴의 실제 사용에 있어서 여러개의 Visitor를 사용하여 하나의 구조에 있어 여러가지의 오퍼레이션을 구현할 수도 있다.

3. Interpreter System Overview

다음 그림 3은 본 논문에서 제안하는 XML Pull Parser Generator를 이용한 Interpreter 시스템의 구조도이다.

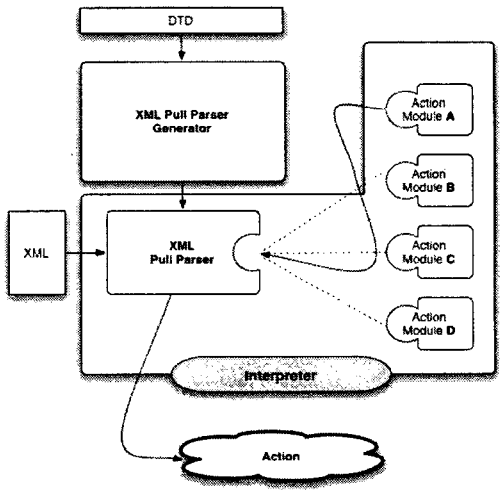

그림 3. Interpreter 시스템 구조도

Interpreter 시스템은 크게 두 부분으로 나뉜다. 첫번째는 DTD를 입력받아 XML Pull Parser가 생성되는 XML Pull Parser Generator 부분이고, 두번째는 Action Module 부분이다. 입력받은 DTD에 의해 XML Pull Parser에서 생성된 Pull 파서는 Interpreter 시스템에서 입력받은 XML 문서를 바탕으로 행해질수 있는 행동에 따라 적합한 각 Action Module을 탑재함으로써 Interpreter 시스템으로의 구성이 가능하게 되고 사용자는 XML 문서에 기술된 컨텐츠를 Action Module에서 기술된 구체적인 행동(Action)으로 사용할 수 있게 된다.

4. XML Pull Parser Generator

본 논문에서 제안하는 XML Pull Parser Generator의 구조는 그림 4와 같다.

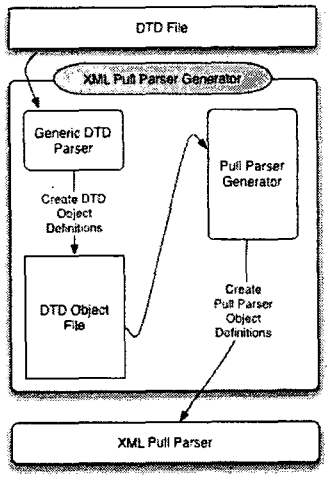

그림 4. XML Pull Parser Generator

XML Pull Parser Generator는 DTD 파일을 읽어들여 XML Pull Parser를 생성한다. 생성되는 Pull 파서는 Visitor 패턴의 구조를 가진다. DTD 파서는 DTD 파일을 읽어들여 분석한 후 DTD Object를 정의하고 XML Pull Parser Generator의 결과물인 Pull 파서에 사용될 Visitor 객체의 오퍼레이션으로 사용될 Object 파일을 생성한다. 이때 XML Pull Parser Generator에서 생성되는 Pull 파서는 미리 정의된 DTD를 기반으로 생성되므로 XML 문서의 유효성 검증을 파싱 단계가 아닌 파서 제작 단계에서 해결할 수 있다.

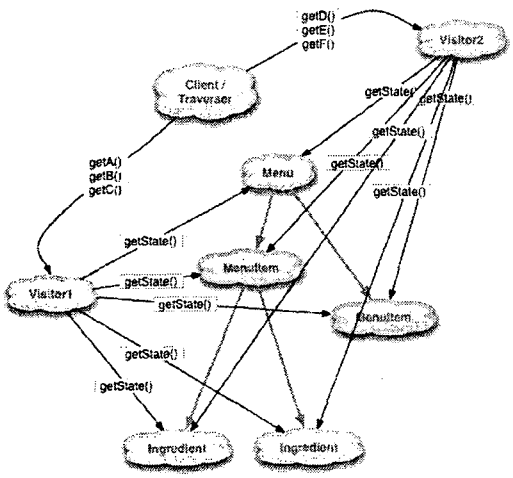

그림 5. Visitor 패턴을 사용한 객체 구조의 예

그리고 그림 5에서 볼수 있듯이 Visitor 디자인 패턴을 기반으로 생성된 Pull Parser는 파싱에 관한 부분과 Action Module간의 결합도가 낮다. 그러므로 각 XML 문서의 컨텐츠를 실행할 Action에 관한 코드를 기술한 Visitor만을 추가, 삭제할 수 있고, 이미 작성된 Action Module에서 기능을 추가, 삭제하고 싶을 때에는 Visitor 클래스에 단순히 메서드만을 추가, 삭제하여 손쉽게 Interpreter 시스템의 기능을 축소, 확장할 수 있어 XML 문서 및 Pull 파서의 활용범위를 넓힐 수 있다.

5. 결론 및 향후 연구과제

본 논문은 XML 문서의 효율적이고 빠른 파싱을 위한 파서를 생성하는 Parser Generator에 대해 제안하였다. 그리고 이를 위해 Pull 파싱 모델을 채택하였다. 본 논문에서 제안한 XML Pull Parsing Generator는 기존의 Pull 파서들이 가지지 못하는 XML 문서의 유효성 검사를 Pull 파서 생성 과정에서 주어진 DTD만을 파싱하는 Pull 파서를 생성함으로써 해결하였으며, Visitor 디자인 패턴을 사용하여 각 모듈들 간의 의존성을 떨어트려 파서 기능의 확장을 손쉽게 하였다.

그리고 XML Pull Parser Generator를 통해 생성된 Pull 파서에 Action Module을 탑재할 수 있게 구성하여 향후 Interpreter 시스템으로의 확장이 가능하다. 하지만 Interpreter 시스템에서 사용되는 Action Module의 경우 의미론적 부분이 많아 자동화의 어려움이 있어 사용자가 직접 생성해야하는 불편함이 따른다. 이는 향후 연구로 보완해야될 점이다.

참고문헌

[1] http://www.ibiblio.org/xml

[2] Aleksander Slominski, "Design of a Pull and Push Parser System for Streaming", http://www.extreme.indiana.edu/xgws/papers/xml_push_pull/, May, 2001

[3] Dennis M.sosnoski, "XML documents on the run, Part 1,2,3", 04.26.2002

[4] Erich Gamma, Ralph Johnson, Grady Booch, Richard Helm, "Design Patterns: Elements of Reusable Object-Oriented Software", 09.1994