

Cocoon 프레임워크를 기반으로 하는 웹 애플리케이션 개발 도구

A Development Tool for Web Applications based on Cocoon Framework

구흥서
Koo, Heung-Seo

청주대학교 컴퓨터정보공학과

요 약

웹 애플리케이션의 복잡도가 증가하면서, 요구조건이 다양한 상황에서 웹 애플리케이션을 보다 빠르고 쉽게 개발하기 위한 해결책으로 Cocoon, Struts, JSF같은 프레임워크를 이용하는 방안들이 최근 보편화되고 있다. 그러나, 초보자는 XML을 사용하여 아파치 Cocoon 사이트맵 파일을 작성하기 어렵다. 본 논문에서는 초보자들이 Cocoon 사이트맵 파일을 쉽게 작성할 수 있도록 지원하는 Java 서블릿 엔진에서 운영되는 Cocoon 사이트맵 편집기를 구현한다. 이 시스템은 세부분, 즉 사이트맵 메인 편집기, 파이프라인 뷰어, 그리고 사이트맵 트레이서로 구성된다. 사이트맵 메인 편집기는 사용자가 Cocoon 애플리케이션을 위한 컴포넌트 파이프라인, 제너레이터, 그리고 다른 환경설정 정보를 정의하는 XML 문서를 작성할 수 있도록 지원한다. 파이프라인 뷰어는 Cocoon 컴포넌트 파이프라인들의 구조를 그래픽 인터페이스를 통해 파악할 수 있도록 지원한다. 사이트맵 트레이서는 사이트맵 파일에 정의된 Cocoon 파이프라인을 따라 흐르는 SAX 이벤트들을 모니터링 하는 기능들을 제공한다.

Abstract

As the complexity of Web applications have increased significantly, the approach using frameworks such as Cocoon, Struts, JSF, is being suggested to develop Web applications more quickly and efficiently. However, it is difficult for beginners to write Apache Cocoon Sitemap files in XML. We implement a Cocoon Sitemap editor on the Java Servlet engine that allows beginners to write Cocoon Sitemap files more easily. Our system consists of three parts: Sitemap Main editor, Pipeline Viewer, and Sitemap Tracer. Sitemap Main editor support users to write XML configuration documents, so-called Sitemap file, that defines all the components pipelines, generators, and other configuration information for a Cocoon application. pipeline Viewer support users to catch the structure of Cocoon component pipelines by the medium of graphical interface. Sitemap Tracer have features to monitor of SAX events through Cocoon pipelines in a Sitemap file.

Key words : 웹 애플리케이션, 프레임워크, Cocoon, 사이트맵, 파이프라인 디버깅.

1. 서 론

웹 애플리케이션이란 웹 시스템 위에 구축된 애플리케이션으로 웹 인터페이스를 제공하기 때문에 사용자들은 사용법이 단순한 웹 브라우저를 통해 서버쪽에 위치한 비즈니스 로직을 포함하는 애플리케이션을 실행시킬 수 있는 장점을 가진다[1].

그러나, 최근의 웹 애플리케이션들은 초기의 단순한 정보 유포 사이트에서 온라인 트랜잭션, 엔터프라이즈급 스케줄링 시스템 등으로 복잡도가 크게 증가했다[2]. 이것의 해결책으로 썬마이크로시스템즈사는 엔터프라이즈급 Java 애플리케이션을 구현하기 위한 J2EE 애플리케이션 모델을 제시하였다. J2EE 모델은 강력한 유연성과 호환성을 제공하지만, J2EE 기반의 웹 애플리케이션을 구축하려면 개발자들이 익혀야 할 명세(specification)와 기술이 너무 방대하고 다루어

야 할 코드가 많다는 점이 어려운 점 중에 하나로 인식되고 있다[3,4].

최근, 엔터프라이즈급 웹 애플리케이션 뿐만 아니라 요구조건이 다양한 상황에서 웹 애플리케이션을 보다 빠르고 쉽게 개발하기 위한 방안으로 Cocoon[5], Struts[6], JSF[7]같은 프레임워크를 이용하는 방안들이 보편화되고 있다. 프레임워크 기반으로 웹 애플리케이션을 개발하면 코딩과 디버깅 시간을 줄일 수 있을 뿐만 아니라, 어떤 디자인패턴을 사용할 지, 특정 기능을 어떻게 구현할 것인지, 발생한 기술적 문제를 어떻게 해결할 것인지 등을 결정하는데, 소요되는 시간을 절감하고 생산성을 향상시킬 수 있기 때문이다[8].

Apache 그룹에서 개발하고, 현재에도 지속적으로 발전시키고 있는 XML 기반 웹 애플리케이션 프레임워크인 Cocoon은 웹 애플리케이션 개발 및 관리비용을 현저하게 줄이고, 하나의 XML 문서를 다양한 클라이언트에 적합하게 변환·배포할 수 있는 우수한 기능을 제공하며, 프레임워크 선택시 중요한 요소인 오픈소스로서의 이점을 가지고 있다. 그러므로, 관련문헌[9,10,11,12]에서 알 수 있듯이 XML을 기

접수일자 : 2005년 10월 12일
완료일자 : 2005년 12월 5일

반으로 하는 많은 개발현장에서 Cocoon을 활용하고 있으며, 최근 크게 성장하고 있는 웹서비스 분야에서도 활용 가능성이 높아지고 있다. 그러나 Cocoon 프레임워크 기반으로 웹 애플리케이션을 개발할 때 핵심요소인 사이트맵(sitemap)은 작성이 복잡하고, 효과적인 코드를 작성하기 위해서는 높은 수준의 Cocoon과 XML 관련 기반 지식을 필요로 하기 때문에 초보자들이 웹 애플리케이션 개발시 Cocoon 프레임워크의 활용 이점을 충분히 발휘하기 어렵다.

본 논문은 Cocoon 프레임워크를 기반으로 복잡한 웹 애플리케이션을 개발할 때 필요한 Cocoon 자원들(resources)의 효율적 관리 및 시각적 개발환경을 지원하여 사용자 편의성을 증대시킨 Cocoon 사이트맵 편집기의 개발에 대해 기술한다. 우리 연구결과인 사이트맵 편집기는 사이트맵 작성시 필요한 기능과 개발 프로세스의 연관성을 고려하여 메인 편집기능을 수행하는 사이트맵 메인 에디터와, 컴포넌트 파이프라인의 흐름을 일목요연하게 파악하고 효율적으로 관리할 수 있는 파이프라인 뷰어(Pipeline Viewer), 그리고 생성·편집된 XML 형태의 사이트맵 파일의 매칭정보를 테스트하고 오류를 디버깅할 수 있는 사이트맵 트레이서(Sitemap Tracer)로 구성된다.

본 논문의 나머지 부분은 다음과 같이 구성된다. 2장은 본 연구를 이해하는데 필요한 Apache Cocoon 관련 배경지식을 기술하고, 3장에서는 사이트맵 편집기의 설계 부분을 설명하며, 4장에서는 구현 및 사용자 인터페이스 부분을 설명한다. 그리고, 마지막으로 6장에서 결론 및 향후 연구에 대해 기술한다.

2. 배경지식

Apache 그룹의 오픈소스 프로젝트인 Cocoon은 XML 콘텐츠를 XSLT(extensible stylesheet language transformations)를 사용하여 다양한 형식의 웹 콘텐츠를 동적으로 출판할 수 있도록 허용하는 XML 기반 웹 애플리케이션 프레임워크로서 콘텐츠와 로직, 그리고 프리젠테이션이 분할된 웹 애플리케이션을 생성하는 플랫폼을 제공한다. 이 장에서는 Cocoon의 시스템 구조와 XML 출판 메커니즘, 그리고 Cocoon 프레임워크 기반의 웹 애플리케이션 개발시 핵심요소인 사이트맵에 대해 살펴본다.

2.1 시스템 구조

Cocoon의 시스템 구조는 그림 1과 같이 운영환경인 서버 및 엔진과 Cocoon Publishing Framework, Components, Avalon Component Management, 그리고 환경설정 파일들로 구성되어 있다. Cocoon Publishing Framework는 서버 및 엔진으로부터 전달된 사용자 요청을 Components에 정의된 일련의 해당 컴포넌트를 이용하여 처리하고, Components는 Cocoon 컴포넌트들, XML 파서와 같은 표준 컴포넌트, 그리고 사용자가 직접 작성한 사용자 정의 컴포넌트들을 포함하며, 이 컴포넌트들은 Avalon Component Management에 의해 관리된다. Cocoon 시스템은 시스템에서 사용되는 다양한 컴포넌트를 관리하는 cocoon.xconf와 컴포넌트들을 이용한 출판 작업을 정의한 sitemap.xmap 파일 등의 환경설정 파일을 이용하여 웹 콘텐츠 출판과정을 정의 및 관리한다[13,14].

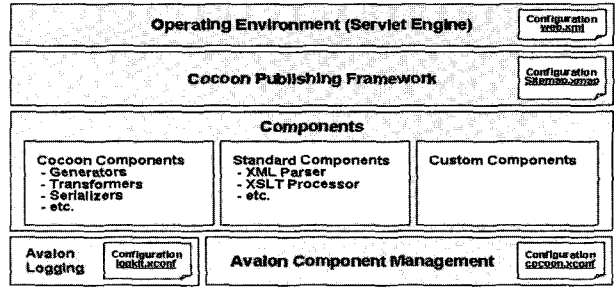


그림 1. Cocoon의 시스템 구조.
Fig. 1. System architecture of Cocoon.

2.2 컴포넌트 파이프라인

그림 2에 나타난 것처럼 Cocoon은 파이프라인 모델을 통해 XML 문서의 생성 및 변환하는 각각의 처리과정을 거쳐 XML 문서를 출판한다. 이를 위해 Cocoon은 URI와 일련의 파이프라인들의 연결정보를 사이트맵이라 불리는 중앙 환경설정 파일에 명시하여 처리하고, 파이프라인을 구성하는 Generator, Transformer, Serializer 컴포넌트들간의 연동은 SAX(Simple API for XML) 이벤트를 통해 이루어지는 메커니즘을 가진다[13,14].

■ **Generator** : 모든 파이프라인은 반드시 하나의 Generator로서 시작되며, Generator가 XML 문서를 읽어 들여 SAX 이벤트를 생성한다. 여러 형태의 데이터 소스들로부터 XML을 추출하는 다양한 Generator가 사용 가능하다.

■ **Transformer** : 파이프라인에 따라 0 또는 하나 이상의 Transformer가 사용될 수 있으며, Generator로부터 전달받은 이벤트에 적절한 처리를 수행한 후 SAX 이벤트를 재생성하여 다음 컴포넌트에 전달한다. 대표적인 Transformer로는 XSLT 변환을 수행하는 XSLT 변환기, 데이터베이스 정보를 추출하는 SQL 변환기 등이 있다.

■ **Serializer** : Serializer는 이전 컴포넌트로부터 전달받은 SAX 이벤트를 클라이언트의 브라우저에 전송하거나 디스크 저장에 적합한 형태의 출력스트림으로 변환한다.

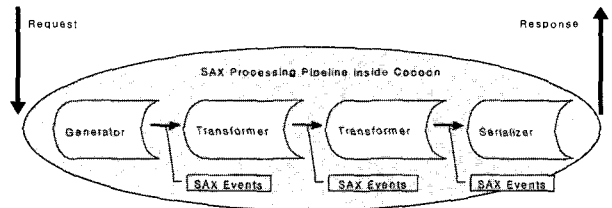


그림 2. Cocoon 파이프라인의 처리과정.
Fig. 2. Process of Cocoon pipeline.

2.3 사이트맵

2.2절에서 기술한 바와 같이 Cocoon은 URI와 일련의 파이프라인들의 연결정보를 사이트맵이라 불리는 중앙 환경설정 파일에 명시하여 처리하는 출판 메커니즘을 가진다. 사이트맵은 W3C XML 표준을 따르는 환경설정 파일로서 컴포넌트 파이프라인에 사용하기 위해 미리 정의된 컴포넌트들을 배치하고, 생성된 컴포넌트들의 조합으로서 정의되는 파이프라인들을 배치하는 Cocoon에서 웹 애플리케이션 개발의 핵심이라 할 수 있다. 그림 3은 사이트맵 파일 구조의 예를 나타낸 것이다.

```

<map:sitemap xmlns:map="http://apache.org/cocoon/sitemap/1.0">
  <map:components>
    <!-- component declarations -->
    <map:generators/>
    <map:readers/>
    <map:transformers/>
    <map:actions/>
    <map:serializers/>
    <map:actions/>
    <map:matchers/>
    <map:selectors/>
  </map:components>
  <map:pipelines>
    <!-- pipeline definitions -->
  </map:pipelines>
</map:sitemap>
    
```

그림 3. Cocoon Sitemap 구조의 예.
Fig. 3. Example of Cocoon Sitemap structure.

3. 사이트맵 편집기의 설계

3.1 설계시 고려사항

사이트맵 편집기는 사용환경이 다르기 때문에 일반 소스 코드 편집기와 다른 몇가지 기능을 필요로 한다. 첫 번째는 Cocoon 프레임워크 컴포넌트 기반의 웹개발 개념을 적용하고 있기 때문에[13], Cocoon 프레임워크에서 제공하는 기본 컴포넌트들과 사용자가 필요에 따라 추가하는 사용자 정의 컴포넌트들에 관한 정보를 관리 및 제공해야 한다. 이러한 기능은 사용자가 파이프라인을 구성하는데 필요한 다양한 기본 컴포넌트들과 사용자 정의 컴포넌트들의 이름, 컴포넌트 타입, 그리고 컴포넌트들과 매핑되는 구현 클래스의 위치정보 등을 기억하지 않아도 되도록 한다. 두 번째, 작성된 사이트맵 파일은 웹 출판되는 콘텐츠의 유형과 복잡도, 콘텐츠의 개인화 정도, 그리고 출력 디바이스의 다양성에 따라 다수의 파이프라인으로 구성된 복잡한 형태가 될 수 있기 때문에, 사이트맵을 구성하는 파이프라인들의 구성 및 흐름(flow)을 시각적으로 파악할 수 있는 파이프라인 흐름에 대한 그래픽 인터페이스 기능이 요구된다. 세 번째는 작성된 사이트맵에

의해 생성된 웹 콘텐츠와 컴포넌트 파이프라인을 통해 흐르는 SAX(Simple API for XML) 이벤트들을 디스플레이 윈도우에 렌더링하여 디버깅 할 수 있는 디버깅 기능이 필요하다.

3.2 클래스 다이어그램

그림 4는 사이트맵 편집기의 주요 클래스들의 클래스 다이어그램을 나타낸 것이다. 사이트맵 편집기는 MDI(Multiple Document Interface)를 기반으로 하는 프로그램으로 여러개의 뷰 윈도우를 가질 수 있다.

Sitemap Editor Frame 클래스는 Sitemap Editor 클래스의 getInstance()에서 자식 프레임 윈도우로 사용할 Sitemap Editor Panel을 등록함으로써 이루어지며, Editor Panel Listener를 통해 프레임 윈도우에서 이벤트가 수행되고 New Editor Panel Menu를 통해 자식 프레임 윈도우의 메뉴가 구성된다. 사이트맵 편집기의 자식 프레임 윈도우는 사이트맵 파일을 열거나, 새로 만들 때 하나씩 생성되며, MDI이므로 여러개의 자식 프레임 윈도우가 생성되어 동시에 여러 문서의 편집을 가능하게 해주고, 자식 프레임에 대해 생성된 여러 뷰와 자식 컨트롤, 메시지 전달 및 윈도우 핸들을 관리하는 역할을 수행한다.

Sitemap Model 클래스의 주된 기능은 응용 프로그램에서 사용할 데이터인 사이트맵 XML 문서를 DOM 트리로 생성하여 유지하는 역할을 수행하는 것이며, Sitemap Editor Panel Impl 클래스는 Sitemap Editor Panel 클래스를 상속받아 구현한 클래스로 생성된 사이트맵의 DOM 트리를 관리하며 해당 XML 문서의 열기 및 저장 등에 관한 기능을 수행한다.

Sitemap Edit Card Panel 클래스는 사이트맵 문서의 구성요소들을 각 요소별로 나누어 효율적으로 관리할 수 있도록 Components 클래스와 Pipeline 클래스에 해당 요소를 추출하여 전달하는 역할을 수행한다. Components 클래스는 J Combo Box를 사용하여 Sitemap Edit Card Panel 클래스로부터 전달 받은 사이트맵 DOM 트리의 Component 요소를 선택적으로 보여주며 순서이동, 삽입, 삭제 등의 기능을 제공한다. ComboBoxListener는 JComboBox에 적용되는 이벤트들에 대한 행동이 구현되어 있는 이벤트 핸들러이다. Pipeline 클래스는 JTree를 사용하여 SitemapEditCardPanel 클래스로부터 전달 받은 사이트맵 DOM 트리의 해당 요소를 트리 형태로 보여주고 삽입, 삭제, 복사, 붙여넣기 등의 편집

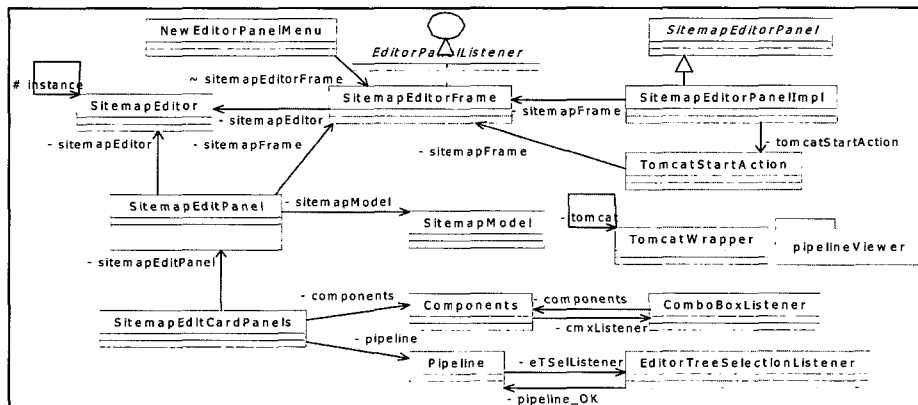


그림 4. 사이트맵 편집기 클래스 다이어그램.
Fig. 4. Class Diagram of Sitemap editor.

이 가능하며 JList를 사용하여 선택 요소 전후에 추가될 수 있는 요소들을 쉽게 파악하고 삽입하는 기능을 제공한다. EditorTreeSelectionListener는 JTree에 적용되는 이벤트들에 대한 행동이 구현되어 있는 이벤트 핸들러이다.

TomcatWrapper 클래스는 서블릿 컨테이너인 톰캣을 구동시키기 위한 기능을 제공하며, TomcatStartAction 클래스가 TomcatWrapper 클래스를 호출하여 톰캣을 시작시키는 역할을 수행한다. 그림 4에 표현된 pipeline viewer 패키지가 톰캣으로부터 실행되는 pipeline viewer 폴더를 나타내며 JSP로 구현된 파이프라인 뷰어는 톰캣이 시작됨에 따라 JSP 문서의 처리가 가능한 상태가 된다. 사이트맵 편집기는 파이프라인 뷰어를 통하여 사이트맵의 DOM 트리를 W3C SVG(Scalable Vector Graphics)[15] 형태로 보여줌으로써 전체적인 구조를 쉽게 알아볼 수 있게 하는 기능을 제공한다.

4. 시스템의 구현

본 연구에서 구현한 사이트맵 편집기는 사이트맵 작성시 필요한 기능과 작성 프로세스의 연관성을 고려하여 메인 편집 기능을 수행하는 사이트맵 메인 편집기(Sitemap Main Editor), 파이프라인을 효율적으로 관리할 수 있도록 하는 파이프라인 뷰어(Pipeline Viewer), 그리고 생성·편집한 사이트맵 파일의 매칭정보를 테스트하고 오류를 디버깅할 수 있는 사이트맵 트레이서(Sitemap Tracer)로 구성된다.

4.1 사이트맵 메인 편집기

사이트맵 메인 편집기는 높은 수준의 기반 지식이 없더라도 Cocoon 애플리케이션의 사이트맵을 효과적으로 작성 또는 편집할 수 있도록 개발된 메인 편집기로서 사이트맵의 핵심기능인 컴포넌트 등록 및 관리와, 등록된 컴포넌트들을 이용한 파이프라인의 구성을 다이얼로그 방식의 시각적 인터페이스로서 제공한다.

4.1.1 사이트맵 메인 편집기

사이트맵 메인 편집기는 사이트맵 작성시 오류 가능성과 작성시간을 줄이기 위해 그림 5처럼 개발 목적에 따른 여러 유형의 기본 사이트맵을 템플릿 형태로 지원하고, Cocoon에서 제공하는 많은 컴포넌트들의 정보를 XML 파일 형태로 제공하여 편집기에 자동 적재, 단순 선택에 의해 파이프라인을 구성할 수 있는 편의성을 제공한다. 또한 버전별 기본 컴포넌트 정보를 XML 스키마 파일 형태로 명세화하여 추가하고자 하는 별도의 사용자 정의 컴포넌트나 Cocoon의 향후 버전을 지원할 수 있도록 하였다.

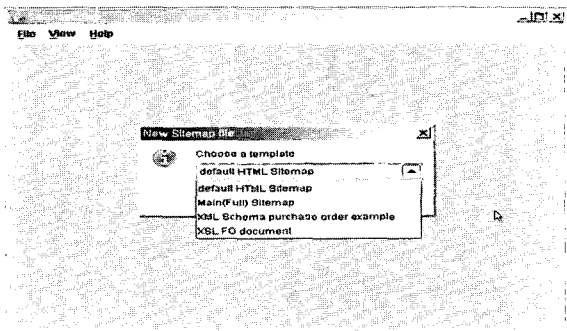


그림 5. 템플릿 선택 사용자 인터페이스.
Fig. 5. Template selection user interface.

4.1.2 컴포넌트 선언 및 파이프라인 구성

사이트맵의 파이프라인을 구성하려면 필요한 컴포넌트들이 사이트맵에 미리 선언되어야 한다. 따라서 새로운 컴포넌트를 선언할 경우, 사용자는 복잡한 구조의 사이트맵 파일을 파악하고 있어야 하며, 적절한 위치에 컴포넌트 타입, 컴포넌트 이름, 구현 클래스 파일의 위치정보, 그리고 디폴트 컴포넌트 등의 정보를 XML 규약에 맞게 정확히 명시하여야 한다. 이는 XML과 Cocoon에 관한 충분한 지식을 갖추지 않은 경우 많은 오류와 시행착오를 갖게 되는 원인이 된다. 사이트맵 메인 편집기는 Cocoon의 기본 컴포넌트뿐만 아니라 사용자가 개발한 사용자 정의 컴포넌트를 손쉽게 선언하여 사용할 수 있도록 그림 6처럼 직관적인 인터페이스를 제공한다.

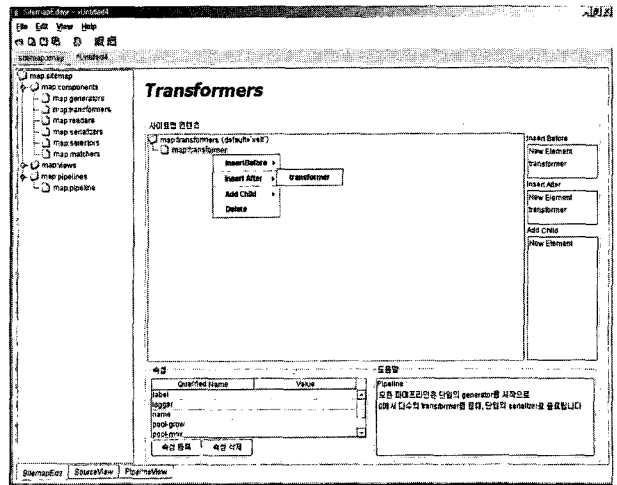


그림 6. 컴포넌트 정의 사용자 인터페이스.
Fig. 6. Component definition user interface.

컴포넌트들을 선언한 후 이를 이용하여 파이프라인을 구성하게 된다. 그림 7은 XML 파일에 XSLT를 적용하여 HTML 문서를 출력하기 위해 Cocoon의 XML Generator, XSLT Transformer, HTML Serialiser로 구성된 간단한 파이프라인을 생성하는 화면이며, 그림 8은 컴포넌트 선언과 파이프라인 등록을 통해 완성된 사이트맵 파일의 내용을 보이고 있다.

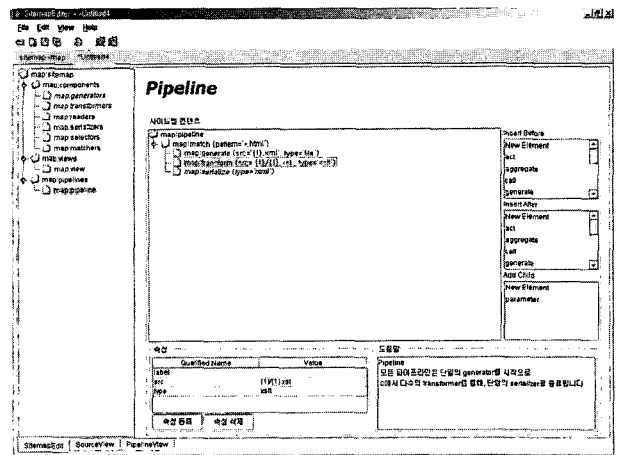


그림 7. 파이프라인 생성 사용자 인터페이스.
Fig. 7. Pipeline generation user interface.

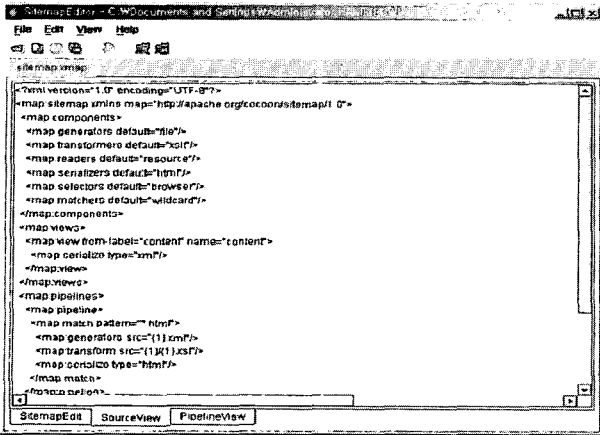


그림 8. 그림 7에서 생성된 사이트맵의 예.
Fig. 8. Example of Sitemap generated in fig. 7.

4.2 파이프라인 뷰어

사이트맵 편집기에서의 결과는 웹 콘텐츠의 복잡도에 따라 다수의 파이프라인으로 구성된 복잡한 형태가 될 수 있으며, 중복 정의되거나 URI와 파이프라인의 잘못된 매핑으로

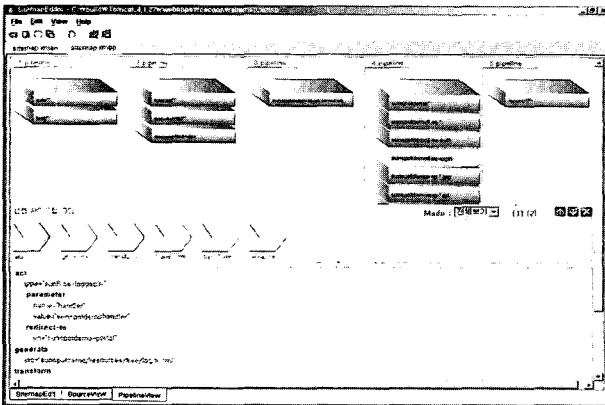


그림 9. 파이프라인 뷰어를 통한 사이트맵 출력 화면.
Fig. 9. Output of Sitemap through Pipeline Viewer.

인해 부적절한 결과를 생성하는 등 관리 측면에서의 많은 비용을 초래할 수 있다. 따라서 복잡한 콘텐츠를 서비스하는 웹 애플리케이션의 경우 각각의 파이프라인의 흐름을 일목요연하게 파악할 수 있는 도구가 필요한데, 파이프라인 뷰어(Pipeline Viewer)가 이러한 목적을 수행한다. 그림 9는 앞에서 생성한 사이트맵 파일의 파이프라인의 흐름을 SVG를 사용하여 그래픽적으로 표현한 파이프라인 뷰어의 출력 화면이다.

4.3 사이트맵 트레이서

사이트맵 트레이서(Sitemap Tracer)는 Cocoon 기반 웹 애플리케이션에서 사이트맵 파일의 매핑 정보와 파이프라인의 SAX 이벤트를 모니터링하는 디버깅 도구로서, 그림 10에 나타난 것처럼 파이프라인의 수행 결과를 출력하는 웹 브라우저 뷰와 사이트맵의 전반적인 구조를 쉽게 파악할 수 있도록 하는 트리구조의 사이트맵 뷰, 그리고 웹 브라우저에 나타난 URL 정보를 바탕으로 사이트맵에서 매칭되는 패턴을 검색하고, 해당 패턴의 소스 및 매핑 정보, 파라미터 정보를 출력하는 디버깅 뷰로서 구성된다.

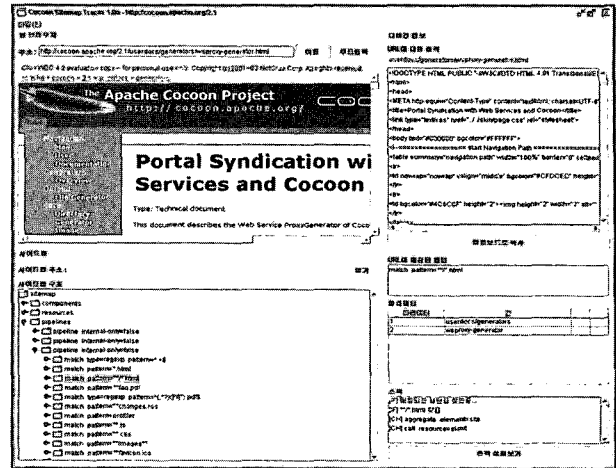


그림 10. 사이트맵 트레이서의 사용자 인터페이스.
Fig. 10. Sitemap Tracer user interface.

5. 평가

Cocoon 프레임워크 기반 웹 애플리케이션 개발 시 핵심인 사이트맵을 작성하려면 그림 11에 나타난 것과 같은 54개의 클래스를 적절하게 활용해야 하지만, 구문오류를 제거하고 사이트맵 컴포넌트들을 중복되지 않게 잘 구조화하는 작업은 어려운 작업이다. 따라서 이러한 작업을 지원하는 도구의 중요도는 매우 높지만 일반적으로 오픈소스를 지원하는 도구들이 다양하지 못하고, 그 종류도 제한적이어서 Cocoon 기반 애플리케이션 개발 시 활용할 수 있는 도구가 전무한 실정이다.

본 연구의 결과인 Cocoon 사이트맵 편집기는 이러한 환경에서 사용할 수 있는 메인 에디터, 파이프라인 뷰어, 그리고 사이트맵을 테스트 할 수 있는 사이트맵 트레이서를 포함하며, 초보자들 또는 숙련자들이 Cocoon 사이트맵 작성이 활용할 수 있는 유용한 도구이다. 그림 12에 나타난 것처럼

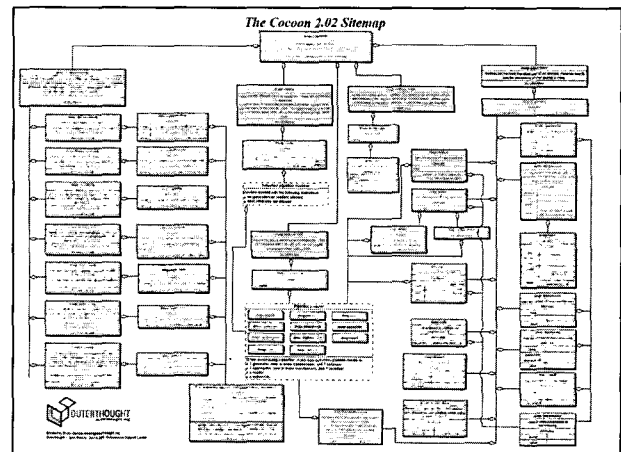


그림 11. Cocoon 사이트맵의 클래스 다이어그램
Fig. 11. Class Diagram for Cocoon Sitemap.

메인 편집기는 사이트맵 코드 작성시 사용자가 정해진 프로세스에 따라 컴포넌트를 선택함으로써 구문오류를 포함하지 않는 코드생성이 가능하게 되며, 일반 소스코드 편집기를 사

용하는 것보다 작업의 생산성을 크게 향상시킬 수 있을 것으로 기대된다. 메인 편집기 단계의 결과코드에는 구문오류는 없지만, 컴포넌트들의 속성값의 오류나 파이프라인 구조상의 오류를 포함할 수 있다. 사이트맵 트레이서는 사용자 요청 URL 중심으로 파이프라인 프로세스의 추적을 지원하므로 파이프라인 구조나 속성값의 오류를 쉽게 찾아낼 수 있도록 지원하므로, 코드 수준에서 디버깅을 하는 것보다 쉽고 빠른 디버깅 작업이 가능하다. 사이트맵 트레이서 단계를 지나면 웹 애플리케이션으로 동작하는 데는 문제가 없으나 파이프라인 구조상의 개선점을 포함할 수 있다. 파이프라인 뷰어는 생성된 사이트맵 코드의 구조적 리팩토링의 필요성 여부를 시각적으로 판단할 수 있는 그래픽 인터페이스를 지원한다. 그러므로, 초보자 수준에서는 작성된 코드의 파이프라인 구조를 쉽게 파악할 수 있도록 도움 받으며, 숙련자들도 코드 수준에서 보다는 파이프라인의 전체적 구조를 파악함으로써 구조의 리팩토링의 필요성 여부를 용이하게 판단할 수 있을 것이다.

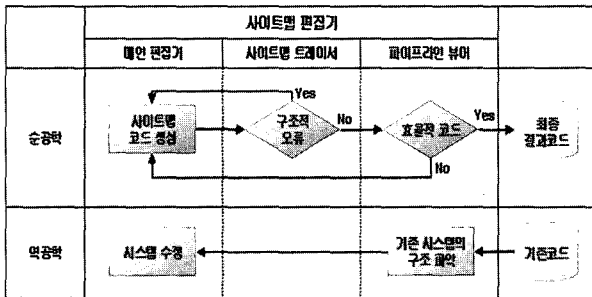


그림 12. Cocoon 사이트맵 편집기의 기능
Fig. 12. Features of Cocoon Sitemap.

6. 결 론

본 논문에서는 생산성 향상을 위해 프레임워크 기반으로 웹 애플리케이션을 개발하는 최근 동향을 반영하여, Apache 그룹에서 개발하고 발전시키고 있는 XML 기반 웹 애플리케이션 프레임워크인 Cocoon의 사이트맵 편집기를 설계·구현하였다. 구현한 사이트맵 편집기는 사이트맵 작성시 필요한 기능과 개발 프로세스의 연관성을 고려하여 메인 편집기능을 수행하는 사이트맵 메인 에디터와, 컴포넌트 파이프라인의 흐름을 일목요연하게 파악하고 효율적으로 관리할 수 있는 파이프라인 뷰어, 그리고 생성·편집된 XML 형태의 사이트맵 파일의 매칭정보를 테스트하고 오류를 디버깅할 수 있는 사이트맵 트레이서로 구성되어 있다.

본 사이트맵 편집기를 통해 Cocoon과 XML 관련 기반 지식이 많지 않은 초보자들도 Cocoon 애플리케이션의 사이트맵을 쉽고 빠르게 작성할 수 있기 때문에, Cocoon 프레임워크의 활용 이점을 취할 수 있게 될 것이다. 또한 기존의 사이트맵도 역공학 과정을 통해 컴포넌트 파이프라인의 흐름을 일목요연하게 파악할 수 있기 때문에 유지관리가 쉬워지는 장점을 가진다.

향후에는 이 도구를 다양한 Cocoon 기반 웹 애플리케이션 개발에 적용해 개발 프로세스를 보다 정형화하는 연구를 수행하고, 이러한 결과를 사이트맵 편집기에 반영하여 사용자 편의성과 개발 효율성을 높일 계획이다.

참 고 문 헌

- [1] J. Conallen, Modeling Web Applications with UML, Addison-Wesley, 1999.
- [2] A. Ginige, "Web Engineering", IEEE, 2001.
- [3] Mohamed E. Fayad, Douglas C. Schmidt, and Ralph E. Johnson, Building Application Frameworks, Wiley Computer publishing, 1997.
- [4] Allan McNaughton, J2EE Development For the Rest of Us, IBM, 2005.6.30.
- [5] Apache Cocoon, <http://cocoon.apache.org/>.
- [6] Jakarta Struts, <http://struts.apache.org/>.
- [7] Eric Armstrong, et al., "Chapter 17: JavaServer Faces Technology", in The J2EE 1.4 Tutorial, <http://java.sun.com/j2ee/1.4/docs/tutorial/doc/index.html>.
- [8] Andrei Cioroianu, Frameworks To The Rescue, Mastering J2EE Application Development Series, Oracle, 2005.
- [9] Joao Costa, Kim Hansen, Susana Cabaco, Cocoon Vs MVC, Faculdade de Ciencias e Tecnologia, 2003.1.10.
- [10] Sunny Yaldiz, Evaluation of Web Application Development Frameworks and Object-Relational Mappers: A Case Study, Technical Univ. Hamburg-Harburg, Germany, 2004.2.16.
- [11] Carsten Ziegeler, Matthew Langham, Building XML Portals with Cocoon, <http://www.xml.com/pub/a/2002/07/24/xmlportal.html>, 2002.7.24.
- [12] Evan Lenz, Cocoon and 4Suite for Content Management: The Best of Both Worlds at Seattle University School of Law, <http://www.xmlportfolio.com/xml europe2003/>.
- [13] Bill Brogden, Conrad D'Cruz, Mark Gaither, Cocoon 2 Programming, SYBEX Inc., 2003.
- [14] Lajos Moczar, Jeremy Aston, Cocoon Developer's Handbook, Developer's Library, 2003.
- [15] SVG, <http://www.w3.org/Graphics/SVG/>.

저 자 소개



구흥서(Koo, Heung-Seo)
2003년 : 인하대학원 전산학과 (이학박사).
2004년~현재 : 청주대학교
컴퓨터정보공학과.

관심분야 : 데이터베이스, 정보 모델링,
웹애플리케이션 프레임워크.

Phone : 043) 229-8492
Fax : 043) 229-8432
E-mail : hskoo@cju.ac.kr