XACML을 적용한 Web 기반 기업 Application 시스템을 위한 접근제어 설계

양경돈°, 이희조
고려대학교 컴퓨터학과
alpha514@empal.com°, heejo@korea.ac.kr

Design of Access Control for Web based Enterprise Application System Using XACML

Donnie Yang°, Hee Jo Lee
Dept. of Computer Science & Engineering, Korea University
1, 5-ga, Anam-dong, Sungbuk-gu, Seoul, 136-701, Korea

요 약

기업의 많은 활동들이 인터넷을 기반으로 하는 시스템으로 점차로 넓어지면서, 기업 내 산재되어 있는 많은 Application 시스템 상호간 연계되거나 통합하기도 한다. 시스템의 연계나 통합에 있어 중요한 요소 중 기업 내 시스템 자원에 대한 접근 제어에 대한 문제가 중요한 이슈 중 하나이다. 이를 위해 각 기업이나 그룹에서는 여러 접근 제어 기술을 구현한 솔루션을 도입하고 있지만, 기업 내 표준에 적합하게 구현되어 있지 않거나, 기업의 표준에 적합하게 구현이 되어 있다면 고비용을 지불해야 되는 문제점으로 인해, 기업에서 직접 기업 내 자원에 대한 접근 제어를 구현하여 사용하는 실정이다. 이에 본 논문에서는 현재의 접근 제어 기술 중 XML에 기반하며 표준화가 정립된 XACML 접근 제어 기술을 활용하여 웹 기반 기업 Application 시스템의 자원에 대해 접근 제어를 할 수 있는 방안을 연구하였다. 아울러 향후 기업의 웹 기반 애플리케이션 시스템의 상호 연계나 통합을 위한 접근 제어를 추진할 때 도움이 될 수 있고자 한다.

1. 서 론

많은 기업들이 인터넷 환경의 급속한 발달로 기업 내 시스템을 웹 기반 환경의 시스템으로 전환을 하거나 새로운 시스템을 도입할 때 웹 기반 환경의 시스템으로 도입하는 추세이다. 기업 내 다양한 시스템이 많아짐에 따라 기존의 시스템과 웹 기반 시스템 간 또는 웹 기반시스템들 간의 연계작업이나 통합작업이 차츰 많아지게 되었고, 이러한 시스템의 연계나 통합에 있어 중요한 여러 요소 중 기업 내 시스템 자원에 대한 접근제어에 대한 문제가 중요한 이슈로 등장을 하게 되었다. 기업 내 시스템의 연계나 통합작업의 일환으로 접근제어를 구축하기 위해 접근제어 관련 솔루션을 도입하기도 하지만 기업의 원하는 요건을 충족시키지 못하거나, 요건을 충족할 경우 비용이 높아지게 되는 경우가 많아, 결국 기업 자체적으로 필요한 시스템간의 상호 접근제어를 일일이 구축해야하는 실정이다. 이렇게 전사적 차원의 접근제어와 관련된 표준 정의 없이 기업 내 시스템 구축 시 자체적으로 접근제어 관련 표준을 정의하여 구축하다보니 표준에 대한 많은 문제점이 도출하게 되었고, 차후 시스템들에 대한 유지보수가 힘들어지는 경우가 빈번히 발생을 하였다. 이러한 상황을 해결하기 위해 여러 가지 접근제어 보안 기술 중 웹서비스 표준화 기구인 OASIS에서 표준화하여 발표한 XACML[1]을 적용하여 문제 해결에 접근할 수 있다. 현재 기업의 개발 환경에서 주를 이루는 객체 지향 개발 방법과 개발 패턴에 XACML 접근제어를 연동하여 웹 기반 기업 Application 시스템들을 위한 표준화된 접근제어 방안을 연구하였다.

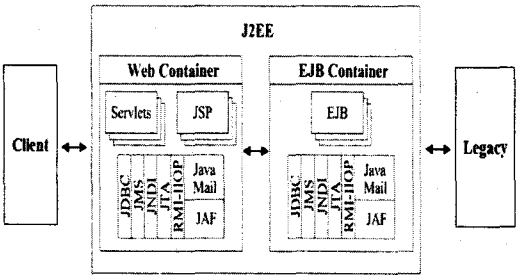
본 논문에서는 서론에 이어 2장에서 현재 기업 시스템 구축 시 많은 수를 차지하는 객체 지향 개발 방법과 생산성 향상을 위한 개발 패턴 그리고 XACML에 관한 이론적인 내용을 설명한다. 3장에서는 현재 개발되는 접근제어에 문제점을 찾아보고 XACML을 적용한 접근제어 모델을 정의한다. 마지막으로 결론 및 향후 연구 방향을 제시한다.

2. 관련연구

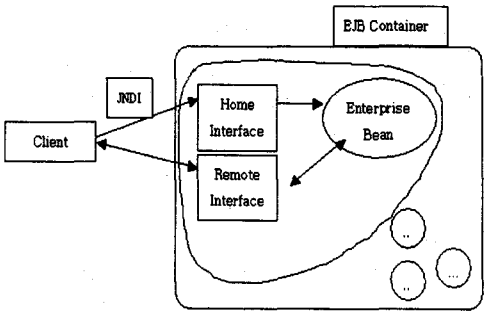
2.1 EJB 환경의 개발 방법

현재 시스템 구축시 주로 사용되는 개발 방법으로는 객체 지향 개발 방법으로, Sun Microsystems사에서 웹 환경을 고려한 기업형 개발 방법을 집약하여 J2EE(Java2 Enterprise Edition) 개발 프레임워크를 고안하여 제시하였다[2]. 최근 기업들은 이 J2EE의 기술 요소 중 하나이며 서버 측의 분산 컴포넌트 환경을 기본으로 하는 EJB(Enterprise Java Beans)[3] 환경을 주로 사용하고 있다. EJB란 Enterprise라는 이름에서 느낄 수 있듯이, 대규모 프레임워크를 위한 자바 기술의 일부로 분산 환경 하에서 Application을 개발, 배포, 실행하기 위한 아키텍처이다. EJB 개발 환경의 장점으로는 퍼포먼스, 확장성, 보안, 분산

트랜잭션처리, 재사용성, 호용성에서 향상을 들 수 있으며, 멀티티어간의 역할 분담 과정에서 발생할 수 있는 상호 신뢰성 문제 또한 해결하여 다양한 트랜잭션서비스를 제공하는 장점을 가지고 있다. [그림 1]은 J2EE구성 개념도 이며 [그림 2]는 EJB의 아키텍처의 구조도 이다.



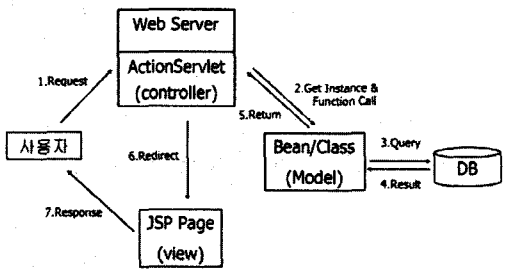
[그림 1] J2EE 구성 개념도



[그림 2] EJB 아키텍처 구조도

2.2 MVC Model 2 개발 패턴

최근에는 인터넷 및 분산 환경에서 개발 패턴을 시스템에 적용하는 연구에 많은 관심이 고조되고 있다[4][5]. 또한 기업의 시스템 구축 시에도 개발 패턴을 이용하는 사례가 점점 증가하고 있다. 현재 기업 시스템 구축시 가장 많이 사용되는 개발 패턴으로는 MVC(Model, View, Control) Model 2[6]방식이다. 개발 패턴을 사용하는 장점으로는 상호 운영성, 유연성, 확장성, 유지보수성, 재사용성과 생산성을 높일 수 있기 때문이다. MVC Model 2 패턴으로 구축된 웹 Application 시스템은 기능들을 캡슐화 하여 변경에 대한 영향을 최소로 하는 구조이므로 확장성, 유지보수성 면에서 좋은 방법으로 인식되고 있다[7]. MVC Model 2 구조는 View와 비즈니스 객체를 분리하여 웹 개발 프로젝트를 작성하는 것을 기본적으로 하여 비즈니스 객체의 상태가 변경될 경우 비즈니스 객체 개발이 View와 연관되지 않도록 하기 위함이다. [그림 3]는 MVC Model2 패턴을 사용하는 웹 기반 개발 환경의 구성도 이다.



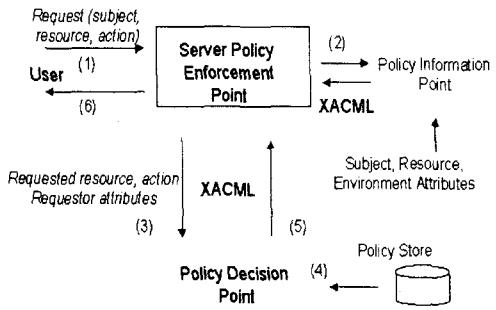
[그림 3] MVC Model 2 패턴을 사용한 웹 기반 개발 환경 구성도

2.3 XACML 접근 제어

XACML(eXtensible Access Control Markup Language)은 접근제어 정책을 통해 보안이 요구되는 자원에 대해 미세한 접근 제어 서비스를 제공 할 수 있는 XML[8]기반의 언어이다. XACML의 목적은 인터넷 상의 접근제어 서비스를 위한 다양한 제품들 및 그 제품들의 서로 다른 환경들 사이에서 일관되게 적용할 수 있는 권한부여(authorization) 정책을 제공하고, 그 정책을 통하여 기존의 다양한 환경 및 방식을 가진 접근 제어 제품들에 상호 운영성을 제공하기 위한 것이다. XACML의 정의에 따라 각각의 사용자별 자원에 대한 접근 정책을 수립하고 적용할 수 있다. 접근에 대한 허가 또는 거부와 같은 단순한 접근제어를 하는 것이 아니라 보다 미세한 접근 제어 모델을 제공한다[9]. XACML은 XML로 기술된 정책 언어와 접근 제어 결정 요구/응답(request/response) 언어로 구성된 접근 제어에 대한 표준이다. 정책 언어는 규칙, 정책 및 정책집합 등에 관한 통상적인 접근 제어 요구사항들에 대해 기술하고 있으며 함수들, 데이터 타입들 및 조합 논리(combining logic)등에 대해서도 정의하고 있다. 요청 언어는 어떤 개체(subject)가 특정 자원(resource)에 대해서 특정한 동작(action)을 수행할 수 있는 지에 대한 질의를 구성할 수 있게 하고 응답 언어는 요청에 대한 결과를 표현하는데 사용하며 응답은 허용(permit), 거절(deny), 부정(indeterminate), 비적용(not applicable)등의 4가지 결과로 표시된다. 어떤 접근 제어 어플리케이션이 XACML과 호환된다고 하면 요청하는 문맥이 XACML의 요청 언어로 기술되고, 응답 받은 XACML로 기술된 문맥을 이해할 수 있음을 뜻한다. 호환되지 않는 접근 제어 어플리케이션이라면 요청/접근 문맥을 XACML 요청/응답 언어로 기술된 문맥으로 바꿀 수 있는 변환이 필요하다[10]. XACML의 정책에서 리소스는 XML을 사용하여 표현되는 어떠한 객체도 될 수 있으며 XACML은 XPath[11]나 LDAP 등 다양한 프로토콜과 함께 바인딩하여 사용될 수 있으며 필요하다면 어떠한 새로운 프로토콜과도 함께 사용

될 수 있다. XACML TC(Technical Commitee)에서는 XACML로 정의된 기술로 정책과 인증을 표현하기 위한 XML 스키마[12]를 제정하고 있다.

XACML은 공통의 산업 명세를 만드는 국제적인 컴소시엄인 OASIS(the Organization for the Advancement of Structured Information Standards)에 의해 2005년 2월 버전 2.0의 표준안이 완성된 상태이고, 현재는 버전 3.0의 개발이 진행 중이다[1]. [그림 4]는 XACML의 프로세스 흐름도[13]이며, [그림 5]는 파일 접근을 위한 권한 속성 구조의 Privilege Statements를 XACML규칙에 의해 인코딩한 예제이다.



[그림 4] XACML의 프로세스 흐름도

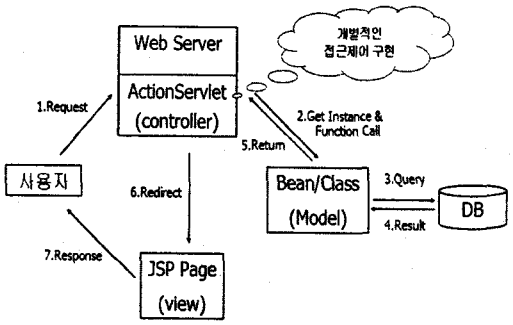
3. 표준화된 XACML을 적용한 접근제어 구축 방법의 연구

3.1 개별적인 접근제어 구축 방법

최근 기업에서 EJB와 MVC Model 2 개발 패턴을 도입하여 시스템을 구축하는 추세이다. 이에 맞추어 기업 내 다양한 시스템들과의 시스템 연계나 통합이 많아지고 있으며, 시스템들의 자원에 대한 접근제어 관련 개발도 많아지는 추세이다. [그림 6]에서와 같이 접근제어 구현에 있어 Controller부에 해당하는 ActionServlet의 상위 클래스에서 사용자의 역할을 찾아 객체에 대한 접근에 적합한 역할을 부여 받은 사용자인지를 판단하여 객체에 대한 접근을 허가 또는 부인하여 응답을 하위의 해당 ActionServlet에게 보내고, 이 해당 ActionServlet에서 받은 응답에 맞추어 다음 단계를 개발하는 방식으로 진행되고 있다. 이 방법에서는 연계되거나 통합되어야 할 해당 시스템 간에 포인트 투 포인트 방식으로 개발이 이루어짐으로 개발 속도가 빠르다는 장점을 가지고 있으나, 접근 권한에 대한 정책이 바뀔 경우 해당 클래스의 소스코드를 바꿔야 하는 문제점을 내포하고 있다. 더불어 시스템에 개별적인 접근 권한 정책을 가지고 개발이 진행 되며, 객체에 대한 세밀한 접근 제어를 구현하기가 무척 어려운 문제점도 내포하고 있다.

```
<Rule RuleId="File-Privilege-Rule" Effect="Permit">
  <Target>
    <Subjects>
      <Subject>
        <SubjectMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:x500Name-equal">
          <AttributeValue DataType="urn:oasis:names:tc:xacml:1.0:data-type:x500Name">
            CN=Park jaewon ,OU=Soongsil Uni User,OU=Class 2,O=setest,C=KOREA
          </AttributeValue>
          <SubjectAttributeDesignator AttributeId="urn:oasis:names:tc:xacml:1.0:subject:subject-id"
DataType="urn:oasis:names:tc:xacml:1.0:data-type:x500Name" />
        </SubjectMatch>
      </Subject>
    </Subjects>
    <Resources>
      <Resource>
        <ResourceMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:anyURI-equal">
          <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#anyURI">
            setestftp://test.setest /data/eamtest/results.dat
          </AttributeValue>
          <ResourceAttributeDesignator AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-id"
DataType="http://www.w3.org/2001/XMLSchema#anyURI" />
        </ResourceMatch>
      </Resource>
    </Resources>
    <Actions>
      <Action>
        <ActionMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
          <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">
            Read
          </AttributeValue>
          <ActionAttributeDesignator AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"
DataType="http://www.w3.org/2001/XMLSchema#string" />
        </ActionMatch>
      </Action>
    </Actions>
  </Target>
</Rule>
```

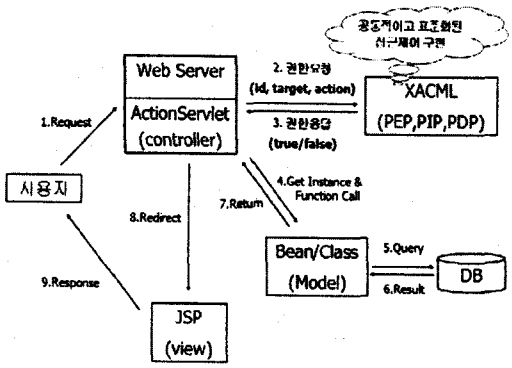
[그림 5] XACML 규칙에 의해 인코딩한 파일 접근 Privilege Statements



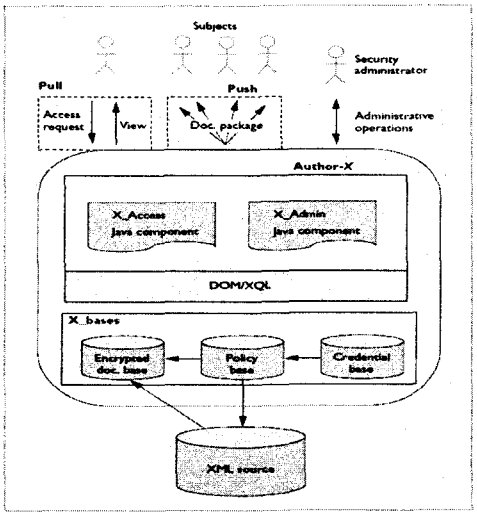
[그림 6] 개별적인 접근제어 구현

3.2 XACML을 적용한 접근제어 구축 방법

앞에서 언급된 사항의 문제점을 해결하기 위해 표준화가 확립된 XACML 접근제어 개념을 도입하면 [그림 7]과 같은 방법을 찾을 수 있다. [그림 8]과 같은 Author-X System 아키텍처 방법을 활용하여[14] 사용자에 대한 접근 권한 정책과 시스템 자원과 대한 정보를 XML포맷의 파일로 각각의 Policy 저장소와 Information 저장소에 관리를 한다. 이전 방법과의 차이점은 Controller부에 해당하는 ActionServlet에서 XACML 접근제어 관련 클래스를 통해 접근 권한 여부를 묻는다. XACML의 PEP는 PIP통한 자원에 대한 정보와 PDP를 통해 결정된 정책 결과(PDP는 'permit' 또는 'deny' 만을 통보)를 가지고 비교하여 PEP는 PDP결과가 'permit'이면 해당 자원에 대한 EJB 인스턴스를 반환하고 'deny'일 경우 거부 메시지를 반환한다. ActionServlet에서 XACML의 서비스 결과 거부 메시지 받으면 거부와 관련된 View로 리다이렉트 되며, 해당 자원의 EJB 인스턴스가 반환되었을 경우 요구했던 Action 을 수행하여 사용자가 원하는 View를 보내준다.



[그림 7] XACML을 적용한 접근제어 구현



[그림 8] Author-X의 시스템 아키텍처

3.3 개별적인 접근제어 구축 방법과 XACML을 적용한 접근제어 구축 방법의 비교

기존의 개별적인 접근제어 구축 방법에 비해 새로 제시하는 XACML을 적용하여 접근제어를 구축하는 방안의 이점을 살펴보면 다음과 같다. 우선 접근 제어와 관련된 개발 방법의 표준화를 제공한다. 결국 표준화로 인하여 시스템간의 연계와 통합에 있어 상호운영성, 유연성, 확장성, 유지보수성, 재사용성을 높일 수 있게 되었다. 더불어 기업이 요구하는 수준의 정책을 구현함에 있어 선택의 폭 넓어 졌다. 그리고 XML의 장점인 이식성과 확장성의 이점을 고스란히 승계하여 시스템의 접근 제어에 있어 타 시스템과의 연계와 통합에 있어 어렵지 않게 개발을 할 수 있게 되었다. 개별적인 접근제어 구축 방법과 XACML을 적용한 접근제어 구축 방법의 간단히 비교를 하면 다음의 [표 1]과 같다.

[표 1] 두 방법의 비교

비교 요소	개별적 접근제어	XACML을 사용한 접근제어
권한 정책 수정	어렵다.	쉽다.
권한 정책의 유연성	결여 되어 있다.	세밀한 접근 제어 가능.
접근제어의 표준화	상황에 따라 다름.	OASIS 단체에 의한 표준.
확장성	나쁨.	좋음(XML의 장점이다).
개발 속도	빠름.	시간 조금 걸림.(표준에 대한 이해 필요)
유지보수성	나쁨.	좋음.

시스템 의존성	높음.	낮음(XML의 장점이다).

4. 결론 및 향후 과제

XACML은 XML문서 또는 많은 다양한 시스템 자원에 대한 세밀한 접근을 제어하여 접근자의 특징적인 행위에 대한 정책을 수립하고 실행함으로써 효율적인 자원 관리를 할 수 있는 접근제어 표준 기술이다. 본 논문에서는 웹 기반 기업 Application 시스템을 위한 접근제어 방안에 XACML을 적용하는 방안을 제시하였다. 앞으로 계속적으로 증가하는 시스템과 이 시스템으로 인해 비약적으로 증가하는 사용자 인증 정보는 시스템의 비효율성과 신뢰성 저하를 야기할 것이다.

따라서, 증가하는 데이터와 이에 따른 시스템의 연계와 통합 그리고 유지보수는 중요한 문제이다. 많은 시스템과 다양한 프레임워크에서 XML을 이용한 보안 프레임워크가 개발되고 있다. XML포맷을 이용한 보안은 확장성과 유연성을 가지고 여러 분야에서 응용될 수 있다.

향후 과제로는 본 논문에서 제시한 XACML 접근제어를 사용하여 웹 기반 기업형 Application 시스템을 구현하여 제시한 방안의 효율성을 검증하는 것이다.

참고문헌

[1] OASIS, XACML 2.0 Specification, http://www.oasis-open.org/specs/index.php#xacmlv2.0, February, 2005.

[2] Sun Microsystems, Java 2 Enterprise Edition Technology, http://java.sun.com/products/j2ee/index.jsp

[3] Sun Microsystems, Enterprise Java Beans Technology, http://java.sun.com/products/ejb/index.jsp

[4] D. C. Schmidt, "Experience Using Design Patterns to Develop Reuseable Object-Oriented Communication Software", Communication of ACM(Special Issue on Object-Oriented Experiences), vol. 38, October, 1995.

[5] F. Buschmann, R. Meunier, H. Rohnert, P. Sommerlad, and M. Stal, "Pattern-Oriented Software Architecture - A System of Patterns", Wiley and Sons, 1996.

[6] Steve Burbeck, "Application Programming in Smalltalk-80 : How to use Model View Controller(MVC). Available", http://st-www.cs-uiuc.edu/users/march/st-doc/mv.htm, January, 1992

[7] 권기현, "Model 2 프레임웍 기반 웹 에플리케이션 자동 생성 시스템 설계", 한국컴퓨터 산업교육학회 논문집, 제4권, 제1호, 2003.

[8] World Wide Web Consortium(W3C), "Extensible Markup Language(XML)", http://www.w3.org/XML/, May, 2006.

[9] Markus Lorch, Seth Proctor, Rebekah Lepro, "First Experiences Using XACML for Access Control in Distributed Systems", ACM Workshop on XML Security, October, 2003.

[10] 김주한,문기영, "XML 기반 접근제어 기술 동향", 한국정보보호학회지, 제13권, 제4호, 2003.

[11] World Wide Web Consortium(W3C), "XML Path Language(Xpath)", http://www.w3.org/TR/XPath/, May, 2006.

[12] World Wide Web Consortium(W3C), "XML Schema", http://www.w3.org/XML/Schema/, May, 2006.

[13] Phil Griffin, "Introduction to XACML", http://dev2dev.bea.com/pub/a/2004/02/xacml.html, February, 2004.

[14] E. Berition, S.Castano, E.Ferrai, "Secruring XML documents with Author-X", IEEE internet Computing, vol. 5, no. 3, May/June, 2001.