XCAP 서버 기능 설계 및 구현

현욱, 박선옥, 이일진, 강신각
*한국전자통신연구원

Design and Implementation of XCAP Server

Wook Hyun, Sunok Park, IlJin Lee ,Shingak Kang
*Electronics and Telecommunications Research Institute

E-mail : whyun@etri.re.kr

요 약

XCAP(XML Configuration Access Protocol)은 IETF의 SIMPLE WG에서 현재 표준화 과정 진행 중에 있다. 네트워크에 위치한 많은 통신 어플리케이션들은 Request를 처리하는 시점에서 사용자 별 정보를 접근하는 것을 필요로 하며 사용자별 정보는 네트워크 내에서 유지되며 end 사용자에 의해 관리되도록 하고 있다. XCAP은 이러한 정보들을 관리하고 조직하기 위한 방법으로 제안된 프로토콜로써 HTTP 기반에서 동작한다. 각 정보들은 특정 어플리케이션에서 각자에게 필요한 형태의 XML 데이터로 규격화 되어 있으며, 어플리케이션마다 unique한 ID인 AUID(Application Unique ID)를 할당 받게 된다. 그리고 이러한 새로운 Application Usage들은 사용하고자 하는 XML의 스키마와 default namespace, MIME Type, Validation Constraints, Data Semantics, Naming Conventions, Resource Interdependency, Authorization Policy등을 규정하도록 되어 있다. XCAP 서버는 단말들로부터 이러한 configuration 정보들을 전달받아 조직화하여 유지 관리하는 기능을 수행하게 된다.

본 고에서는 XCAP 서버의 기능을 구현하기 위하여 기능의 정의 및 블록별 기능 분리를 통한 설계 방법 및 구현 방법에 대하여 논하고자 한다.

ABSTRACT

XCAP(XML Configuration Access Protocol) which has been proposed in IETF is based on both XML and HTTP protocol. XCAP server maintains user's configuration information for specific application which is described by XML. This protocol can be applied to many application servers for adapting user's preferences. There can be many way to interwork with other application servers. In this paper, we will talk about the experience of designing and implementation of XCAP server and the way of interwork with application servers.

키워드

XCAP, XCAP server

Ⅰ. 서 론

XCAP(XML Configuration Access Protocol)[1]은 IETF의 SIMPLE WG에서 제안되어 현재 표준화 과정 진행중에 있다. 많은 통신 어플리케이션들은 Request를 처리하는 시점에서 사용자 별 정보를 접근하는 것이 필요하며 사용자별 정보는 네트워크 내에서 유지되며 end 사용자에 의해 관리되도록 하고 있다. 이러한 정보들을 관리하고 조직하기 위한 방법으로 제안된 프로토콜로써 HTTP 기반에서 동작한다. 각 정보들은 특정 어플리케이션에서 각자에게 필요한 형태의 XML 데이터로 포맷팅이 되어 있으며, 어플리케이션 마다 unique한 ID인 AUID(Application Unique ID)를 할당 받게 된다. 그리고 이러한 새로운 Application Usage들은 사용하고자 하는 XML의

스키마와 default namespace, MIME Type, Validation Constraints, Data Semantics, Naming Conventions, Resource Interdependency, Authorization Policy등을 규정하도록 되어 있다.

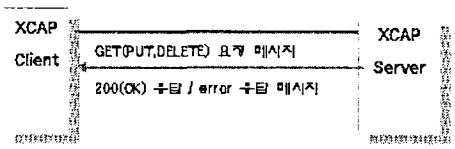

그림 1 XCAP protocol

위 그림에서 보는 바와 같이 XCAP Client는 세가지 메소드를 사용하여 원하는 데이터를 조작하며 저장시킬수가 있으며, XCAP 서버는 대개 웹서버 위에서 구동되며 사용자의 요청에 따른 데이터를 유지 관리하게 된다. 이 과정에서 인증 및 인가의 처리를 위하여 HTTP Digest Authentication을 사용하게 된다. 만약 Request를 처리함에 있어 문제가 생길 경우 적절한 error response를 발생시킨다. XCAP은 네트워 기반 어플리케이션에서 다양하게 사용될 수 있다. 이를테면, 특정 어플리케이션의 설정(configuration)정보를 XCAP 서버에 저장을 하고 실행되는 시점에서 이 서버로부터 해당 정보를 가져와 어플리케이션의 동작을 조정할수 있다. 이렇게 함으로써 사용자의 이동에도 동일한 환경을 구축할 수 있게 되는 장점이 있다. 현재 IETF에서 XCAP Usage 로 제안되어 표준화 과정중에 있는 presence authorization policy 조정, resource list 목록 관리등의 기능들은 프레즌스 서비스의 한 부분으로써 사용될 것이다.

II. XCAP 서버 기능 설계

이번 장에서는 XCAP 서버 기능 설계를 위하여 XCAP 서버가 가져야 할 기능과 각 기능별 블록 구성관계를 설명하도록 한다.

가. XCAP 서버 기능 정의

XCAP 서버는 XCAP 클라이언트로부터의 요청을 수신받아 사용자의 데이터를 관리해주는 기능을 담당한다. 기본적으로 XCAP은 HTTP기반위에서 동작하는 프로토콜이므로 XCAP 서버는 HTTP 1.1을 지원할 수 있어야 하며 다음과 같은 기능을 지원할 수 있어야 한다.

■ XCAP 데이터 저장 및 변경

HTTP의 PUT 메소드에 의해 XCAP 데이터가 전달되며 이경우에는 HTTP URI에 node를 지정하지 않고 document만 지정함으로써 문서를 대체시킬수 있어야 한다. node가 명시된 경우에는 해당 node를 찾아 PUT 메시지의 body에 포함된 XML 데이터를 그 위치에 반영해 주어야 한다.

■ XCAP 데이터 검색 및 인출

HTTP의 GET 메소드에 의해 특정 정보를 요청하게 되는데 node가 명시된 경우 특정 document의 node를 Xpath 쿼리를 통해 찾아내는 기능을 필요로 한다. node가 명시되지 않은 경우에는 해당 document의 모든 정보를 그대로 전달하면 된다.

■ XCAP 데이터 부분 삭제 및 전체 삭제

HTTP의 DELETE 메소드에 의해 특정 정보의 삭제를 처리하게 되는데, HTTP URI의 node를 명시함으로써 특정 부분만을 삭제할 수 도있고 문서 전체를 삭제할수 있어야 한다.

아래 그림은 XCAP 클라이언트 서버간의 동작을 개략화하여 도시한 것이다.

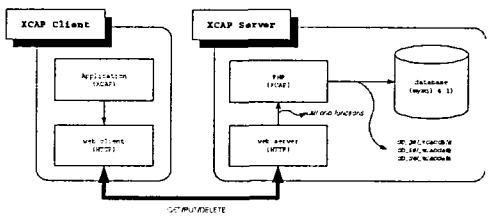

그림 2 XCAP 클라이언트 및 서버간 동작

나. XCAP 서버 기능별블록 설계

XCAP 서버의 기능을 수행하기 위하여 아래 그림과 같은 형태로 제공하는 기능별로 블록을 나누어 보았다.

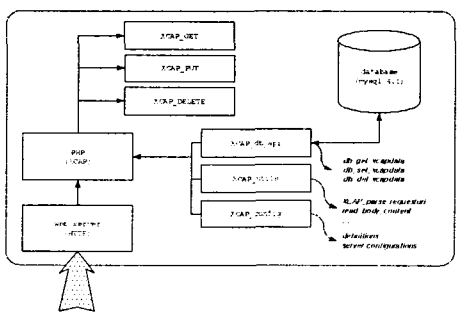

그림 3 XCAP 서버 기능 블록

위 그림에서 보는 바와 같이XCAP 서버는 웹서버 기반에서 별도의 PHP 또는 CGI 스크립트등으로 구동될 수 있다. XCAP 서버의 주요 기능인 GET, PUT, DELETE 처리 루틴과 데이터베이스에 접근하기 위한 접속부, URI 파싱과 응답 메시지 생성등에 관련된 기능을 하는 XCAP 유틸리티 함

수군, XCAP 서버의 동작을 지정하는 설정 함수 군등으로 그 기능들을 나누었으며, XCAP_GET, XCAP_PUT, XCAP_DELETE 모듈들에서 유기적으로 각 함수들을 호출하여 기능을 제공할 수 있게 하였다.

III. XCAP 서버 구현

리눅스기반에서 아파치 2.0 웹 서버와 PHP4.1, Mysql 4.1 을 활용하여 XCAP 서버기능을 구현하였다.

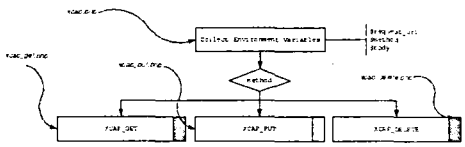

그림 4 XCAP 요청 처리부

XCAP Client로부터의 요청들은 위 그림과 같은 형태로 메소드에 따라 분기되어 처리되어진다. XCAP 서버는 기본적으로 웹서버위에서 구동이 가능하며, 본 구현물은 리눅스 기반의 아파치 웹서버위에서 구축되어져 있다. 기존 웹서버를 이용함으로써 웹서버가 제공해주는 각종 환경 변수와 편의적인 함수들을 그대로 사용할 수 있는 장점이 있다. 그리고 PHP는 자체적인 XML 엔진등을 비롯한 많은 분야를 망라하는 함수들을 제공하고 있으므로 구현의 복잡도 완화 및 구현시간 단축을 이룰 수 있었다. 위 그림에서 보는 바와 같이 XCAP Request를 수신하게 되면 XCAP 서버는 우선 해당 요청으로부터 가용한 정보들을 추출해 낸다음 적절한 함수를 분기시켜주게 되며, 각 분기 함수들의 동작은 이 절에서 설명될 것이다.

가. GET 메소드 처리

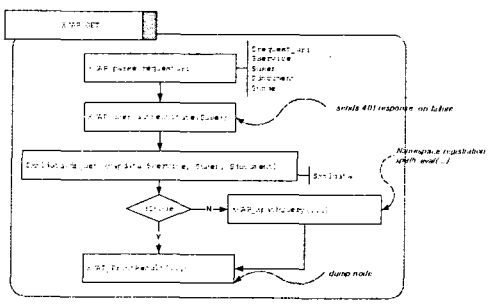

그림 5 GET method processing

HTTP GET을 통한 XCAP 문서의 인출 또는 문서의 부분 내용에 대한 인출 요구가 오면 다음과 같은 절차를 거쳐 처리된다.

① HTTP Request-URI의파싱을 통해 service, user, document, node 값을 분리해낸다.
② 인증여부를 확인한다. 인증을 통과하지 못하는 경우에는 401 응답을 전송하고 작업 종료한다.
③ 데이터베이스에서 저장되어 있는 XCAP 데이터를 인출한다.
④ node 값이 없으면 인출된데이터를 그대로 전송한다
⑤ node 값이 있으면 Xpath_Query를 통해 인출된 데이터 중 node가 지칭하는 부분을 전송한다.

나. PUT 메소드 처리

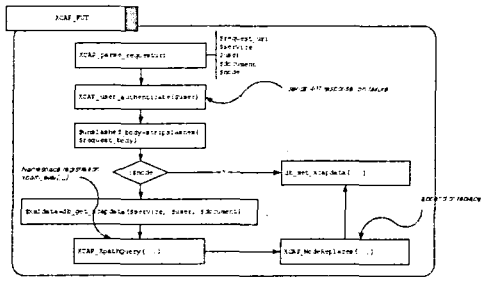

그림 6 PUT method processing

① HTTP Request-URI의 파싱을 통해 service, user, document, node 값을 분리해낸다.
② 인증여부를 확인한다. 인증을 통과하지 못하는 경우에는 401 응답을 전송하고 작업 종료한다.
③ body에 실려온 데이터에서 slash를 제거한다.
④ node의 값이 없으면 body에 실려온 XCAP 데이터를 그대로 데이터베이스에 저장한다. 만약 기존에 해당 문서와 연관된 데이터가 있더라도 새로운 데이터로 대체된다.
⑤ node의 값이 있으면 데이터베이스에서 저장되어 있는 XCAP 데이터를 인출하고 Xpath_Query를 통해 인출된 데이터 중 node가 지칭하는 부분을 갱신한다.

다. DELETE 메소드 처리

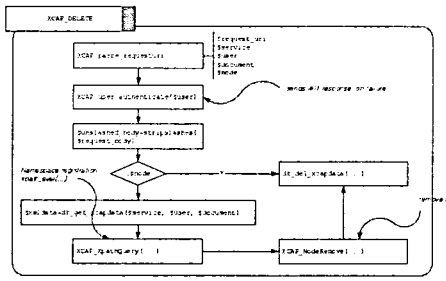

그림 7 DELETE method processing

① HTTP Request-URI의 파싱을 통해 service, user, document, node 값을 분리해낸다.
② 인증여부를 확인한다. 인증을 통과하지 못하는 경우에는 401 응답을 전송하고 작업 종료한다.
③ body에 실려온 데이터에서 slash를 제거한다.
④ node의 값이 없으면 저장되어 있는문서를 제거한다.
⑤ node의 값이 있으면 데이터베이스에서 저장되어 있는 XCAP 데이터를 인출하고 Xpath_Query를 통해 인출된 데이터 중 node가 지칭하는 부분을 제거한다.

IV. XCAP Usage "pres-rules" 응용

프레즌스 서비스를 위한 서비스 모델은 크게 두가지로 누뉠수 있다. 프레즌스 정보를 해당 사용자에게 직접 통지 받는 단말간 서비스 모델이 있을수 있으며, PA 서버로부터 통지받는 서버를 경유한 서비스 모델이 있을수 있다. 본장에서 기술하는 프레즌스 권한 정책을 위한 XCAP Usage는 단말간 서비스 모델에서는 의미가 없는 것이며, 서버를 경유한 프레즌스 서비스 모델에서 필요한 기능이다.

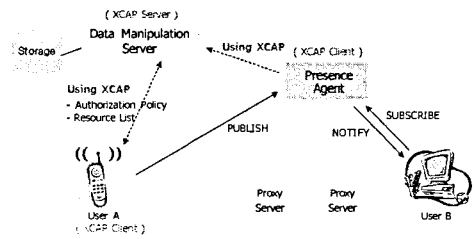

그림 8 XCAP pres-rules app-usage

PA 서버는 특정 사용자에 대한 SUBSCRIBE 메시지를 수신하게 되면, 해당 사용자의 프레즌스 정보를 항상 제공하는 것이 아니라, 요청한 와쳐가 허용된 사용자인지 먼저 체크를 하게 되며, 허용된 사용자에게만 요청한 프레즌스 정보를 제공한다. 이때, 각 사용자들에 대한 프레즌스 정보 권한 정책을 관리하기 위해 프레즌스 권한 정책을 위한 XCAP Usage 문서가 제안되어 표준화중에 있다.

각각의 사용자는 PA에 등록해둔 자신의 프레즌스 정보로의 접근 권한 정책을 XCAP 표준기술을 이용하여, XCAP 서버에 등록하게 된다. 이때, 프레즌스 권한 정책이라함은 어떤 사용자에게 어떤 프레즌스 정보를 언제 제공할것인지를 명시하는 정책이다. 이를 기술하기 위해, "urn:ietf:params:xml:ns:pres-rules"라는 별도의 XML 스키마가 정의되어 있으며, XML 문서는 n개의 <rule>을 명시한다. 각각의 <rule>은 3가지 파트로 다시 나뉘게 되며, 어떤 사용자에게 이 룰을 적용할것인지<conditions> 이라는 노드를 각각의 사용자를 기술한다. 또한 해당 사용자들로부터의 SUBSCRIBE 메시지를 허용할것이지 거절할것인지, 어떻게 처리할것인지 <actions>이라는 노드를 통해 기술하며, 허용된 경우, 해당 사용자에게 어떤 프레즌스 정보를 제공할것인지 필터링 정보를 기술하기 위해 <transformations> 노드를 사용한다. [2]

V. 결 론

본 고에서는 XCAP 서버의 기능을 구현하기 위한 설계 및 실제 구현에 사용된 모델을 설명하였다. XCAP 프로토콜은 SIP기반 프레즌스 시스템을 비롯한 다양한 분양에 응용되어 네트웍에 로직을 제공하는데 유용한 프로토콜이다. 현재 IETF에서는 "pres-rules"를 비롯하여 "resource-list"등 다양한 application-usage에 대한 표준화를 진행중에 있다.

참고문헌

[1] IETF, draft-ietf-simple-xcap-07.txt, "The Extensible Markup Language (XML) Configuration Access Protocol"
[2] IMPP 응용을 위한 XCAP 표준기술 동향, 주간기술동향 02-07, 박선옥