

	(19) 대한민국특허청(KR) (12) 공개특허공보(A)	(11) 공개번호 10-2012-0139955 (43) 공개일자 2012년12월28일
(51) 국제특허분류(Int. Cl.) G06F 17/00 (2006.01)		(71) 출원인 인하대학교 산학협력단 인천광역시 남구 인하로 100, 인하대학교 (용현동)
(21) 출원번호 10-2011-0059516		(72) 발명자 최원익 인천광역시 연수구 청능대로 175, 우성아파트 11 6동 1802호 (연수동)
(22) 출원일자 2011년06월20일 심사청구일자 2011년06월20일		장병국 인천광역시 남구 재념이길 143, 205동 302호 (용현동, 문화)
		김민철 인천광역시 남구 용현5동 우성아파트101동 811호
		(74) 대리인 특허법인무한

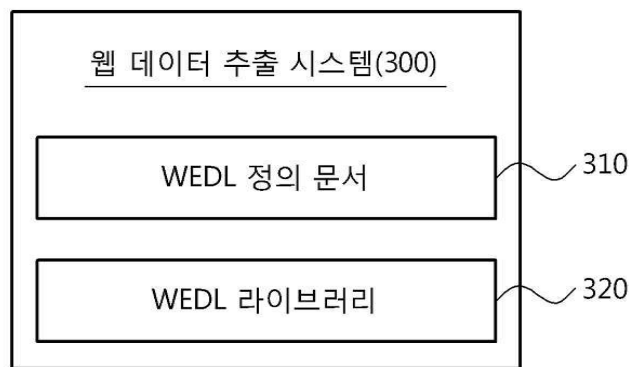
전체 청구항 수 : 총 6 항

(54) 발명의 명칭 모바일 환경에서의 웹 데이터 추출을 위한 시스템

(57) 요약

모바일 환경에서의 웹 데이터 추출을 위한 시스템이 개시된다. 웹 데이터 추출 시스템은 웹 서버에 접근하는 방법과 웹 데이터를 추출하는 방법이 WEDL(Web Extraction Definition Language) 문서로 기술되고, WEDL 문서를 기반으로 웹 서버에 직접 접속한 후 웹 서버로부터 수신된 웹 페이지로부터 데이터를 추출하는 WEDL 라이브러리를 포함할 수 있다.

대표도 - 도3



특허청구의 범위

청구항 1

모바일 환경에서의 웹 데이터 추출을 위한 시스템에 있어서,

웹 서버에 접근하는 방법과 웹 데이터를 추출하는 방법이 WEDL(Web Extraction Definition Language) 문서로 기술되고,

상기 WEDL 문서를 기반으로 상기 웹 서버에 직접 접속한 후 상기 웹 서버로부터 수신된 웹 페이지로부터 데이터를 추출하는 WEDL 라이브러리를

를 포함하는 웹 데이터 추출 시스템.

청구항 2

제1항에 있어서,

상기 WEDL 문서는,

상기 웹 페이지를 요청 및 추출하는 과정을 순서대로 정의하는 요소인 PROCEDURE element로 구성되는 것

을 특징으로 하는 웹 데이터 추출 시스템.

청구항 3

제2항에 있어서,

상기 PROCEDURE element는,

상기 웹 페이지를 요청하고 응답 결과를 수신하는 요소인 REQUEST element, 및 상기 응답 결과로 수신된 웹 페이지에서 웹 데이터를 추출하는 요소인 EXTRACT element로 구성되는 것

을 특징으로 하는 웹 데이터 추출 시스템.

청구항 4

제3항에 있어서,

상기 REQUEST element에는 요청 페이지 URL, 요청 방법(GET 또는 POST), 기본 요청 폼(Baseform), 요청 필드 값이 포함되고,

상기 EXTRACT element에는 상기 웹 페이지 상에 존재하는 테이블 형태의 데이터를 추출하는 요소인 TABLEROWS element, 상기 웹 페이지 상에 존재하는 일반 문자열 형태의 데이터를 추출하는 요소인 TEXT element, 상기 웹 페이지 상에 존재하는 <FORM> 태그를 추출하는 요소인 FORM element가 포함되는 것

을 특징으로 하는 웹 데이터 추출 시스템.

청구항 5

제3항에 있어서,

상기 PROCEDURE element는,

상기 추출된 웹 데이터에 대하여 유효성을 검사하는 요소인 VALIDATE element, 상기 추출된 웹 데이터의 변환을 처리하는 TRANSFORM element 또는 FILTERS element, 사용자의 입력 또는 선택을 입력 받는 요소인 VIEW element 중 적어도 하나의 요소를 더 포함하여 구성되는 것

을 특징으로 하는 웹 데이터 추출 시스템.

청구항 6

제1항에 있어서,

상기 WEDL 라이브러리는,

상기 웹 페이지에 대한 요청 및 추출 작업을 담당하는 클래스인 WebClient 클래스로 구성되고,

상기 WebClient 클래스는 상기 WEDL 문서를 각각의 요소 별로 객체화 하여 저장하는 WEDL 문서 객체(WEDL Document Object), 상기 웹 서버로부터 수신된 HTML 문서를 텍스트 형태로 저장하는 수신 문서 데이터(Received Document Data), 상기 HTML 문서의 전처리 과정을 거쳐 DOM 모델로 변환된 문서를 저장하는 수신 문서 객체(Received Document Object), 실행 중인 프로시저(procedure) 정보를 저장하는 프로시저 정보 테이블(Procedure Context Table), 상기 웹 페이지 요청 시 필요한 필드 값과 추출 결과를 저장하는 가변 테이블(Variable Table)로 이루어지는 것

을 특징으로 하는 웹 데이터 추출 시스템.

명세서

기술분야

[0001] 본 발명의 실시예들은 일반 웹 브라우저를 대상으로 제작된 웹 페이지로부터 사용자가 원하는 정보를 추출하여 모바일 환경에서 활용할 수 있는 방법에 관한 것이다.

배경기술

[0002] 스마트폰 및 tablet PC 등의 모바일 기기가 활성화됨에 따라 많은 웹사이트에서 모바일 전용 페이지를 별도로 제작하고 있다. 웹사이트에서 모바일 전용 페이지를 제공할 경우 모바일 기기에 최적화 된 사용자 인터페이스를 통하여 사용자는 편리하게 서비스를 이용할 수 있다. 그러나 모바일 페이지를 제공하지 않는 경우에는 작은 휴대전화 화면에서 일반 웹 페이지를 사용하는데 불편함이 따른다.

[0003] 웹 데이터 추출(Web Scrapping, Web Harvesting) 기술은 인터넷상에 웹 페이지로 존재하는 정보들을 대상으로 유용한 정보를 추출하는데 널리 이용되고 있다. 대표적인 예로 온라인 가격 비교, 날씨 데이터 추출, Web Content Mash-up 등이 있다.

[0004] 현재 많은 모바일 응용프로그램들이 이러한 기술들을 이용하여 인터넷상에 공개된 웹 페이지로부터 필요한 정보를 가져와 이를 모바일 환경에 맞게 표시해 주는 방식으로 동작하고 있다. 일반적으로 PC용으로 제작된 웹 페이지에 접속하여 HTML 페이지를 가져온 후 XML 분석기(XML parser) 또는 정규 표현식(regular expression)을 이용하여 필요한 정보를 추출하게 된다. 이를 구현하는 과정에서 프로그램의 소스코드에 데이터 추출을 위한 방법 및 절차가 구현 언어(Java, C 등)의 코드를 통해 직접적으로 기록되고 이 경우 웹 사이트 주소, 구조 또는 접근 절차 변경으로 인한 프로그램 유지 보수 시 소스 코드를 수정하고 재 컴파일 해야 한다는 단점이 있다. 또한 다양한 기기를 대상으로 개발이 필요할 경우 기기 별로 별도의 코드를 작성해야 하는 문제점이 있으며 유사한 종류의 정보를 각각 다른 형태로 제공하는 여러 사이트를 통합할 경우(예: 버스 배차정보, 쇼핑몰 가격 비교, 커뮤니티 웹사이트 등) 각각의 사이트 별로 별도의 코드를 작성해야 하는 문제점이 있다.

[0005] 기존 연구에서는 모바일 용 데이터 제공을 위해 별도의 프록시(proxy) 서버에서 웹 데이터 추출 과정을 거쳐 추출된 데이터를 데이터베이스에 저장하고 제공하는 방식이 제안되었다. 그러나 이러한 방법은 실시간 처리를 필요로 하는 데이터나 사용자의 개인 정보 등 보안상 민감한 데이터를 처리하는데 문제가 있다.

[0006] 도 1에 도시된 바와 같이, 프록시(proxy) 서버(110)를 이용한 방법의 경우 모바일 기기(120)는 프록시 서버(110)에 데이터를 요청한다. 이때, 프록시 서버(110)는 기존에 추출된 데이터가 존재할 경우 해당 데이터를 제공하고 그렇지 않을 경우 정보를 제공하는 웹 서버(130)로부터 웹 페이지를 받아 데이터 추출 작업을 거친 후 모바일 기기(120)에 데이터를 제공한다.

[0007] 이 방식은 요청 및 응답 데이터가 제3자의 서버를 통해 제공되기 때문에 안전하지 않을 수 있다. 또한, 프록시 서버에 저장된 정보를 제공할 수 있기 때문에 개인화된 데이터나 실시간 정보 데이터를 이용하는데 한계가 있다.

발명의 내용

해결하려는 과제

[0008] 모바일 기기 상에서 웹 데이터 추출을 직접 처리하는 방법을 제안하며 이를 위해 웹 페이지로부터 데이터를 추출하는 방법 및 절차를 기술하는 정의 언어(Web Extraction Definition Language, WEDL)를 설계한다.

[0009] 또한, 모바일 환경에서의 웹 데이터 추출을 위하여 설계된 WEDL 문서를 기반으로 동작하는 웹 클라이언트 라이브러리를 구현한다.

과제의 해결 수단

[0010] 모바일 환경에서의 웹 데이터 추출을 위한 시스템은 웹 서버에 접근하는 방법과 웹 데이터를 추출하는 방법이 WEDL(Web Extraction Definition Language) 문서로 기술되고, WEDL 문서를 기반으로 웹 서버에 직접 접속한 후 웹 서버로부터 수신된 웹 페이지로부터 데이터를 추출하는 WEDL 라이브러리를 포함할 수 있다.

[0011] 일측에 따르면, WEDL 문서는 웹 페이지를 요청 및 추출하는 과정을 순서대로 정의하는 요소인 PROCEDURE element로 구성될 수 있다.

[0012] 다른 측면에 따르면, PROCEDURE element는 웹 페이지를 요청하고 응답 결과를 수신하는 요소인 REQUEST element, 및 응답 결과로 수신된 웹 페이지에서 웹 데이터를 추출하는 요소인 EXTRACT element로 구성될 수 있다.

[0013] 또 다른 측면에 따르면, REQUEST element에는 요청 페이지 URL, 요청 방법(GET 또는 POST), 기본 요청 폼(Baseform), 요청 필드 값이 포함되고, EXTRACT element에는 상기 웹 페이지 상에 존재하는 테이블 형태의 데이터를 추출하는 요소인 TABLEROWS element, 상기 웹 페이지 상에 존재하는 일반 문자열 형태의 데이터를 추출하는 요소인 TEXT element, 상기 웹 페이지 상에 존재하는 <FORM> 태그를 추출하는 요소인 FORM element가 포함될 수 있다.

[0014] 또 다른 측면에 따르면, PROCEDURE element는 추출된 웹 데이터에 대하여 유효성을 검사하는 요소인 VALIDATE element, 추출된 웹 데이터의 변환을 처리하는 TRANSFORM element 또는 FILTERS element, 사용자의 입력 또는 선택을 입력 받는 요소인 VIEW element 중 적어도 하나의 요소를 더 포함하여 구성될 수 있다.

[0015] 또 다른 측면에 따르면, WEDL 라이브러리는 웹 페이지에 대한 요청 및 추출 작업을 담당하는 클래스인 WebClient 클래스로 구성될 수 있다. 이때, WebClient 클래스는 WEDL 문서를 각각의 요소 별로 객체화 하여 저장하는 WEDL 문서 객체(WEDL Document Object), 웹 서버로부터 수신된 HTML 문서를 텍스트 형태로 저장하는 수신 문서 데이터(Received Document Data), HTML 문서의 전처리 과정을 거쳐 DOM 모델로 변환된 문서를 저장하는 수신 문서 객체(Received Document Object), 실행 중인 프로시저(procedure) 정보를 저장하는 프로시저 정보 테이블(Procedure Context Table), 웹 페이지 요청 시 필요한 필드 값과 추출 결과를 저장하는 가변 테이블(Variable Table)로 이루어질 수 있다.

발명의 효과

[0016] 웹 페이지 요청, 추출 및 변환 절차를 정의하는 WEDL 문서 구조를 제안하고, 제안된 구조의 WEDL 문서를 처리하는 라이브러리를 구현함으로써 일반 웹 브라우저를 대상으로 제작된 웹 페이지로부터 사용자가 원하는 정보를 추출하여 모바일 환경에서 활용할 수 있다.

[0017] 웹 페이지 중 필요한 데이터를 추출하기 위한 요청 및 추출 절차를 WEDL 문서로 정의하고 정의된 절차에 따라 웹 데이터 추출을 수행함으로써 애플리케이션 코드와 처리 과정 정의가 분리될 수 있고 이로 인하여 애플리케이션 유지 보수가 보다 용이하다.

도면의 간단한 설명

[0018] 도 1은 프록시 서버를 이용하여 웹 데이터를 추출하는 방법을 설명하기 위한 도면이다.

도 2는 본 발명의 일실시예에 있어서, WEDL를 이용하여 모바일 기기 상에서 웹 데이터 추출을 직접 처리하는 프로세스를 설명하기 위한 도면이다.

도 3은 본 발명의 일실시예에 있어서, WEDL 기반의 웹 데이터 추출 시스템을 설명하기 위한 도면이다.

도 4는 본 발명의 일실시예에 있어서, WEDL 문서 구조의 일례를 도시한 도면이다.

도 5는 본 발명의 일실시예에 있어서, WEDL 문서에서 사용 가능한 요소 중 REQUEST element 구조를 도시한 도면

이다.

도 6은 본 발명의 일실시예에 있어서, WEDL 문서에서 사용 가능한 요소 중 EXTRACT element 구조를 도시한 도면이다.

도 7은 본 발명의 일실시예에 있어서, WEDL 문서에서 사용 가능한 요소 중 VALIDATE element 구조를 도시한 도면이다.

도 8은 본 발명의 일실시예에 있어서, WEDL 문서에서 사용 가능한 요소 중 TRANSFORM element 구조를 도시한 도면이다.

도 9는 본 발명의 일실시예에 있어서, WEDL 문서에서 사용 가능한 요소 중 FILTERS element 구조를 도시한 도면이다.

도 10은 본 발명의 일실시예에 있어서, WEDL 문서에서 사용 가능한 요소 중 VIEW element 구조를 도시한 도면이다.

도 11은 본 발명의 일실시예에 있어서, WEDL 문서를 처리하는 WEDL 라이브러리 구조의 일례를 도시한 도면이다.

도 12는 본 발명의 일실시예에 있어서, WEDL 실행 과정의 일례를 도시한 순서도이다.

발명을 실시하기 위한 구체적인 내용

- [0019] 이하, 본 발명의 실시예를 첨부된 도면을 참조하여 상세하게 설명한다.
- [0020] Web Data Extraction(Web Data Scrapping, Web Harvesting) 기술은 인터넷 상에 존재하는 웹 페이지로부터 필요한 정보를 추출하는 기술을 말한다. 일반적으로 웹 페이지는 HTML 태그를 이용하여 구성되어 있고 실제 정보 외에 디자인을 위한 의미 없는 요소가 존재한다. 이러한 웹 페이지 내부에서 의미 있는 데이터만을 추출하여 사용자에게 제공하기 위해 여러 기술들이 사용된다.
- [0021] 그 중 하나인 WIDL(Web Interface Definition Language)은 1997년 webMethods, Inc.에 의해 W3C에 제안되었다. 이는 문서 기반의 웹 페이지를 대상으로 서비스 기반 구조를 구현하기 위한 메타 언어(meta language)이다. XML 형태의 언어이며 웹 페이지에서 제공하는 서비스에 대한 요청 형태와 응답 형태를 정의하여 응용프로그램간 데이터를 상호 교환할 수 있도록 한다.
- [0022] 그리고, WIDL은 입력 바인딩(input binding)과 출력 바인딩(output binding)을 정의하여 서비스 호출 방법 및 데이터 추출 방법을 기술하고 있다. 추출할 데이터를 정의하는 방법으로 DOM(Document Object Model) 및 정규 표현식(regular expression)이 사용된다. DOM을 이용하여 정의하는 경우 해당 문서의 최상위 element부터 명시해야 하며 복잡한 문서의 처리가 어렵다는 단점이 있다. 또한 추출된 데이터를 일관된 형태로 변환하는 기능을 제공하지 않는다.
- [0023] WIDL에서 제공하는 Service Chain을 통하여 여러 단계의 웹 요청 처리가 가능하며 Condition을 통하여 웹 페이지 추출의 성공 및 실패 여부를 확인할 수 있다.
- [0024] 이후, 1998년 웹 문서 처리를 위한 프로그래밍 언어인 WebL이 제안되었다. WebL은 객체지향 스크립트 언어(object-oriented scripting language)를 통하여 Service Combinator 및 Markup Algebra를 제공한다. Service Combinator는 웹 요청 방식을 나타내는 언어이며 Markup Algebra를 통해 구조적인 데이터와 비 구조적인 데이터를 추출한다. 검색엔진에 사용되는 웹 크롤러(web crawler), 메타 검색 엔진(meta search engine), 페이지 변환기(page transducer), 쇼핑 로봇(shopping robot)등을 빠르게 개발하는 용도로 사용되도록 제안되었다.
- [0025] 이외에도, 모바일 사용자를 위한 웹서비스 페이지 개인화 기법, 위젯 서비스를 위한 오픈 API 프록시 서버 시스템 등이 제안되었다. 제안된 두 가지 방법은 데이터 처리 및 저장을 위한 별도의 프록시 서버를 구성하여 처리함으로써 처리 속도가 빠른 장점이 있으나 프록시 서버에 추출된 데이터를 저장하여 제공함으로써 실시간 정보 처리 및 신용 카드 번호, 개인 정보 등의 처리에 부적합한 단점이 있다.
- [0026] 또한, 모바일 환경에서의 웹 추출 관련 연구로는 이동 단말을 위한 적응적 웹 문서 변환, 모바일 환경을 위한 웹 콘텐츠 추출기법 설계 및 구현 등이 있다.

- [0027] 본 실시예에서는 기존에 제안된 방식의 문제를 해결하기 위해 웹 페이지로부터 데이터를 추출하는 절차 및 방법을 기술한 WEDL(Web Extraction Definition Language)을 제안하고, 제안된 정의 문서를 기반으로 모바일 기기 상에서 직접 동작하는 웹 클라이언트 라이브러리를 구현할 수 있다.
- [0028] 도 2는 본 발명의 일실시예에 있어서, WEDL를 이용하여 모바일 기기 상에서 웹 데이터 추출을 직접 처리하는 프로세스를 설명하기 위한 도면이다.
- [0029] 도 2에 도시된 바와 같이, WEDL를 이용하는 방법의 경우 정보 요청은 모바일 기기(220)와 실제 정보를 제공하는 웹 서버(230) 사이에서 통신이 이루어진다. 모바일 기기(220)는 WEDL 문서에 명시된 웹 서버 주소와 접근 방법 및 절차를 통하여 웹 서버(230)에 직접 접속하고 수신된 웹 페이지로부터 데이터를 모바일 기기(220) 내에서 자체적으로 추출하여 사용자에게 정보를 제공한다. 이때, 웹 서버(230)에서 SSL(Secure Socket Layer) 보안 연결을 제공할 경우 더욱 안전하게 데이터를 주고 받을 수 있다.
- [0030] 이하에서는, 웹사이트 접속 방법 및 데이터 추출 방법을 기술하는 언어인 WEDL에 대하여 설명한다. 도 3은 본 발명의 일실시예에 있어서, 웹 데이터 추출 시스템(200)을 도시된 것이다.
- [0031] 도 3에 도시된 바와 같이, 웹 데이터 추출 시스템(300)은 크게 WEDL 정의 문서(310)와 WEDL 문서를 처리하는 WEDL 라이브러리(320)로 구성된다. WEDL 정의 문서(310)에는 웹사이트 주소, 요청 방법(GET 또는 POST), 요청 매개변수(parameter) 및 웹 페이지 추출 방법 등을 기술하며, WEDL 라이브러리(320)는 정의된 문서를 읽어 웹 요청을 수행하고 웹 데이터를 추출한다.
- [0032] WEDL 문서에서 사용 가능한 주요 element는 표 1과 같다.

표 1

element	설명
WEDL	WEDL 정의 문서의 최상위 element
CONFIG	웹 데이터 추출 과정에서 사용되는 설정 값을 정의하는 element
PROCEDURE	웹 데이터 추출 과정을 정의하는 element
REQUEST	웹 요청을 정의하는 element
EXTRACT	데이터 추출 방법을 정의하는 element
VALIDATE	추출된 데이터의 유효성을 검증하는 element
TRANSFORM	추출된 데이터의 형태를 변환하는 element
VIEW	사용자 상호 작용(user interaction)을 위한 View를 표시하는 element

[0033]

표 2

```

<?xml version="1.0" encoding="UTF-8"?>
<wedl version="1" id="kobus" description="kobus" parsing data file="
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:namespaceSchemaLocation="wedl.xsd">
  <config key="providerCode" type="string">KB</config>
  <config key="enableSelectCardType" type="boolean">false</config>
  <config key="maxReserveSeatCount" type="integer">45</config>
  <procedure id="schedule">
    <request method="post" action="http://www.kobus.co.kr/web/03_reservation/reservation01_1.jsp">
      <param name="TER_FR">${inDepartureCode}</param>
      <param name="TER_TO">${inDestinationCode}</param>
      <param name="Tim_date_Year" type="date" format="yyyy">${inDepartureDate}</param>
      <param name="Tim_date_Month" type="date" format="MM">${inDepartureDate}</param>
      <param name="Tim_date_Day" type="date" format="dd">${inDepartureDate}</param>
      <param name="TIM_TIM_I" type="date" format="HHmm">${inDepartureDate}</param>
      <param name="BUS_GRA_I">${inGrade}</param>
      <param name="pCnt_100">1</param>
      <param name="pCnt_050">0</param>
    </request>
    <extract id="extractScheduleTable" append="true">
      <tableRows>
        <xpath>//table//tr[count(td)=5][position()=1]</xpath>
        <column columnIndex="0" var="time"/>
        <column columnIndex="1" var="grade">
          <filters outvar="gradeCode">
            <mapping>
              <mapvalue>
                <regex group="0">^일반고속$</regex>
                <replacement>0</replacement>
              </mapvalue>
              <mapvalue>
                <regex group="0">^부들고속$</regex>
                <replacement>1</replacement>
              </mapvalue>
            </mapping>
          </filters>
        </column>
        <column columnIndex="2" var="company"/>
        <column columnIndex="3" var="remainSeat"/>
      </tableRows>
      <form var="reserveForm">
        <xpath>//form[contains(@name, 'SeatForm')</xpath>
      </form>
    </extract>
    <extract>
      <form var="nextPageForm">
        <xpath>//form[@name="PageForm"]</xpath>
      </form>
    </extract>
    <extract>
      <tableRows>
        <xpath>//table//tr[count(td)=4]/td[contains(., '구분')]/../tr</xpath>
        <column columnIndex="1" rowIndex="2" var="normalFare"/>
        <column columnIndex="2" rowIndex="2" var="excelFare"/>
        <column columnIndex="3" rowIndex="2" var="nightExcelFare"/>
        <column columnIndex="1" rowIndex="3" var="childFare"/>
        <column columnIndex="1" rowIndex="4" var="takeTime"/>
        <column columnIndex="1" rowIndex="5" var="distance"/>
      </tableRows>
    </extract>
  </procedure>
</wedl>

```

[0034]

[0035] WEDL 문서 구조

[0036] 도 4는 WEDL 문서 구조를 도시된 것이다.

[0037] 도 4에 도시된 바와 같이, WEDL 문서는 WEDL element를 최상위 element로 가지며 하위 element로 CONFIG element와 PROCEDURE element를 가진다. CONFIG element는 웹 데이터 추출 과정에서 필요한 설정 값을 정의할 수 있으며 생략이 가능하다. PROCEDURE element에는 웹 페이지 요청 및 추출 과정을 순서대로 정의하며 반드시 1개 이상 존재해야 한다. PROCEDURE element는 ID 속성으로 구분되며 라이브러리를 이용하여 ID 값으로 해당 procedure를 실행할 수 있다.

- [0038] WEDL 라이브러리에는 웹 페이지 요청 과정에서 필요한 데이터 웹 추출 과정을 통해 추출된 데이터가 저장되는 테이블이 존재한다. procedure를 호출 하기 전 라이브러리에서 제공하는 setValue 메서드를 통해 필요한 입력 필드 값을 설정하고 웹 데이터 추출 과정이 완료되면 getValue 메서드를 통해 추출된 데이터를 사용한다. 또한 중간 과정에서 필요한 변수들은 \${var_name[0]} 형태로 PROCEDURE 하위 element에서 사용할 수 있다.
- [0039] 표 2는 상기한 구조의 WEDL 예시를 나타낸 것이다.
- [0040] 웹 페이지 요청
- [0041] 웹 페이지 요청 및 응답 결과 수신은 REQUEST element를 통해 처리한다. 도 5에 도시된 바와 같이, REQUEST element에는 요청 페이지 URL, 요청 방법(GET 또는 POST), 기본 요청 폼(Baseform) 및 요청 필드 값을 포함할 수 있다. Baseform은 기존에 추출된 HTML form의 action URL, 요청 방법(method) 및 필드 값들을 그대로 상속 받으면서 특정 필드의 값만 교체하여 요청할 경우 사용한다. 이는 여러 페이지로 분리된 form들의 연속적인 요청을 처리할 때 유용하게 사용될 수 있다. REQUEST의 유일한 하위 element인 PARAM element를 통해 요청 시 전송할 필드 값을 정의할 수 있다. PARAM element의 type 및 format 속성을 사용하여 날짜 등의 간단한 formatting도 가능하다.
- [0042] 웹 데이터 추출
- [0043] 웹 페이지 요청 후 요청한 데이터(HTML 페이지)가 정상적으로 수신될 경우 웹 데이터 추출 과정을 진행할 수 있다. HTML 페이지는 전처리 과정을 통해 well-formed XML 문서로 변환한 후 XML parser로 처리하거나 XML parser에서 XML 문법 오류 정정 기능을 지원할 경우 전처리 과정 없이 처리가 가능하다. XML parser에서는 XPath expression을 통한 페이지 탐색 기능을 제공하므로 XPath expression을 정의함으로써 데이터의 위치를 명시할 수 있다. XPath를 이용할 경우 DOM(Document Object Model)과 같이 전체 경로를 지정할 필요 없이 중간 element부터 지정이 가능한 장점이 있다. (예: //form[@id='form_id']) DOM으로 element를 지정한 경우 웹 사이트의 레이아웃이 조금만 변경되어도 전체 경로가 달라질 수 있는 반면 XPath로 중간 element부터 지정할 경우 추출할 데이터를 제외한 레이아웃이 변경되어도 동일한 XPath 표현식을 사용할 수 있다. 또한 XML 형태의 문서가 아닌 XPath로 추출이 불가능한 Java Script 또는 일반 텍스트의 추출을 위해 정규 표현식(regular expression)을 사용할 수 있다.
- [0044] 웹 데이터 추출은 EXTRACT element를 통해 이루어 진다. 도 6에 도시된 바와 같이, EXTRACT element의 하위 element로는 TABLEROWS, TEXT, FORM 등의 element를 가진다. TABLEROWS element 및 FORM element는 XPath를 통해 해당 추출할 데이터의 위치를 지정할 수 있고 TEXT는 XPath 표현식 또는 정규 표현식(regular expression)을 통해 위치 지정이 가능하다.
- [0045] 이때, TABLEROWS element는 웹 페이지상에 존재하는 테이블 형태의 데이터를 추출할 때 사용된다. HTML 페이지상의 테이블은 <TABLE> 태그와 <TR>, <TD> 등의 태그로 구성되어 있으며 <TR> 태그의 XPath 표현식과 열 인덱스를 지정함으로써 테이블의 특정 열을 모두 추출할 수 있다. 또한 행 인덱스를 추가로 지정하여 특정 행과 열에 존재하는 데이터를 추출 가능하다.
- [0046] 그리고, TEXT element는 웹 페이지상에 존재하는 일반 문자열 형태의 데이터를 추출한다. 데이터의 위치를 XPath 표현식으로 정의하거나 정규표현식(regular expression)을 사용하여 지정할 수 있다.
- [0047] 또한, FORM element는 웹 페이지상에 존재하는 <FORM> 태그를 추출할 때 사용된다. 추출된 FORM 데이터는 추후에 연속적인 요청이 필요할 경우 REQUEST element의 Baseform 속성을 통해 요청될 수 있다.
- [0048] TABLEROWS과 TEXT의 하위 element에 FILTERS element를 추가하여 추출되는 데이터를 변환하는 기능을 수행할 수 있다. FILTERS element는 크게 mapping 작업과 formatting 작업을 지원하며 MAPPING element 및 FORMAT element를 정의하여 처리한다. MAPPING은 특정 패턴의 텍스트를 치환하는 역할을 수행하며 FORMAT은 날짜 등과 같이 일정한 포맷을 가지는 텍스트를 날짜 객체로 변환이 가능하다. FILTERS element는 데이터 변환(Transform element)에서도 사용된다
- [0049] 데이터 유효성 검사
- [0050] EXTRACT element를 사용하여 웹 데이터 추출 작업을 진행한 후 데이터가 올바르게 추출되었는지 확인하기 위하여 VALIDATE element를 정의하여 유효성을 검사할 수 있다. 유효성 검사 과정에서 지원하는 형태로는 크게 존재 유무와 특정 값의 비교 연산이 수행될 수 있으며 조건을 충족하지 않을 경우 지정한 메시지의 예외가 발생된다. 도 7을 참조하면, VALIDATE 하위 element인 EXISTS element를 사용하여 특정 변수의 존재 유무를 확인할

수 있고 COMPARE element를 사용하여 두 값의 일치 여부 또는 숫자 형태의 값일 경우 대소 비교가 가능하다.

[0051] 데이터 변환

[0052] 추출된 데이터의 변환은 도 8에 도시된 TRANSFORM element를 통하여 데이터 추출 작업 후 별도로 처리할 수 있다. 데이터 변환은 데이터 추출 과정에서 도 9에 도시한 바와 같이 FILTERS element를 정의하여 변환을 처리할 수 있으나 복잡한 처리 과정의 경우 도 8의 TRANSFORM element를 통해 별도 처리할 수 있다. 변환 과정에서 사용되는 필터는 여러 개의 필터를 사용할 수 있으며 정의된 순서에 따라 순차적으로 데이터 변환을 수행하게 된다.

[0053] User Interaction

[0054] 프로시저(procedure) 수행 과정 도중 사용자의 입력 또는 선택이 필요할 경우 도 10에 도시된 바와 같이 VIEW element를 정의하여 처리 중간에 사용자의 입력을 받은 후 계속해서 웹 데이터 추출 과정을 진행할 수 있다. VIEW element의 name 속성을 통하여 사용자 입력을 받는 view의 이름을 지정할 수 있으며 WEDL 라이브러리는 프로시저(procedure) 처리 도중 view가 필요할 경우 이벤트를 발생하여 사용자에게 적절한 view를 표시하고 사용자의 입력이 완료 되면 프로시저(procedure)의 다음 과정을 계속해서 진행하게 된다.

[0055] 본 실시예에서는 제안된 WEDL 문서에 따라 웹 페이지를 요청하고 추출하는 라이브러리를 개발하기 위하여 Java를 사용한다. Java 라이브러리로 개발할 경우 최근에 각광 받는 모바일 플랫폼인 Android 애플리케이션 개발이 쉽게 사용할 수 있는 장점이 있다.

[0056] HTML 문서는 태그로 구성된 마크업(mark-up) 언어로 구성되어 전처리 과정을 거쳐 XML 문서로 변환될 수 있고 이는 XML 분석기(parser)를 통해 분석될 수 있다. XHTML의 경우 문서 자체가 XML 문서이기 때문에 전처리 과정이 필요하지 않으나 다수의 웹 페이지들이 표준을 준수하지 않는 경우가 있으므로 전처리 과정은 필수적이다.

[0057] 본 실시예에서는 웹 페이지 추출 과정 전에 HTML 전처리를 위하여 HtmlCleaner 라이브러리를 사용하고 XML 문서 분석을 위하여 JDOM 및 Jaxen을 사용할 수 있다. 아이폰과 같은 C언어 기반의 platform의 경우 libxml2 등의 라이브러리를 통해 동일한 작업을 수행 가능하다.

[0058] 도 11은 WEDL 라이브러리 구조를 도시한 것이다.

[0059] 도 11에 도시된 바와 같이, 전체적인 웹 요청 및 추출 작업을 담당하는 클래스인 WebClient 클래스의 내부에는 WEDL Document Object, Received Document Data, Received Document Object, Procedure Context Table, Variable Table이 존재한다.

[0060] WEDL Document Object는 WEDL 문서를 각각의 element별로 객체화 하여 담고 있는 객체이다. Received Document Data는 웹 요청 후 서버로부터 수신한 HTML 문서를 텍스트 형태로 저장하고 전처리 과정을 거쳐 DOM 모델로 변환된 Document는 Received Document Object로 저장된다. 이 객체를 통하여 XPath 질의를 수행하고 웹 데이터를 추출할 때 사용된다. Procedure Context Table은 다수의 프로시저(procedure) 처리를 위해 실행 중인 프로시저(procedure) 정보를 저장하고 있는 테이블이다. Variable Table은 웹 요청 시 필요한 필드 값 및 추출 결과를 저장하는 테이블로써 변수명과 값으로 구성된 테이블 형태이다.

[0061] WEDL Document Object

[0062] WEDL Document Object는 WEDL 문서를 Java 객체 형태로 저장하고 있는 변수이다. XML 문서 구조와 동일한 구조로 되어있으며 프로시저(procedure) 수행 시 이 객체로부터 데이터를 읽어 순서대로 작업을 수행하게 된다. PROCEDURE element의 하위 element인 REQUEST, EXTRACT, VALIDATE, TRANSFORM, VIEW element는 Operation 클래스를 상속받아 각각 RequestOperation, ExtractOperation, ValidateOperation, TransformOperation, ViewOperation 클래스로 구현되어 있다. 이러한 Operation들은 WebOperationThread 클래스에서 별도의 스레드로 처리된다.

[0063] Variable Table

[0064] Variable Table에는 웹 페이지 요청 시 필요한 필드 값 및 추출 결과가 저장된다. 내부에 저장 가능한 데이터 형태로는 문자열, 날짜, HTML 폼 데이터 등이 있으며 이러한 데이터의 배열 형태로도 저장이 가능하다.

[0065] 저장된 데이터는 WEDL 문서에서 참조하여 사용이 가능하다. \${var_name} 형태로 단일 변수를 참조하며 \${var_name[0]} 형태로 배열의 특정 값을 참조할 수 있다. 폼 형태의 변수의 경우

`${form_name[0][field_name]}` 형태로 값을 참조한다. 이러한 형태의 변수는 REQUEST element의 action이나 PARAM element의 값을 정의할 때 사용한다.

[0066] Procedure Context

[0067] Procedure 처리 과정 중 다른 Procedure 처리를 위해 WebClient 내부에 Procedure Context Table을 유지한다. Procedure가 시작되면 해당 Procedure에 대한 새로운 Procedure Context를 생성하고 이를 Procedure Context Table에 추가한다. Procedure Context에는 현재 실행중인 Procedure 및 Operation의 인덱스가 저장되어 하나의 Procedure 실행 중 다른 Procedure 실행이 가능하다.

[0068] WebClient Event

[0069] 프로시저(procedure) 실행 과정 중 수행되는 모든 Operation들은 표 3과 같이 별도의 스레드로 처리된다. 스레드 실행 중 프로시저(procedure) 및 Operation의 시작, 종료, 실패 시 지정한 핸들러를 통해 이벤트를 처리할 수 있다. 또한 프로시저(procedure) 실행 과정 중 VIEW element를 만났을 경우 view를 표시하기 위한 이벤트가 발생하며 해당 이벤트에 대한 인터페이스를 구현함으로써 view를 표시하는 등의 구현이 가능하다.

표 3

메서드	설명
onProcedureStart	Procedure가 시작될 때 실행되는 메서드
onOperationStart	Operation이 시작될 때 실행되는 메서드
onOperationFinish	Operation이 종료될 때 실행되는 메서드
onProcedureFinish	Procedure가 종료될 때 실행되는 메서드
onProcedureFail	Procedure 실행 중 오류가 발생하였을 때 실행되는 메서드. (VALIDATE 수행 도중 유효성 검사에 실패한 경우 포함)
onProcedureNeedTo DisplayView	Procedure 수행 과정 중 VIEW element를 만났을 경우

[0070]

[0071] 실행 과정

[0072] 도 12는 WEDL 실행 과정의 일례를 도시한 순서도이다.

[0073] WEDL Document Object를 WebClient 클래스에 설정한 후 runProcedure 메서드를 통해 프로시저(procedure)를 시작한다. 프로시저(procedure) 또는 Operation이 시작 및 종료 될 경우 해당 이벤트가 발생되고 Operation들은 비동기적으로 별도의 스레드를 통해 수행된다. 수행 과정 중 사용자 상호 작용(User Interaction)이 필요할 경우 view를 표시하여 사용자로부터 입력을 받고 입력 받은 값은 setValue 메서드를 통하여 WEDL Library의 Variable Table에 저장되게 된다. 이후 doNextOperation 메서드를 통해 프로시저(Procedure)의 남은 Operation을 수행한다.

[0074] 따라서, 본 발명에서는 웹 페이지 요청, 추출 및 변환 절차를 정의하는 WEDL 문서 구조를 제안하고, 제안된 구조의 WEDL 문서를 처리하는 라이브러리를 구현한다. 라이브러리의 동작을 검증하기 위하여 실제 웹사이트를 대상으로 WEDL 문서를 설계하고 실행을 확인하였다. 기존에는 웹 페이지 중 필요한 데이터를 추출하기 위해 요청 및 추출 절차를 복잡한 코드를 통해 구현함으로써 애플리케이션의 유지 보수가 힘들어지는 단점이 있다. 이 부분을 WEDL 문서로 정의하고 정의된 절차에 따라 웹 데이터 추출을 수행함으로써 애플리케이션 코드와 처리

과정 정의가 분리될 수 있고 이를 통해 애플리케이션 유지 보수를 보다 쉽게 할 수 있을 것으로 기대된다.

[0075] 본 발명에서는 안드로이드 플랫폼을 대상으로만 구현하였으나 추가로 아이폰 등의 다른 플랫폼을 대상으로 라이브러리를 구현할 경우 동일한 WEDL 문서를 통해 cross-platform 환경에서 동작하는 애플리케이션을 구현할 수 있음은 물론이다. 또한 WEDL 정의를 확장하여 모바일 환경이 아닌 일반 서버 환경에서 동작하는 라이브러리를 구현할 경우 웹 상에 존재하는 유용한 데이터를 서버에 데이터베이스 형태로 저장하는 것도 가능하다.

[0076] 이상과 같이 본 발명은 비록 한정된 실시예와 도면에 의해 설명되었으나, 본 발명은 상기의 실시예에 한정되는 것은 아니며, 본 발명이 속하는 분야에서 통상의 지식을 가진 자라면 이러한 기재로부터 다양한 수정 및 변형이 가능하다.

[0077] 그러므로, 본 발명의 범위는 설명된 실시예에 국한되어 정해져서는 아니 되며, 후술하는 특허청구범위뿐 아니라 이 특허청구범위와 균등한 것들에 의해 정해져야 한다.

부호의 설명

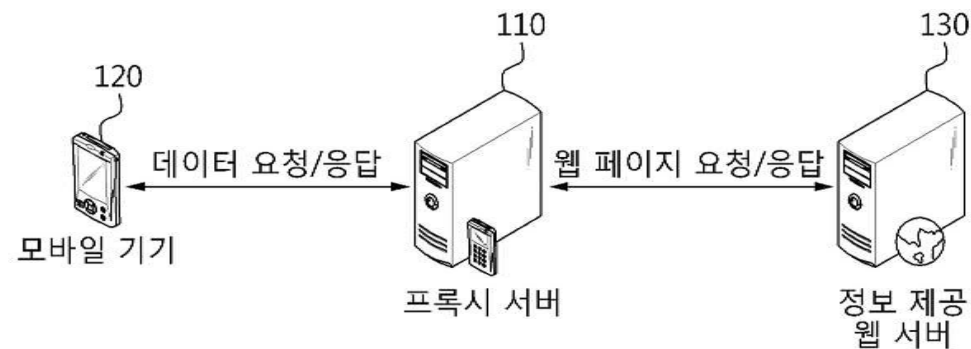
[0078] 300: 웹 데이터 추출 시스템

310: WEDL 정의 문서

320: WEDL 라이브러리

도면

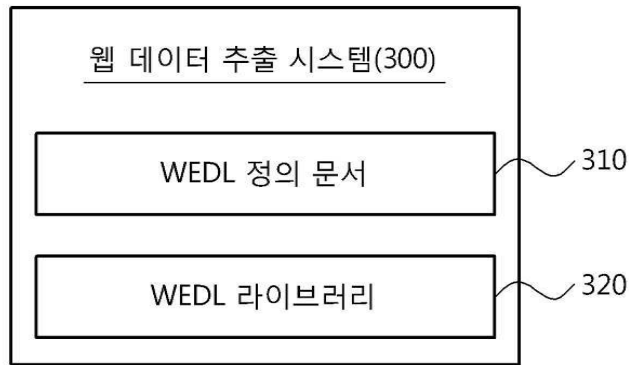
도면1



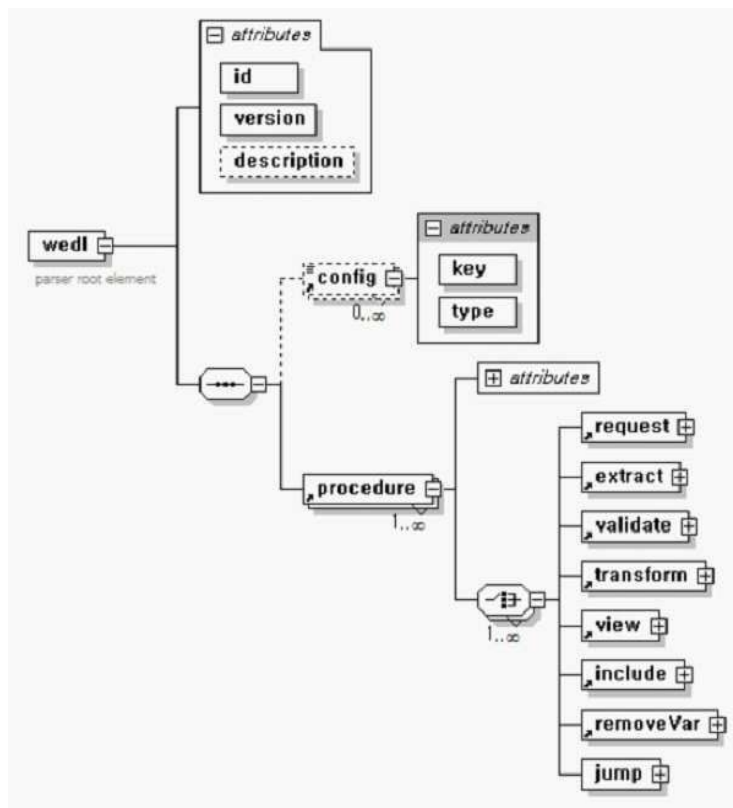
도면2



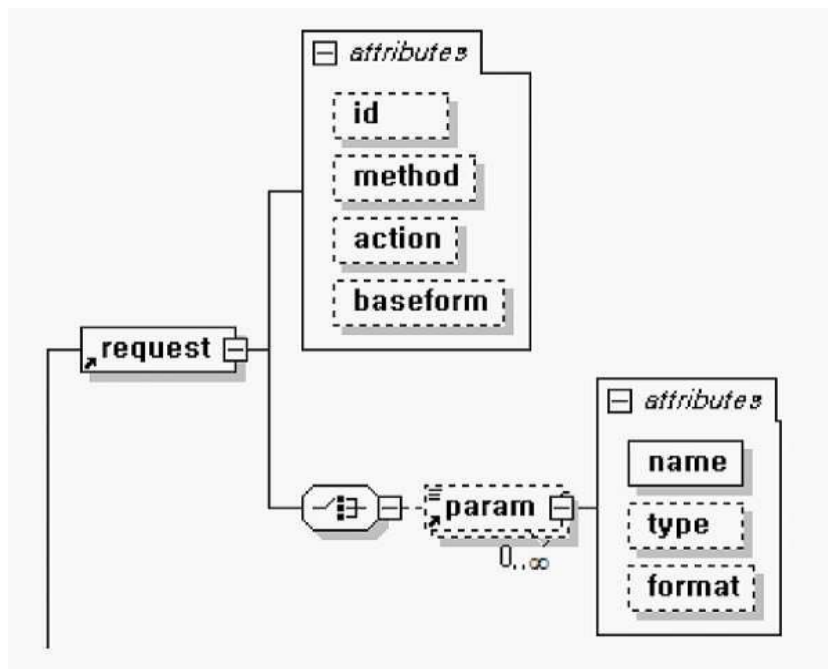
도면3



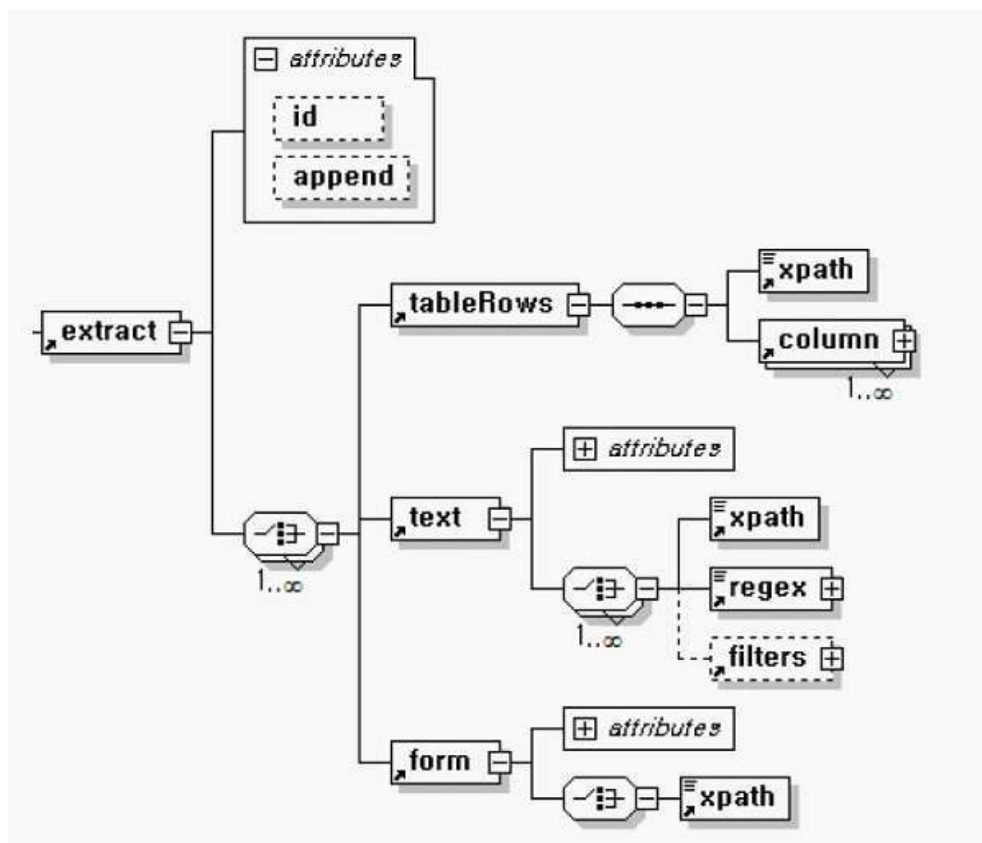
도면4



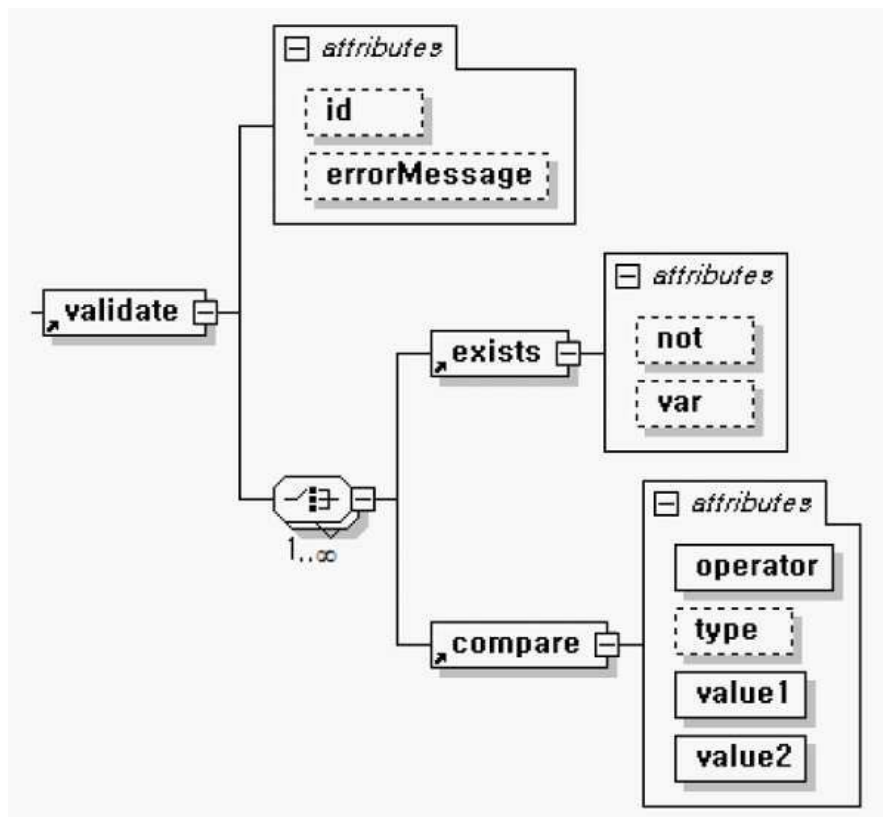
도면5



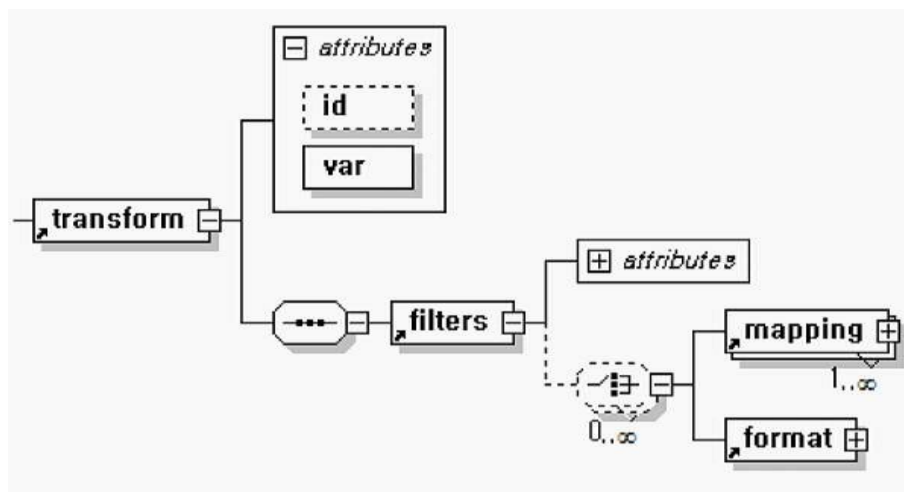
도면6



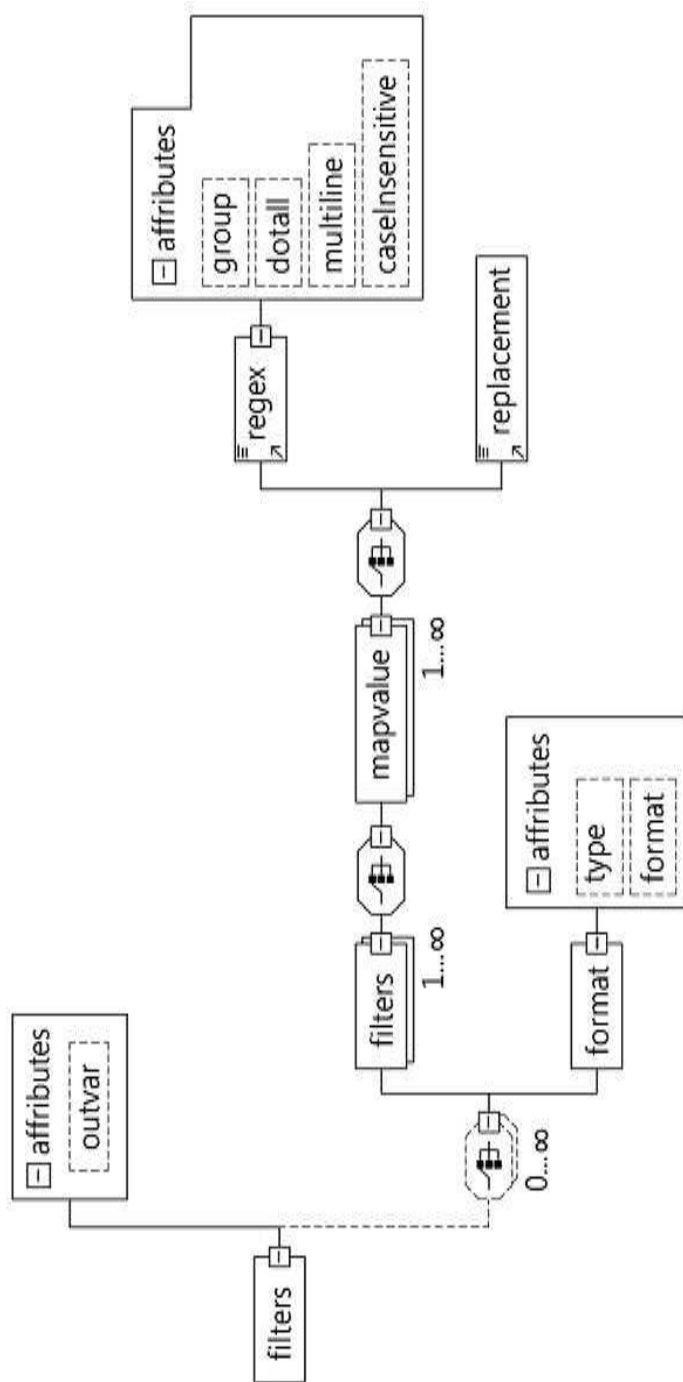
도면7



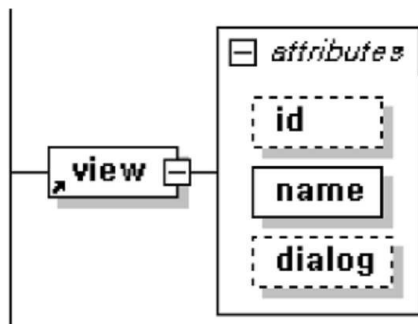
도면8



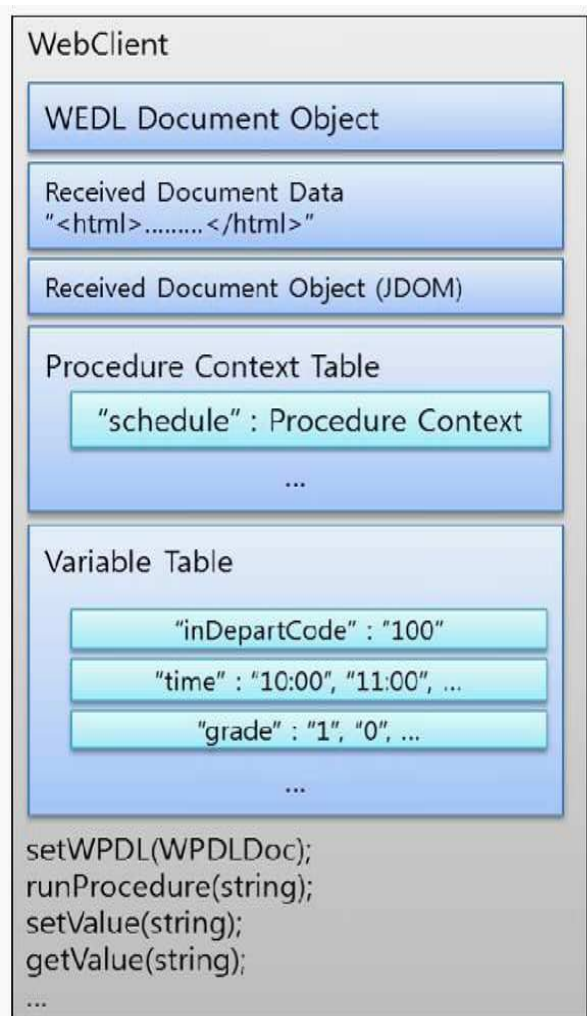
도면9



도면10



도면11



도면12

