

XML 프로토콜

충남대학교 이경하* · 이규철**

1. 서론

1990년 WWW이 소개된 이후에, WWW은 인터넷의 보급과 사용에 많은 영향을 미쳐왔다. 원래의 개발 목적과 같이 현재 WWW의 주된 사용 내용은 네트워크 상에 산재해 있는 문서 또는 멀티미디어 데이터에 대한 접근성을 보장하는 것이었다. 그러나 웹의 폭발적인 성장으로 인하여 그 규모가 커짐에 따라 많은 어플리케이션들이 웹과 결합해 가고 있으며, 어플리케이션 간 데이터 교환에서도 웹을 적용하기 시작하였다. 이렇게 분산 환경 하에서 어플리케이션간 통신 수단으로써 웹을 이용하고자 하는 흐름은 XML[1]이 출현한 이후에 보다 가속화되고 있다.

W3C가 1998년 권고안(Recommendation)으로 채택한 이래 XML은 “새 시대의 ASCII”라고 일컬어지는 바와 같이 그 개방성과 상호운용성으로 인하여 널리 보급되어 사용되어지고 있다. 단순히 사용자에게 보여질 문서를 제작하는데 사용되는 것뿐만 아니라, 어플리케이션간 데이터 교환에서의 포맷으로도 그 활용의 폭이 넓어지고 있다. 이러한 XML의 사용에 있어서 문제점은 XML 그 자체에 대한 표준화 및 개발은 이미 종료되었으나, XML을 이용한 데이터 전달 형식 및 매체는 결정되지 않았다는 것이다[2,3]. OASIS에서 배포한 “XML Conformance Test suit”[4]를 통하여 XML 프로세서가 지원하는 XML 표준에 대한 상호운용성은 보장할 수 있었던 반면에, 어플리케이션간 교환에 사용되어지는

XML 응용에 대한 표준은 정해져 있지 않아 어플리케이션에서 사용하는 XML 응용끼리의 상호운용성은 보장할 수가 없다. 현재까지 나와 있는 XML 응용으로는 Coins, WDDX, XMOP, SOAP, XML-RPC 등과 같이 서로 호환되지 않는 XML 응용들이 있어 왔으며, 이에 따라 W3C는 2000년 2월에 어플리케이션 간 데이터 교환에 사용되어질 XML 응용, 즉 “XML 프로토콜”에 대한 워킹 그룹을 결성하여 이에 대한 표준안을 작성할 예정이다.

본 고에서는 이와 같은 XML을 기반으로 한 분산 환경에서의 어플리케이션간 데이터 교환을 지원하는 XML 프로토콜의 구조와 기능에 대하여 살펴보고, 이에 대한 후보안으로써 WDDX, XML-RPC와 SOAP을 언급한다. 마지막으로 XML 프로토콜 표준화 동향을 살펴보도록 한다.

2. 컴포넌트 모델로서의 XML

컴포넌트 소프트웨어의 주된 목적은 소프트웨어 개발에 있어서 재사용을 증가시키고 개발자간의 협동(collaboration)과 협조(cooperation)를 증가시키기 위한 것이다. 이러한 컴포넌트 기술이 가져야 할 가장 중요한 특징으로는 상호운용성(Interoperation)이 있으며 이 상호운용성은 아래와 같이 계층적으로 구별해 볼 수 있다[5].

메모리 상의 상호운용성이란 오브젝트 또는 컴포넌트의 메모리 상의 표현을 표준화함으로써 컴포넌트 포맷에 상관없이 VM(Virtual Machine) 상에서 가질 수 있는 호환성을 뜻하며, 소스 코드 상의 상호운용성은 컴포넌트 서비스에 접근할 때 필요한 API를 표준화함으로써 얻어지는 소스

* 학생회원

** 종신회원

In-Memory Interoperation
Source Code Interoperation
Type Information Interoperation
Wire Interoperation

그림 1 컴포넌트 상호운용성의 4계층

코드 레벨에서의 호환성을 의미한다. 타입 정보에 따른 상호운용성은 컴포넌트를 사용하는 각 시스템 또는 플랫폼에서 데이터 타입에 대한 호환을 의미하며, 네트워크에서의 상호 운용성(wire interoperation)은 분산 환경에서 컴포넌트 사용시 호환성을 유지하기 위한 네트워크 프로토콜을 의미한다.

현재 가장 널리 사용되고 있는 컴포넌트 기술로는 Microsoft의 COM/DCOM, OMG의 CORBA, Java의 Beans가 존재하나, 이러한 컴포넌트 기술들은 서로 상호 호환되지 않는다. 예를 들어 메모리 상의 상호운용성을 지원하기 위해서는 서로 동일한 VM 위에서 동작해야만 하며, 소스 코드상에서의 상호운용성을 지원하기 위해서는 컴포넌트 서비스 접근시 필요한 API끼리 서로 호환이 가능해야만 한다. 또한 각 컴포넌트가 지원하는 데이터 타입끼리 호환이 되어야 하며, 사용하는 프로토콜끼리도 호환 가능해야만 한다. 더욱이 이러한 컴포넌트 기술들은 웹과의 호환성을 직접 가지지 못한다. 따라서 웹을 데이터 교환에 사용하는 어플리케이션에서는 별도의

프로토콜을 지원하도록 요구받게 된다. 이러한 컴포넌트 기술 간의 호환성 문제와 웹과의 호환성 문제를 해결하기 위하여 XML을 이용하고자 하는 많은 노력이 있어왔다.

XML은 플랫폼 및 프로그래밍 언어에 독립적이고, 확장 가능하며, 그 자체만으로도 데이터와 메시지 전달을 하는데 있어 표현 수단으로 사용될 수 있다. 또한 DOM과 SAX와 같은 표준 API를 통한 접근 방법을 제공하고, XML Schema나 여러 XML Vocabulary를 통하여 데이터 타입을 확장시킬 수 있으며, 무엇보다도 HTTP 프로토콜을 이용함으로써 현재의 웹 환경에 곧바로 이식할 수 있는 특징을 가진다. 표 1은 XML과 다른 컴포넌트 기술들을 비교한 것이다[5].

3. XML 프로토콜

XML 프로토콜(XP)은 어플리케이션 간 교환 또는 컴포넌트 모델로서 XML을 이용하고자 하는 노력의 일환이다. XML 프로토콜은 다음의 4가지 기능을 지원함으로써 어플리케이션 간 교환을 가능케 한다[6].

- 데이터 교환 시 데이터를 캡슐화
- RPC와 같은 외부 함수 호출의 지원
- 비구조적 데이터에 대한 직렬화
- HTTP 프로토콜을 그대로 이용

표 1 XML과 컴포넌트 기술간의 비교

	XML	COM	Java	CORBA
In-Memory Interoperation	W3C DOM, SAX	The COM API	프로그래밍 언어 자체	Object Interface
Text-based Type Information Interoperation	DTD/XMLSchema	COM IDL	프로그래밍 언어 자체	OMG IDL
Binary Type Information Interoperation	DTD/XMLSchema	Type 라이브러리	.class 파일	-
API-level Type Information Interoperation	XMLSchema	LoadTypeLib, ItypeLib	java.lang.reflect	Interface Repository
Wire Interoperation	XML(over HTTP, raw TCP, 또는 message-based protocols)	DCOM(DCE based) over raw TCP, SPX	RMI/JRMP 또는 RMI/IIOP 또는 RMI/HTTP	IIOP over raw TCP

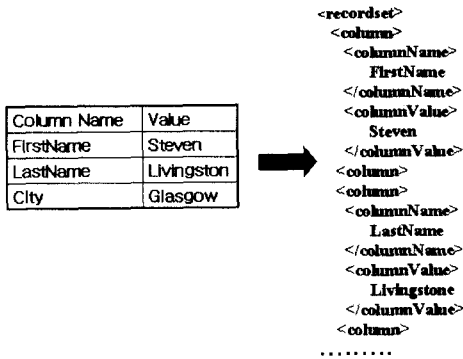


그림 2 XML을 이용한 데이터 직렬화

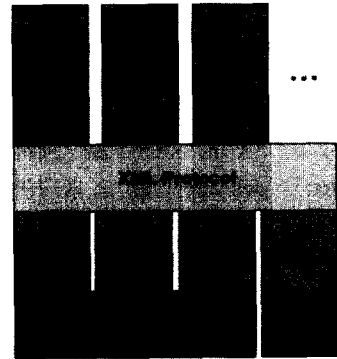


그림 4 XML 프로토콜 계층도

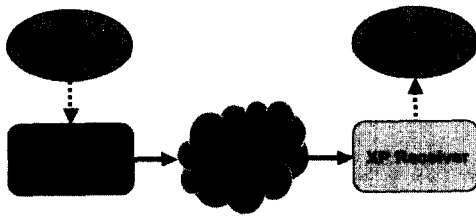


그림 3 XML 데이터의 교환

XML 프로토콜은 실제 어플리케이션 간 교환에 사용되어지는 각 XP 메시지들의 형식과 처리 규칙에 대한 정의이며, XP 바인딩은 이러한 XP 메시지들을 해당 프로토콜에서 전송하고자 할 때 사용되어지는 규칙의 집합이다. XP 모듈은 XP 블록과 XP 핸들러로 구성된다. XP 블록은 XP 프로세서가 XML 프로토콜을 처리하는 단위이며, XP 메시지는 XP 블록들로 이루어진다. 그림 5는 XP 메시지의 구조이다.

XML 프로토콜은 데이터 교환시 해당 데이터를 XML 태그로 캡슐화한다. 이 때, 데이터베이스의 테이블과 같은 구조적인 데이터는 태그의 표현으로써 직렬화되는데, 일단 직렬화된 데이터는 전송받은 시스템 쪽에서 각 태그가 가지는 시맨틱을 기반으로 XP 프로세서를 통하여 다시 데이터 구조를 재구성해 낼 수 있다. 그림 2는 데이터베이스 테이블의 데이터를 XML을 이용하여 직렬화하는 예이며 그림 3은 직렬화된 데이터를 교환/처리하는 예를 그림으로 보인 것이다.

[7]에 따르면, XML 프로토콜은 여러 함수와 서비스를 정의하는 XP 모듈 집합을 가질 수 있는 프레임워크로 정의된다. XML 프로토콜 모듈이 정의하는 함수나 서비스로는 캐싱, 라우팅, 보안 처리 및 구매 신청서의 제출과 같이 특정 분야에서의 이벤트 처리 등이 있다. XML 프로토콜 그 자체는 단순하고 간략하게 설계되어지며, 복잡한 연산들은 추가된 XP 모듈이 지원하게 된다. 그림 4는 XML 프로토콜 계층도이다.

XP 메시지는 XML 프로토콜에서 통신의 단위가 되며, XML 프로토콜이 전송하는 데이터 스트림은 XP 메시지의 집합으로 이루어진다. 이

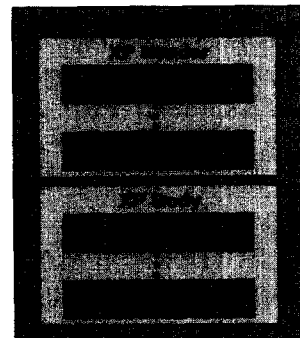


그림 5 XML 프로토콜 메시지 구조

XP 메시지는 XP 헤더와 XP 바디로 구성된 XP 인벨롭(XP envelope)으로 구성된다. XP 바디는 최종 XP 수신 측에 전송될 콘텐츠를 담으며, XP 블록들로 구성되어 있다. XP 헤더는 XP 바디에 대한 추가적인 설명과 옵션을 정의하는 메타 데이터이며, XP 바디, XP 헤더 모두 XP 블록으로 구성되어진다. XP 블록은 XP 프로세서가 XP 메시지를 처리할 때 작업의 논리적 단위이며, 각 XP 블록의 시맨틱은 XP 모듈에서 정의된다. 이러한 XP 블록 중 특별한 형태로는

XP Fault가 존재할 수 있다. XP Fault는 XP 프로세스가 해당 XP 블록을 처리할 때 발생하는 오류에 대하여, XP 메시지를 생성한 측에 반환하는 오류 처리 구문이다.

XP 메시지는 XML 프로토콜 메시지 경로(XML Protocol Message Path)를 통하여 발신 측에서 수신 측으로 전달되어진다. 발신 측은 XP 메시지를 XP 프로세서를 통하여 생성하고 이를 적절한 XP 바인딩을 통하여 수신 측에 전달한다. 수신 측은 해당 XP 메시지를 XP 바인딩을 통하여 추출하여 이를 XP 프로세서를 통해 어플리케이션에게 전달해 준다. 이 메시지 경로에는 XP Intermediary가 존재하는데 이는 XP 메시지를 다른 수신측에 포워드해주는 역할을 수행한다.



그림 6 XML 프로토콜 메시지 경로

4. XML 프로토콜의 예

4.1 WDDX[8,10]

WDDX(Web Distributed Data eXchange)는 프로그래밍 언어 사이에서의 복잡한 데이터 구조를 교환하기 위하여 고안된 XML 프로토콜의 한 예로서 XML을 이용하여 데이터 직렬화를 통하여 데이터를 전달한다. 직렬화된 메시지는 데이터에 대한 타입 정보와, 변수 이름, 변수 값을 포함한다. 그림 7은 WDDX의 한 인스턴스이다.

그러나, WDDX는 데이터 직렬화에 대한 방법만이 정의되어 있을 뿐, 리모트 메소드 호출의 기능이 없으며, XP 바인딩이 HTTP로 한정되어 있는 단점이 존재하며, 지원하는 데이터 타입 또한 한정되어 있다.

```
<!DOCTYPE wddxPacket SYSTEM "wddx.dtd">
<wddxPacket version='0.9'>
  <header/>
  <data>
    <var name="title">
      <string>Professional XML</string>
    </var>
  </data>
</wddxPacket>
```

그림 7 WDDX의 예

4.2 XML-RPC[9,10]

XML-RPC(eXtensible Markup-up Language Remote Procedureal Call)은 리모트 머신에 있는 메소드를 호출하는, 즉 RPC의 기능을 XML로 구현한 XML 프로토콜의 한 후보안이다. XML-RPC는 리모트 머신에 대한 함수 호출을 XML을 이용하여 캡슐화하여 전달하고, 함수 결과 또한 XML 형태로서 전달받는다. XML-RPC는 기존의 HTTP 프로토콜을 그대로 이용하고, XML 표준을 이용함으로써 프로그래밍 언어에 독립적인 특성을 가진대[8].

XML-RPC의 구현은 HTTP POST 방식을 기반으로 하여 이루어진다. 이전 데이터의 전송은 Base64 인코딩 방식을 이용하며, 데이터 타입 또한 XML 태그를 이용하여 표현한다. 기존의 HTTP 프로토콜을 XP 바인딩으로 이용할 수 있음으로써, XML-RPC는 방화벽에서의 특정 프로토콜에 대한 거부를 피하면서 분산 환경을 지원할 수 있는 특징을 가진다. 그림 8은 XML-RPC에서의 Request/Reply의 형태이다. 그러나, XML-RPC는 자체 데이터 타입을 이용함으로써, 데이터 타입에 대한 상호운용성을 보장할 수 없으며, 기존의 컴포넌트 모델에서의 Introspection 기능을 지원하지 못하는 단점을 가진다.

POST /RPC2 HTTP/1.0	HTTP/1.0 200 OK
User-Agent: Frontier/5.1.2(WinNT)	Connection: close
Host: betty.userland.com	Content-length: 342
Content-Type: text/xml	Date: Fri,17 Jul 1998 19:55:08 GMT
Content-length: 181	Server: Userland Frontier/5.1.2-WinNT
<?xml version="1.0"?>	<?xml version="1.0"?>
<methodCall>	<methodResponse>
<methodName>	<params>
myMethodName</methodName>	<param>
<params>	<value>
<param>	<array>
<value><string>	<data>
myParameter</string></value>	<value><string>myParameter
</param>	</string></value>
</params>	<value><double>76570
</methodCall>	</double></value>
	</data>
	</array>
	</value>
	</param>
	</params>
	</methodResponse>
Request	Response

그림 8 XML-RPC에서 Request/Response

4.3 SOAP[10~13]

SOAP(Simple Object Access Protocol)은 플랫폼 독립적으로 인터넷 상에 분산되어 있는 서비스, 오브젝트, 또는 서버에 접근할 수 있도록 HTTP 프로토콜을 지원하는 XML 프로토콜의 한 예이다. SOAP은 1998년 마이크로소프트에 의해 제안되어, IETF와 W3C에 표준안으로 제출되었으며, 마이크로소프트의 BizTalk 서버에 적용되는 등 상업적으로도 다른 XML 프로토콜에 비해 많은 성과를 보이고 있다. 또한 W3C는 [7]에서 SOAP이 XML 프로토콜이 가져야 할 요구사항을 모두 충족할 수 있도록 수정이 된다면, XML 프로토콜의 권고안으로써 SOAP을 그대로 적용할 가능성을 보이고 있다.

SOAP은 WDDX와 같은 데이터의 직렬화뿐만 아니라 XML-RPC와 유사한 원격 함수 호출, 기존의 컴포넌트 모델이 지원하는 Introspection 기능을 수용하고 있으며, 지원하는 데이터 타입은 XMLSchema[14]를 확장한 형태를 사용함으로써 좀 더 나은 확장성을 지원한다. 또한 SOAP은 기존의 HTTP 프로토콜을 그대로 XP 바인딩으로 이용할 수 있을 뿐만 아니라, SMTP, MSMQ, FTP와 같은 다양한 XP 바인딩을 지원한다. 그림 9는 SOAP에서의 Request/Reply 형식과 이에 대한 각 HTTP 헤더 형식을 보이고 있다.

SOAP은 “HTTP Extension Framework”의 기능을 이용하여 HTTP 헤더에 특정 어플리케이션을 위한 추가적인 식별자를 붙일 수 있다. 이러한 확장된 식별자는 확장 네임스페이스로 동작함으로써 이후에 발생할지 모르는 충돌을 해결할 수 있다. 그림 10은 SOAP을 위하여 확장된 HTTP 헤더의 예이다.

SOAP 클라이언트는 먼저 웹 서버에 위와 같은 HTTP 헤더를 SOAP Request를 보내게 된다. 만약 웹 서버가 클라이언트에게 “501 Not Extended” 또는 “510 Not Implemented” 라는 response를 보내게 되면, 클라이언트는 표준 POST 방식으로 다시 Request를 보낸다. 여기에서 Man 헤더는 두 자리의 prefix(ns)를 가진다. 이 prefix는 각 확장된 HTTP 헤더에서 사용되어지며, XML에서의 네임스페이스와 같이 이름

```

POST /StockQuote HTTP/1.1
Request
Host: www.stockquotesserver.com
Content-Type: text/xml
Content-Length: nnnn
SOAPMethodName: Some:Namespace-URI#GetLastTradePrice

<SOAP:Envelope
  xmlns:SOAP="urn:schemas-xmsoap-org:soap.v1">
  <SOAP:Body>
    <m:GetLastTradePrice xmlns:m="Some-Namespace-URI">
      <symbol>DIS</symbol>
    </m:GetLastTradePrice>
  </SOAP:Body>
</SOAP:Envelope>

Response
HTTP/1.1 200 OK
Content-Type: text/xml
Content-Length: nnnn

<SOAP:Envelope
  xmlns:SOAP="urn:schemas-xmsoap-org:soap.v1">
  <SOAP:Body>
    <m:GetLastTradePriceResponse
      xmlns:m="Some-Namespace-URI">
      <return>34.5</return>
    </m:GetLastTradePriceResponse>
  </SOAP:Body>
</SOAP:Envelope>

```

그림 9 SOAP에서의 Request/Response

```

M-POST /StockQuote HTTP/1.1
Man: "http://schemas.xmlsoap.org/soap/envelope/"; ns=NNNN
Content-Type: text/xml; charset="utf-8"
Content-Length: nnnn
NNNN: SOAPAction: "http://electrocommerce.org/abc#MyMessage"

<SOAP-ENV:Envelope...

HTTP/1.1 200 OK
Ext:
Content-Type:text/xml; charset="utf-8"
Content-Length: nnnn

<SOAP-ENV:Envelope...

```

그림 10 SOAP을 위하여 확장된 HTTP 헤더의 예 충돌을 방지하기 위한 목적으로 사용된다.

5. 결론

XML 프로토콜은 XML 응용의 새로운 도입 분야이다. 기존의 웹이 단순히 사용자에게 콘텐츠 제공을 목적으로 하였으며, HTML이 이러한 콘텐츠 제작의 도구로서 사용이 되었다면, XML 프로토콜은 어플리케이션이 웹을 통신 수단으로서 이용할 수 있도록 지원하는 도구이다. 기존의 분산 컴포넌트 기술과는 달리, XML 프로토콜은

기존의 웹 환경을 그대로 이용할 수 있으며, 보다 나은 상호운용성을 지원한다. W3C는 이러한 XML 프로토콜의 필요성을 인식하여 XML 프로토콜 워킹 그룹을 결성하였으며, [16]에서 보는 바와 같이 기존의 많은 XML 프로토콜 또는 분산 컴포넌트 기술을 검토하여 왔다. 현재는 XML 프로토콜에 대한 요구사항이 워킹 드래프트로 나와 있는 상태이며, W3C는 2001년 4월까지 후보 권고안(Candidate Recommendation)을 발표하고, 2001년 10월에는 최종적으로 권고안을 발표할 예정이다[6].

그러나 XML 프로토콜은 이러한 장점에도 불구하고 XML 메시지에 따른 검증(validation)에 따른 성능의 저하와, 텍스트 기반의 통신 프로토콜을 이용함으로써 기존의 컴포넌트 기술보다 높은 대역폭을 요구받는 문제점을 아직 가지고 있다. 또한, XML 프로토콜은 ebXML과 같은 특정 분야에서의 XML vocabulary와 메시지 전달 구조에서 일부분 동일한 기능을 가지며 이에 따른 경쟁이 있을 것이라 예측된다[14,15]. 때문에 W3C는 XML 프로토콜 표준 자체는 단순화시켜 경량화할 것이며, 일반적인 도메인에서의 지원을 목적으로 한다. 대신에, 특정 분야에 대해서는 XML Vocabulary를 지원할 수 있도록 XP 모델을 확장하는 방법을 취한다.

웹 상에서의 어플리케이션간 교환의 필요성이 나날이 증대되고 있고, 또한 이를 지원하기 위한 기술이 필요함에 따라서 XML은 HTML의 기능을 대체하는 것뿐만 아니라, 메시지 교환에서의 표준으로서 적용되어질 것이다. 이에 따라 많은 어플리케이션들에서의 웹 적용이 보다 수월해질 것이며, 또한 기존의 컴포넌트 기술을 대체할 수 있는 새로운 가능성을 보이고 있다.

참고문헌

- [1] W3 Consortium, "W3C Recommendation: Extensible Markup Language 1.0", <http://www.w3.org/TR/1998/REC-xml-19980210>
- [2] Edd Durmbill, XML.com, O'really & Associates Inc, "The State of XML", XML.com., <http://www.xml.com/pub/a/2000/06/xmldope/keynote.html>
- [3] 하이테크 정보, "웹 분산컴퓨팅 시대 열린다", <http://www.hitech.co.kr>, 1998.10.23.
- [4] XML Conformance Subcommittee, "XML Conformance Test Suite", Working Draft, OASIS, 12 July 1999, <http://www.oasis-open.org/committees/xmltest/testsuite.htm>
- [5] Don Box, DevelopMentor Inc, "Lessons from the Component Wars: An XML Manifesto", September 1999, <http://msdn.microsoft.com/xml/articles/xmlmanifesto.asp>
- [6] W3 Consortium, "XML Protocol Working Group Charter", <http://www.w3.org/2000/09/XML-Protocol-Charter>.
- [7] W3 Consortium, "W3C Working Draft: XML Protocol Requirements", <http://www.w3.org/TR/2000/WD-xml-reqs-20001219>, December 19, 2000.
- [8] Simeon Simeonov, Allaire Corporation, "WDDX: Distributed Data for the Web", <http://www.wddx.org/DistributedDataForWeb.html>", December 7, 1998.
- [9] Dave Winer, Userland Software Inc, "XML-RPC Specification", <http://www.xml-rpc.com/spec>
- [10] Richard Anderson, et al, "Server to Server", pp. 497-574, Professional XML, wrox press, 2000.
- [11] W3 Consortium, "W3C Note 08 May 2000 : Simple Object Access Protocol (SOAP) 1.1", <http://www.w3.org/TR/2000/NOTE-SOAP/>
- [12] Pete Loshin, "Web Services and the Simple Object Access Protocol", December 1999, http://msdn.microsoft.com/xml/general/soap_webserv.asp
- [13] Aaron Skonnard, "SOAP: The Simple Object Access Protocol", January 2000, "http://www.microsoft.com/mind/0100/soap/soap.asp"
- [14] W3 Consortium, "W3C Candidate Recommendation: XML Schema Part 2 Datatypes", <http://www.w3.org/TR/>

xmlschema-2/

- [14] Edd Dumbill, XML.com, O'really & Associates Inc, "XML Protocols", <http://www.xml.com/pub/a/2000/05/10/protocols/index.html>
- [15] Edd Dumbill, XML.com, O'really & Associates Inc, "Distributed XML, The role played by XML in the next-generation Web", <http://www.xml.com/pub/a/2000/09/06/distributed.html>
- [16] Eric Prud'hommeaux, "XML Protocol Matrix", <http://www.w3.org/2000/03/29-XML-protocol-matrix>

이경하



1998 충남대학교 정보통신공학과(공학사)
 2000 충남대학교 대학원 정보통신공학과(공학석사)
 현재 충남대학교 대학원 컴퓨터공학과 박사 과정
 관심분야: 데이터베이스, Java, XML, 정보통합, WWW
 E-mail: bart@ce.cnu.ac.kr

이규철



1984 서울대학교 컴퓨터공학과(공학사)
 1986 서울대학교 대학원 컴퓨터공학과(공학석사)
 1990 서울대학교 대학원 컴퓨터공학과(공학박사)
 1989.3~현재 충남대학교 컴퓨터공학과 교수
 1994.3~1995.6 미국 IBM ALmaden Research Center 객원 연구원
 1995.8~1996.8 미국 Syracuse University, CASE Center 객원 교수
 1997.1~1998.1 학술진흥재단 부설 첨단학술정보센터관건 교수
 관심분야: 데이터베이스, XML, 정보통합, 멀티미디어 시스템
 E-mail: kcleee@ce.cnu.ac.kr

● 제11회 고속통신망 워크샵 ●

- 일 자 : 2001년 2월 14 ~ 16일
- 장 소 : 용평 리조트
- 주 최 : 정보통신연구회
- 문 의 처 : 한국과학기술원 전산학과 조동호 교수
 Tel. 042-869-3467
<http://bnlab.kaist.ac.kr/hsn2001>