

에너지 정보 교환을 위한 esXML (energy system eXtensible Markup Language) 개발

Development of esXML for Energy Information Exchange

김정숙* · 구현우**

Jung-Sook Kim* and Hyun-Woo Koo**

* 김포대학 IT학부

** 동국대학교 컴퓨터공학과

요 약

미래 디지털 환경에서 에너지 관리 시스템은 예기치 못한 정전이나 과부하 같은 응급 상황 이벤트들이 발생하면 이를 실시간으로 대처할 수 있어야 하고, 원격 검침과 같은 소비자 서비스 이벤트들도 신속하게 제공되어야 한다. 따라서 다양한 에너지 이벤트 정보들을 실시간으로 전달할 수 있는 단순하면서 장치 독립적인 정보 교환 모델이 필요하다. 이에 본 논문에서는 XML을 기반으로 객체 지향 모델링을 사용하여 장치와 이벤트 모델링을 분리한 esXML을 개발하였다. 실험 결과 독립적이고 효율적으로 정보 교환을 할 수 있었다.

키워드 : esXML, 에너지 관리 시스템, 에너지 정보 교환, 장치 모델링, 이벤트 모델링

Abstract

In future digital environment, energy management system will meet the real-time capability to process the emergency events, unexpected blackouts or over-load, and the high speed to provide the consumer service events such as remote meter reading. According to, energy management system needs the simple and independent information exchange model which can transmit various energy event information in real-time. In this paper, we developed an esXML that was divided into two modelings, device modeling and event modeling, based on XML using object-oriented modeling. As a result of experiments, the system was able to exchange information independently and efficiently.

Key Words : esXML, Energy Management System, Energy Information Exchange, Device Modeling, Event Modeling

1. 서 론

현재 정보통신 기술의 눈부신 발달로 미래 디지털사회의 요구를 충족시킬 수 있는 에너지 시스템의 확보와 이를 제어하는 지능형 양방향 통신 네트워크와의 통합 시스템 개발이 필요하다. 특히 에너지 시장 환경의 변화가 소비자 중심으로 변하고 있어, 소비자가 원하는 수요관리 및 추가적인 서비스 기능이 제공되어야 한다. 따라서 이러한 요구사항들을 충족시킬 수 있는 에너지 관리 시스템을 개발하고 이를 기반으로 에너지 소비자의 서비스 개선과 동시에 대수용가가 수익을 창출할 수 있는 부가서비스 애플리케이션을 개발하는 일이 필요하다. 특히 에너지는 자연재해 등과 같은 예상할 수 없는 상황에서 정전과 과부하 같은 응급 상황이 발생할 수 있다. 이때 신속하게 대처하지 않으면 많은 피해가 발생할 수 있다.

따라서 이러한 응급상황에 신속하게 대처할 수 있는 장치

들과 에너지 관리 시스템간의 지능형 양방향 통신이 가능하여야 한다. 응급 상황이 발생한 장치들은 신속하게 응급 상황을 에너지 관리 시스템에게 전달하여야 하고, 에너지 관리 시스템은 발생한 응급 상황에 대해 정확하고 실시간으로 대처할 수 있는 방안이 마련되어야 한다. 뿐만 아니라 언제 어디서나 사용자가 원하는 정보 즉 에너지 사용량이나 부하 관리를 위한 정보 등을 실시간으로 송·수신할 수 있어야 한다. 이러한 에너지 관리 시스템 개발을 위해서는 먼저 표준화된 에너지 관리 시스템 아키텍처를 수립하고, 수립된 표준 아키텍처를 기반으로 양방향 통신을 가능하게 하는 시스템으로 개발되어야 한다. 특히 에너지 관련 정보의 교환에 대한 표준을 추진하여, 기존의 다양한 형태의 데이터교환에 따른 한계를 극복하고 누구나, 어떠한 플랫폼에서도 데이터를 교환할 수 있는 기반을 제공할 수 있어야 한다[1, 2, 3, 4]. 이러한 요구들을 해결하기 위해 연구들이 진행되고 있으나, 현재 검침기와 몇 가지 종류의 장치들을 제외한 대부분의 에너지 장치들은 에너지 시스템에서 발생한 이벤트를 직접 처리할 수 있는 기능을 탑재하고 있지 않다. 특히 에너지 장치들의 종류가 다양하며, 에너지 사용량이 많은 대수용가들은 에너지 사용량 등을 수시로 점검할 필요가 있고, 또한 원격 에너지 시스템에 대한 정보가 필요하거나 에너지 시스템을 제어해야 할 필요성이 대두되고 있다. 이렇게 점차 증

접수일자 : 2008년 12월 10일

완료일자 : 2009년 4월 6일

* 교신 저자

본 논문은 2009학년도 김포대학의 연구비 지원에 의하여 연구되었습니다.

가하고 있는 에너지 정보 교환과 다양한 종류의 에너지 장치들에서 효율적으로 이벤트를 처리할 수 있는 기능을 탑재한다고 하면, 가능한 간단하면서도 장치에 독립적인 에너지 정보 교환 모델 개발 또한 필요하다. 그러나 에너지 장치들과 에너지 관리 시스템간의 발생하는 이벤트들을 처리할 때, 효율적인 정보 송·수신을 할 수 있기 위해 필요한 정보교환 모델에 대한 연구는 거의 찾아 볼 수 없다.

이에 본 논문에서는 에너지 장치와 에너지 관리 시스템에서 발생하는 이벤트들을 처리하고자 할 때 발생하는 정보들을 효율적으로 교환할 수 있도록 하기 위해 웹의 표준이면서 데이터를 간결하게 모델링할 수 있는 XML[5, 6]을 기반으로 단순하면서 플랫폼에 독립적인 esXML을 객체지향 모델링 기법을 이용해 개발하였다. 객체 모델을 사용하여 물리적인 장치를 논리적으로 모델링하고, 장치로부터 발생하는 이벤트들도 모델링을 하였다. 이러한 모델링을 개발하기 위해 필요한 기본 자료형(primary data type), 참조 자료형(reference data type) 그리고 기본 자료형과 참조 자료형의 혼합형태인 사용자 정의 자료형(user defined data type)을 정의하였다. 또한 본 논문에서 개발한 esXML을 실험하기 위해 미래 에너지 시스템을 판넬에 축소하여 제작한 후 이들로부터 에너지 사용량을 정기적으로 가져오는 예와 비정기적으로 사용자가 요구하는 정보를 송·수신 하는 예를 제시하였으며, 이를 실험하기 위해 에너지 관리 시스템을 개발하여 적용한 결과 요구된 정보를 효과적으로 송·수신할 수 있었다.

본 논문의 구성은 다음과 같다. 먼저 2절에서 관련 연구를 설명하고, 3절에서 효율적인 에너지 정보 교환을 위해 개발한 esXML을 기술하고, 4절에서 본 논문에서 개발한 esXML을 적용한 예들과 실험 결과를 보여주고, 마지막으로 결론을 내리고 향후 연구과제를 살펴본다.

2. 관련 연구

본 절에서는 XML을 기반으로 다양한 분야에서 개발된 응용 예들을 살펴보고자 한다. XML은 서로 다른 시스템 특히 인터넷에 연결된 시스템끼리 데이터의 송·수신과 저장을 효율적으로 지원하기 위해 제작된 마크업 언어로, HTML의 한계를 극복할 목적으로 만들어졌다. 따라서 XML을 이용한 응용들이 많이 개발되고 있으며, 응용의 예들을 살펴보면 ebXML(electronic business eXtensible Markup Language)[7], rcXML(remote control eXtensible Markup Language)[8], voiceXML(voice eXtensible Markup Language), XBEL(XML Bookmark Exchange Language), SMIL(Synchronized Multimedia Integration Language) 그리고 MathML(Mathematical Markup Language)[9] 등이 있다. ebXML은 UN/CEFACT(United Nations Centre for Trade Facilitation and Electronic Business)와 OASIS(Organization for the Advancement of Structured Information Standards)가 주도하여 기업의 규모나 지역적 위치에 관계없이 인터넷을 통해 거래할 수 있도록 하는 규약들의 모음으로 거래 메시지의 교환, 거래 메시지의 수립, 공통의 조건에 의한 데이터 통신과 비즈니스 프로세스 정의 및 등록을 할 수 있다. 기존의 표준들은 대부분 단순히 문서만을 표준화하여 사용하는 것과는 달리 ebXML에서는 재활용 수준을 문서수준뿐만 아니라, 시나리오 수준까지 확대해 사용한다. 즉 비즈니스 전 과정을 모델

링하여 시나리오를 작성하고, 이 시나리오에 따라서 B2B 거래를 자동화해 실행한다. rcXML은 원격제어로 작동되는 모든 가전제품에 대한 제어 명령을 XML 기반의 확장성 표기 언어로 만든 것이다. 따라서 이 rcXML을 해석할 수 있는 미들웨어인 rcXML 인터프리터를 내장한 가전제품은 제어 시나리오를 rcXML 문서 파일을 만들어 유·무선 통신을 이용하여 보내면 이 제어 명령을 그대로 수행할 수 있게 된다. 따라서 원격 제어 서비스 제공자는 그 가전제품에 대하여 전혀 몰라도 rcXML 기반으로 다양한 서비스를 개발할 수 있는 것이다. 그리고 VoiceXML은 음성 인식 기술을 통해 인터넷과 사용자가 상호작용하도록 하는 XML을 기반으로 하는 음성 응용 기술이다. 음성 합성, 디지털 오디오, 음성 다이얼로그 입력 인식, 말하기 입력 기록 등의 특징을 갖고 있으며 웹 기반 개발과 콘텐츠 전달이라는 장점을 대화형 음성 응답 응용에 도입을 시도하고 있다. XBEL은 웹 브라우저 북마크를 저장하는 단순하고 대중적인 포맷이다. Python/XML special interest group (SIG)은 XML 초기에 XBEL을 개발했다. XBEL은 XML의 계층을 사용하여 중첩된 폴더와 북마크를 모델링하는데, 각각 타이틀과 URL의 프로퍼티를 갖고 있다. 벤더들이 필요할 경우 상세들을 추가할 수 있도록 후크를 제공하여 확장성을 높인다. SMIL은 대화식 멀티미디어 표현을 작성하기 위한 XML 기반 언어이며, W3C의 조정 하에 있는 한 그룹에 의해 개발되었고 이 그룹에는 CD-ROM, 양방향TV, 웹, 오디오 및 비디오 스트리밍 산업계 등의 대표자들이 포함되어 있다. 최초의 SMIL 공식 초안은 1997년 11월에 발표되었다. MathML는 XML 응용 중 하나로 수학 수식을 표현하기 위한 마크업 언어이다. 수식을 웹에서 표현하기 위해 만들어졌으며, 단순한 수식의 표기뿐만 아니라 수식의 의미를 표현할 수 있다. 위의 예들처럼 다양한 분야에서 XML을 기반으로 개발된 정보 교환 모델이 있으나, 아직까지 에너지 정보 교환을 효율적으로 하기 위해 개발된 모델은 찾아보기 어렵다.

3. esXML 개발

3.1 시스템 구성도

다음 그림 1은 에너지 관리 시스템의 개략적인 구성요소를 보여준다. 에너지 관리 시스템은 크게 단말기기, 서비스 제공자(portal server, provider)와 서비스 수용자(client)로 구성된다. 서비스 제공자 내에 단말기기, 서비스 제공자 그리고 관리자(administrator)로 구성되고 서비스 수용자 측에는 전달 받은 메시지를 사용자에게 보여주기 위한 인터페이스와 메시지를 처리하기 위한 모듈로 구성된다.

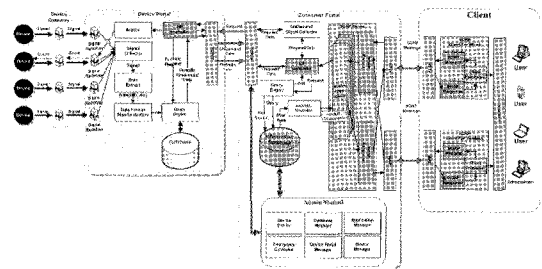


그림 1. 에너지 관리 시스템 구성도
Fig. 1. Block diagram of energy management system

3.2 esXML을 위한 자료형 정의와 장치 및 이벤트 모델링

esXML은 객체 모델이라는 구조로 모든 요소를 모델화한다. 객체 모델에서는 모델화되는 모든 정보가 다양한 객체로 나뉘어 있으며 객체는 구성요소(element) 집합들로 구성된다. 또한 그 객체들 자체는 계층 구조로 그룹화 되어 분류되므로서 처리하기 쉽고 다양한 분야에 응용이 가능하다. 장치를 esXML로 모델링하기 위해서 크게 두 가지 부분에 대한 모델링이 진행되어야 한다. 첫 번째로 물리적 단말 장치들을 표현하기 위한 장치 모델링(device modeling)과 장치에서 발생하는 이벤트 또는 사용자 및 관리자의 장치 관리에 필요한 명령 처리를 위한 이벤트 모델링(event modeling)을 진행한다. 모델링을 하기 위해 먼저 에너지 시스템에서 정보를 표현할 수 있는 자료형을 정의하였다.

1) 자료형 정의

자료형은 기본 자료형, 참조 자료형 그리고 기본 자료형과 참조 자료형의 혼합형태인 사용자 정의 자료형으로 구성된다. 다음의 표 1에 esXML에서 제공되는 자료형을 요약하여 보여주고 있다.

표 1. esXML에서 제공되는 자료형

Table 1. esXML data types

구분	자료형	크기	범위	의미/기타
Primary	bool	1byte	true/false	
	signed int	1byte	128(27)~127(27-1)	정수를 표현
		2byte	-32768(-215)~32767(215-1)	
		4byte	-231~231-1	
		8byte	-263~263-1	
	unsigned int	1byte	0~255(28-1)	0을 포함한 양의 정수를 표현
		2byte	0~65535(216-1)	
		4byte	0~4294967295(232-1)	
		8byte	0~264-1	
	real	4byte	3.4*10 ⁻³⁷ ~3.4*10 ³⁸	크기에 따라 표현되는 정밀도가 다르며 고정 소수점과 배정 소수점 표현
		8byte	1.7*10 ³⁰⁷ ~1.7*10 ³⁰⁸	
	str	variant		가변길이 유니코드 문자 집합 표현
	list/enum	variant		비순서/순서화된 집합 표현
	datetime	8byte	1975-01-01~9999-12-31	다양한 형식으로 표현 가능
Reference	ID	6byte		예) 회사-년도 Serial no.
	ref			HTML 링크 태그를 포함
User defined	Identifier	variant		기본 자료형과 참조 자료형을 혼합하여 새로운 자료형으로 표현

① 기본 자료형

기본 자료형은 정수형, 실수형, 불(boolean)형, 문자열형, 리스트/열거형과 날짜형으로 구성되며, 표 2는 정수 자료형에 대한 자세한 내용을 보여주고 있다.

표 2. 정수 자료형

Table 2. Integer data type

자료형 (type)	속성(Property)		값의 범위
	signed	size (bits)	
int	true	8	128(27)~127(27-1)
		16	-32768(-215)~32767(215-1) / default
		32	-2147483648(-231)~2147483647(231-1)
		64	-263~263-1
	false	8	0~255(28-1)
		16	0~65535(216-1)
		32	0~4294967295(232-1)
		64	0~264-1
스키마 정의의 예	<element name="KW", type="int" size="16" signed="false"/> <element name="fee", type="int" size="64" signed="false"/> <element name="height", type="int", size="16", signed="true"/>		
사용 예	<KW>100</KW> <fee>1234567890</fee> <height>-10</height>		

② 참조 자료형

참조 자료형은 다른 데이터를 참조하기 위한 자료형으로 ID는 계층적 구조로 표현할 수 있으며 속성으로 계층적 구조 표현 여부를 결정할 수 있다. 계층적 구조의 표현은 "@"을 중심으로 앞에는 각각의 장치를 구별하기 위해 표현하는 장치ID 부분이 나타나며, 뒤에는 각 장치가 위치에 있는 장소를 표현하도록 구성되어진다. 여기서 장치ID 부분의 계층 구조와 장소의 계층 구조는 "."으로 구분하여 표현한다. 그리고 ref는 다른 객체를 참조하기 위한 개체로서 HTML의 링크 태그를 포함한다. 다음의 표 3은 참조 자료형을 자세히 기술하여 보여주고 있다.

표 3. 참조 자료형의 상세 설명

Table 3. A detailed description of reference data type

자료형 (type)	속성(Property)		설명
	속성명	값	
ID(Identification)	Unique	true	name을 중복하여 사용할 수 없음
		false	name을 중복하여 사용할 수 있음
	Hierarchy	true	name을 계층적 구조로 표현할 수 있음
		false	name을 계층적 구조로 표현할 수 없음
스키마 정의의 예	<element name="UserID" type="ID" Hierarchy="false" Unique="true"/> <element name="DeviceID" type="ID" Hierarchy="true" Unique="false"/>		
사용 예	<UserID>Hong Gil Dong</UserID> <DeviceID>Monitor123.MyCom@MyHome</DeviceID>		

③ 사용자 정의 자료형

사용자 정의 자료형은 기본 자료형과 참조 자료형을 사용자가 필요에 따라 적절하게 혼합하여 사용하는 것으로 다음의 그림 2에 대표적인 예인 “사용자 정보”를 보여주고 있다.

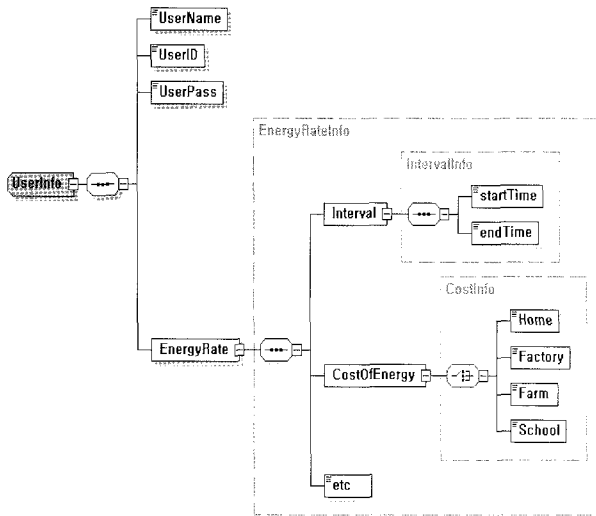


그림 2. 사용자 정의 자료형의 예 : UserInfo

Fig. 2. Example of user defined data type : UserInfo

④ 메소드(method)

메소드는 장치의 동작 또는 기능을 표현하며, 일반 객체 지향 프로그래밍 언어와 같이 함수명, 인자리스트(<argument>)와 반환값(<return>)으로 구성된다. 아래의 표 4와 5에 인자 리스트와 반환값에 대한 예를 보여주고 있다.

표 4. 인자 리스트의 예

Table 4. Example of argument

<arguments>			
자료형 (type)	속성(Property)		설명
	속성명	값	
모든 자료형	name	String	프로그래밍언어에서의 변수명과 같음
	default	Variable	자료형에 따른 기본값을 지정할 수 있음
스키마정의 예	<argument name="SlitentMode" type="int" size="16" signed="false" default="0"/> <argument name="RightNow" type="bool" default="true"/>		
사용 예	<arguments> <SlitentMode>1</SlitentMode> <RightNow>0</RightNow> </arguments>		

표 5. 반환값 예

Table 5. Example of return value

<returns>			
자료형(type)	속성(Property)		설명
	속성명	값	
모든 자료형	name	String	프로그래밍언어에서의 반환 변수와 같음
스키마정의 예	<return name="Result" type="bool"/>		
사용 예	<returns> <Result>>false</Result> </returns>		

2) 장치 모델링

정의된 자료형을 사용하여 esXML 대상이 되는 물리적 단말 장치들을 논리적으로 표현하고 모델링하는 방법이 필요하다. 장치 모델링이란 장치의 특성을 정의하는 XML 스키마를 뜻하며, 이는 단순히 하나의 데이터로 표현하기 어렵다. 즉 장치의 정보는 다양한 정보의 집합으로 표현되며, 이 집합의 각각의 원소에 대한 표현을 하기 위한 모델로서 객체 모델을 사용한다. 객체는 구성요소의 집합으로 구성되며, 각각의 구성요소는 기본 자료형과 참조 자료형으로 구성되는 특성을 가진다. 물론 객체안에 다른 객체를 포함할 수 있다.

표 6. 장치 모델링의 구성요소들

Table 6. Components of device modeling

	구성 자료형 이름	기본 자료형	설명
Device Info	EnergyType	eID	에너지 타입 정보
	DeviceID	ID	장치 식별코드
	User	UserInfo	장치를 사용하는 사용자
	Position	PositionInfo	장치의 위치
	DataOfInstallation	DateTime	장치의 설치 날짜 및 시간
	State	enum	장치의 현 상태
	Energy	EnergyInfo	장치의 에너지 사용량
	Price	PriceInfo	장치의 에너지 단가에 대한 정보
[Function]	Method	장치의 기능	

위의 표 6은 esXML에서 정의한 장치 모델링 구성요소들을 보여주고 있으며, 다음 그림 3은 이를 esXML로 표현한 것이다.

```
<complexType name="DeviceInfo">
  <sequence>
    <element name="EnergyType" type="eID" /> <
element name="DeviceID" type="ID" Unique="true"
Hierachy="true"/>
    <element name="User" type="UserInfo"/>
    <element name="Position" type="PositionInfo"/>
      <element name="DateOfInstallation"
type="datetime" format="YYYYMMDD"/>
    <element name="State" type="DeviceState"/>
    <element name="Energy" type="EnergyInfo"/>
    <element name="Price" type="PriceInfo"/>
    <element name="Function" type="Method"/>
  </sequence>
</complexType>
```

그림 3. 장치 모델링을 위한 esXML
Fig. 3. esXML for device modeling

3) 이벤트 모델링

esXML에서 이벤트 발생 및 처리는 W3C의 권고안을 기본적으로 따른다. 이벤트는 esXML 문서내의 구성요소들과 연관되어 비동기적으로 발생되는 모든 것을 일컫는다. 마우스의 클릭과 구성요소가 잘못된 값을 가지는 경우와 같은 오류 이벤트와 응급 상황이 발생한 응급 이벤트인 alarm을 발생시켜야 하는 경우, 장치 자체의 속성을 변경하는 장치 제어(handle) 이벤트 등이다. 그 외 다양한 이벤트를 정의하여 추가할 수도 있다. 이벤트는 W3C의 권고안인 이벤트의 DOM 모델로 처리되며, 다음과 같은 단계를 거쳐 이벤트 발생을 감지하고 처리한다. Target이라 불리는 element에서 이벤트가 발생되면 observer라 불리는 element에서 감지하고 이벤트 처리 여부를 결정한다[9]. 다음의 표 7은 이벤트에 대한 모델링을 표현한 것이다.

표 7. 이벤트
Table 7. Event

	속성(Attribute)	
	속성명	설명
event	ID	이벤트 식별을 위한 ID
	Type	이벤트의 종류를 표현하는 속성
	targetId	이벤트가 발생한 장치 ID를 표현하는 속성
	[observer]	이벤트를 감지하는 위치를 표현하는 속성 생략가능하며 생략이 되면 기본적으로 디바이스 게이트웨이를 뜻함
	[handler]	이벤트를 처리하는 대상을 표현하는 속성 [user/administrator/device/devicegateway/deviceportal]
	[defaultAction]	이벤트가 발생되어 기본적으로 처리해야 하는 내용을 표현하는 속성, 생략 가능 [ignore/cancel/perform]
	[propagate]	이벤트의 발생을 다른 element에서 전달할 것인가를 나타내는 속성 [stop/continue]

① Alarm 이벤트

Alarm 이벤트는 디바이스 게이트웨이 또는 장치 자체에서 판단하여 사용자나 관리자에게 경고 메시지를 전달한다. Alarm 이벤트는 긴급한 메시지일 가능성이 높음으로 최소

한 정보만으로 메시지를 구성한다.

```
Alarm 이벤트에 대한 응답 메시지

HTTP/1.1 200 OK
Content-Type : text/xml; charset = utf-8
Date: Thu, 04 Apr 2002 11:34:01 GMT
Content-Length : 587
ETag: "0a7ccac50cbc11:1aad"
Server : localhost
<?xml version="1.0" encoding="utf-8">
<esXML version="1.0" type="event">
  <event type="Alarm" id="e_20081010" targetid="power123",
observer="GW123", handler="Administrator",
defaultAction="perform", propagation="continue">
    <arguments>
      <argument name="state">OverPower</argument>
      <argument name="method">ControlDevice</argument>
    </arguments>
  </event>
</esXML>
```

그림 4. Alarm 이벤트에 대한 응답 메시지
Fig. 4. Response message of Alarm event

② 오류 이벤트

오류 이벤트는 요청에 필요한 작업을 하지 못하는 경우를 "error"라 하고 오류 메시지를 사용자나 관리자에게 전달한다. 다음의 그림 5는 오류 이벤트를 esXML로 표현하여 보여주고 있다.

```
<event type="error" id="e_20081010" targetid="NULL",
observer="CP123", handler="Administrator",
defaultAction="perform", propagation="continue">
  <arguments>
    <argument name="method">DeviceView</argument>
    <argument name="code">400</argument>
  </arguments>
</event>
```

그림 5. 오류 이벤트에 대한 esXML
Fig. 5. esXML of error event

③ 장치 제어 이벤트

장치 제어 이벤트는 관리자가 장치 자체 설정을 변경하는 명령(온도 설정, 온도 변경, 가격 변경, 가격 설정, 시간 설정 등)을 전달하기 위한 것이다. 이 이벤트는 에너지 관리 시스템의 On-Demand Signal Collector가 사용자에게서 받은 요청을 디바이스 게이트웨이의 Arbiter에게 전달하고 Arbiter는 장치에게 명령을 전달한다. 구성 속성은 오류 이벤트와 동일하며, 다음의 그림 6에 온도를 40도로 설정하라는 이벤트 예를 보여주고 있다.

```
<event type="handle" id="e_20081010" targetid="power123",
observer="CP123", handler="Administrator",
defaultAction="perform",
propagation="stop">
  <arguments>
    <argument name="method">SetDegree</argument>
    <argument name="degree">40</argument>
  </arguments>
</event>
```

그림 6. 장치 제어 이벤트에 대한 esXML 예
Fig. 6. esXML example of device handle event

Acknowledgement는 장치 제어 이벤트의 처리 결과를 요청자에게 전달하는 메시지로 표 8에서 이에 대한 구성을 보여주고, 그림 7은 장치 제어 결과에 대한 응답 메시지를 표현한 것이다.

표 8. 장치 제어 이벤트에 대한 acknowledgement
Table 8. Acknowledgement of device handle event

	구성 속성	기본 자료형	설명
Event_Ack	DeviceID	DeviceID	이벤트를 처리할 장치 ID
	eventID	EventID	이벤트 ID
	Action	List	이벤트 처리 결과 accept/ignore/reject/rerequest
	EventTime	DateTime	이벤트 처리 시간(format : default)
	Result		처리 결과

```

응답(acknowledgement)

POST /path/to/some/DeviceInfo.cs/ HTTP/1.1
Host: localhost
Content-Type : application/x-www-form-urlencoded
Content-Length : 43
ETag: "0a7ccac50cbc11:1aad"
<?xml version="1.0" encoding="utf-8">
<esXML version="1.0" type="Event_Ack">
<DeviceID>power123</DeviceID>
<EventID>e_20081010</EventID>
<Action>Accept</Action>
<EventTime>2008-10-10 12:00:00</EventTime>
<Result>
<ModifiedDegree>27</ModifyDegree>
<Difference>9</Difference>
</Result>
</esXML>
    
```

그림 7. 장치 제어 이벤트 응답 메시지
Fig. 7. Response message of device handle event

3.3 통신과 보안

esXML은 XML을 기반으로 개발되어 기본적인 통신방법과 보안은 XML과 동일하다. 통신은 HTTP와 SOAP과의 연계를 통신을 진행하지만 기본적으로 필요한 명령은 요청과 응답으로 나눌 수 있다. 요청은 사용자가 서버에게 자료가 필요하다는 메시지를 전달하는 것을 뜻한다. 자료를 단순히 요청하는 경우도 있고 매개 변수를 전달하여 특정한 이벤트나 메서드를 호출하는 경우도 있으며, 요청 메시지는 아래의 내용을 반드시 포함하고 있어야 한다.

- Request method (type of request) : GET or POST
- Relative URI of resource requested

- HTTP Version
- Virtual host name

응답은 서버가 사용자의 요청을 받아 해당하는 자료를 이용하여 응답 메시지를 생성하고 이를 사용자에게 전달하는 것을 뜻한다. 사용자의 요청 형태에 따라 단순 자료 전달 또는 특정한 이벤트나 메서드 호출 결과를 사용자에게 전달한다. 그리고 esXML은 다음과 같은 보안 기능을 제공한다. 접근하고자 하는 사용자의 실체를 확인하는 인증(authentication), 네트워크를 통해 전송되고 수신되는 실제 데이터를 보호하기 위한 암호화(encryption), 접근을 인증 받은 사용자가 실제 esXML 객체를 사용할 수 있는 권한이 있는지를 확인(permission)하는 기능과 사용자의 객체접근 권한 레벨 및 사용자 계정을 관리할 수 있는 기능을 제공한다.

4. 실험 및 결과

4.1 실험 환경과 방법

본 논문에서 개발한 esXML을 실험하기 위해 미래 에너지 시스템을 축소판 판넬을 제작하였다. 제작된 시스템은 10개의 에너지 장치를 탑재하고 있으며, 무선 통신을 통해 장치와 에너지 관리 시스템간의 통신을 수행한다. 그리고 에너지 관리 시스템은 웹에서 동작하고, Linux 환경에서 Oracle과 JAVA[10] 및 Flex[11]을 사용하여 개발하였다. 에너지 관리 시스템은 이벤트 기반으로 동작하며, 장치들의 각각을 제어할 수 있는 이벤트를 발생시킬 수 있고, 또한 각 장치의 정보를 가져올 수 있을 뿐만 아니라 직접 수유관리를 위해 장치들의 전원을 공급 및 차단할 수 있는 기능을 제공한다. 동작 방법은 먼저 장치에서 발생된 이벤트인 경우, 필요한 정보가 esXML로 표현되어 디바이스 게이트웨이를 거쳐 에너지 관리 시스템으로 전달된다. 에너지 관리 시스템에서는 수신된 정보를 데이터베이스에 저장하고, 이를 처리한다. 반대로 에너지 관리 시스템에서 발생한 경우, 전송해야 할 정보를 esXML로 표현하여 디바이스 게이트웨이로 전송하면, 디바이스 게이트웨이에서 각 장치로 전송한다. 특히 장치에서 발생한 이벤트들 중 응급 상황 이벤트인 경우, 디바이스 게이트웨이에서 처리하고, 처리된 결과를 에너지 관리 시스템에게 전달할 수도 있다. 다음의 그림 8은 정기적으로 장치에서 에너지 관리 시스템으로 에너지 사용량을 전달해 주는 예를 보여주고, 그림 9는 비정기적인 이벤트 요청에 대한 결과를 보여주고 있다.

4.2 이벤트 처리

1) 정기적인 이벤트

장치에서 에너지 관리 시스템으로 15분마다 정기적으로 전달되는 분 단위 에너지 사용량에 대한 응답 메시지이다.

```

응답

HTTP/1.1 200 OK
Content-Type : text/xml; charset = utf-8
Date: Thu, 04 Apr 2002 11:34:01 GMT
Content-Length : 587
ETag: "0a7ccac50cbc11:1aad"
Server : localhost
<?xml version="1.0" encoding="utf-8">
<esXML version="1.0" type="PeriodicData">
  <EnergyType> Power </EnergyType>
  <DeviceID>power123</DeviceID>
  <Energy>
    <StartTime>20070711:120101</StartTime>
    <EndTime>20070711:120115</EndTime>
    <MountOfUsing>10</MountOfUsing>
    <MountOfUsing>11</MountOfUsing>
    <MountOfUsing>9</MountOfUsing>
    <MountOfUsing>11</MountOfUsing>
    <MountOfUsing>10</MountOfUsing>
    <MountOfUsing>15</MountOfUsing>
    <MountOfUsing>12</MountOfUsing>
    <MountOfUsing>1</MountOfUsing>
    <MountOfUsing>5</MountOfUsing>
    <MountOfUsing>12</MountOfUsing>
    <MountOfUsing>13</MountOfUsing>
    <MountOfUsing>9</MountOfUsing>
    <MountOfUsing>10</MountOfUsing>
    <MountOfUsing>10</MountOfUsing>
  </Energy>
</esXML>
    
```

그림 8. 정기적인 이벤트 응답 예
Fig. 8. Response example of periodic event

2) 비정기적인 데이터 전송의 예
사용자가 이벤트를 통해 "DP123"이 관리하고 있는 "power123"의 정보와 현재 사용한 에너지량을 요청하는 요청 메시지이다.
다음의 그림 10은 요청을 받은 에너지 관리 시스템이 데이터베이스에서 "power123"의 정보를 찾아 "power123"의 소유자, 설치 위치, 설치 일자, 상태 및 에너지 사용량을 사용자에게 전달해 주는 메시지를 보여주고 있다.

```

요청

POST /path/to/some/DeviceInfo.cs/ HTTP/1.1
Host: localhost
Content-Type : application/x-www-form-urlencoded
Content-Length : 43
ETag: "0a7ccac50cbc11:1aad"
<?xml version="1.0" encoding="utf-8">
<esXML version="1.0" type="event">
  <event type="reading" id="e_20081010" targetid="power123",
observer="DP123", handler="Administrator",
defaultAction="perform", propagation="stop">
  <arguments>NULL</arguments>
</event>
</esXML>
    
```

그림 9. 비정기적인 이벤트 요청 예
Fig. 9. Request example of irregular event

```

응답

HTTP/1.1 200 OK
Content-Type : text/xml; charset = utf-8
Date: Thu, 04 Apr 2002 11:34:01 GMT
Content-Length : 587
ETag: "0a7ccac50cbc11:1aad"
Server : localhost
<?xml version="1.0" encoding="utf-8">
<esXML version="1.0" type="DeviceInfo">
  <EnergyType> Power </EnergyType>
  <DeviceID>power123</DeviceID>
  <User>
    <UserName>Hong Gil Dong</UserName>
    <UserID>HGD123</UserID>
    <UserPass>*****</UserPass>
    <Address>Seoul</Address>
  </User>
  <Position>
    <Floor>1</Floor>
    <RoomNumber>A101</RoomNumber>
  </Position>
  <DateOfInstallation>20071010</DateOfInstallation>
  <State>TurnOff</State>
  <Energy>
    <DataOfMeasurement>20081010:120101</DataOfMeasurement>
    <MountOfUsing>100</MountOfUsing>
  </Energy>
</esXML>
    
```

그림 10. 비정기적인 이벤트 응답 예
Fig. 10. Response example of irregular event

4.3 적용 결과

본 논문에서 개발한 에너지 관리 시스템은 이벤트 기반 시스템으로 사용자가 원하는 에너지 장치에 필요한 이벤트를 선택하고, 선택한 이벤트 발생에서 요구되는 정보를 설정해 주면 된다. 다음 그림 11은 이벤트 설정 화면이며, 다음 그림 12는 이벤트 실행 결과를 보여주고 있다.

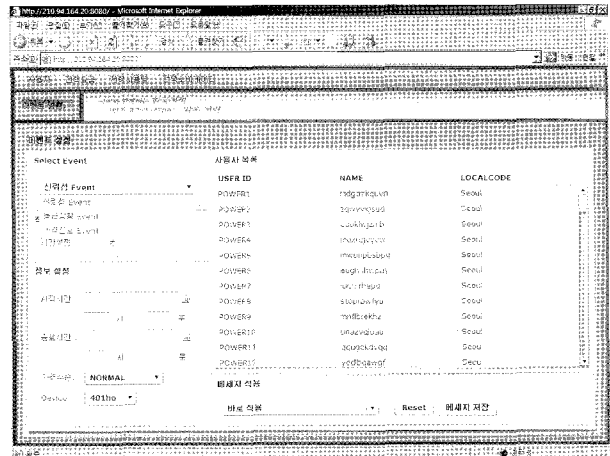


그림 11. 이벤트 설정
Fig. 11. Event setting

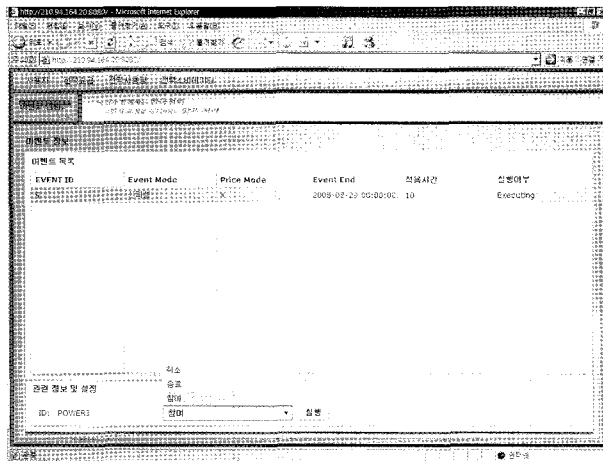


그림 12. 이벤트 실행 결과
Fig. 12. Event execution result

5. 결론 및 향후 연구과제

본 논문에서는 변화하고 있는 에너지 시장 환경에 부합되도록 사용자 중심의 정보 및 부가 서비스 제공을 효율적으로 할 수 있는 에너지 관리 시스템에서 요구되는 정보 교환 모델링을 위해 esXML을 개발하였다. 미래의 에너지 관리 시스템은 다양한 에너지 장치들에서 발생하는 이벤트들을 효율적으로 처리할 수 있어야 하고, 원격 에너지 시스템에서 필요한 정보를 손쉽게 확인해 볼 수 있어야 할 뿐만 아니라, 직접 원격 에너지 시스템을 제어할 수 있어야 한다. 이러한 환경에서 특히 플랫폼에 상관없이 자유롭게 정보 교환이 이루어지기 위해서는 에너지 관련 정보의 교환에 대한 표준을 XML 기반으로 수행하여, 기존의 다양한 형태의 정보교환에 따른 한계를 극복하고 누구나, 어떠한 플랫폼에서도 정보를 교환할 수 있는 기반을 제공해야 한다.

본 논문에서 개발한 esXML은 웹 표준인 XML이 가지고 있는 장점을 그대로 유지하면서 간단하고 효율적으로 정보를 교환할 수 있도록 모델링을 하였다. 객체 지향 모델링 기법을 이용하여 물리적인 장치와 에너지 관리 시스템에서 발생한 이벤트를 처리할 수 있는 모델링을 분리하여 개발하였다. 본 논문에서 개발한 장치 모델링을 적용하면 다양하고 많은 종류의 에너지 장치들의 특성을 에너지 관리에 필요한 정보들로 구성된 장치 모델링에서 요구되는 특성들만을 논리적으로 정형화하므로 장치 독립적인 표현이 가능하다. 그리고 에너지 관리 시스템에서 발생하는 여러 종류의 이벤트에 따른 정보 교환 메시지를 이벤트 모델링을 통해 메시지를 교환하므로 메시지 형식이 단일화되어 일관될 뿐만 아니라, 이벤트 처리기에서 이벤트 "type" 속성만을 먼저 해석하여 어떠한 처리를 해야 하는지를 신속하게 알 수 있으므로 효과적으로 정보를 교환할 수 있다. 이러한 장치와 이벤트 모델링을 정의할 수 있도록 기본 자료형, 참조 자료형 그리고 기본 자료형과 참조 자료형의 혼합형태인 사용자 정의 자료형을 정의하였다. 그리고 esXML을 실험하기 위해 미래 에너지 시스템을 판넬에 축소하여 제작하였고, 에너지 관리 시스템을 개발하여 적용한 결과 요구된 정보를 효과적으로 송·수신할 수 있었다.

향후 연구과제는 에너지 시스템에 직접 적용될 수 있도록 확장하는 일과, 수요자 중심의 부가 서비스 제공을 위한

부가 서비스 개발 및 이를 실현할 수 있는 사용하기 편리한 인터페이스를 개발할 예정이다.

참고 문헌

- [1] Intelligrid Consortium within EPRI, "Phase I of the Intelligrid project white papers : Integrated Energy and Communications System Architecture, Communications Architecture for Distributed Energy Resource in Advanced Distribution Automation, Consumer Portal, Fast Simulation and Modeling", *Intelligrid Consortium within EPRI(Electric Power Research Institute)*, pp. 1-300, 2005.
- [2] Kurt Yeager, Stephen Gehl, Brent Barker, "The Role of Smart Power Technologies in Global Electrification", *19th Proc. of the World Energy Congress*, pp. 1-20, 2004.
- [3] David Becker, "Common Information Model(CIM) : CIM 10 Version", *EPRI*, pp. 1-263, 2001.
- [4] Lori Hogg, "Business Intelligence for Enterprise Energy Management", *Itron White Paper*, pp. 1-5, 2007.
- [5] 박형일 외 옮김, *Beginning XML 3/E*, 사이텍미디어, 2005.
- [6] 김종민, *Power XML 실전 Programming*, 와이미디어, 2007.
- [7] 한국전자거래진흥원, <http://www.kiec.or.kr>.
- [8] 최중인 외 10명, 원격제어서비스용 XML 국제 표준 및 인터프리터 개발, *정보통신부 최종연구보고서*, pp. 5-72, 2005.
- [9] W3C, <http://www.w3.org/TR/>.
- [10] 김시연 옮김, *자바 네트워크 프로그래밍*, 한빛미디어, 2005.
- [11] 바닐로라이, *Flex 3 Knowhow BIBLE*, 대림, 2008.

저자 소개



김정숙(Jung-Sook Kim)
1993년 : 동국대학교 컴퓨터공학과 학사
1995년 : 동국대학교 컴퓨터공학과 석사
1999년 : 동국대학교 컴퓨터공학과 박사
2000년 ~ 현재 : 김포대학 IT학부 부교수

관심분야 : 유전 알고리즘, 지능형 에이전트, 분산/병렬 시스템

Phone : +82-31-999-4659
Fax : +82-31-989-4387
E-mail : kimjs@kimpo.ac.kr



구현우(Hyun-Woo Koo)

2002년: 동국대학교 컴퓨터공학과 학사

2004년: 동국대학교 컴퓨터공학과 석사

2009년: 동국대학교 컴퓨터공학과 박사

2006년~현재: 동국대학교 시간강사

관심분야 : 분산/병렬 시스템, 실시간 시스템, 지능 로봇

Phone : +82-2-2260-3338

E-mail : hwgoo@dongguk.edu