

DTD 자동 생성 기법을 이용한 USN 미들웨어 설계 및 구현

남시병*, 권기현*, 유명한*

Design and Implementation of USN Middleware using DTD Generation Technique

Si-Byung Nam*, Ki Hyeon Kwon*, Myung-Han Yu*

요약

웹 서비스 어플리케이션 기반 모니터링 시스템에서 빈번히 발생하는 데이터 구조 변경에 따른 코드 재생산에 확장성 및 오류 복구능력 향상을 위하여 DTD(Document Type Definition) 자동 생성 기법을 이용한 모니터링 시스템 설계방법을 제안하였다. 제안한 DTD 자동 생성 기법은 센서 데이터 구조가 수정되어도 서버 측에서 동적 서버사이드 스크립트(Dynamic Server-side Script)로 DTD정보를 자동으로 생성해 주기 때문에 데이터 손실과 오류에 대한 능동적인 대처가 가능하며, AJAX(Asynchronous JavaScript and XML)를 이용한 XML 데이터 파싱(Parsing)을 구성하여 대단위 데이터 전송이나 다양한 모니터링 시스템 분야에 데이터 손실이나 손상 확인 등의 예외처리 및 데이터 구조의 확장과 축소가 용이하였다. 제안된 시스템은 일시적인 데이터 깨짐 상황에서 정상화 복구에 필요한 평균시간이 기존의 XML을 이용한 방법보다 약 44.8ms 감소되어 오류 복구처리 속도가 향상됨을 알 수 있었다.

▶ Keyword : DTD, USN, 모니터링 시스템

Abstract

In the monitoring system based on web service application, it is faced with the problems like code reproduction, difficult scalability and error recovery derived from the frequent change of data structure. So we propose a technique of monitoring system by DTD(Document Type Definition) automatic generation. This technique is to use dynamic server-side script to cope with the change of sensor data structure, generate the DTD dynamically. And it also adapt the AJAX(Asynchronous JavaScript and XML) for XML data parsing, it can support mass data transmission and exception processing for data loss and damage. This technique shows the result of recovery time is decreased

• 제1저자 : 남시병 • 교신저자 : 권기현

• 투고일 : 2011. 12. 03, 심사일 : 2011. 12. 22, 게재확정일 : 2012. 01. 03.

* 강원대학교 전자정보통신공학부(Dept. of Information & Communication Engineering, Kangwon National University)

about 44.8ms in case of temporary data failure by comparing to the conventional XML method.

▶ Keyword : DTD, USN(Ubiquitous Sensor Network), Monitoring System

I. 서론

USN(Ubiquitous Sensor Network) 하드웨어 등 관련 기술의 빠른 발전 속도에 비해, USN을 관제하는 모니터링 소프트웨어 분야의 발전은 낮은 편이다[1].

모니터링 시스템은 운영 방법상 크게 PC 어플리케이션과 웹 서비스 어플리케이션으로 구분된다. PC 어플리케이션은 웹 서비스 어플리케이션에 비해 폭넓게 상용화된 방법이지만 PC상에 어플리케이션 소프트웨어가 반드시 설치되어 있어야 하는 단점이 있다[2, 3, 4].

웹 서비스 어플리케이션은 서비스용 웹 서버를 두고 인터넷에 연결된 원격 모니터링 PC의 웹 브라우저를 통하여 접속하는 방법으로서 점차 웹 서비스를 이용한 모니터링 방법이 활성화되고 있다. 그러나 웹 브라우저 내에서 실행하는 웹 서비스 특유의 보안 제약, 오류에 대한 복구 능력, UI 표현능력의 한계점으로 인하여 실시간 정보 전달이 필요한 분야에서 사용하기에는 한계가 있다[5].

본 연구에서는 위에서 서술한 웹 서비스 어플리케이션 기반 실시간 USN 모니터링 시스템에서 나타나는 저조한 오류 복구능력에 대한 단점을 보완하기 위해 DTD(Document Type Definition) 자동 생성 기법을 이용한 실시간 USN 데이터 모니터링 품질 개선방법을 제안한다. 모니터링 시스템과 관제 서버 사이의 통신에서 오류가 발생하였을 때 기존의 DTD를 포함하지 않는 XML(eXtensible Markup Language) 데이터 구조만으로 구성된 시스템들은 실시간 적으로 상세한 오류내역을 검출하기가 어렵다[6, 7].

USN 모니터링 분야에서 XML 데이터 구조 검증을 위한 DTD는 데이터 구조가 수정될 때 마다 DTD정보를 함께 수정해 주어야 하는 번거로움으로 인하여 현장에서 사용되지 않고 있다. 하지만 본 연구에서 제안한 DTD 자동 생성 방법을 이용할 경우 데이터 구조가 수정되어도 서버 측에서 DTD정보를 자동으로 수정 생성해 주기 때문에 데이터 손실과 손상 등 오류에 대한 품질 개선이 가능하며, 부가적으로 데이터 구조 변경에 대한 확장성을 향상시킬 수 있다.

본 논문의 구성은 서론에 이어 II장에서는 USN 및 기존 모니터링 시스템에 대한 전반적인 내용을 서술하고 III장에서 본 논문에서 제안한 DTD 자동 생성 구조를 설명한다. IV장, V장에서는 전체 시스템 구성과 실시간 USN 모니터링의 오

류 복구 성능에 대하여 기존의 시스템들과 본 연구에 제안한 시스템을 비교하고 차이점을 분석한다. 끝으로 VI장에서는 향후 USN 모니터링 시스템의 개선점에 대해 서술하고 결론을 맺는다.

II. 관련 연구

모니터링 시스템은 운영 방법에 따라 PC기반 모니터링 시스템과 웹기반 모니터링 시스템으로 구분할 수 있다. PC 기반 모니터링 시스템은 다양한 부가기능 구현이 가능한 점이 장점이나 플랫폼 의존적이며 별도의 유지비용이 요구된다. 웹기반 모니터링 시스템은 플랫폼 비의존적이나 느린 업데이트 및 반응속도, 불안정성 (스크립트 오류, 서버 장애, 네트워크 오류 등에 대한 처리가 어려움) 등의 문제점을 가지고 있다. 모니터링 시스템의 종류별 장단점은 표 1과 같다.

표 1. 모니터링 시스템 장/단점 비교
Table 1. The Characteristics of Monitoring System

시스템 종류	장점	단점
(1) PC어플리케이션 기반	-다양한 부가기능 구현가능 -빠른 업데이트 및 반응속도	-플랫폼 의존적 -별도의 설치과정 필요 -많은 비용이 드는 데이터구조확장, 검증
(2) Web서비스 기반	-별도의 프로그램 설치과정 불필요 -플랫폼 비의존적 -상대적으로 쉬운 구현방법	-느린 업데이트 및 반응속도 -불안정성 -제한적인 부가기능 구현 -많은 비용이 드는 데이터구조확장, 검증
(3) 제안시스템	<(2)의 장점 외> -빠른 업데이트 및 반응속도 -동적 DTD생성을 통해 보다 안정적인 전송 및 데이터 표출 보장 -자유로운 데이터 구조 확장, 검증	-제한적인 부가기능 구현

제안된 시스템은 관리자가 데이터 구조와 DTD 정보 관련 파일을 직접 수정하지 않더라도 동적으로 UI상에서 각 모니터링 요소에 대한 추가 및 수정작업이 가능토록 하는 모니터링 시스템으로 웹기반 모니터링 시스템의 장점 이외에 빠른 업데이트 및 반응속도 및 DTD 자동 생성을 통해 보다 안정적인 전송 및 데이터 표출 보장하는 시스템이다.

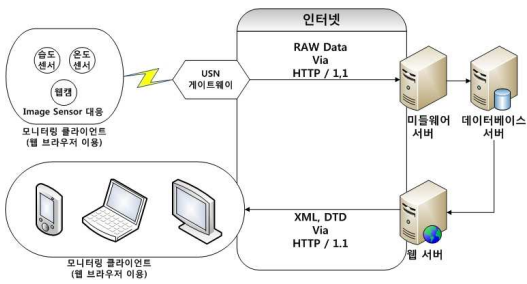


그림 1. 웹 기반 USN 모니터링 시스템의 전체 구성도
Fig. 1. The Structure of Web based USN Monitoring System

제안된 시스템은 게이트웨이 역할을 동시에 수행 가능한 상시전원 환경 및 ARM 플랫폼 기반의 센서 노드와, 이에 연결된 온도 및 습도 센서, USB 웹 카메라 캡처를 통해 640x480의 고해상도 이미지를 연속된 스트림으로 전송하는 이미지 센서를 구성하였으며, 이를 이용하여 서버로 데이터를 전송하여 미들웨어가 그 데이터를 받아서 DB에 축적하고 모니터링 시스템에 제공토록 구성하였다.

취득된 센서 데이터는 HTTP를 이용 모니터링 서버로 보내지고 모니터링 서버에서 데이터베이스 및 파일시스템에 맞게 저장하고, 데이터를 각 필드 구성에 맞는 DTD 형식으로 가공하여 사용자 브라우저에 전송한다.

III. DTD 자동 생성 구조

웹 기반 모니터링의 문제점 중 하나로 지적되는 XML 데이터 구조의 검증능력과 확장성을 위해 데이터 전송 과정에서 DTD 정보를 동적으로 생성, 추가하고 데이터 수신 및 UI 표현 과정에서 수신된 데이터의 유효성을 검증한다.

3.1. DTD 자동 생성

기존의 모니터링 어플리케이션 시스템은 미리 정형화된 형태의 XML 패키지를 로드하여 파싱 후 화면에 표시하는 방법이다.

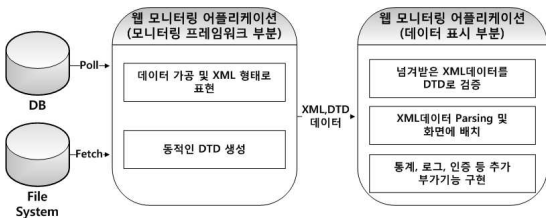


그림 2. DTD 자동 생성 기반 웹 모니터링 시스템의 구성
Fig. 2. Web based USN Monitoring System using DTD Generation

제안된 시스템은 그림 2와 같이 서버사이드 스크립트로 구현된 DTD와 XML 데이터 구조를 통해 실시간 적으로 패키지 검증이 용이하도록 설계하였으며, 그림 3에서와 같이 데이터 필드 갱신 시 쉽게 동적으로 구조를 변경할 수 있는 장점이 있다.

```

<?php
$SensorData =
    $database->getSensorData($sensorNode, $sensorType);
$SensorDS = $database->getSensorDataStructure(0);
$dtdObject = new DTDGeneration($SensorDS, $SensorData);
$dtdObject->Create();
echo $dtdObject->Print();
?>
    
```

그림 3. PHP에서의 DTD 자동 생성
Fig. 3. DTD example generated by PHP

DTD 구조 생성을 위한 별도의 라이브러리(DTDGeneration 클래스)를 PHP로 구현하였으며, 이 라이브러리를 이용하여 센서 데이터의 구조에 맞는 DTD 구조를 생성 및 제공하게 된다. DTD 생성을 위한 라이브러리는 그림 4와 같이 구성되어진다.

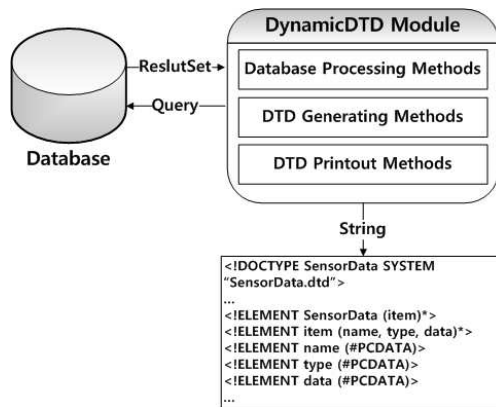


그림 4. DTDGeneration 라이브러리의 구조 및 동작
Fig. 4. The Library Structure of DTD Generation

DTDGeneration 라이브러리는 데이터베이스에서 모니터링 대상 센서데이터들의 구조에 대한 정보를 가져오거나 설정하는 데이터베이스 처리 메서드, 데이터베이스로부터 받은 데

이더로 DTD 구조를 생성하여 문자열 형태로 변환하는 DTD 생성 메서드, XML 데이터에 DTD 정보를 취합하고 최종 출력처리를 담당하는 DTD 출력 메서드로 이루어지게 된다.

클라이언트에서는 XML 데이터와 함께 서버에서 생성된 DTD 정보를 가져온 뒤, 매번 데이터를 갱신할 때 마다 그 데이터가 올바른지 검증한다. 만약 데이터의 구조가 올바르지 않을 경우 서버 측에 시간을 두고 재전송을 요청하거나 모니터링 화면상에 디버깅을 위한 데이터 깨짐에 대한 상세 정보를 제공할 수 있다.

3.2. DTD 예외처리

클라이언트 측 XML 파서는 기본적으로 DTD 정보가 올바르지 않을 경우에는 예외 처리를 통해 파싱작업이 중단되도록 설정되어 있다. 최종적으로 구성된 클라이언트측 검증 코드는 그림 5와 같다. XML 파서는 데이터를 로드한 후 해당 XML 구조 내에 DTD가 존재하는지 여부를 판단하여 검증을 수행하게 되며, DTD와 비교한 결과 오류가 있거나 검증작업 도중 다른 치명적인 문제가 발생한다면 `parseError` 객체에 오류 코드(`errorCode`)와 상세한 오류내용(`reason`)에 대한 정보를 남기게 된다. 아무런 문제가 없다면 `errorCode`는 0으로 남게 된다.

만약 DTD 정보가 포함되지 않았다면, 오류 검증 과정은 UI 표현단계로 미루어지게 된다. 이 경우 XML 데이터의 각 필드별 데이터 유무를 판단하여 단순 오류발생 여부를 판별할 수 있게 된다.

```

xml = new ActiveXObject("Microsoft.XMLDOM");
xml.async = false;
xml.validateOnParse = true;
xml.loadXML(xmlData);

if(xml.parseError.errorCode != 0) {
    // 문제 발생 (xml.parseError.reason = 상세 오류정보)
}
else { // 정상적인 XML 데이터 }
    
```

그림 5. DTD를 이용한 XML 구조 검증 코드 (IE)
Fig. 5. The XML Validation using DTD Generation(IE)

3.3. XML 데이터 구조

```

<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE SensorData SYSTEM "SensorData.dtd">
<sensorData version="1.0-unicode">
    <location>193.31,126.725,247.173</location>
    <name>교량 이미지센서 1</name>
    <type>image</type>
    <data>
        <value>uimg_37826_0001.jpg</value>
        <parseAs>string, image/x-jpeg</parseAs>
        <time>2010/01/01 00:00:00</time>
    </data>
</sensorData>
    
```

그림 6. 센서 데이터 전송형식 (이미지센서)
Fig. 6. The Sensor Data Transfer Example (Image Sensor)

모니터링 클라이언트에서 설정된 갱신주기에 따라 HTTP 1.1 프로토콜을 이용하여 서버 측에 XML 데이터와 DTD 정보를 요청하는 Request를 보내면 서버는 이에 대한 응답으로 HTTP Status Code와 현재 시간, 데이터 만료일, 패킷의 길이, 기타 부가적인 헤더를 포함한 결과 값을 텍스트 형태로 반환하게 된다. AJAX[8]의 경우, HTTP Status Code를 우선적으로 분석하여 서버 오류 발생여부 및 오류의 종류를 파악하고 그에 적합한 예외처리 작업을 수행할 수 있다.

센서데이터의 전송과 표현을 위한 가장 기본적인 XML의 구성은 그림 6과 같다. 최상단에는 DTD 정보를 담은 서버사이드 스크립트 파일에 대한 경로를 정의하였으며, 이하의 필드는 모두 데이터베이스 상에 저장되어진 필드 정의 정보를 바탕으로 자동으로 구성되어 관리자가 필요에 따라 필드를 추가적으로 생성하여 코드 수정 없이도 바로 모니터링 시스템에 반영할 수 있다.

각 센서데이터는 `sensorData` 필드의 하위에 위치하며, GIS 연동 등에 필요한 센서노드 위치정보(`location`), 센서노드 명(`name`), 센서노드 종류(`type`), 데이터(`data`) 필드로 이루어진다. 데이터 필드는 다시 하부에 수치(`value`), 처리대상 데이터 종류 지정(`parseAs`), 수신시간(`time`) 필드를 두게 된다. 이러한 모든 구조는 데이터베이스에 필드 정보를 정의함으로써 자동으로 생성되어진다.

XML을 검증하기 위한 DTD 정보 또한 XML 데이터와 동일하게 서버 측에 동적으로 생성되어진다. DTD가 손상되면 XML의 Parsing과정 전체에 영향을 주게 되므로 예외처리 과정에 주의가 필요하다. 그림 7은 그림 6의 XML 구조에

따라 자동으로 생성된 DTD 정보를 나타낸다.

이렇게 생성된 DTD 정보와 XML 구조를 조합하여 그림 5와 같이 검증 작업을 수행할 수 있다. PHP의 경우 XML 문서의 검증과정은 코드 상에서 DOMDocument::validate()를 호출함으로써 간단하게 이루어지며, 해당 메시지의 실행 결과는 true(성공) 혹은 false(실패)만으로 리턴되어진다. AJAX의 경우 자체적으로 DTD를 인식하여 Validation을 수행하기 때문에 검증 코드가 불필요하며, Internet Explorer 브라우저의 경우 디버깅 등 추가적인 필요에 의해 XMLHttpRequest 오브젝트의 검증 관련 메시지와 코드를 불러올 수 있다.

```

<!ELEMENT sensorData
  (location, name, type, data)*
  >
<!ELEMENT location (#PCDATA)
  >
<!ELEMENT name (#PCDATA)
  >
<!ELEMENT type (#PCDATA)
  >
<!ELEMENT data (value, parseAs, time)*
  >
<!ELEMENT value (#PCDATA)
  >
<!ELEMENT parseAs (#PCDATA)
  >
<!ELEMENT time (#PCDATA)
  >
<!!ATTLIST sensorData version CDATA #REQUIRED
  >
    
```

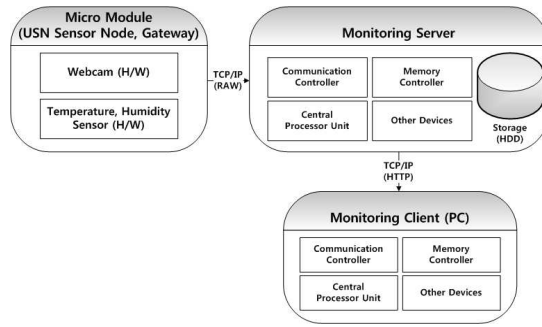
그림 7. 그림 6의 센서 데이터에 따른 DTD 정보
Fig. 7. The DTD of Sensor Data

IV. 시스템 구현

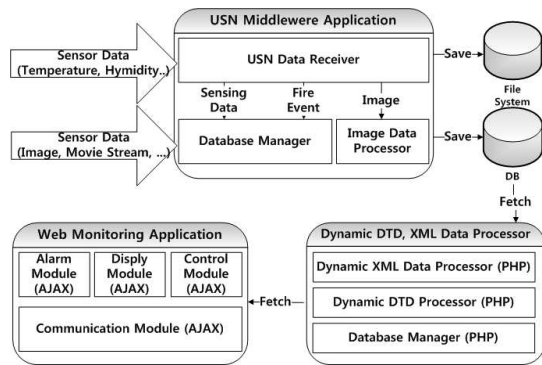
4.1 시스템 구성도

그림 8은 USN Middleware, Web Monitoring을 포함하며 USN Node 및 Gateway H/W의 Firmware를 제외한 나머지 하드웨어 및 소프트웨어의 구성을 나타낸다.

USN 센서노드 및 게이트웨이 역할을 담당하는 마이크로 모듈은 장착된 웹캠과 온/습도 센서의 데이터를 TCP/IP를 통해 서버 측에 전달하며, 서버는 이를 수신하고 처리하여 데이터베이스 및 파일 형태로 스토리지에 저장하게 된다. 수집된 데이터는 데이터베이스에서 추출된 후 XML과 DTD 정보의 형태로 가공되어 클라이언트측에 HTTP를 이용하여 전달되며, 이 데이터를 이용하여 모니터링 어플리케이션은 각종 상황처리와 화면 표시, 제어 등의 통신작업을 수행하게 된다.



(a) USN 시스템의 하드웨어 구성도
(a) H/W Structure



(b) USN 시스템의 소프트웨어 구성도
(b) S/W Structure

그림 8. 제안 USN 시스템의 H/W 및 S/W 구성도
Fig. 8. The System Structure of H/W and S/W

구성된 USN 시스템은 일반적으로 가장 많이 도입되어 사용되어지고 있는 분야인 재난재해 및 이상상황 감시를 주목적으로 하였다. 모니터링 클라이언트를 제외한 모든 소프트웨어는 GNU/Linux 운영 체제상에서 구동되어지며, PHP 언어로 코딩되었다. 데이터베이스에는 MySQL 5.1 버전을 사용하였다. 웹 서버는 Apache Web Server 2.2 버전을 사용하였다.

4.2 센서 전달 패킷 구조

USN 센서노드에서 수집된 데이터는 게이트웨이를 거쳐 별도의 변환절차 없이 미들웨어로 TCP/IP를 통해 아래와 같은 패킷 형태로 전송된다.

CMD(3)	Node ID(1~3)	Type(1~6)	Data(1~100)	Date/Time(19)	CR/LF
--------	--------------	-----------	-------------	---------------	-------

그림 9. 마이크로모듈-미들웨어간 센서데이터 패킷 구조
Fig. 9. The Packet Structure of Sensor Data between Micro Module and Middleware

패킷은 가변 길이 특성을 가지며 각 필드는 콤마(,) 로 구분된다. 데이터의 시작은 3바이트의 CMD 필드로 구분 가능하며, 끝은 캐리지 리턴(CR)과 라인피드(LF) 문자로 구분 가능하다. 패킷을 수신한 서버측 미들웨어는 그림 9와 같이 데이터를 식별하여 DB에 저장한다.

4.3 모니터링 애플리케이션

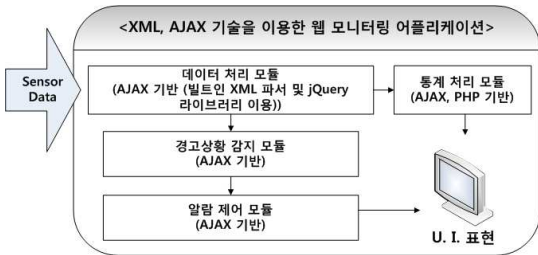


그림 10. 모니터링 애플리케이션의 모듈 구성
Fig. 10. The Module Structure of Monitoring Application

사용자 브라우저에서 구동되는 웹 모니터링 애플리케이션은 AJAX를 이용하여 관제 서버에서 데이터(DB 및 파일시스템에서 데이터를 추출하여 구성된 XML 및 DTD 데이터)를 가져오고, 이를 각 처리를 담당하는 세부 모듈로 넘겨주는 역할을 담당한다. 세부 모듈의 구성은 그림 10과 같다.

4.4 데이터 수신 절차

관제 서버에서 데이터를 가져오는 모든 과정은 데이터 처리 모듈이 담당하며, 경고상황 감지 모듈과 통계 처리 모듈에서 수신된 데이터를 전달받는다. 알람 제어 모듈은 경고상황 감지 모듈에서 경고상황이 발생할 경우 동적으로 로드된다. 각각의 모듈들은 독립적으로 UI에 접근하여 자신이 담당하는 UI 부분을 갱신한다.

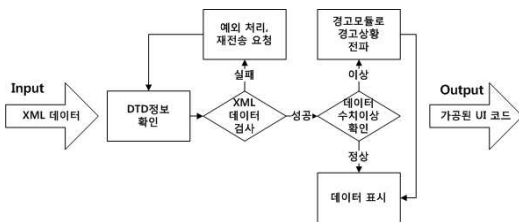


그림 11. 모니터링 애플리케이션의 데이터 수신처리 절차
Fig. 11. The Data Receiving Procedure of Monitoring Application

모니터링 애플리케이션의 데이터 수신 절차는 그림 11과 같이 왼쪽에 위치한 모니터링 서버로부터 XML 형태의 센서 데이터를 수신한 후 헤더부분에 내장된 DTD 정보가 있는지 판단하여 이를 바탕으로 수신한 XML 데이터 전체를 검사한다. 검사에 실패할 경우 예외 처리를 통해 현재 화면에 표시되는 센서데이터 수치의 업데이트 작업을 일단 보류하고, 미리 설정되어 있는 갱신 시간보다 빠르게 재전송을 요청한다. 검사에 성공하면 해당 데이터의 수치 중 이상점이 있는지(예: 화재상황 감지) 확인한 후, 문제가 있으면 경고모듈로 경고상황을 전파하여 관리자가 이메일 혹은 SMS를 통해 알림을 받을 수 있도록 한다. 이후 화면에 표시되는 센서데이터 수치 및 경고상황을 새 값으로 갱신한다.

4.5 데이터베이스 연동

본 연구에서는 DTD 데이터를 데이터베이스에 저장된 필드 정의 정보를 바탕으로 PHP 스크립트를 이용하여 동적으로 생성하여 제공한다는 특징이 있다. 데이터베이스는 MySQL Workbench 소프트웨어를 이용하여 그림 12와 같이 설계되었다. 센서 노드 하드웨어 정보를 저장하는 SensorNode 테이블과 응용 가능한 센서의 정보를 저장하는 Sensor 테이블, 상기 두 테이블간의 n:n 연결 정보를 저장하는 AttachedSensorList 테이블, 미들웨어 S/W에서 수집된 센서데이터가 저장되는 SensorData 테이블로 구성된다.

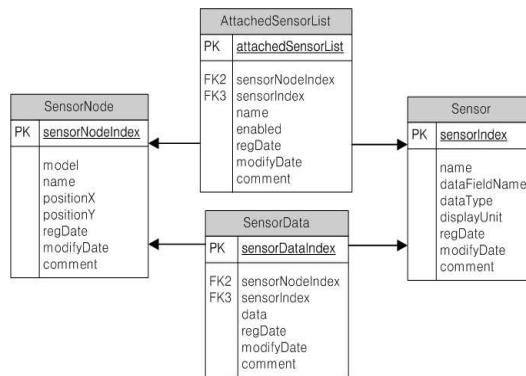


그림 12. USN 시스템의 데이터베이스 구성도
Fig. 12. The Diagram of Database Modeling

시스템이 DTD와 XML 구조를 자동 생성할 수 있으므로, 관리자는 SensorNode 테이블에 저장된 센서노드 하드웨어 정보와 Sensor 테이블에 저장된 각 센서의 정보를 AttachedSensorList 테이블과 연결, 조합하여 별도의 프로토콜 수정 없이도 센서노드에 장착되는 센서의 종류를 클릭이

나 간단한 정보 입력만으로 동적으로 조정할 수 있다.

4.6 모니터링 화면

DTD 자동 생성을 검증하기 위한 실제 웹 모니터링 시스템 소프트웨어는 그림 13, 14와 같이 구현하였다. 구현된 정보 시스템은 이미지 센서의 영상 데이터와 관리자 설정에 따른 센서별 텍스트 데이터 모니터링 기능을 포함하고 있으며, 온도가 영하(-)로 떨어지거나 습도가 20% 밑으로 떨어지게 되면 경고 이미지가 표시된다.

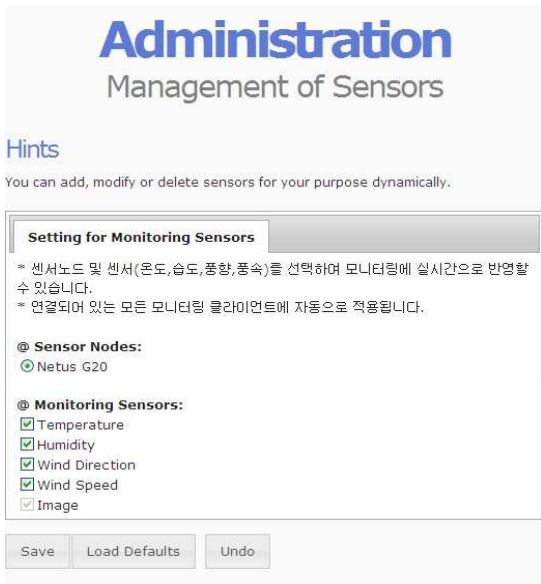
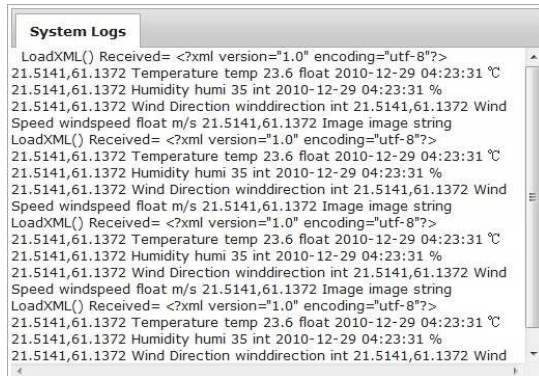


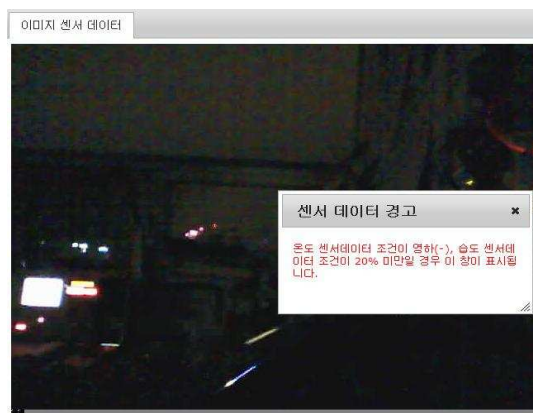
그림 13. 모니터링 대상 센서 항목 동적 수정 페이지
Fig. 13. The Sensors Management of Monitoring Server

구현된 모니터링 시스템은 사용자가 웹 브라우저에 시스템 주소를 입력함으로써 접속할 수 있다. 화면에 위치한 버튼이나 스크롤바 등 구성요소를 클릭 및 터치하여 추가 기능을 사용할 수 있다.

위로부터 좌측에 위치한 이미지 센서 데이터 셀에서 모니터링 대상 영상을 볼 수 있으며 우측의 기타 센서 데이터 셀에서 현재 현장의 온도 및 습도, 기타 센서 상황을 실시간으로 모니터링할 수 있다. 아래에 위치한 세 개의 버튼은 순서대로 모니터링 대상 센서 항목을 동적으로 수정할 수 있는 관리자 모드와 XML 및 DTD 원본 정보를 표시할 수 있는 기능이다. 맨 아래에 위치한 셀에서는 웹 모니터링 시스템의 데이터 처리와 오류 복구 등 활동 기록을 열람할 수 있다.



(a) System Log



(b) 센서 데이터 경고

그림 14. 모니터링 서버
Fig. 14. The Monitoring Server

“모니터링 항목 변경..” 버튼을 클릭하거나 터치하면 그림 13과 같은 모니터링 대상 센서 항목 설정화면이 나타나게 된다. 현재 모니터링 세션뿐만 아니라 연결되어진 다른 모든 클라이언트에 영향을 주게 되며, 모니터링에 사용되어지는 DTD 구조와 XML 구조를 실시간으로 수정하게 된다.

경고 기능이 원활히 동작하는지 실험하기 위해 온도센서의 범위가 영하(-)로 떨어지거나 습도가 20% 밑으로 떨어지게 되면 그림 14와 같은 센서 데이터 경고창이 표시되도록 설정하였다.

V. 성능평가

5.1 환경 설정

성능 평가 시뮬레이션을 수행하기 위해서, 표 2와 같이

USN 센서노드 및 게이트웨이 역할을 수행하는 ARM 기반 마이크로모듈은 Netus G20 모델을 사용하였다[9].

표 2. 실험 환경 구성 하드웨어 사양
Table 2. The Hardware Specification for Experiment

명칭	사양
USN 센서노드 및 게이트웨이	Built on the Atmel ARM9 @ 400Mhz CPU module Netus G20-L
온도/습도 센서	SHT71 Humidity and Temperature Sensor
Webcam	Logitech WEBCAM C910 FULL HD 1080P 2XSTEREO MICS/10MP
DB, Web 서버	Intel x86, Dual core, 2GB RAM, 160GB HDD
모니터링 PC	Intel x86, Dual core 1.6GHz, 1GB RAM, 80GB HDD

여기에 입출력 인터페이스를 이용하여 온도/습도 센서와 USB 기반 웹캠을 장착하였다. DB, Web 서버는 인텔 x86 기반 데스크톱 시스템으로 Intel사의 듀얼코어 2.6GHz 중앙처리장치와 2GB의 RAM, 160GB의 하드디스크로 구성되었다. 모니터링 PC는 Intel사의 듀얼코어 1.6GHz 중앙처리장치와 1GB의 RAM, 80GB의 하드디스크를 가진 노트북 PC를 사용하였다.

그림 1에서와 같이 Netus G20 마이크로모듈과 DB, Web 서버는 인터넷망으로 연결되어 있으며, 서버와 모니터링 PC간의 연결은 크로스(Cross)된 100Mbps 이더넷 인터페이스로 구성하였다.

DTD 자동 생성 기반 USN 모니터링 시스템을 검증하기 위한 실험은 클라이언트에서 모니터링 시스템에 접속하여 스크립트의 구문별 처리속도와 정상동작 여부를 측정하는 방식으로 진행하였다. 실제 데이터 전송 및 오류검출, 재전송 요청과정에서 소요되는 시간을 측정하기 위하여 어플리케이션 코드 안에 실행시간 측정용 코드를 포함 하였으며, 추가적으로 Microsoft Internet Explorer Developer Tools 응용프로그램을 사용하였다.

5.2 XML 구조와 DTD 자동 생성 이용 XML 처리 속도 비교

표 3. 기존 XML 구조와 DTD를 이용한 XML 구조의 처리속도 평균치 차이(단위: ms)
Table 3. The Average Processing Time between XML and DTD+XML based Structure(ms)

구분	XML	DTD정보 포함 XML	차이
데이터 수신	3.13	9.38	+6.25
데이터 검증	0.00	4.69	+4.69
데이터 표시	3.13	1.56	-1.57

기존 XML 구조와 DTD를 포함한 XML 구조의 처리속도를 비교하기 위해, 데이터 구조를 임의로 설정하여 전송하였다. 실험에 사용된 XML 데이터 구조는 센서의 물리적 위치, 센서명, 센서 종류, 센서 데이터값 및 데이터타입 지정, 수신 시간 필드로 이루어져 있다. 표 3에 의하면 기존의 별도 추가 처리과정 없이 바로 데이터를 수신하는 XML 구조와 DTD를 포함한 XML 구조의 처리속도 차이를 측정할 결과 평균치를 기준으로 DTD를 포함한 XML 구조가 더 느린 것으로 나타났다. 하지만 센서 데이터 필드 및 DTD 검증 정보를 자동으로 생성 가능한 부가기능을 포함한 것에 비해, 상대적으로 속도의 지연이 크지 않은 편이다.

5.3 데이터 깨짐 상황에서의 비교

일시적인 깨짐 현상을 포함한 데이터로부터 기존 XML 구조와 DTD를 포함한 XML 구조의 재전송 요청 후 처리속도를 비교하기 위해, 데이터 구조를 임의로 설정하여 전송하였다.

설정된 데이터는 XML에서 name 필드의 데이터를 생성 처리하던 중 데이터베이스 오류 및 웹 서버 오류, 네트워크 오류 등 다양한 문제로 인해 정상적인 처리가 도중에 중단된 상태로 클라이언트까지 도달하는 상황을 가정하였다.

정상 데이터 사이에 깨진 데이터를 한 차례만 삽입하여 전송한 후 이에 대한 오류 검출 및 재전송 요청을 통해 정상적인 데이터로 복구하는데 걸리는 평균 시간을 측정할 결과, 기존의 XML 기반 웹 모니터링 시스템은 깨짐 여부를 미리 검출해내지 못하므로 잘못된 XML 데이터가 Parser 및 UI 표현 단계까지 들어간 후 예외 처리를 통해 재전송을 요청하기 때문에 446.6ms의 시간이 소요되었다. 제안한 시스템은 깨짐 여부를 미리 검출해내서 바로 재전송을 요청하였고 Parser가 동작하기 전에 올바른 데이터를 수신하기 때문에 검증한 결과 402.8ms의 시간이 소요되었다. 이로 인해 43.8ms의 시간차이로 오류 발생 상황에서 제안된 시스템이 오류 복구율이 더 좋은 것으로 나타났다.

기존의 USN 웹 모니터링 시스템은 XML 데이터의 깨짐이 발생할 경우 별도의 재전송 요청 과정이 없으므로 실제로 정상화에 소요되는 시간은 표 4의 측정치보다 더 길어진다[10].

표 4. 데이터 깨짐 상황의 오류복구 평균시간(단위: ms)
Table 4. The Average Recovery Time of Corrupted XML(ms)

	XML	DTD정보 포함 XML	차이
재전송 및 정상 데이터 수신까지 소요된 시간	446.6	402.8	-44.8

5.4 오류 복구 시간 비교

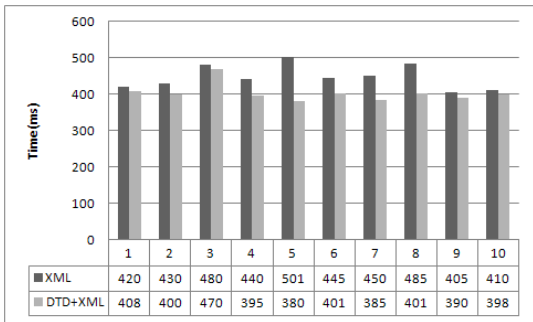


그림 15. XML기반 시스템과 제안한 시스템의 오류복구 시간
 Fig. 15. The Failure Recovery Time between XML and XML+DTD System

그림 15는 XML기반 시스템과 제안한 시스템의 오류복구 시간을 검증하기 위하여 USN으로부터 매 초마다 약 317바이트의 센서 데이터를 갱신하는 재해 모니터링 서비스 환경에서 센서 스트림 데이터 317바이트를 연속하여 10회 전송하면서 측정된 결과이다.

기존의 XML 구조 기반 모니터링 시스템에 비해, 새로 제안한 시스템은 데이터를 수신하여 출력에 걸리는 시간이 약 9.37ms 정도의 차이로 느렸으나 오류 발생 상황의 복구에서는 44.8ms 더 빠른 양상을 보였다. 실시간적인 깨짐 검출과 센서 데이터 필드의 실시간 업데이트 등 다양한 기능을 부가적으로 제공할 수 있음에도 처리속도상의 손실이 미세하다는 점은 장점으로 볼 수 있다.

VI. 결 론

모니터링 시스템의 오류 복구능력을 개선하기 위하여 DTD 정보를 자동 생성하는 기법을 제안하였다. 제안된 시스템은 데이터베이스의 노드 및 센서 정보에 따라 자동으로 생성되는 DTD 정보와 XML 구조 설계를 이용하여 복잡한 소프트웨어 수정 절차 없이도 스스로 필드를 확장하고 그에 대한 깨짐 등의 문제점을 검사하여 재전송을 요청하는 모니터링 시스템이다.

시스템을 운영하는 관리자가 전문적인 소프트웨어 지식이 없어도 내부적으로 사용되는 데이터 구조 및 최종 모니터링에 표현될 화면 구성 항목을 필요에 따라 동적으로 구성할 수 있으며, 부가적으로 USN 웹 모니터링에서 문제점으로 지적되던 안정성과 확장성 문제를 해결할 수 있다.

이 기법이 USN 모니터링 시스템에 도입되면 우수한 확장성에 의해 전반적인 시스템 유지보수 비용절감과 전송오류 복구에 효과적일 것으로 기대된다. 향후 하드웨어에 대한 전문 지식이 없이도 센서노드에 대한 센서 추가 및 제거, 네트워크 구성 및 전반적인 유지보수가 가능하기 때문에 USN 분야에 파급 효과는 극대화될 것이다.

참고문헌

- [1] Ian F. Akyildiz, Tommaso Melodia, Kaushik R. Chowdhury, "A survey on wireless multimedia sensor networks Original Research Article," Computer Networks, Vol. 51, no. 4, 14, pp. 921-960, March 2007
- [2] Daniela Ballari, Monica Wachowicz, Arnold K. Bregt, Miguel Manso-Callejo, "A mobility constraint model to infer sensor behaviour in forest fire risk monitoring", Computers, Environment and Urban Systems, 9 August 2011
- [3] L.C. Carvalho, S.O. Bernardo, M.D.M. Orgaz, Y. Yamazaki, "Forest Fires Mapping and Monitoring of current and past forest fire activity from Meteosat Second Generation Data", Environmental Modelling & Software, Vol. 25, no. 12, pp. 1909-1914, December 2010
- [4] Sudipta Bhattacharjee, Prमित Roy, Soumalya Ghosh, Sudip Misra, Mohammad S. Obaidat, "Wireless sensor network-based fire detection, alarming, monitoring and prevention system for Bord-and-Pillar coal mines", Journal of Systems and Software, 16 September 2011
- [5] Joachim Schnier, "XML Introduction", Flash XML Applications, pp. 3-9, 2008
- [6] Dirceu Cavendish, K. Selçuk Candan, "Distributed XML processing: Theory and applications", Journal of Parallel and Distributed Computing, Vol. 68, no. 8, pp. 1054-1069, August 2008

[7] S. Abiteboul, G. Gottlob, M. Manna, "Distributed XML design Original Research Article", Journal of Computer and System Sciences, Vol. 77, no. 6, pp. 936-964, November 2011

[8] Ali Mesbah, Arie van Deursen, "A component- and push-based architectural style for ajax applications", Journal of Systems and Software, Vol. 81, no. 12, pp. 2194-2209, December 2008

[9] "Netus G20", ACME Systems, <http://www.acmesystems.it/?id=NETUSG20>

[10] M. Benghanem, "Measurement of meteorological data based on wireless data acquisition system monitoring", Applied Energy, Vol. 86, no. 12, pp. 2651-2660, December 2009

[11] Won-Ik Park, Young-Kuk Kim, "A Sensor Data Management System for USN based Fire Detection Application", Journal of the Korea society of computer and information, Vol.16 no.5 pp. pp.135-145, 2011

[12] YH Han, "A Study on Monitoring of Bio-Signal for u-Health System", Journal of the Korea society of computer and information, Vol.6 no.3 pp.9-15, 2011



권 기 현

1993년 : 강원대학교 전자계산학과 이
학사.
1995년 : 강원대학교 전자계산학과 이
학석사
2000년 : 강원대학교 컴퓨터과학과 이
학박사
2002년~현재 : 강원대학교 전자정보
통신공학부교수
관심분야 : 임베디드시스템, 패턴인식
Email: kweon@kangwon.ac.kr



유 명 한

2009 : 강릉원주대학교 컴퓨터공학과
공학사
2011 : 강원대학교 전자공학과 공학석사
현 재 : 파란 소프트 팀장
관심분야 : USN, 임베디드시스템
Email : greatynh@gmail.com

저 자 소 개



남 시 병

1979 : 단국대학교 전자공학과 공학사
1982 : 단국대학교 전자공학과 공학석사
1994 : 단국대학교 전자공학과 공학박사
현 재 : 강원대학교 전자정보통신공학부
교수
관심분야 : 임베디드 시스템, 센서네트
워크, 인터페이스, 패턴인식
Email : sbnam@kangwon.ac.kr