

웹을 기반으로 하는 자동화 방법에 관한 연구

박종락* · 정원** · 박관희***

Abstract

Over the past decade, the unprecedented internet revolution has dramatically altered many core aspects of our daily business and personal lives. Many of these fundamental changes have completely transformed traditional communication mechanism, boosted productivity, opened up new marketing opportunities, and given birth to entirely new applications in the world of eCommerce and eBusiness. But Many web applications doesn't connect with real-world device such as lights, pumps, and Motors available over the Internet. In this paper We introduce XML and presents way it can be used with I/O system.

keyword : e-business, xml, I/O system,

I. 서론

전자상거래는 기존의 상거래를 전자적인 수단으로 대체하기 위한 것으로 국가사회의 새로운 인프라로 자리잡아 가고 있다. 기업과 소비자간 (B2C) 전자상거래는 기업이 제공하는 상품이나 서비스를 인터넷/웹상에서 소비자가 구매할 수 있도록 제공하는 것으로, 급속히 활성화되어 보편적인 것이 되었다. 반면에, 기업간 (B2B) 전자상거래는 독립된 기업간에 상거래 정보를 전자적인 수단으로 교환하여 업무를 처리하는 것으로, 아직 초기단계에 있다. 기업간에 교환되는 문서 등과 같은 업무 정보의 표준화가 생산성과 정확성을 좌우한다고 할 수 있으며, 전자상거래 시대의 국제 경쟁력의 요체이다. 이제 정보 교환을 XML로 처리하는 일은 자연스러운 것이 되고 있습니다. 또한 기업 내부의 정보화 시스템과 웹 사이트를 연결하여 거래를 자동화 하기 위한 XML 미들웨어 제품도 등장하고 있다. 하지만 완전한 의미의 자동화는 이러한 미들웨어들과 실제 명세서와 관련된 부품을 직접 연결할 수 있는 하드웨어 기능의 XML 데이터 인식 시스템이 필요한데 이를 통한 웹과 창고의 부품을 연결 제어할 수 있어야 한다. 이러한 제어는 Either I/O를 통해서 가능하다. 따라서 본 연구는 XML을 이용한 Either I/O에 제어에 관한 연구이다.

* 대구대학교 산업공학과 박사과정

** 대구대학교 자동차·산업·기계 공학부 교수

*** 대구대학교 금융·회계·보험금융학부 교수

II. 관련 기술

Either I/O 시스템이란 실세계의 아날로그, 디지털 또는 시리얼데이터를 발생 시키는 Ethernet 네트워크나 인터넷 상의 어떠한 장비들로부터 이를 이용하여 XML(Extensible Markup Language) 파일을 사용하거나 만드는 것을 말한다. 다양한 XML 제품군들을 이용한 I/O시스템들로부터 모니터나 기록을 위한 데이터 베이스 XML파일을 만들 수 있어야 한다. 아래의 <그림1>은 XML제품군의 적용들로부터 Ethernet I/O 시스템으로 직접 Ethernet 연결을 통해 연결한 연결도인데 이러한 시스템에는 중간 연결용 컴퓨터나 미들웨어 또는 기타 다른 요소들이 필요가 없다.



기업데이터베이스 Ethernet I/O 시스템

<그림 1> Ethernet I/O 시스템 연결도

1. XML

XML은 메타 데이터를 간단하게 표현하며 사람이 쉽게 읽을 수 있는 태그를 사용하는 표준이다. XML은 산업의 폭넓은 범위에서 사용되며, XML은 데이터를 교환하기에 매우 좋은 프로그램이다. 마이크로 소프트, 오라클, IBM, 선마이크로 시스템과 같은 기타 주요 회사들은 데이터 베이스나 기타 다른 소프트웨어에 적용을 위해 XML에 대한 전적인 지원을 하고 있다. 그 한 예로서 마이크로 소프트사의 BizTalk를 들 수 있다.

XML파일은 태그데이터를 포함하는 단순하며 구조화된 ASCII 텍스트 파일이다. 태그의 정보에는 브라우저에서 어떻게 나타내는가에 대한 정보만 들어 있는 HTML 파일과는 다르며, XML은 사용자가 정의하고 기술한 개인적인 데이터 요소의 태그를 가지고 있다. 데이터는 계층적인 구조를 가지고 있으며 태그와 계층구조는 적용할 때에 XML 파일에서 개별적인 사용이나 인식을 할때에 서로 밀접한 상관성을 만들 수 있다.

XML 파일은 쉽게 읽을 수 있어야 한다. 아래 <그림2>는 공장 기계 관한 정보상태를 나타내는 XML 파일의 예이다.

```
<?xml version="1.0" standalone="yes"?>
<Workstation>
  <MotorStatus> On</MotorStatus>
  <MotorTemperature Units="F">450
  </MotorTemperature>
  <RPM>3000</RPM>
  <TankPressure Units="PSI">200
  </TankPressure>
</Workstation>
```

<그림 2> XML을 이용한 기계 정보 표현

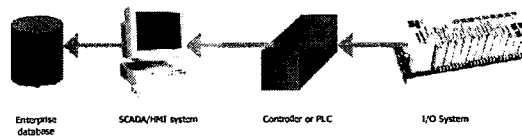
위의 <그림 2>에서 첫 번째 라인에서 나타내는 선언문은 XML파일 버전 1.0 기준에 작성 되었음을 나타내며 “standalone” 파라미터는 다른 외부 문서를 참조하지 않았다는 것을 나타낸다. 그리고 아래의 라인들은 실제 데이터 요소를 포함하고 있는데 <Workstation>는 루트 요소를 나타내며, 루트 요소 안에 <MotorTemperature>와 <RPM> 과 같은 종속적인 태그들이 있다.

2. XML파일 접근

XML의 가장 근본적인 목적과 가장 효과적인 사용은 어떠한 확대되고 있는 XML군들의 적용들에 대한 데이터 교환을 쉽게 하는 것이다. 예를 XML군의 기업용 데이터베이스는 직접적으로 원격 I/O시스템에 접근과 온도값과 또는 다른 현실 세계의 값들을 데이터 베이스로 의 반영이 가능해야 한다. 또한 XML군의 적용은 또한 HTTP서버에서 원격 파일을 열수가 있어야 하며, 파서가 가능하며 포함된 XML파일의 조작이 가능해야 한다. 또한 데이터 베이스의 작업을 위해서 마이크로 SQL 서버 2000이나 오라클 8i/9i 데이터 베이스를 사용하면 더욱 효과적이다.

이러한 사용자의 적용부분과 I/O시스템의 직접적인 연결은 현실세계의 데이터를 얻기위한 간단한 접근법은 기존의 전통적인 미들웨어나 다른 요소들을 이용한 접근법 보다 훨씬 쉬우며 효과적이다.

<그림 3>은 전통적인 시스템의 연결도로서 미들웨어로 PLC나 컴퓨터 시스템을 통한 연결로서 그 복잡하고 제어가 매우 어려웠다.



<그림 3> 전통적인 I/O시스템의 연결도

3. I/O시스템에서의 XML 사용법

데이터 베이스 상의 I/O 포인터 데이터를 이용하여 모니터링하기 위해서는 웹브라우저나 기타 XML군의 적용한 XML파일을 만들기 위해서는 현재의 I/O 포인트의 메모리 맵을 이용하여야 하며 이 파일을 I/O 시스템에 저장 가능해야 한다. 또한 이 파일은 I/O시스템의 IP주소와 URL에 있는 파일 명을 이용하여 원격 접속 또한 가능해야 한다. XML파일은 I/O시스템의 펌웨어로 웹페이지를 위해 생성이 가능해야 하며 이 파일은 PC나 네트워크 상의 다른 PC에도 저장 가능해야 한다. 또한 이미 정의된 XML파일을 이용하여 I/O시스템의 환경설정이나 변경이 가능해야 하며 또한 자신만의 XML 파일을 생성할 수 있어야 하며 네트워크를 통하여서 I/O 시스템의 펌웨어에 파일의 업로드가 가능해야 한다.

설정된 포인트 값들을 인터넷 익스플로러를 이용하여 디스플레이가 가능하며 또한 변화된 I/O포인트 값의 상태도 XSL(Extensible Stylesheet Language)를 이용하여 PC나 네트워크상의 다른 서버에도 표현이 가능해야 한다. <표 1>은 I/O시스템에서 XML상에 데이터를 사용하기 위한 메모리 맵의 파라미터를 나타낸다.

<표 1> 메모리 맵을 위한 파라미터

파라미터	설 명
adr	메모리 맵의 주소로 8개의 16진수로 표현한다. 예)adr=F0800040
fmt	수치 데이터의 양식을 나타낸다. fmt=%08x 8개의 16진수로 데이터로 앞에는 0으로 채워진다. fmt=%3.1f 데이터는 소수점 포인터로 3개의 숫자로 하나의 숫자가 소수점 아래에 위치한다. fmt=%d 정수형 데이터를 말한다. fmt=%u Unsigned 정수형 fmt=%s 문자열 데이터 fmt=ip IP주소 데이터 fmt=%t 날짜와 시간데이터(YYYY-MM-DD HH:MM:SS:00)
onoff or on and off	on/off 또는 불린(Boolean) 상태를 나타낸다. 만약 onoff가 정의되어 있지 않다면 0은 off이고 그 외 0이 아닌 모든 값은 on을 나타낸다. 분리된 on과 off의 문자에는 슬래쉬(/)를 앞에 가지고 있다. onoff=ON/OFF 또는 on=on&off=off
bytes	바이트의 수를 나타내는데 유효 값은 1,2, 또는 4. 디폴트 값 4

4. XML 파일 생성

XML파일을 생성하기 위해서 아래의 단계를 따라야 한다.

① 파일의 포함되어야 할 목록과 요소를 결정한다.

- 파일의 첫 번째 라인에는 "<?xml"로 시작한다. 이 문자는 매우 엄격하게 지켜져야 한다.
- 문자로 "<html"은 포함하지 않아야 한다.
- 파일 명은 .xml 확장자명을 가지고 있어야 한다.

② XML파일은 자신의 고유한 정의를 이용한 태그를 이용하여 만들며 포인트를 읽기 위하여 메모리 맵을 플러그인한다.

메모리맵을 플러그 인하기 위해서는 시작을 다섯 문자 "<!--#" 와 끝나는 문자 "-->" 으로 나타낸다.

파라미터의 구분은 앰퍼샌트(&)로 구분된다. 아래의 표는 메모리맵의 플러그 인을 위하여 사용할 수 있는 파라미터를 나타낸다.

다음 <그림 4>는 0포인트와 1포인트에 위치한 2개의 스위치값의 상태를 읽기 위해 메모리 맵 플러그 인을 사용한 XML파일의 예를 나타낸다. 메모리 맵의 플러그 인은 하나의 XML파일에 128번까지 사용할 수 있다.

```

<?xml version="1.0" standalone="yes"?>
<SwitchStatus>
  <Switch>
    <Name> A </Name>
    <State><!--#MemMap
adr=F0800000&onoff=On/Off--> </State>
  </Switch>
  <Switch>
    <Name> B </Name>
    <State><!--#MemMap
adr=F0800000&onoff=On/Off--> </State>
  </Switch>
</SwitchStatus>

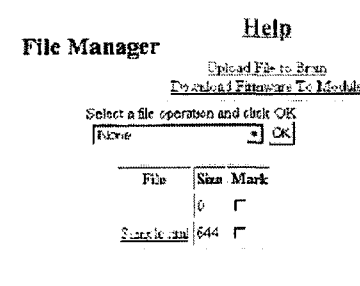
```

<그림 4> 메모리 맵 플러그 인을 이용한 XML파일

5. I/O 시스템에 XML 파일 전송

XML파일이 완성되면 파일을 I/O시스템의 펌웨어 부분으로 파일을 전송을 해야하며, 파일 전송을 위해서 IP 주소를 알아야 한다. 그리고, I/O시스템을 위한 웹페이지의 위치도 알아야 한다.

- ① I/O 시스템이 같은 네트워크 상에 있는 경우 인터넷 익스플로러에서 I/O 시스템이 있는 URL의 웹페이지를 연다.
- ② 페이지가 나타나면, I/O 시스템의 IP를 입력하여 연결한다.
- ③ 파일 관리를 클릭한다.
- ④ 파일관리 페이지가 나타나면 I/O 시스템에 전송하기를 눌러준다.



<그림 5> 파일 관리 웹페이지

- ⑤ 정의된 파일 필드에서 I/O시스템에 나타내고자 하는 파일명을 적어준다. 파일명은 공백이 없어야 하며 확장자 명이 xml로 되어 있어야 한다.
- ⑥ 원본 파일 필드에서 컴퓨터에 있는 파일명의 경로를 적어준다.
- ⑦ 전송 버튼을 클릭한다.

위와 같이 하면 XML 파일은 I/O 시스템에 전송된다.

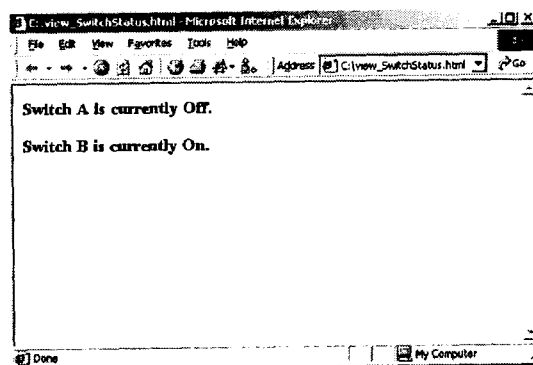
6. 웹에서 XML파일 보기

I/O 시스템에 로드된 XML파일을 인터넷 웹브라우저에서 볼수가 있는데 포맷된 형식이나 포맷

되지 않은 형식이든 가능하다. 포맷되지 않은 XML 파일을 보기 위해서는 <그림 5>의 파일 관리 화면에서 클릭하면 된다. 또 다른 방법은 인터넷 익스플로러 주소 필드에서 URL을 적어 주면 된다. SwitchStatus.xml파일을 IP 주소 203.244.154.64를 가진 I/O시스템에서 보고자 할때는 URL 주소 필드에 http://203.244.154.64/SwitchStatus.xml 을 입력하면 볼수가 있다. 포맷되지 않은 XML 데이터는 아래 <그림 6>과 같이 나타나며 포맷 되어 나타나는 경우에는 <그림 7>과 같이 나타난다.

```
<?xml version="1.0" standalone="yes"?>
<SwitchStatus>
  <Switch>
    <Name>A </Name>
    <State>off </State>
  </Switch>
  <Switch>
    <Name>B </Name>
    <State>on </State>
  </Switch>
</SwitchStatus>
```

<그림 6> 포맷되지 않은 XML 파일



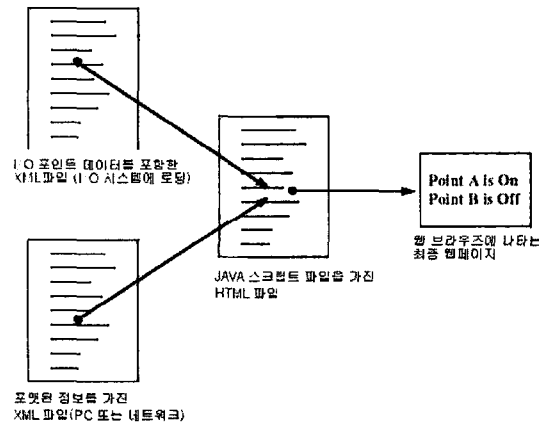
<그림 7> 포맷된 XML 파일

<그림 7>과 같이 인터넷 익스플로러에서 표현하기 위해서는 PC나 서버상의 컴퓨터에 저장되어 있는 XSL(extensible Style Language)파일을 이용하였다. XML파일에는 단지 데이터만 들어 있으며 웹브라우저에서 어떻게 나타내어야 할지에 대한 포맷에 대한 정보는 들어 있지 않다. XSL에는 웹 상에서 나타낼 XML파일에 대한 포맷에 대한 지시가 들어 있다. 웹상에서 포맷된 형태로 XML 데이터 파일을 나타내기 위해서는 적어도 3개의 파일이 요구된다.

- I/O시스템에 XML 파일이 있어야한다.
- PC나 웹서버 네트워크 상에 1개 이상의 XSL파일이 있어야 한다.
- 하나의 HTML파일이 PC나 네트워크 상의 서버에 있어야 한다.

HTML 파일은 웹브라우저 상에서 열수가 있으며,<그림 8>과 같이 파일을 웹브라우저에서 열면

데이터를 가진 XML파일에 XSL 포맷 양식과 조합된 자바 스크립트를 통해서 웹브라우저에 나타낼 수 있다.



<그림 8> XML, XSL, HTML 파일의 연결도

<그림 9>는 PC나 다른 웹서버 상에서 저장되어 데이터의 결과를 보기 위한 소스파일로 XML 로딩 부분과 XSL 파일 로딩, 그리고 보기위한 파일의 변환으로 나누어져 있다. 본 파일은 HTML 코드에 자바 스크립트로 구현하였다.

```

<html>
<body>
<script language="javascript">
//Load XML
var xml= new ActiveXObject("microsoft.XMLDOM")
xml.async=false
xml.load(http://203.244.154.64/SwitchStatusData.xml")
//Load XSL
var xsl= new ActiveXObject("microsoft.XMLDOM")
xsl.async=false
xsl.load("SwitchStatus.xsl")
//Transform
document.writ(xml.transformNod(xsl))
</script>
</body>
</html>

```

<그림 9> StatusSwitch.html 파일

아래 <그림10>은 XML데이터에 대한 정보를 가지고 있으며 데이터를 웹 브라우저에 나타내기 위해서 HTML 코드로 작성하였다. 이 파일 역시 PC에 저장되어 있다. "SwitchStatus/Switch"는 SwitchStatus의루트 엘리먼트를 나타내며 그 다음으로 "Switch"태그가 I/O 시스템에 저장되어 있다는 것을 나타낸다. "Name"과 "Status"는 I/O시스템의 XML 파일의 요소를 나타낸다.

```

<? xml version ="1.0"?>
<xsl.stylesheet
xmlns:xsl="http://www.w3.org/TR/WD-xsl">
<xsl:template match="/">

<html>
<body>
<xsl:for-each select="SwitchStatus/Switch">
<h3> Switch <xsl:value-of select="Name" /> is
Currently <xsl:value-of select="Status" />.</h3>
</ xsl:for-each>
</body>
</html>

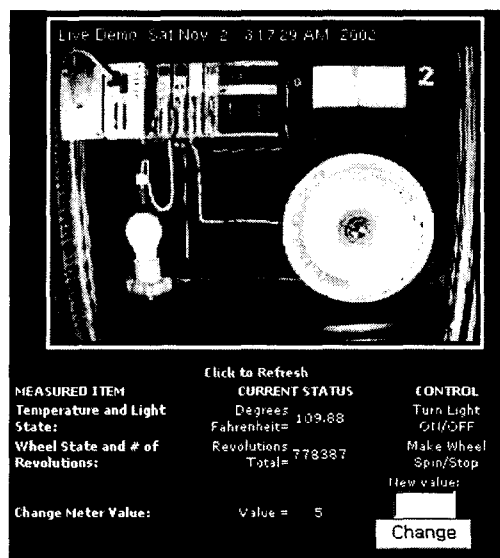
</xsl:template>
</xsl:stylesheet>

```

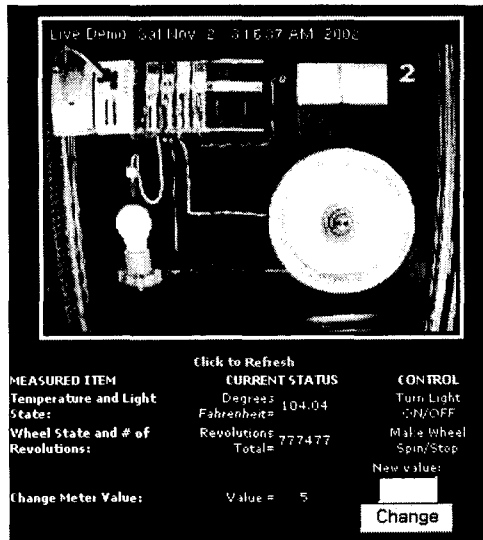
<그림10> StatusSwitch.xml 파일

Ⅲ. 실험 결과

<그림 11>은 <그림 12> 앞에서의 언급한 방법들을 이용하여 Opto 22 SNAP Ethernet I/O 시스템을 이용하여 구현한 예이다. 테스트 결과 인터넷의 속도에 따라 차이가 조금씩 있지만 원격 제어가 가능하다는 것을 알 수 있었다. <그림 11>온도와 전구를 off 하고 원반도 정지 시킨 경우이다. 화면의 우측 하단부에 있는 컨트롤 상태를 클릭하여 변경시켰을 때 <그림 12>와 같이 전구에 빛이 들어오며, 온도의 변화와 원반의 움직임을 나타내고 있다.



<그림 11> OFF 상태의 실험 환경



<그림 12>On 상태의 실험 환경

IV. 결론

웹 브라우저를 통한 실세계의 증가는 전자 상거래의 필요성이 증가 하면 할수록 그 필요성이 제기 되고 있다. 하지만 많은 전자 상거래와 인터넷의 개발 범위가 웹 범주 중심의 개발만을 지속 하여 왔다. 또한 일부 개발된 시스템의 경우를 보면 PC와 미들웨어를 통해서 특별히 개발된 소프트웨어만을 이용하여 연결이 가능하였다. 본 연구는 현재 새로운 웹프로그래밍 언어인 XML을 통하여 웹브라우저와 현실 세계의 데이터 직접적으로 연결하고 데이터를 형성할 수 있음을 보여주고 있다. 또한 실 세계의 데이터를 직접적으로 제어하여 그 효용성이 매우 높다고 할 수 있다. 앞으로 본 연구를 바탕으로 전자 상거래와 공장 자동화를 손쉽게 연결할 수 있으므로 이 두분야를 접목하는데 매우 훌륭한 가교 역할을 할 것으로 기대 된다.

V. 추후 연구과제

웹을 기반으로 하는 시스템은 그 반응이 일정하지 않으며 LAN의 상태에 따라 많이 변한다. 따라서 안정적인 반응속도와 자동화를 위한 속도의 개선 부분에 더 많은 연구가 필요할 것으로 예상된다.

참고 문헌

- [1] S. Abitebou, P. Buneman, D. Suciu, Data on the Web, Form Relations to Semistructured Data and XML, Morgan Kaufman, Los Altos, CA, 2000.
- [2] Extensible Markup Language(XML) 1.0, W3C Recommendation 10 February 1998, <http://www.w3.org/TR/REC-xml>.

- [3] I. Macherius, Java applications: XQL Language and Persistent W3C-DOM, <http://www.oasis-open.org/cover/Macherius19990329.html>
- [4] P. Buxmann, L. Martin Diaz, E. Wustner: XML-based supply chain management-as SIMPLEX as it is -, in Proceedings of the 35th Hawaii International conference on System Science(HCSS 2002), 2002.
- [5] B2B 전자 상거래@ XML, 김형도. 2000