

XML과 XSLT를 이용한 레거시 기어 설계 시스템의 확장에 관한 연구

정태형(한양대)*, 박승현(한양대 대학원)

Extension of legacy gear design systems using XML and XSLT

Tae Hyong Chong(Hanyang Univ.), Seung Hyun Park(Graduate School, Hanyang Univ.)

ABSTRACT

As computer-related technologies have been developed, legacy design systems have not been appropriate for new computing environment. Therefore, it is necessary that most of them are either modified or newly developed. However, this requires quite much amount of cost and time. This paper presents a method of extending legacy design system without modification using XML and XSLT. In order to apply the developed method, a good example of legacy design systems, AGMA gear rating system has been extended so as to be suitable for the distributed computing environment. An XML document for AGMA gear rating process is defined. It is transformed to the form of the input document of AGMA gear rating system by XSLT processor according to the transformation rules defined in the AGMA gear rating XSLT document. After that, AGMA gear rating system reads this input document and generates an output document in the server. These operations are automatically executed by the external legacy system controller without user interactions. Using these operations, AGMA gear rating web service has been developed based on SOAP and WSDL to provide the functions of legacy AGMA gear rating system through the distributed network. Any system or user can implement AGMA gear rating process independently to the platform type, without making it for oneself, by simply referring the AGMA gear rating web service via Internet.

Key Words : Design Methodology(설계 방법론), Gear(치차), Legacy System(레거시 시스템), XML(eXtensible Markup Language), XSLT(XML Stylesheet Language Transformation), SOAP(Simple Object Access Protocol), WSDL(Web Service Description Language), Web Service

1. 서론

컴퓨터의 도입과 함께 공학 설계 활동을 지원하기 위한 많은 시스템들이 개발되어왔다. 이러한 시스템들은 CAD, FEM 등의 분야와 같이 상용으로 제작되는 시스템에서 필요에 따라서 자체적으로 제작되는 시스템까지 그 종류와 형태가 매우 다양하다.

그러나 컴퓨터 관련 기술들이 발전함에 따라서 오래 전에 개발된 시스템들은 새로운 컴퓨터 환경에 맞지 않게 되었다. 이렇게 더 이상 개발이 되지 않는 오래된 시스템을 레거시 시스템(legacy system)이라고 한다. 일반적으로 레거시 시스템들은 새로 개발된 시스템에 비해서 사용이

불편하거나, 또는 변경되지 않으면 아예 작동하지 않기도 한다.

대부분의 상용 시스템의 경우 새로운 환경에 맞도록 계속 개발이 되면서 보다 더 강력한 기능을 제공해주지만, 이러한 시스템들은 일반적인 목적으로 사용되도록 개발되었기 때문에 실제 공학 설계 환경에서는 수많은 설계 지원 시스템들이 자체적으로 개발되었다. 그러나 이러한 시스템들은 오래된 컴퓨터 환경에 맞도록 개발된 것이 많아서 새로운 컴퓨터 환경에 맞도록 변경하거나 새로 제작하려면 상당한 비용과 시간이 필요하다.

이에 본 연구에서는 기존의 레거시 시스템을

변경하거나 새로 제작하지 않고도 인터넷 기반의 분산 환경으로 발전하고 있는 새로운 컴퓨터 환경에서 사용할 수 있도록 XML(eXtensible Markup Language)^[1]과 XSLT(XML Stylesheet Language Transformation)^[2]를 이용하여 레거시 시스템을 확장하는 방법에 대한 연구를 수행하고, 레거시 기어 설계 시스템을 대상으로 선정하여 개발된 방법론을 적용하였다.

2. 공학 설계에서의 XML

2.1 XML과 XSLT의 개요

XML은 1998년 W3C(World Wide Web Consortium)에서 SGML(Standard Generalized Markup Language)^[3]보다 간결하면서 HTML(Hypertext Markup Language)보다는 구조적인 마크업 언어 개발의 산물로서 제안된 메타 마크업 언어이다. XML의 가장 큰 특징은 HTML과는 달리 자신만의 태그를 정의할 수 있다는 것이다. 각 적용 분야에 맞는 태그를 정의함으로써 XML을 이용하여 복잡한 정보를 구조적으로 표현할 수 있고, 따라서 XML은 문서와 데이터 컨테이너의 특성을 모두 가진다.

XML은 기본적으로 텍스트 문서이므로 플랫폼과 시스템에 독립적이다. 또한 XML은 유니코드를 사용하므로 다양한 문자를 표현할 수 있고 언어에 중립적이다. 따라서 XML은 상이한 플랫폼, 시스템간의 정보 교환에 매우 유리하다.

DTD(Document Type Definition)^[1]는 XML 문서의 구조와 규칙을 정의해놓은 문서로서 XML 문서의 검증에 사용된다. DTD를 사용하면 컴퓨터 파서(computer parser)가 자동으로 XML 문서의 문법 등을 검증해주므로 시스템이 XML 문서를 사용할 때 발생할 수 있는 오류를 사전에 막을 수 있다.

XML은 자신만의 태그를 정의할 수 있으므로 한 분야에서도 정의하는 방법에 따라서 여러 가지 문서 형태가 나타날 수 있다. XSLT는 XML문서를 다른 문서 형태로 변환하기 위한 목적으로 개발되었다. XSLT를 이용하면 하나의 XML문서를 다른 형태의 XML 문서, HTML, 심지어는 복잡한 형태의 텍스트 문서로 쉽게 변

환할 수 있다. 이때 원본 XML문서는 전혀 변경되지 않고 겉보기 형태만 달라지기 때문에 여러 개의 XSLT 문서를 각각 적용하여 하나의 XML 문서를 다양한 형태로 나타낼 수 있다. 또한 XSLT는 XML문서의 한 형태이므로 XML의 특징을 모두 가지고 있다. 이렇게 XSLT를 활용하면 서로 다른 형태의 문서를 사용하는 시스템간의 정보 교환을 보다 용이하게 할 수 있다.

이와 같은 XML의 장점을 공학 설계에 적용하려는 연구들이 최근 들어 다양하게 시도되고 있다^{[4],[5]}.

2.2 SOAP

본 연구에서는 분산 환경을 지원하기 위해서 SOAP(Simple Object Access Protocol)^[6]을 사용하였다. SOAP은 분산 환경 하에서 정보 교환을 위한 경량 프로토콜이다. SOAP은 Fig. 1과 같이 XML 형식의 메시지를 주고받는 형태를 취하고 있어서 서로 다른 플랫폼간에도 통신이 가능하다.

```
POST /StockQuote HTTP/1.1
Host: www.stockquotesserver.com
Content-Type: text/xml; charset="utf-8"
Content-Length: nnnn
SOAPAction: "Some-URI"

<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
  <SOAP-ENV:Body>
    <m:GetLastTradePrice xmlns:m="Some-URI">
      <symbol>DIS</symbol>
    </m:GetLastTradePrice>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Fig. 1 Example of SOAP Message Embedded in HTTP Request

또한 HTTP(Hyper Text Transfer Protocol)를 사용하므로 CORBA(Common Object Request Broker Architecture)나 DCOM(Distributed Component Object Model)과는 달리 방화벽의 제한을 받지 않으므로 어떤 곳에 있는 시스템과도 통신할 수 있다.

2.3 웹서비스(Web Service)

웹서비스는 특정 시스템의 기능을 인터넷을 통해 서비스 형태로 제공하는 시스템 모델이다. 웹서비스는 SOAP과 XML을 기반으로 하기 때문에 언어, 플랫폼에 상관없이 어떠한 시스템이든지 웹서비스가 제공하는 기능을 사용할 수 있다. 특히 SOAP을 사용하기 때문에 방화벽을 통과할 수 있어서 전 세계의 어떠한 곳에서도 웹 서비스에 접근할 수 있다.

웹서비스를 제공하는 서버에는 서버가 제공하는 네트워크 서비스에 대해서 기술해 놓은 XML 형식의 WSDL(Web Services Description Language)^[7] 문서가 존재한다. WSDL 문서에는 SOAP 메시지를 처리하는 수신자(listener), 서버가 제공하는 메소드(method) 이름, 파라미터, 반환값에 대한 정보가 기술되어 있다. SOAP 서버와 SOAP 클라이언트는 서버 상에 존재하는 WSDL문서를 참조하여 전송할 SOAP 메시지를 생성하고, 이렇게 생성된 SOAP 메시지를 주고받음으로써 웹서비스가 제공하는 기능을 사용하게 된다. Fig. 2는 본 연구에서 사용된 WSDL 문서의 일부이다.

```
<?xml version='1.0' encoding='UTF-8' ?>
<definitions name='AGMAWebService'
  targetNamespace='http://tempuri.org/wsd/'
  xmlns:wsdlns='http://tempuri.org/wsd/'
  xmlns:typens='http://tempuri.org/type'
  xmlns:soap='http://schemas.xmlsoap.org/wsd/soap/'
  xmlns:xsd='http://www.w3.org/2001/XMLSchema'
  xmlns:stk='http://schemas.microsoft.com/soap-toolkit/wsdl-extension'
  xmlns='http://schemas.xmlsoap.org/wsd/'>
  <types>
    <schema targetNamespace='http://tempuri.org/type'
      xmlns='http://www.w3.org/2001/XMLSchema'
      xmlns:SOAP-ENC='http://schemas.xmlsoap.org/soap/encoding/'
      xmlns:wsd='http://schemas.xmlsoap.org/wsd/'
      elementFormDefault='qualified'>
    </schema>
  </types>
  <message name='clsAGMARatingService.AGMARatingService'>
    <part name='AGMARatingXML' type='xsd:string' />
  </message>
  <message
    name='clsAGMARatingService.AGMARatingServiceResponse'>
    <part name='Result' type='xsd:string' />
  </message>
```

Fig. 2 The WSDL file of AGMA gear rating webservice

3. 레거시 시스템의 확장

3.1 레거시 시스템의 확장의 필요성

공학 설계 환경에서는 복잡한 계산을 수행하기 위한 많은 시스템이 존재하는데 상용의 대형 시스템을 제외하고는 오래된 컴퓨터 환경 하에서 작동하는 레거시 시스템이 상당수 있다. 이러한 레거시 시스템들은 새로운 컴퓨터 환경이 제공하는 편리하고 강력한 환경을 사용할 수 없을 뿐만 아니라 새로 개발되는 시스템과 연동하기도 힘들다. 이러한 이유로 야에 새로 시스템을 제작하는 경우가 종종 발생한다.

그러나 기존의 시스템을 현재 환경에 맞도록 새로 개발하기 위해서는 상당한 노력과 시간을 투입해야 한다. 특히 시스템이 안정화되는데 많은 시간이 소모되므로 개발에 걸리는 시간이 길어질수록 새로운 개발로 얻어지는 효과가 더욱 미미해진다. 따라서 기존의 시스템을 변경하지 않고 다른 시스템과 연동시킬 수 있도록 확장할 수 있다면 시스템을 다시 제작하는 것에 비해서 비용과 시간 면에서 상당한 이점을 가질 수 있다.

3.2 레거시 시스템의 확장 방법

본 연구에서는 AGMA 기어 강도 평가를 수행하는 레거시 기어 설계 시스템을 선정하고 이를 분산환경을 지원하도록 확장하는 방법에 대한 연구를 수행하였다.

AGMA 기어 강도 평가 시스템은 사용자가 필요한 값을 직접 입력하거나 특정 형식의 텍스트 문서를 입력으로 받아들여 AGMA의 기어 강도 평가 방법에 따라서 강도 평가를 수행하는 시스템이다. 이 시스템은 Fig. 3과 같이 DOS 환경에서 작동하는 시스템으로 매우 오래 전에 개발되었기 때문에 사용하기가 매우 불편할 뿐만 아니라 다른 시스템과의 연동이 거의 불가능한 상태이다.

본 연구에서는 AGMA 기어 강도 평가 시스템을 제어하는 레거시 시스템 제어 모듈을 제작하고, 이 모듈이 자동으로 시스템을 제어하게 함으로써 사용자의 개입 없이 AGMA 기어 강도 평가 시스템을 자동화하였다. 또한 AGMA 기어

강도 평가 XML 문서를 정의하고 레거시 시스템의 기능을 웹서비스화 함으로써 기존 시스템의 변경 없이 분산 환경 하에서 작동하도록 확장하였다.

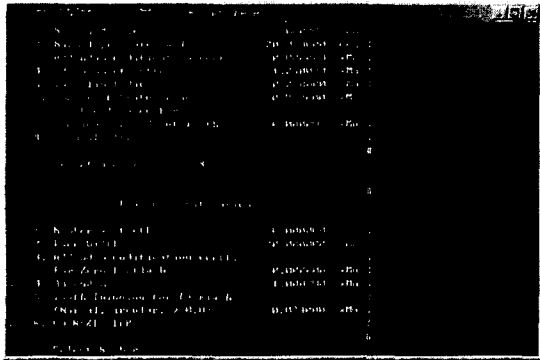


Fig. 3 Example of legacy gear design system

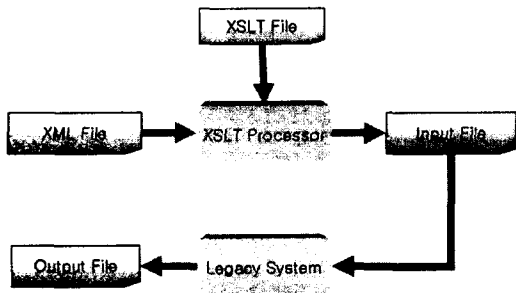


Fig. 4 The flow of document transformation

확장된 시스템에서의 문서의 흐름은 Fig. 4와 같다. Fig. 4에서 사용자 또는 외부의 다른 시스템이 Fig. 5와 같이 AGMA 기어 강도 평가 XML 문서를 생성하면, 이 문서는 Fig. 6과 같이 미리 정의된 AGMA 기어 강도 평가 XSLT 문서에 정의된 변환 규칙에 따라 XSLT 프로세서에 의해 AGMA 기어 강도 평가 시스템이 요구하는 입력 문서로 변환된다. 이렇게 생성된 입력 문서를 AGMA 기어 강도 평가 시스템이 읽어들이서 계산 결과 문서를 서버에 생성한다. 이때 XSLT의 특성과 같이 AGMA 기어 강도 평가 XML 문서 자체가 변경되는 것이 아니라 겉보기 형태만이 변경되는 것이므로 원본 XML 문서는 변경되지 않는다

```

<?xml version="1.0"?>
<AGMA>
  <Gear_type>1</Gear_type>
  <Gear_tooth_type>2</Gear_tooth_type>
  <Cutter_type>1</Cutter_type>
  <Operating_specification>
    <Pinion_speed unit="rpm">870</Pinion_speed>
    <Transmitted_power unit="KW">4.5</Transmitted_power>
    <Number_of_load_cycles>5.22e8</Number_of_load_cycles>
  </Operating_specification>
  <Cutter_specification>
    <Normal_module unit="mm">2.5</Normal_module>
    <Normal_pressure_angle unit="deg">20</Normal_pressure_angle>
    <Standard_helix_angle unit="deg">25</Standard_helix_angle>
    <Addendum_modification_coefficient unit="Mn">0</Addendum_modification_coefficient>
    <Addendum_of_cutter unit="Mn">1.25</Addendum_of_cutter>
    <Tooltip_radius unit="Mn">0.25</Tooltip_radius>
    <Amount_of_protuberance unit="Mn">0</Amount_of_protuberance>
    <Stock_allowance_for_grinding_per_side_of_tooth unit="Mn">0</Stock_allowance_for_grinding_per_side_of_tooth>
  </Cutter_specification>

```

Fig. 5 Example of a AGMA gear rating XML document

```

<?xml version='1.0'?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/TR/WD-xsl">
  <xsl:template match="/">y
  <xsl:value-of select="//Gear_type"/>
  <xsl:value-of select="//Gear_tooth_type"/>
  1
  <xsl:value-of select="//Pinion_speed"/>
  2
  n
  <xsl:value-of select="//Transmitted_power"/>
  3
  <xsl:value-of select="//Number_of_load_cycles"/>
  4
  <xsl:value-of select="//Cutter_type"/>
  1
  <xsl:value-of select="//Normal_module"/>
  2
  <xsl:value-of select="//Normal_pressure_angle"/>
  3
  <xsl:value-of select="//Standard_helix_angle"/>
  4

```

Fig. 6 AGMA gear rating XSLT document

이러한 변환 방법은 변환 모듈을 제작하는 것에 비해서 여러 가지 장점을 가진다. XML과 XSLT를 사용하기 때문에 플랫폼, 시스템의 특징에 상관없이 해당 플랫폼의 XSLT 프로세서만 있으면 입력 문서를 생성할 수 있는 것이 가장 큰 장점이다. 그러나 변경 모듈을 제작해서 사용한다면 각 플랫폼마다 하나씩 변경 모듈을 만들

어야 한다. 또한 XSLT는 문서 기반의 변환 규칙이므로 프로그램 소스 코드 수준에서 문서를 변환시키는 변환 모듈을 제작하는 것보다 훨씬 빠르고 간편하게 변환 기능을 구현할 수 있다. XML은 웹과 데이터베이스와의 연동이 매우 유리하므로 이러한 방식은 높은 유연성을 가진다.

4. 시스템 구현

이러한 방법론을 기반으로 Fig. 7과 같이 AGMA 기어 강도 평가 웹서비스를 개발하였다.

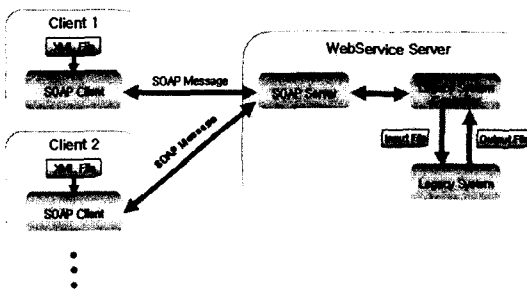


Fig. 7 The architecture of proposed system

클라이언트 내부에 존재하는 SOAP 클라이언트가 AGMA 기어 강도 평가 XML 문서를 SOAP 형식으로 전송하면, SOAP 서버는 이를 받아서 레거시 시스템 컨트롤러에게 전달한다. 레거시 시스템 컨트롤러는 수신된 AGMA 기어 강도 평가 XML 문서를 AGMA 기어 강도 평가 XSLT 문서의 변환 규칙에 따라 XSLT 프로세서를 이용하여 AGMA 기어 강도 평가 시스템의 입력 파일로 변환한다. AGMA 기어 강도 평가 시스템은 생성된 입력 문서를 읽어들이어서 계산을 수행하고 그 결과를 텍스트 문서 형태로 서버의 저장 공간에 생성한다. 이렇게 생성된 계산 결과를 레거시 시스템 컨트롤러가 읽어들이어서 SOAP 서버로 전달하고, SOAP 서버는 이를 SOAP 클라이언트로 전송한다.

AGMA 기어 강도 평가 웹서비스의 기능은 SOAP을 통해서 참조되기 때문에 플랫폼에 독립적이고 어떠한 시스템이든지 클라이언트가 될 수 있다. 하나의 AGMA 기어 강도 평가 웹서비스를 여러 시스템에서 반복적으로 사용될 수 있으므로 새로 개발되는 시스템에 AGMA 기어 강도 평가

기능을 직접 만들지 않고 쉽게 추가할 수 있다.

본 연구에서는 지역 응용 프로그램과 웹 응용프로그램의 두 개의 클라이언트 시스템을 제작해서 AGMA 기어 강도 평가 웹서비스를 참조하게 함으로써 이러한 확장성을 확인하였다. Fig. 8은 AGMA 기어 강도 평가 웹서비스를 참조하는 지역 응용 프로그램이다. 사용자나 다른 외부 시스템이 미리 생성한 AGMA 기어 강도 평가 XML 문서를 읽어들이어서 웹서비스 서버로 전송하면 그에 해당하는 AGMA 기어 강도 계산 결과가 반환된다. Fig. 9는 ASP.NET(Active Server Page.NET)^[8]을 이용하여 지역 응용 프로그램과 유사한 기능을 웹 응용프로그램으로 제작한 것으로 지역 응용프로그램과는 달리 웹브라우저만 있으면 어떤 곳에서든지 AGMA 기어 강도 평가 웹서비스를 사용할 수 있다.

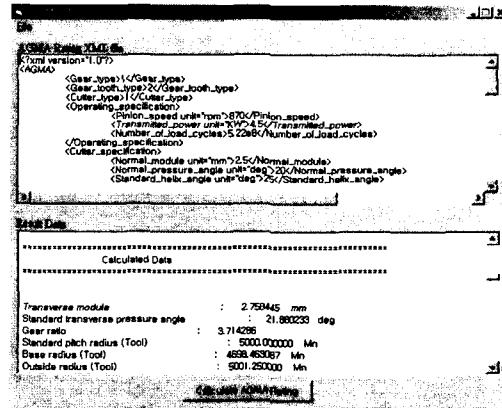


Fig. 8 Client system of - in form of local program

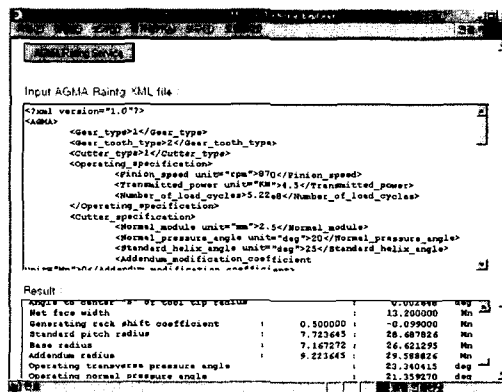


Fig. 9 Client system of - in form of web application

5. 결 론

본 연구에서는 XML과 XSLT를 이용하여 레거시 기어 설계 시스템을 분산환경 하의 새로운 컴퓨터 환경에서 사용할 수 있도록 확장하는 방법에 대한 연구를 수행하였다. XML과 XSLT를 이용함으로써 플랫폼에 독립적으로 보다 쉽게 AGMA 기어 강도 평가 시스템의 입력 파일을 생성할 수 있다. 또한 AGMA 기어 강도 평가 시스템의 기능을 웹서비스화 하여 플랫폼에 상관 없이 다양한 시스템에서 SOAP을 통해 AGMA 강도 평가 기능을 사용할 수 있도록 하였다. 차 후 개발되는 기어 설계 지원 시스템들은 AGMA 기어 강도 평가 기능이 필요한 경우 AGMA 기어 강도 평가 웹서비스를 참조하게 함으로써 쉽게 그러한 기능을 추가할 수 있다.

본 연구에서 제안하는 방법과 같이 기존의 레거시 기어 설계 시스템을 변경하거나 새롭게 만들지 않고 재활용한다면 새로운 컴퓨터 환경에 맞도록 기존 시스템을 고치는데 필요한 비용과 시간을 상당히 감소시킬 수 있을 것이다.

참고문헌

1. XML, Extensible Markup Language, <http://www.w3c.org/XML>
2. XSLT, XSL Transformations, <http://www.w3c.org/TR/xslt>
3. SGML, Standard Generalized Markup Language Extensible Markup Language, <http://www.w3.org/MarkUp/SGML>.
4. 정태형, 안준영, "XML을 이용한 기어 장치 설계 정보의 구조적 표현", 한국공작기계학회 2000년도 추계 학술대회 논문집, 2000.
5. 정태형, 박승현, "웹 기반 환경에서 XML을 이용한 기어 설계 데이터의 통합", 한국정밀공학회 2001년도 춘계 학술대회 논문집, 2001
6. SOAP, Simple Object Access Protocol, <http://www.w3c.org/TR/SOAP/>
7. WSDL, Web Service Definition Language, <http://msdn.microsoft.com/xml/general/wsdl.asp>.
8. ASP.NET, Active Server Page.NET <http://msdn.microsoft.com/vstudio/>