

분산 설계 환경 하에서 XML과 XSLT를 이용한 레거시 기어 설계 시스템의 확장

정태형*, 박승현⁺

(논문접수일 2002. 1. 16, 심사완료일 2002. 5. 18)

Extension of Legacy Gear Design Systems Using XML and XSLT in a Distributed Design Environment

Tae-Hyong Chong*, Seung-Hyun Park⁺

Abstract

As computer-related technologies have been developed, legacy design systems have not been appropriate for new computing environment. It is necessary that most of them are either modified or newly developed. However, these activities require quite much amount of cost and time. This paper presents a method of extending legacy design systems to the internet without any modification using XML and XSLT. We have been extended legacy systems in the two viewpoints. First, an XML document has been defined to present the input information of a legacy system which is executed on the consol environment - MS DOS, for example. Also, an XSLT document has been generated to transform an XML document to the input document of the legacy system. An XML document is transformed to the input document by XSLT processor according to the transformation rules defined in the XSLT document. This technique to generate input documents is independent to the platform type and facilitates to link legacy systems to other systems. Second, a legacy system controller has been made to control a legacy system and developed a web service to extend it and its controller. The legacy system controller operates it automatically. The web service provides its functions to other systems via internet. We have applied the developed methodologies to the legacy gear design system which calculates AGMA gear rating and made AGMA gear rating web service.

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

* 주저자, 한양대학교 기계설계학과 (cthoon@joongbu.ac.kr)
주소: 133-791 서울시 성동구 행당1동 17번지, Tel: 02-2290-0458
⁺ 한양대 대학원

1. 서론

컴퓨터의 도입과 함께 공학 설계 활동을 지원하기 위한 많은 시스템들이 개발되어 왔다. 이러한 시스템들은 CAD, FEM 등의 분야와 같이 상용으로 제작되는 시스템에서부터 필요에 따라 자체적으로 제작되는 소규모 시스템까지 그 종류와 형태가 매우 다양하다.

그러나 컴퓨터 관련 기술들이 발전함에 따라 오래 전에 개발된 시스템들은 새로운 컴퓨터 환경에 적합하지 않게 되기 시작하였다. 오래 전의 워크스테이션이나 MS-DOS 환경에서 작동하는 프로그램들이 그러한 예인데, 이러한 프로그램들은 현재의 그래픽 기반의 인터페이스를 사용하고, 인터넷을 지원하며, 다른 시스템과 연동할 수 있는 최신 시스템에 비해서 사용이 매우 힘들뿐만 아니라 더 이상 개선이 이루어지지 않았기 때문에 현재의 최신 컴퓨터에서는 동작이 되지 않는 경우도 있다. 이와 같이 더 이상 개선이 이루어지지 않는 오래된 시스템을 레거시 시스템(legacy system)이라고 한다.

대부분의 상용 설계 지원 시스템들은 컴퓨팅 환경의 발전에 맞춰서 끊임없이 개선되어 그만큼 더 강력하고 편리한 기능을 제공한다. 그러나 설계 환경에는 이러한 상용 시스템 이외에도 필요에 따라 자체적으로 개발된 수많은 설계 지원 시스템들이 존재한다. 이러한 시스템들은 상용 시스템과는 달리 개선이나 재개발이 되는 경우가 적어서 오래 전에 개발된 시스템들은 대부분이 레거시 시스템화된다. 이러한 레거시 시스템들은 사용하기에 불편할 뿐만 아니라 새로 개발되는 다른 시스템에 제공하기도 힘들다. 이러한 이유로 인하여 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 가진다.

이에 본 논문에서는 기존의 레거시 시스템을 변경하거나 새로 제작하지 않고 인터넷 기반의 분산 환경으로 발전하고 있는 새로운 컴퓨팅 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 시스템에 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 장점이 있다. 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, 시스템이 혼재된 분산 환경에서의 정보 교환에 매우 유리하다.

DTD(Document Type Definition)⁽⁴⁾는 XML 문서의 구조와 규칙을 정의한 문서로서 XML 문서의 검증에 사용된다. DTD를 사용하면 컴퓨터 파서(computer parser)가 자동으로 XML문서의 문법 등을 검증해주므로 시스템이 XML 문서를 사용할 때 발생할 수 있는 오류를 사전에 막을 수 있다. 그러나 DTD는 SGML의 유산으로 남겨진 것으로 XML 문법을 따르지 않고 자신만의 문법을 사용하기 때문에 이미 존재하는 다양한 XML 기반 기술의 지원을 받을 수가 없었다. 따라서 XML 기반의 문법 검증 기술의 필요성이 증대되었고, W3C의 작업의 결과로 XML 스키마(XML Schema)⁽⁵⁾가 개발되었다. XML 스키마는 XML 문법을 따르고 있어서 각종 XML 기반 기술을 그대로 사용할 수 있을 뿐만 아니라 DTD보다 더 뛰어난 표현력을 가진다.

XML은 자신만의 태그를 정의할 수 있으므로 같은 정보를 포함하고 있더라도 정의하는 방법에 따라서 여러 가지 형태의 XML 문서가 생성될 수 있다. XSLT는 XML문서를 다른 문서 형태로 변환하기 위한 목적으로 개발된 XML 기반의 기술이다. XSLT 문서는 XML 문서를 변환하는데 필요한 여러 가지 규칙 및 변환 방법에 대한 내용을 XML 문법에 따라 기술한다. XSLT를 이용하면 하나의 XML문서를 다른 형태의 XML 문서, HTML, 심지어는 복잡한 형

태의 텍스트 문서로 쉽게 변환할 수 있다. XSLT를 활용하면 XML 문서의 내용과 시각적 표현 형식을 분리시킬 수 있다. 즉 XSLT는 원본 XML문서를 전혀 변경하지 않고 시각적으로 보여지는 형태만 바꾸기 때문에 여러 개의 XSLT 문서를 각각 적용하여 하나의 XML 문서를 다양한 형태로 나타내는 것을 가능하게 한다. 또한 XSLT는 XML 문서 형식을 따르고 있기 때문에 XML의 장점을 그대로 가진다. 이렇게 XSLT를 활용하면 분산 환경에서 서로 다른 형태의 문서를 사용하는 시스템간의 정보 교환을 보다 용이하게 처리할 수 있다.

이와 같은 XML의 장점을 공학 설계에 적용하는 것에 관한 일련의 연구들을 저자 등이 수행한 바 있다^(6,7,8). 본 논문에서는 XML을 이용하여 기어 설계 정보를 표현하는 것에 대한 연구를 수행하였고, XML을 이용하여 다양한 설계 지원 시스템을 연동 및 원격 기어 모델링 서비스의 개발에 관한 연구를 수행하였다.

2.2 SOAP

본 논문에서는 분산 환경을 지원하기 위해서 SOAP(Simple Object Access Protocol)⁽⁹⁾을 사용하였다. SOAP은 분산 환경 하에서 정보 교환을 위한 경량 프로토콜이다. SOAP은 Fig. 1과 같이 XML 형식의 메시지를 주고받는 형태를 취하고 있어서 서로 다른 플랫폼간의 통신이 가능하다.

또한 HTTP(Hyper Text Transfer Protocol)를 사용하므로 CORBA(Common Object Request Broker Architecture)

나 DCOM(Distributed Component Object Model)같은 다른 분산 환경 기술과는 달리 방화벽의 제한을 받지 않으므로 어떤 곳에 있는 시스템과도 통신이 가능하다.

2.3 웹 서비스(Web Service)

웹 서비스는 특정 시스템의 기능을 인터넷을 통해 서비스 형태로 제공하는 분산 환경 기술이다. 웹 서비스에서는 이전의 분산 환경 기술과는 달리 시스템이 제공하는 기능이 현실 세계의 서비스와 비슷한 개념으로 취급된다. 즉 시스템이 제공하는 기능들이 클라이언트 시스템에게 서비스의 형태로 제공된다. 따라서 필요한 기능이 있을 경우 그 기능을 제공하는 웹 서비스가 존재하면 직접 개발할 필요 없이 해당 웹 서비스를 참조함으로써 원하는 기능을 쉽게 추가할 수 있다.

웹 서비스를 제공하는 서버에는 서버가 제공하는 네트워크 서비스에 대해서 기술해 놓은 XML Scheme 형식의 WSDL(Web Services Description Language)⁽¹⁰⁾ 문서가 존재한다. WSDL 문서에는 SOAP 메시지를 처리하는 수신자(listener), 서버가 제공하는 서비스의 이름, 파라미터의 이름과 형식, 반환값(return value)에 대한 정보가 기술되어 있다.

SOAP 서버와 SOAP 클라이언트는 서버 상에 존재하는 WSDL문서를 참조하여 전송할 SOAP 메시지를 생성하고, 이렇게 생성된 SOAP 메시지를 주고받음으로써 서버와 클라이언트간의 통신이 이루어진다.

웹 서비스는 SOAP과 XML을 기반으로 하기 때문에 언어, 플랫폼에 상관없이 어떠한 시스템이든지 웹 서비스가 제공하는 기능을 사용할 수 있다. 특히 SOAP을 사용하기 때문에 방화벽을 통과할 수 있어서 전 세계의 어떠한 곳에서도 웹 서비스에 접근할 수 있다.

3. 레거시 시스템의 확장

3.1 레거시 시스템의 확장 방법

본 논문에서는 2가지 관점에서 레거시 시스템 확장 방법을 개발하였다.

첫째, XML과 XSLT를 이용하여 레거시 시스템의 입력 문서를 자동으로 생성할 수 있게 함으로써 다른 시스템과 연동할 수 있는 기반을 구축한다. MS-DOS 환경에서 실행되는 레거시 시스템의 경우에는 데이터 입력의 편리성을 위해 입력 데이터가 기록되어 있는 입력 문서를 활용하는 경

```

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 An example of SOAP message embedded in HTTP request

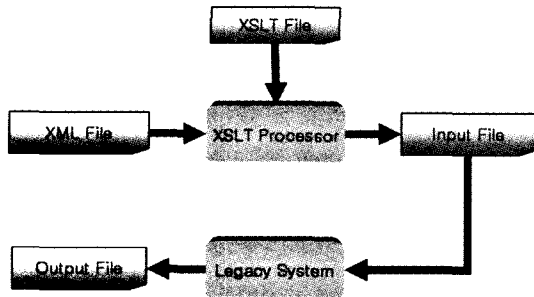


Fig. 2 The flow of document transformation

우가 많다. 입력 문서를 자동으로 만들 수 있다면 사용자의 개입 없이 다른 시스템과 자동으로 연동할 수 있다. 이에 XML과 XSLT를 이용하여 플랫폼에 상관없이 입력 문서를 자동으로 생성하는 방법을 개발하였다.

Fig. 2는 입력 문서의 변환 과정을 보여준다. 사용자 또는 외부의 다른 시스템이 XML파일을 생성하면 XSLT 프로세서는 XSLT 문서를 읽어들이 레거시 시스템의 입력 문서로 변환하기 위한 규칙을 분석하고, 이 규칙에 따라 XML 문서 내부의 XML 요소들의 변경 또는 재배치 등의 작업을 거쳐 레거시 시스템의 입력 문서를 생성한다. 이렇게 생성된 입력 문서는 레거시 시스템에 의해서 처리되어 최종적으로 계산 결과 문서가 생성된다.

이러한 변환 방법은 문서 변환 모듈을 제작하는 것에 비해서 여러 가지 장점을 가진다. XML과 XSLT를 사용하기 때문에 플랫폼, 시스템의 특징에 상관없이 입력 문서를 생성할 수 있다. 만약 문서 변환 모듈을 제작해서 사용한다면 각 플랫폼마다 하나씩 문서 변환 모듈을 만들어야 한다. 새로운 플랫폼이 등장한다면 그 플랫폼에 해당하는 변환 모듈을 개발해야 한다. 또한 XSLT는 XML문서 기반의 변환 규칙이므로 프로그램 소스 코드 수준에서 문서를 변환시키는 문서 변환 모듈을 제작하는 것보다 훨씬 빠르고 간편하게 변환 기능을 구현할 수 있다. XML을 다른 시스템과의 통신 수단으로서 활용함으로써 웹이나 데이터베이스와의 연동에 매우 유리할 뿐만 아니라 다른 시스템과 쉽게 연동시킬 수 있는 장점도 가진다.

둘째, 레거시 시스템을 제어하는 레거시 시스템 컨트롤러를 제작하여 이를 웹 서비스로 구축한다. 레거시 시스템은 시스템이 설치되어 있는 컴퓨터에서 사용자가 직접 작동해야만 하는데, 이러한 작업을 레거시 시스템 컨트롤러가 자동으로 처리하게 함으로써 레거시 시스템을 자동화할 수 있

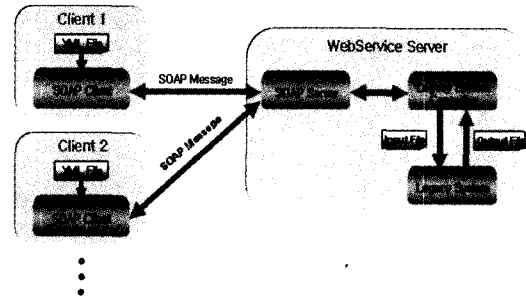


Fig. 3 The architecture of proposed system

다. 또한 레거시 시스템 컨트롤러를 웹 서비스 형식으로 외부에 노출시킴으로써 레거시 시스템이 제공하는 기능이 필요한 경우 직접 개발할 필요 없이 웹 서비스를 참조하여 바로 구현할 수 있도록 한다.

Fig. 3은 확장된 레거시 시스템의 구조를 보여준다. 이 구조에서 외부의 다른 시스템들이 클라이언트가 되고 레거시 시스템과 레거시 시스템 컨트롤러가 서버 역할을 수행한다. 클라이언트 내부에 존재하는 SOAP 클라이언트가 XML 문서를 SOAP을 통해 전송하면, SOAP 서버는 이를 받아서 레거시 시스템 컨트롤러에게 전달한다. 레거시 시스템 컨트롤러는 수신된 XML 문서를 XSLT 문서의 변환 규칙에 따라 XSLT 프로세서를 이용하여 레거시 시스템의 입력 파일로 변환한다. 레거시 시스템은 생성된 입력 문서를 읽어들이어서 계산을 수행하고 그 결과를 서버의 저장 공간에 생성한다. 이렇게 생성된 계산 결과를 레거시 시스템 컨트롤러가 읽어들이어서 SOAP 서버로 전달하고, SOAP 서버는 이를 SOAP 클라이언트로 전송한다.

본 논문에서 제안하는 레거시 시스템 확장 방법은 XML과 웹 서비스를 기반으로 하기 때문에 플랫폼에 독립적인 특징을 가진다. 기존의 레거시 시스템은 하나의 플랫폼에서만 작동하게 되지만 웹 서비스와 XML이라는 표준 기술을 사용함으로써 UNIX나 다른 OS를 사용하는 플랫폼에서도 MS-DOS 기반의 레거시 시스템의 기능을 활용할 수 있다. 또한 기존의 레거시 시스템은 외부 제어 모듈을 통해 자동화되므로 사용자의 개입이 필요하지 않다.

이러한 확장 방법은 시스템을 새로 제작하는 것에 비해 비용과 시간을 현저하게 감소시킬 수 있다. XML 문서의 제작과 레거시 시스템 제어 모듈의 개발은 레거시 시스템을 다시 개발하는 것에 비해 매우 적은 노력과 시간을 필요로 한다. 또한 같은 기능을 여러 번 구현할 필요가 없고, XML

을 사용하기 때문에 다른 시스템과 연동하기 쉬울 뿐만 아니라 기존의 시스템을 변경하지 않으므로 시스템의 안정화 기간도 짧아진다.

4. 시스템 구현

본 논문에서는 AGMA 기어 강도 평가를 수행하는 레저 시 기어 설계 시스템에 개발된 방법론을 적용하여 기존 시스템을 확장하였다.

AGMA 기어 강도 평가 시스템은 사용자가 필요한 값을 직접 입력하거나 특정 형식의 텍스트 문서를 입력으로 받아 들어 AGMA의 기어 강도 평가 방법에 따라 강도 평가를 수행하는 시스템이다. 이 시스템은 Fig. 4와 같이 MS-DOS 환경에서 작동되는 시스템으로서 매우 오래 전에 개발되었기 때문에 사용하기가 불편할 뿐만 아니라 다른 시스템과의 연동이 불가능하다.

Fig. 5와 같이 AGMA 기어 강도 평가에 필요한 정보를 표현하는 XML 문서를 정의하였다. 이 XML 문서에는 피니언, 기어의 형상 및 재질에 관한 정보와 각종 인자들에 대한 정보들이 각각 하나의 XML 요소(element)로 표현되어 있다. AGMA 기어 강도 평가 XML 문서는 Fig. 6의 AGMA 기어 강도 평가 XSLT 문서의 변환 규칙에 따라 Fig. 7과 같이 AGMA 기어 강도 평가 시스템의 입력 문서 형식으로 변환된다.

AGMA 기어 강도 평가 시스템을 자동으로 제어하는 레저 시 시스템 컨트롤러 모듈을 개발하고 이를 웹 서비스로 구축하였다. Fig. 8과 같이 AGMA 기어 강도 평가 웹 서비스에 대한 정보를 기술하는 WSDL 문서를 생성하였다. 외

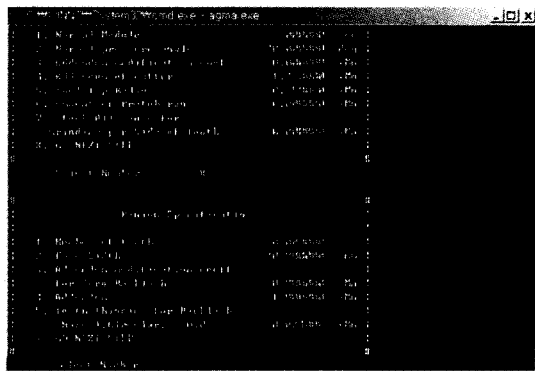


Fig. 4 AGMA gear rating calculation system

부 시스템에서 AGMA 기어 강도 평가 기능이 필요한 경우 직접 제작하지 않고 AGMA 기어 강도 평가 WSDL 문서를 참조하여 쉽게 기능을 추가할 수 있다. 이때 SOAP 서버와 클라이언트간의 통신을 위한 SOAP 메시지는 WSDL 문서를 기반으로 필요에 따라 내부적으로 자동으로 생성된다.

AGMA 기어 강도 평가 웹 서비스의 기능은 SOAP을 통해서 참조되기 때문에 플랫폼에 독립적이고 어떠한 시스템 이든지 클라이언트가 될 수 있다. 또한 방화벽을 뛰어 넘을 수 있기 때문에 기업 내부의 인트라넷뿐만 아니라 전 세계 어떤 곳에서도 이 서비스를 사용할 수 있다. 하나의 AGMA 기어 강도 평가 웹 서비스를 여러 시스템에서 반복적으로 사용될 수 있으므로 새로 개발되는 시스템에 AGMA 기어 강도 평가 기능을 직접 만들지 않고 쉽게 추가할 수 있다.

본 논문에서는 지역 응용 프로그램과 웹 응용 프로그램의 두 개의 클라이언트 시스템을 제작해서 AGMA 기어 강도 평가 웹 서비스를 참조하게 함으로써 이러한 확장성을 확인하였다. Fig. 9는 AGMA 기어 강도 평가 웹 서비스를 참조

```

<?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>
  </Operating_specification>
  <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>
  </AGMA>
  </?xml>
  
```

Fig. 5 Example of an 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

```
y
1
2
1
870
2
n
4.5
3
5.22e8
4
1
1
2.50000
2
20.00000
3
25.00000
4
0.00000
:
```

Fig. 7 Input document for AGMA gear rating calculation system

하는 지역 응용 프로그램이다. 사용자나 다른 외부 시스템이 미리 생성한 AGMA 기어 강도 평가 XML 문서를 읽어 들여서 웹 서비스 서버로 전송하면 그에 해당하는 AGMA 기

```
<?xml version='1.0' encoding='UTF-8' ?>
<definitions name='AGMAWebService'
targetNamespace='http://tempuri.org/wsdl/'
xmlns:wsdlns='http://tempuri.org/wsdl/'
xmlns:typens='http://tempuri.org/type'
xmlns:soap='http://schemas.xmlsoap.org/wsdl/soap/'
xmlns:xsd='http://www.w3.org/2001/XMLSchema'
xmlns:stk='http://schemas.microsoft.com/soap-toolkit/wsdl-extension'
xmlns='http://schemas.xmlsoap.org/wsdl/'>
<types>
<schema targetNamespace='http://tempuri.org/type'
xmlns='http://www.w3.org/2001/XMLSchema'
xmlns:SOAP-ENC='http://schemas.xmlsoap.org/soap/encoding/'
xmlns:wsdlns='http://schemas.xmlsoap.org/wsdl/'
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. 8 The WSDL file of AGMA gear rating web service

어 강도 계산 결과가 반환된다. Fig.10은 ASP.NET(Active Server Page.NET)⁽¹¹⁾을 이용하여 지역 응용 프로그램과 유사한 기능을 웹 응용 프로그램으로 제작한 것으로 지역 응용 프로그램과는 달리 웹브라우저만 있으면 어떤 곳에서도 AGMA 기어 강도 평가 웹 서비스를 사용할 수 있다.

두 개의 클라이언트 시스템의 형태는 다르지만 AGMA 웹 서비스의 참조 부분에서는 실질적으로 거의 같다. 또한 직접 AGMA 기어 강도 평가 시스템을 새롭게 제작하는 것에 비해 시스템의 제작 시간이 현저하게 감소된다.

5. 결론

본 연구에서는 XML과 XSLT를 이용하여 레저시 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 웹 서비스화하여 플랫폼에 상관없이 다양한 시스템에서 SOAP을

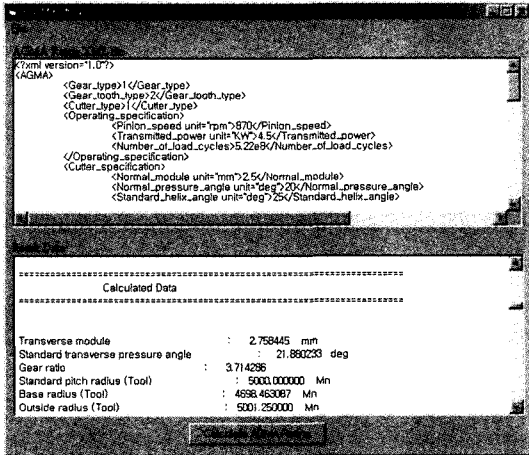


Fig. 9 Client system - local program type

통해 AGMA 강도 평가 기능을 사용할 수 있도록 하였다. 차후 개발되는 기어 설계 지원 시스템들에 대하여 AGMA 기어 강도 평가 기능이 필요한 경우 AGMA 기어 강도 평가 웹 서비스를 참조하게 함으로써 쉽게 원하는 기능을 추가할 수 있다.

본 논문에서 제안하는 방법에 따라 기존의 레거시 기어 설계 시스템을 변경하거나 새롭게 만들지 않고 재활용한다면 새로운 컴퓨터 환경에 맞도록 기존 시스템을 고치는데 필요한 비용과 시간을 상당히 감소시킬 수 있다.

참고 문헌

- (1) XML, Extensible Markup Language, <http://www.w3c.org/XML>
- (2) XSLT, XML Stylesheet Language Transformations, <http://www.w3c.org/TR/xslt>
- (3) SGML, Standard Generalized Markup Language Extensible Markup Language, <http://www.w3.org/MarkUp/SGML>

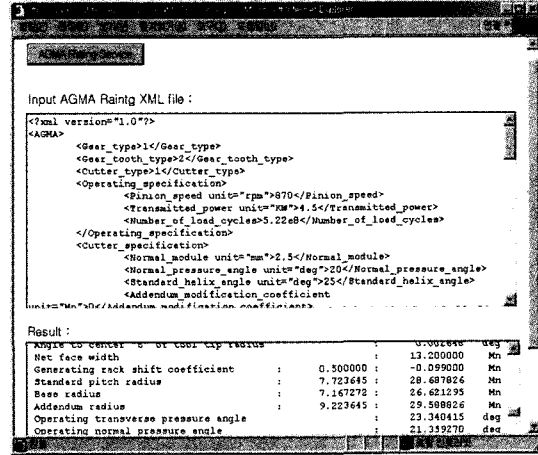


Fig. 10 Client system-web application type

- (4) DTD, Document Type Definition, <http://www.w3.org/XML/1998/06/xmlspec-report>
- (5) XML Schema, <http://www.w3.org/XML/Schema>
- (6) 정태형, 안준영, "XML을 이용한 기어 장치 설계 정보의 구조적 표현", 한국공작기계학회 2000년도 추계 학술대회 논문집, 2000.
- (7) 정태형, 박승현, "웹 기반 환경에서 XML을 이용한 기어 설계 데이터의 통합", 한국정밀공학회 2001년도 춘계 학술대회 논문집, 2001.
- (8) 정태형, 박승현, "XML과 XSLT를 이용한 레거시 기어 설계 시스템의 확장에 관한 연구", 한국공작기계학회 2001년도 추계 학술대회 논문집, pp. 257~262, 2001.
- (9) SOAP, Simple Object Access Protocol, <http://www.w3c.org/TR/SOAP/>
- (10) WSDL, Web Service Description Language, <http://msdn.microsoft.com/xml/general/wsdl.asp>
- (11) ASP.NET, Active Server Page.NET, <http://msdn.microsoft.com/vstudio/>