

인터넷을 활용하기 위한 수치해석 프로그램의 재구성 방법

송희용*·곽영균**·고성호***

Methods of Reconstructing Numerical Analysis Program for Utilizing the Internet

Heeyong Song*, Youngkyun Koak** and Sungho Ko***

Key Words : Numerical Analysis(수치해석), Computational Fluid Dynamics(전산유체역학), Java(자바),
Applet-Servlet Communication(애플릿-서블릿 통신), XML(확장성 표기 언어), Web Service(웹
서비스)

Abstract

The present study introduces some useful methods of implementing the Internet numerical analysis program with existing numerical codes for utilizing the Internet environment. The Internet gives developers good environment for development and release. Several methods were suggested, and some of them were implemented with an existing numerical code named SOLA-VOF, a computational fluid dynamics program to solve two-dimensional transient flow problems with free surface. This was reconstructed with Java technologies and compared with the original one. Java technologies have been applied to development of Internet applications for a long time. The objective of this work is to contrive methods of implementing Internet numerical analysis program with existing numerical codes and confirm the possibility of them. Methods using the applet-servlet communication were suggested and implemented. In addition, the Java web services with XML was introduced, which makes possible the cooperation of components. Although the concept has been suggested and developed for business applications, it can also be used for engineering softwares. Therefore, this study will be a preparation for numerical analysis to participate in engineering web services.

1. 서론

관련 기술들이 빠르게 발전하면서 다른 많은 산업에 적지 않은 영향을 주고 있는 인터넷은 응용 프로그램 개발에서 단지 사용자와 서버를 연결하는 수단으로 활용되어 왔다. 그러나, 인터넷은 프로그램 개발자들에게 훌륭한 배포 및 개발 환경을 제공할 수 있으며 이러한 시도는 최근 들어 XML(eXtensible Markup Language)과 컴포넌트 기반의 협업을 위한 웹 서비스 개념으로 수렴되고 있다. 현재 이를 위한 실행 및 개발 환경의 표준으로서 선 마이크로시스템즈사(Sun Microsystems, Inc.)의 J2EE(Java 2 Enterprise Edition)와 마이크로소프트사(Microsoft, Inc.)의 닷넷(.NET)이 경쟁하고 있다^[1].

인터넷 환경의 장점을 충분히 활용하기 위해서는 결국 인터넷에 배포 및 개발 환경으로서의 역할을 부여해야 하며 이는 어떠한 형태로든지 협업에 참여할 수 있도록 응용 프로그램을 개발해야 한다는 것을 의미한다. 그러나, 수치해석에 사용되는 응용 프로그램들은 아직까지 고전적인 방식으로 개발되고 있다. 이는 필요하지 않아서가 아니라 기존에 만들어진 방대한 양의 프로그램들을 전환하는 작업이 쉽지 않았기 때문일 것이다. 그러나, 이러한 이유로 새로운 방식의 도입을 미룬다면 결국 개발

자들은 더 큰 어려움에 봉착하게 될 것이다. 이제 구조적인 프로그램 개발 방식은 점점 규모가 커지고 복잡해지는 사용자들의 요구에 부응할 수 없으며 인터넷이라는 기반을 제대로 활용할 수도 없다. 반면 객체지향 프로그래밍은 융통성(flexibility)과 재사용(reuse)을 바탕으로 인터넷 환경과 어우러져 소프트웨어 개발자들에게 새로운 전성기를 맞게 하고 있다.

인터넷의 가장 큰 장점은 거리에 구애받지 않고 필요한 자원에 접근 또는 공유할 수 있다는 것이다. 인터넷을 기반으로 하여 만들어진 프로그램은 설치하거나 업그레이드하지 않고 사용할 수 있으며 인터넷에 연결된 다른 자원들과 쉽게 통합되어 대화형화될 수 있다. 뿐만 아니라 물성치나 실험치 같은 대량의 정보를 쉽게 공유할 수 있다. 또한, 인터넷 환경을 개선하기 위한 노력은 많은 연구 기관이나 기업체에 의해 끊임없이 이루어지고 있으며 실제로도 눈에 띄게 발전하고 있다. 인터넷 기반의 응용프로그램은 결국 그 발전에 따른 이득을 고스란히 승계 받을 수 있게 되며 이것은 인터넷을 활용함으로써 얻을 수 있는 최대의 장점일 것이다.

본 연구의 궁극적인 목적은 공학의 한 분야인 수치해석에 인터넷 환경이 갖는 장점을 활용하기 위한 기존 프로그램의 재구성 방법을 고안해 보고 이를 실제 사용되고 있는 수치해석 프로그램에 적용하여 그 가능성 및 효율성을 확인하는데 있다. 인터넷 응용 프로그램은 미래의 웹 서비스를 위한 컴포넌트로 발전하고 있으며 본 연구는 수치해석을 공학용 웹 서비스 개발에 한 일원으로서 참여시키기 위한 하나의 준비 과정이 될 것이다.

* 충남대학교 대학원, s_barram@hanmail.net

** 충남대학교 대학원, camsu95@hotmail.com

*** 충남대학교 기계설계공학과, sunghoko@cnu.ac.kr

2. 기존 프로그램의 재구성 방법

하나의 프로그래밍 언어로 출발했던 자바는 자바 2가 발표되면서 언어가 아닌 인터넷 프로그램의 개발 및 실행 환경이 되고자 하는 의지를 보여주고 있다. 자바의 가장 큰 장점은 시스템에 대한 독립성과 이동성이다. 자바 객체는 시스템의 종류에 상관없이 인터넷에 연결되어 있고 자신을 활성화시켜줄 가상 기계가 존재한다면 언제든지 이동하여 실행될 수 있다. 또 다른 자바의 장점은 재사용이다. 이것은 자바와 같은 객체지향 언어의 일반적인 특성으로서 전문 프로그래머가 아니더라도 손쉽게 자신의 목적에 맞는 프로그램을 개발하고 유지 및 보수할 수 있도록 해 주며 이는 자바의 확장성으로도 설명할 수 있다. 본 연구에서는 자바를 기준으로 하여 재구성 방법을 설명하고 일부를 실제 구현하였다.

2.1 배치와 통신

기존 코드를 자바로 전환하고 이를 사용자 측과 서버 측에 그대로 혹은 간단히 분리하여 배치한 후 이들 사이에 통신을 구현하면 쉽게 인터넷에서 수행되는 수치해석 프로그램을 구현할 수 있다. 사용자 측 프로그램을 애플릿(applet)으로 만들어 사용자의 요청이 있을 때 자동으로 공급되게 하고 서버 측 프로그램은 서블릿(servlet)으로 구현하여 사용자가 애플릿을 통해 계산을 요청할 때 작동되도록 하면 일련의 과정을 자동으로 수행되도록 할 수 있다.

애플릿은 인터넷 브라우저의 GUI(Graphic User Interface)를 통하여 사용자와 서버를 연결해 주는 역할을 수행한다. 그러나, 애플릿이 서버와 통신하는 일은 매우 어려운 일이었다. 자바 기반의 서버 측 프로그램인 서블릿이 등장하면서 이러한 작업이 손쉬워졌으며 요청에 대해 쓰레드를 생성하여 처리하는 서블릿은 기존의 CGI(Common Gateway Interface)에 비해 서버 측 시스템에 비교적 적은 부담을 주는 것이 특징이다.

Fig. 1은 기존 코드를 각각 전처리, 계산, 후처리 과정으로

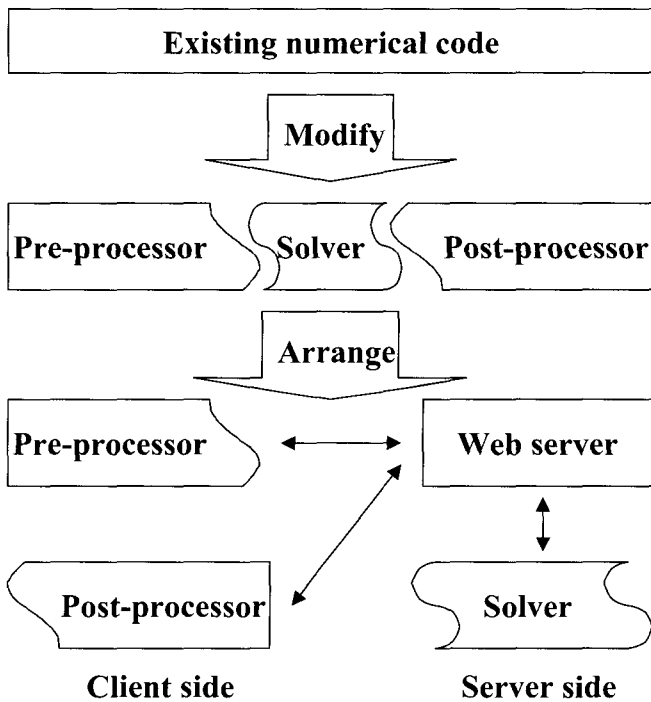


Fig. 1 A reconstructing method using arrangement and communication

분리하고 이를 각각 사용자 측과 서버 측에 배치시킨 후 통신을 구현하는 방법을 보여주고 있다. 서버 측에 있던 사용자 측 프로그램은 인터넷을 통해 사용자가 접근하였을 때 웹 브라우저를 통해 전달되며 사용자의 요청에 따라 서버 측 프로그램이 호출되고 사용자 측과 서버 측 프로그램의 통신에 의해 입력 값, 계산 수행 과정, 결과 값에 대한 정보를 주고받게 된다. 그러나, 이것은 사용자가 인터넷을 통해 서버 시스템의 자원을 사용한다는 점 외에 다른 어떠한 장점도 없으며 인터넷을 제대로 활용하기 위한 다른 시도도 할 수 없게 된다. 또한, 객체 지향 언어를 사용하였을 뿐 구조적 프로그래밍이 갖는 단점을 그대로 가지고 있으며 이것은 결국 유지 및 보수를 어렵게 만든다. 게다가 분리되지 않은 프로그램은 무겁다. 무겁다는 것은 인터넷을 통해 전달되기에 비효율적이고 수많은 사용자들의 요구를 동시에 처리해야 하는 인터넷 프로그램의 특성에 부합하지 않는다는 것을 의미한다. 자바는 언어 자체에서 다중 쓰레드를 지원한다. 다중 쓰레드는 동시에 여러 가지 계산을 수행할 수 있는 개념으로 잘 활용하면 계산 속도를 향상시키기 위한 병렬처리에 활용할 수 있을 것이다. 그러나, 분리되지 않아 무거운 프로그램은 이를 제대로 활용할 수 없다. 그럼에도 불구하고 방대한 양의 기존 자원을 고려한다면 가장 빠르게 인터넷 환경을 이용할 수 있는 방법일 것이다.

2.2 융통성을 부여하기 위한 객체화

배치와 통신을 이용한 방법의 단점을 보완하려면 기존 코드를 자바로 전환하고 이를 분리하여 객체화하여야 한다. 이 방법은 배치시켜 통신을 구현하는 방법에 비해 많은 시간과 노력이 소요된다. 또, 기존 코드나 개발 언어에 대해 많은 지식을 요하는 작업이기도 하다. 그러나, 일단 객체화가 이루어지면 프로그램은 융통성을 부여받게 되어 쉽게 유지 및 보수할 수 있다. 객체 지향의 이론적 틀은 메시지 전달이다. 객체들이 메시지를 주고받는 것으로 시스템이 동작한다. 객체는 자신의 상태를 속성으로 저장하며 객체의 기능에 해당하는 함수를 통해서만 이 속성에 접근할 수 있다. 이와 같은 구조는 적절한 정보의 은폐가 이루어져 견고한 시스템을 구축할 수 있게 한다. 또한, 객체의 직렬화(serialization)를 이용할 수 있어 객체의 상태를 저장하고 이를 인터넷을 통해 전달할 수 있으며 전달된 장소에서 다시 활성화시킬 수 있다. 객체지향 개념은 이처럼 많은 융통성을 가지므로 성능을 향상시키기 위한 여러 가지 시도를 해볼 수 있게 한다.

Fig. 2는 기존 코드를 각각 속성과 기능을 가진 여러 개의 객체로 분리하고 이를 서버 측과 사용자 측에 적절히 배치시키는 방법을 설명하고 있다. 사용자측은 애플릿으로 서버 측은 서블릿으로 구현하는 것이 배치와 통신을 구현하는 방법과 같지만 본 구성에서는 입력 객체, 격자 구성 객체, 전처리 과정 객체,

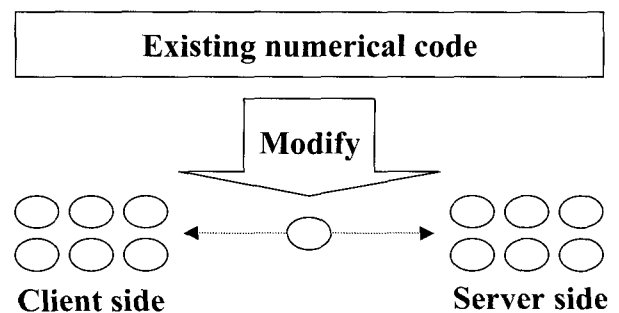


Fig. 2 A reconstructing method using objects.

결과 객체 등 기능과 속성에 따라 분리된 다양한 객체가 생성되어 작동하며 각각의 객체는 쉽게 결합되어 하나의 과정을 수행할 수 있고 분리되어 인터넷을 통해 전달될 수도 있다.

객체를 전달하는 방법이 통신으로 데이터를 보내는 방법에 비해 항상 좋은 것만은 아니다. 결과 객체와 같이 실시간으로 생산해내야 하는 데이터보다 저장해야 하는 데이터가 많은 경우에는 통신을 수행하여 전달하는 방법이 훨씬 더 효율적이다. 그러나, 통신을 구현함에 있어서도 객체로 분리된 경우가 통신 관련 API(Application Programming Interface)를 사용하기에 용이하므로 보다 쉽게 통신을 구현할 수 있다.

2.3 컴포넌트의 협업을 위한 XML의 사용

부품을 조립하듯이 쉽고 빠르게 규모가 큰 프로그램을 개발할 수 있는 장점으로 컴포넌트 기반의 소프트웨어 개발은 각광을 받고 있다. 컴포넌트란 이미 개발된 애플리케이션 조각이며 여러 개가 조립되어 하나의 애플리케이션으로 구성될 수 있다. 컴포넌트는 기본적으로 객체에 그 근간을 두고 있으므로 일단 객체로 시스템을 구성하게 되면 쉽게 컴포넌트로 전환하여 적절히 배치시켜 애플리케이션 서버의 도움을 받을 수 있다. 이의 구현을 위해 J2EE에서는 EJB(Enterprise JavaBeans)를 제공하며 이는 Java RMI(Remote Method Invocation)에 기반을 두고 있다^[2]. Java RMI는 자바기술이 제공하는 분산 객체 기술로 네트워크 상에 떨어져 있는 객체의 함수를 자신의 시스템에 있는 것처럼 호출할 수 있게 한다.

이러한 컴포넌트들이 공급자의 요청에 대해 단일화된 표준 데이터 양식을 생산하여 제공한다면 곧 인터넷 상에서 협업이 이루어지게 되는 것이다. 이러한 과정이 HTTP(Hypertext Transfer Protocol)에 XML이 결합된 SOAP(Simple Object Access Protocol)와 같은 프로토콜을 통해 이루어지는 것이 곧 웹 서비스의 개념이며 표준 데이터 양식인 XML은 이식성과 상호운용성의 특징을 갖는다^[3]. XML은 특정 응용 프로그램에 종속되지 않은 데이터를 제공하게 함으로써 개발을 용이하게 한다. 게다가, 자바는 XML을 다루는 다양한 API를 제공한다.

Fig. 3은 컴포넌트간의 협업을 위해 XML을 사용하는 방법을 보여주고 있다. 사용자의 요청을 받은 애플리케이션 서버는 다른 공급자와의 협업을 통해 계산을 수행하고 결과를 사용자에게

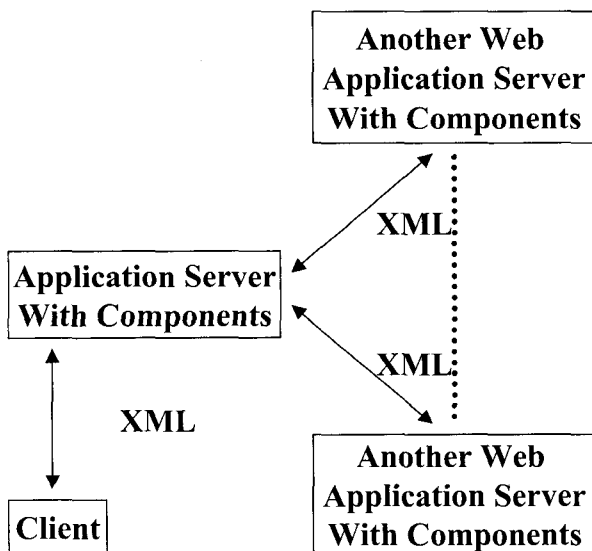


Fig. 3 A reconstructing method using cooperation of components with XML

전달하게 된다. 이 과정에서 개발자가 구현하기 어려운 것들의 많은 부분을 애플리케이션 서버가 대신 담당하여 수행하게 된다. 이는 규모가 큰 프로그램에 적당한 방식으로 작은 규모의 프로그램에 적용하는데는 무리가 있다. 따라서, 본 연구에서는 실제 구현하지 않았다.

3. 재구성된 예제 프로그램

본 연구에서는 실제 사용되고 있는 프로그램을 재구성하여 구현해 봄으로써 그 가능성을 확인하고자 하였다. 예제로 사용된 프로그램은 SOLA-VOF(SOLution Algorithm for Volume Of Fluid)로 불리는 전산유체역학 프로그램으로 자유표면을 포함하는 2차원 비정상 유동을 해석한다^[4]. 배치와 통신을 이용하는 방법과 객체로 분리하는 방법으로 재구성하였으며 실제 브라우저를 통해 인터넷에서 수행될 수 있도록 구현하였다.

3.1 기존 프로그램

SOLA-VOF는 자유표면의 움직임을 고정된 격자만을 이용하여 해석할 수 있게 하여 적은 기억 용량으로 처리할 수 있게 되어 있다. 이때 각 격자를 차지하는 유체의 부피를 나타내는 용적함수를 별도로 계산함으로써 자유 표면의 움직임을 감지할 수 있게 한다. SOLA-VOF는 포트란으로 작성되어 있으며 개별적인 임무를 수행하는 서브루틴으로 분리되어 있다. 그러나, 각 서브루틴은 공통 변수를 사용함으로써 분리를 어렵게 하고 있다.

본 연구에서 여러 개의 GUI 관련 객체와 입력, 격자, 전처리, 계산, 결과 등의 객체가 계산에 참여하도록 재구성하였다. 실제 컴포넌트 기반의 웹 서비스를 구현하려면 계산 과정을 좀 더 세밀하게 분리하여야 한다.

3.2 재구성된 프로그램

배치와 통신을 이용한 방법과 객체화하여 융통성을 부여한 방법은 겉으로 그 차이점이 드러나지 않으며 내부적으로만 차이를 갖게 된다. 모든 방법은 동일하게 인터넷에서 브라우저를 통하여 프로그램에 접근하게 되고 사용자는 다음 Fig. 4와 같은 GUI를 만나게 된다. 애플릿의 GUI는 JFC(Java Foundation Class)를 사용하였다. 자바가 기본적으로 제공하던 GUI인 AWT(Abstract Window Toolkit)는 응용 프로그램에 사용하기에 적당하지 않았다. 그러나, 순수한 자바 GUI를 만들기 위한 여러 업체들의 공동 작업으로 JFC가 등장하면서 자바가 일반 응용프로그램에 사용되는데 부족함이 없음을 보여 주었다.

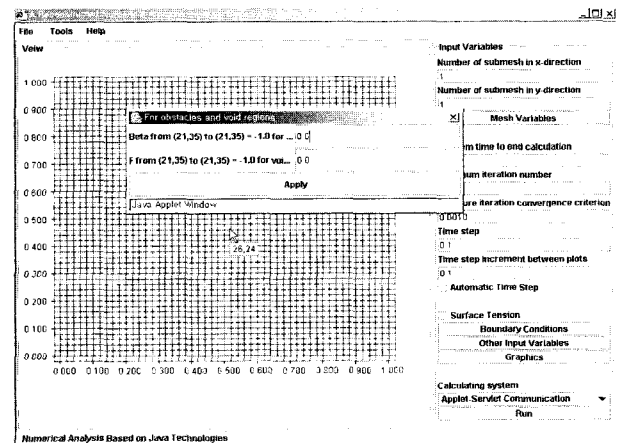


Fig. 4 GUI for pre-processing.

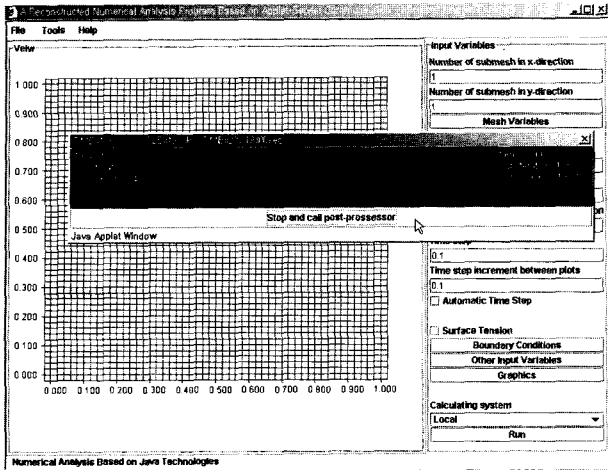


Fig. 5 GUI for managing solver.

Fig. 4와 같은 GUI를 통하여 격자를 생성하고 각 입력 값을 결정한 후 계산 시스템을 선택하여 계산을 요청하게 된다. 계산이 요청되는 순간 통신을 이용하는 방식은 HTTP를 통해 GET 또는 POST로 요청을 보내게 된다. 반면 객체를 전송하는 방식은 입력 값을 입력 객체에 저장하고 이를 직렬화한다. 애플릿은 직렬화된 객체를 서버에 보내게 되며 서버 측에서는 이 직렬화된 객체를 다시 활성화하여 활용할 수 있게 된다.

요청을 받은 서버 측 프로그램은 계산을 수행하는 스레드를 생성하여 계산을 수행한다. 이 과정에서 사용자 측 프로그램에서는 Fig. 5와 같이 서버 측 프로그램과 통신하며 계산과정을 지켜볼 수 있도록 하는 객체가 생성된다. 이 객체는 서버 측에 계산을 중단하도록 요청할 수 있게 하였으며 서버 측 계산이 중단되거나 완료되면 이 사실을 사용자가 확인할 수 있도록 구성하였다.

계산이 완료되었을 때 사용자가 결과를 요청하면 서버 측 프로그램은 사용자 측 프로그램에 결과를 전송하게 된다. 이 과정에서 통신을 구현하는 경우에는 결과를 문자열로 전송하게 되며 객체를 전송하는 경우에는 결과 객체에 결과를 저장하고 이를 직렬화하여 전송하게 된다. 사용자 측 프로그램에서는 문자열을 분석하거나 직렬화되어 전송된 객체를 활성화하여 Fig. 6와 같은 GUI를 통해 사용자가 계산 결과를 활용할 수 있도록 한다. Fig. 6는 장애편을 포함하는 사각 통 안의 회전에 대한 유체유동을 해석하여 그 유선과 속도 벡터를 확인하는 모습이다.

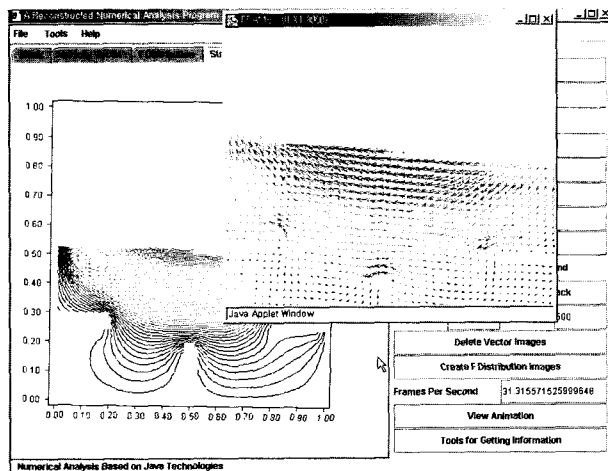


Fig. 6 GUI for post-processing.

4. 결과 및 고찰

본 연구에서 확인하였듯이 기존의 수치해석 코드를 인터넷 안에서 실행되도록 하는 것은 생각보다 쉬운 일이다. 단지 프로그램을 인터넷이 수용할 수 있도록 전환하여 적절히 배치하고 그들간의 통신을 구현하면 가능한 일이다. 또, 전환 작업을 대신 해주는 도구를 사용할 수도 있다. 실제로 그러한 도구들은 존재하며 지금도 개발되고 있다. 그러나, 이러한 방식은 좀 더 나은 형태의 프로그램 개발을 위해 지양되어야 한다. 단순히 전환하여 재배포하고 서로간의 통신을 구현하게 되면 구조적 프로그램 개발 기법이 갖는 단점을 그대로 떠 안아 결국 더 이상 발전하지 못하는 골칫거리 인터넷 콘텐츠로 남게 될 것이다. 본 연구에서 이러한 방식으로 실제 예제를 구현해 보면서 인터넷을 통해 프로그램을 실행하고 결과를 확인할 수 있다는 것 외에 인터넷의 장점을 좀 더 활용하고자 하는 어떠한 시도도 해볼 수가 없었다.

반면, 객체로 분리하여 재배포하는 방식은 구조적 프로그램 개발 기법이 갖는 단점을 개선하였으며 인터넷의 장점을 최대한 활용하기에 충분한 융통성을 가지고 있었다. 특히, 각각의 객체가 배치될 위치의 선정이 자유로웠으며 다양한 API를 쉽게 접근시킬 수 있어 개발 기간 내내 많은 욕심을 갖게 하였다.

5. 결론

기존 수치해석에 인터넷 환경의 장점을 활용하기 위한 방법은 매우 다양하다. 본 연구에서 제시된 방법 외에도 다른 많은 방법들이 가능하며 특히 기존 코드를 수정하지 않고 미들웨어 개념의 중간 프로그램을 만드는 것은 기존 코드의 방대한 양을 고려한다면 현 시점에서 가장 유용한 방식일 것이다. 그러나, 본 연구에서 보았듯이 기존 프로그램을 객체로 분리하지 않으면 인터넷이 갖는 장점을 제대로 활용할 수 없게 된다.

인터넷은 이제 배포 환경의 역할을 넘어 여러 컴포넌트 또는 객체들의 협업을 위한 개발 및 실행 환경으로 거듭나고 있다. 여기에 기존의 수치해석 프로그램이 한 일원으로서 참여하여 인터넷이라는 환경이 갖는 장점을 제대로 활용하려면 속성과 기능에 따라 객체로 분리되고 배치되어야 하며 표준 데이터 양식을 생산하여 전달하여야 한다. 이러한 작업이 원활하게 이루어지기 위해서는 기존의 수치해석 프로그램이 적절히 재구성되어야 할 것이다. 본 연구에서는 몇 가지 재구성 방법을 제시하였고 일부를 실제 구현함으로써 가능성을 확인할 수 있었다.

참고문헌

- [1] C. Vawter and E. Roman, "J2EE vs. Microsoft.NET A comparison of building XML-based web services", Sun Microsystems, Inc. (2001).
- [2] S. Bodoff, D. Green, K. Haase, E. Jendrock, M. Pawlan and B. Stearns, "The J2EE™ Tutorial", Sun Microsystems, Inc. (2002).
- [3] E. Armstrong, S. Bodoff, D. Carson, M. Fisher, D. Green and K. Haase, "The JAVA™ Web Service Tutorial", Sun Microsystems, Inc. (2001).
- [4] B. D. Nichols, C. W. Hirt and R. S. Hotchkiss, "Volume of Fluid(VOF) Method for the Dynamics of Free Boundaries", J. Comput. Phys. Vol. 39 (1981).