웹 서비스 기반의 통합 설계 프레임워크

장원석 김광식 정갑주
건국대학교 인터넷미디어 공학부
{ws822°, beast000}@imc.konkuk.ac.kr jeongk@konkuk.ac.kr

Web Service-Based Integrated Design Framework

Wonseok Jang° Kwangsik Kim Karpjoo Jeong
Division of Internet and Media, Konkuk University

요 약

오늘날 공학 분야에서 한 분야에서만 이뤄지던 연구가 다분야 통합 연구로 바뀌어 가고 있다. MDO (Multi-Disciplinary Optimization) 프레임워크는 각 분야의 설계 도구들 간의 데이터 공유 및 효율적 관리를 위한 기술과 여러 분야가 분산된 환경 하에서 병렬로 작업할 수 있는 컴퓨팅 환경을 말한다. 기존의 MDO 프레임워크는 여러 분야의 설계 도구들을 통합 관리하는 표준 인터페이스가 없고, 이것들의 작업 흐름을 자동으로 통합 관리할 환경이 없다는 문제점이 있다. 본 논문에서는 웹 서비스를 사용하여 각 설계 도구 간의 표준 인터페이스를 제공하고, 워크플로우를 사용하여 이것들을 자동으로 통합 관리하는 웹 서비스 기반 통합 설계 프레임워크를 구현한다.

1. 서 론

오늘날 공학 분야 연구가 인터넷이 발달하고 프로젝트 규모가 커짐에 따라 한 분야에서만 국한되어 진행되던 연구가 지리적으로 분산된 다분야의 사람들이 모여서 통합으로 진행하는 연구의 필요성이 증대되고 있다. 그리고 컴퓨팅 속도와 네트워크의 발달로 공학 분야의 연구나 실험을 컴퓨터가 대신해주는 사례가 늘고 있다. MDO (Multi-Disciplinary Optimization) 는 공학 분야에서 다분야의 공학 설계를 설계 단계부터 통합적으로 진행함으로써, 전체적으로 프로젝트 비용을 줄이고 유지보수를 용이하게 하는 통합 설계 방법론이다. MDO 프레임워크는 각 분야의 설계 도구들 간의 데이터 공유 및 효율적 관리를 위한 기술과 여러 분야가 분산된 환경 하에서 병렬로 작업할 수 있는 컴퓨팅 환경을 말한다[1].

서비스 지향 구조 (Service-Oriented Architecture) 는 비즈니스 프로세스에 초점을 맞추고, 표준 인터페이스를 사용함으로서 IT 환경이 가진 근본적인 기술적 복잡성을 덮어주는 기술이다. 그리고 서비스들의 제공, 요청, 그리고 중계를 지원하는 소프트웨어의 서비스화를 지향하는 구조이다.

그리고 서비스 지향 구조는 잘 정의된 인터페이스를 통해 서비스의 기능과 서비스 호출 규칙 등의 통신 규약을 정의하고, 개방형 표준 인터페이스를 제공함으로서 합의된 기준으로 통신을 가능하게 해주고, 느슨한 결합을 통해 서비스 제공자와 서비스 요청자 간의 완전한 분리를 이루고, 그리고 서비스를 정의할 때, 비즈니스 로직이나 프로세스 단위로 블록화를 시켜서 어플리케이션 기능들을 서비스화를 지향한다.

서비스 지향 구조를 가장 잘 반영한 기술이 바로 웹 서비스이다. 웹 서비스는 서비스 지향 구조 기반의 XML 메시징을 위한 최신 분산 컴포넌트 기술이다. 웹 서비스 기본 요소에는 SOAP, WSDL, UDDI 가 있다. SOAP은 HTTP 상에서 XML 메시징을 하기 위한 표준 경량 프로토콜이다. WSDL은 웹 서비스를 정의하는 표준 언어로 XML 스키마 기반으로 되어있다. UDDI는 웹 서비스를 등록 (Publish) 하고, 검색 (Discovery) 하고, 그리고 연결 (Binding) 해주는 웹 서비스 표준이다[2].

BPEL4WS (Business Process Execution Language for Web Service) 는 웹 서비스를 위한 비즈니스 프로세스를 정의해주는 언어이다, 이것은 웹 서비스 컴포지션 (composition) 표준의 기초가 될 것이다. BPEL4WS는 WSFL (support for graph-oriented processes)과 XLANG (structural constructs for processes)의 최고의 장점들을 패키지로 묶어 매우 자연스러운 방식으로 모든 종류의 비즈니스 프로세스를 구현할 수 있도록 지원한다. BPEL4WS는 구현 언어일 뿐만 아니라, 추상 프로세스 개념을 사용하여 비즈니스 프로세스의 인터페이스를 설명하는데 사용될 수 있다[3].

본 논문은 웹 서비스와 BPEL4WS를 사용하여 서비스 지향 구조 기반의 통합 설계 프레임워크를 구현한다.

본 논문의 구성은 다음과 같다. 제 2장에서는 관련연구를 기술하고, 제 3장에서는 웹 서비스 통합 설계 시스템에 대해서 설명하고, 제 4장에서는 시스템 구현에 대해서 설명하고, 그리고 제 5장에서는 결론 및 향후과제에 대해서 기술한다.

2. 관련연구

ModelCenter는 Virginia Tech Corporate Research Center 에 위치한 Pheonix Integration 에 의해 개발되었다. Pheonix Integration 은 Virginia 공대의 박사과정인 Scott Woyak와 Brett Malone 와 그들의 정신적인 지주인 Arvid Myklebust 기계공학과 교수에 의해 창설되었다. ModelCenter는 Analysis Server와 함께 통합 설계 프레임워크를 이룬다. 현재 ModelCenter는 버전 6.0까지 릴리즈 되어있으며, Analysis Server는 4.1 버전까지 릴리즈 되어 있다. 이것들은 자바 언어로 만들어졌기 때문에 플랫폼에 독립적인 장점을 가지고 있다[4].

3. 웹 서비스 기반 통합 설계 프레임워크

3.1 시스템 요구사항

본 시스템은 지리적으로 분산되어 있는 공학 설계 모듈들을 통합 관리하기 위한 표준 인터페이스를 제공하고, 공학 설계 모듈들의 작업 흐름을 자동으로 제어하기 위한 환경을 제공한

다. 그리고 사용자와 모듈 리스트를 관리하기 위한 관리자 환경이 요구된다.

3.2 시스템 기능

본 시스템의 주요기능은 지리적으로 떨어져 있는 공학 설계 모듈을 호출하기 위한 환경을 제공하는 Invocation Pool Service와 프로세스를 정의하는 Process Definition Service, 설계 모듈의 작업 상태를 보여주는 Monitoring Service와 설계 모듈의 워크플로우를 실행시키고, 최종 결과를 보여주는 Execute Service. 마지막으로 사용자와 모듈 리스트를 관리하는 Administration Service 가 있다.

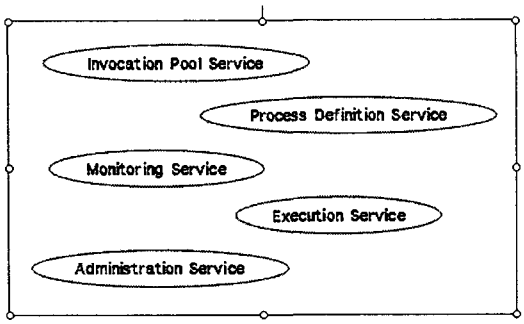

그림 1 시스템 주요기능

3.3 시스템 설계

본 시스템의 시나리오는 그림 1과 같다. 본 시스템은 크게 웹 서버(Web Application Server) 와 서비스 제공자 (Service Provider), 그리고 서비스 중개자 (Service Registry) 로 나뉜다. 먼저, 서비스 제공자에서는 본 시스템에서 제공할 서비스들을 생성하고, 생성한 서비스들을 서비스 중개자 (Service Registry) 에 등록 (Publish) 하고, 그리고 사용자가 웹 인터페이스를 통해 서비스를 찾고 (Discovery), 그리고 웹 서버를 통해 찾은 서비스와 연결 (Binding) 한다.

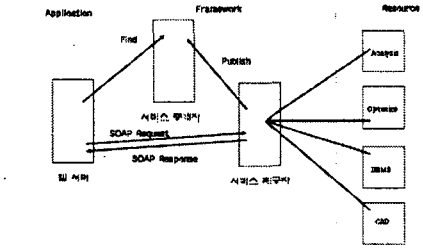

그림 2 시스템 시나리오

본 시스템의 구조는 크게 세 계층으로 나뉜다. 첫째로 Resource Layer를 보면, 이 계층은 각 공학 분야에서 쓰이는 설계 자원들을 포함한다. 이 계층의 특징은 각 자원들이 지리적으로 분산되어 있고, 레가시 환경을 구축하고 있다는 데에 있다. 둘째로 Middleware Layer를 보면, 이 계층에서는 실제로 사용자에게 제공할 서비스들을 가지고 있는 소프트웨어 환경을 제공한다. 마지막으로 Application Layer를 보면, 이 계층에서는 사용자들이 보다 더 쉽게 서비스를 사용할 수 있도록 GUI를 제공한다.

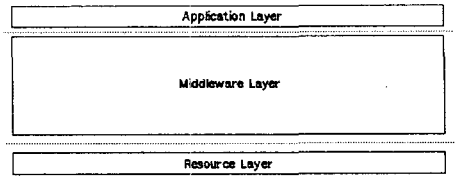

그림 3 1단계 설계

더 자세히 살펴보면, Middleware Layer를 그림 2와 같이 나눌 수 있다. Middleware Layer는 다음과 같이 5 개의 서비스가 제공되는데 먼저, Invocation Pool Service는 웹 서비스를 통해서 지리적으로 분산되어 있는 설계 모듈들을 작동시키고, 각 설계 모듈 간에 표준 인터페이스를 제공한다. 둘째로, Orchestration Service는 아래 계층 (Invocation Pool)에서 생성된 웹 서비스들의 작업 흐름을 WS-BPEL 기반의 BPM (Business Process Manager) [6]를 사용하여 자동으로 통합 제어해주는 역할을 한다. 셋째로, Monitoring Service는 아래 계층 (Orchestration)에서 생성된 Business Process[7]의 상태를 보여주고, 제어하는 역할을 한다. 넷째로 Workspace Service는 공학 설계의 작업을 위한 프로젝트를 생성하고, 아래 계층 (Orchestration)에서 생성한 Workflow Service 들을 동작시키는 역할을 한다. 마지막으로 Administration Service는 User Management Service 와 Module Management Service 가 있는데, 먼저 User Management Service는 사용자 계정을 생성, 삭제, 수정하는 역할을 한다. Module Management Service는 설계 모듈에 관한 정보들을 관리하는 서비스이다.

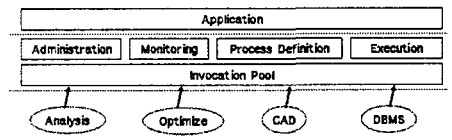

그림 4 2단계 설계

4. 시스템 구현

4.1 Invocation Pool Service

Invocation Pool Service는 원격에 위치한 설계 모듈들을 호출하기 위해 프록시를 생성해주는 서비스이다. 제공할 인터페이스 이름은 'InvocationPool' 이고, 메소드 이름은 'execute' 이고, 그리고 인자 값은 'path' 이다. 기본적으로 RMI 인터페이스를 상속받으며, 원격 호출 방식을 채택한다. 메소드에 들어갈 인자 값 'path' 는 원격 호출할 설계 모듈의 설치 경로에 해당한다.

인터페이스 이름	메소드 이름	인자 값
InvocationPool	execute	path

표 1 Invocation Pool Service 인터페이스

4.2 Process Definition Service

Process Definition Service는 각 설계 모듈의 작업 흐름을 정

의해주는 서비스이다. 실행시킬 모듈을 선택하고, 실행 타입을 선택하고, 그리고 선택한 모듈과 실행 타입으로 XML 기반의 워크플로우 문서 (이하 WS-BPEL) 를 만드는 역할을 한다. initProcess 메소드는 프로세스를 초기화해준다. startProcess 메소드는 프로세스 시작 노드를 추가해준다. endProcess 메소드는 프로세스 끝 노드를 추가해준다. addModule 메소드는 프로세스에 모듈을 추가해준다. 인자 값으로는 실행 모듈의 실행 경로를 갖고 있는 'path' 를 갖는다. setExecutionType 메소드는 실행 타입을 셋팅해준다. 인자 값으로 실행 타입을 갖는 'type' 을 갖는다. getExecutionType 메소드는 현재 노드의 실행 타입을 가져온다. 리턴 값으로 실행 타입을 갖는 'type' 을 갖는다.

인터페이스 이름	메소드 이름	인자 값/리턴 값
ProcessDefinition	initProcess	없음
	startProcess	없음
	endProcess	없음
	addModule	path
	setExecutionType	type
	getExecutionType	type

표 2 Process Definition Service 인터페이스

4.3 Monitoring Service

Monitoring Service 는 설계 모듈의 작동 형태를 알려주는 서비스이다. 제공하는 인터페이스의 이름은 'Monitoring' 이고, 메소드는 'displayStatus' 가 있다. 인자 값으로 'process_id' 를 갖는다.

인터페이스 이름	메소드 이름	인자 값
Monitoring	displayStatus	process_id

표 3 Monitoring Service 인터페이스

4.4 Execution Service

Execution Service는 모듈의 워크플로우를 실행하고, 최종 결과를 텍스트와 차트 형태로 보여주는 서비스이다. 제공하는 인터페이스 이름은 'Execution' 이고, 메소드는 'startWorkflow', 'displayChart', 'displayResult' 가 있다. startWorkflow 메소드는 설계 모듈의 워크플로우를 실행시킨다. displayChart 메소드는 최종 결과를 차트 형태로 사용자에게 보여준다. displayResult 메소드는 최종 결과를 텍스트 형태로 사용자에게 보여준다.

인터페이스 이름	메소드 이름	인자 값
Execution	startWorkflow	없음
	displayChart	없음
	displayResult	없음

표 4 Execution Service 인터페이스

4.5 Administration Service

Administration Service는 User Management Service와 Module Management Service 나뉜다. 먼저 User Management Service는 사용자 관리 관련 서비스이다. 제공하는 인터페이스 이름은 'UserManagement' 이고, 메소드는 'createUser' , 'deleteUser' , 'modifyUser' 가 있다. createUser 메소드는 사용자를 추가해주는 기능을 한다. deleteUser 메소드는 사용자를 삭제하는 기능을 한다. modifyUser 메소드는 사용자를 수정하는 기능을 한다.

인터페이스 이름	메소드 이름	인자 값
UserManagement	createUser	id, pwd, name, SSN
	deleteUser	id
	modifyUser	id

표 5 User Management 인터페이스

그리고 Module Management Service는 모듈 관리 관련 서비스이다. 제공하는 인터페이스 이름은 'ModuleManagement' 이고, 메소드는 'addModule', 'deleteModule', 'modifyModule' 이 있다. addModule 메소드는 설계 모듈 정보를 추가시키는 기능을 하고, 인자 값으로는 모듈을 구분하는 id, 모듈 이름을 갖는 name 을 갖는다. deleteModule 메소드는 설계 모듈 정보를 삭제하는 기능을 한다. 인자로는 id를 갖는다. modifyModule은 설계 모듈 정보를 수정하는 기능을 하고, 인자로는 id를 갖는다.

인터페이스 이름	메소드 이름	인자 값
ModuleManagement	addModule	id, name
	deleteModule	id
	modifyModule	id

표 6 Module Management 인터페이스

5. 결론 및 향후 계획

본 논문에서는 웹 서비스 기반의 통합 설계 프레임워크를 구현함으로서, 지리적으로 떨어져있는 여러 분야의 연구자들이 통합적으로 연구개발 할 수 있는 소프트웨어 환경을 제시하였다. 그리고 그 작업들의 흐름을 자동으로 통합 관리할 수 있는 가능성을 제시하였다. 향후 과제로는 앞으로 더욱더 연구개발하여 비즈니스 분야에서도 사용할 수 있는 범용성 있는 통합 프레임워크를 개발하는 것이다.

참고문헌

[1] P.Scott Zink, "New Approaches to High Speed Civil Transport Multidisciplinary Design and Optimization"
[2] Web Service : http://www.ibm.com/kr/
[3] BPEL4WS : http://www-128.ibm.com/developworks/library/ws-bpelcol1/
[4] ModelCenter : Andrew Thomas Scott, "An evaluation of three commercially available integrated design framework packages for use in the Space Systems Design Lab"