

프로세스 중심의 동적 엔지니어링 웹서비스 지원 방법에 대한 연구

이재열*, 윤장혁**, 이순재***, 김 현****, 김광수*****

A Study on Supporting Process-Centric Dynamic Engineering Web Services

Lee, J. Y.*, Yoon, J.H.***, Lee, S.J.***, Kim, H.**** and Kim, K.S.*****

ABSTRACT

Collaborative and distributed engineering web services are emerging as a viable alternative to the traditional design and engineering process automation. Existing approaches have limitations in supporting long-running engineering transactions, automatic engineering process orchestration and choreography, synchronous and asynchronous conversation, and geometric abstraction for transmission and sharing. In this paper, we present a process-centric engineering web service methodology to overcome these limitations by utilizing BPEL-based (Business Process Execution Language) process templates and coordination broker-based conversation support. This paper discusses the synchronization of engineering web services which can be either peer-centric or process-centric to support long-running engineering transactions and conversation. The process orchestration and choreography broker works as a service dispatching and aggregation mediator for executing process templates, which enables the individual activity of the engineering process to dynamically select and invoke one of the alternative web services through the run-time process brokering. Further, the paper presents how to support collaboration over the running process using conversation policy.

Key words : engineering web services, business process management, BPEL, process orchestration and choreography, conversation policy

1. 서 론

인터넷의 발달로 기업 비즈니스 환경이 글로벌화되면서, 기업들은 인터넷으로 연결된 사이버 공간 상에서 기업 활동을 수행할 수 있는 고도의 환경을 요구하고 있다. 이러한 환경 변화에 따라 기존의 대기업 중심의 고정적이며 수직적이었던 협력업체 또는 공급업체 관계는 점차 유동적이며 수평적인 관계로 전환될 것이다. 즉 각 기업들은 핵심 역량만을 갖고 그 이외의 부문에 대해서는 가장 우수한 전략적 파트너를 사이버 공간 상에서 찾아 이들이 신속한 팀(Agile Virtual Team)을 구성하여 기업 활동을 수행하는 이른

바 가상기업이 출현하게 될 것이다. 가상기업 환경에서 가상 중요한 것은 협업이며, 앞으로 어떠한 형태로 협업을 효과적으로 지원할 것인가가 중요한 이슈로 떠오를 것이다¹⁾.

또한, 정보기술의 발전과 분산된 가상기업 환경은 제품개발 프로세스 환경을 급속히 변화시키고 있다. 그 환경은 글로벌화되고, 네트워크 기반의 분산된 환경을 의미한다. 이러한 변화는 제품 설계 및 엔지니어링 서비스 과정에서 지리적인 한계 및 시간적인 한계를 극복하여 보다 효과적인 커뮤니케이션을 가능케 하고, 보다 폭넓은 제품 정보를 획득하고 교환할 수 있도록 도와주고 있다. 특히, 웹 기술의 발전과 보편화로 인해 향후에는 네트워크 및 분산환경 하에서 협업적 제품개발 및 이와 관련된 많은 작업들이 이루어질 것이다²⁾.

하지만, 지금까지 연구는 주로 기업의 제품개발 및 생산에서의 특정한 부분의 생산성을 높이고 작업을 신속하게 처리하는데 초점이 맞추어져 있었다. 다양한 설계 및 제품정보 시스템의 도입으로 설계 작업이

*중신회원, 전남대학교 산업공학과

**LG-CNS

***포항공과대학교 산업공학과

****중신회원, 한국전자통신연구원 소프트웨어로봇연구팀

*****중신회원, 포항공과대학교 산업공학과

- 논문투고일: 2004. 05. 03

- 심사완료일: 2004. 08. 02

자동화되어 각 단위 업무는 전산화되고 자동화되어서 효율이 높아졌으나, 단위 업무간의 정보의 교환이 원활히 이루어지지 못하고, 단순한 기능 중심의 협업만을 지원하여 업무의 병목 현상이 생기게 되었다¹⁷⁾.

이러한 문제점을 해결하기 위하여 협업프로세스 중심의 효과적인 설계 및 실행은 새로운 제품개발에서 중요한 요소 중의 하나로 새롭게 인식되어 왔으나, 분산컴퓨팅의 부재, 애플리케이션 통합의 어려움 (Enterprise Application Integration, EAI), 이질 시스템간의 커뮤니케이션의 어려움, 협업프로세스 표준의 부재 등으로 인하여 프로세스 중심의 엔지니어링 서비스가 제대로 이루어지지 못하고 있다.

기존의 통합된 시스템에서는 단위 중심의(Point-to-Point 혹은 Peer-to-Peer) 일차원적 트랜잭션을 지원하는 DBMS가 중요하였으나, 협업제품개발 환경하에서는 다중의 비즈니스 파트너간의 협업을 위한 비동기적인 다차원 메시지 전달을 지원할 수 있는 프로세스 중심의 엔지니어링 서비스 환경이 중요하다. 이와 같은 협업프로세스 기반의 애플리케이션 아키텍처의 발전은 Fig. 1(a)와 같은 협업 형태에서 Fig. 1(b)와 같

은 형태로의 발전이 진행되고 있다¹⁷⁾.

또한, 정보기술의 발전으로 XML, WSDL, SOAP, UDDI와 같은 표준접근기술에 기반을 둔 웹서비스 기술의 발전에 힘입어 비즈니스 액티비티의 성격을 지닌 웹서비스 컴포넌트들을 조합하여 복잡한 비즈니스 프로세스 컴포넌트를 모델링하고 구동할 수 있는 BPEL4WS(Business Process Execution Language for Web Services)가 발표되었다¹⁸⁾¹⁹⁾. 이는 단순한 모듈이라는 개념을 가졌던 기존의 웹서비스를 커다란 비즈니스 프로세스를 구성하는 비즈니스 액티비티로 볼 수 있도록 하였다.

따라서, 가상기업 환경하에서 협업적 제품개발을 효과적으로 지원하기 위해서는 프로세스 중심의 서비스가 제공되어야 하며 이를 위해서는 1) 표준화된 엔지니어링 프로세스 정의, 2) 동적인 서비스 실행 및 협업, 3) 플랫폼에 독립적이고 분산 이질환경에서의 응용시스템 간의 효과적인 커뮤니케이션, 4) 프로세스 상에서 제품정보의 공유 및 전송을 효율적으로 지원할 수 있는 방법 등을 개발해야 한다. 특히, 제품정보가 프로세스 상에서 효과적으로 전송 및 공유될 수 있도록 개방적이고 느슨한 결합(Loosely-Coupled)을 지원하는 프로세스를 생성하고 실행하는 방법론 개발이 필요하다. 하지만, 이러한 환경을 효과적으로 지원할 수 있는 프로세스 중심의 엔지니어링 서비스에 대한 연구가 미흡한 실정이다. 따라서 본 연구에서는 협업 제품개발을 효과적으로 지원하기 위한 프로세스 중심의 엔지니어링 웹 서비스 지원 방법을 제시한다. 특히 본 연구에서는 비동기적이고 장시간이 걸리는 엔지니어링 트랜잭션 (Asynchronous and Long-Running Engineering Transaction), 프로세스 제어 및 연동 (Process Orchestration and Choreography), 프로세스 상에서 대화(Conversation) 및 협업 지원에 중점을 둔다. 또한, 서비스 동기화(Synchronization) 관점에서 단위 중심의 (Peer-Centric) 서비스와 프로세스 중심의 (Process-Centric) 서비스의 개념을 제시하고 각 개념에 따른 서비스 실행 및 협업방법을 설명한다. 그리고, BPEL기반의 엔지니어링 프로세스 템플릿을 정의하고, 프로세스 실행 시 동적바인딩(Dynamic Binding)과 대화 및 협업을 지원할 수 있는 방법을 설명한다.

본 논문의 구성은 다음과 같다. 제 2장에서 기존연구 결과를 간략히 살펴본다. 제 3장은 엔지니어링 서비스의 개념과 동기화에 대해 설명하고, 4장에서는 동기화에 따른 엔지니어링 서비스 실행을 구체적으로 설명한다. 제 5장에서는 시스템 구현을 설명하고, 6장에서 결론을 맺는다.

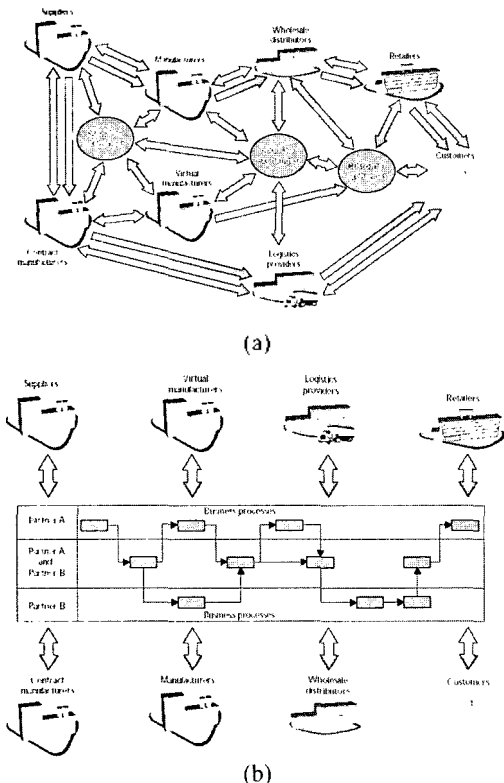


Fig. 1. Change of Collaborative Processes¹⁷⁾.

2. 관련연구

정보기술의 발전에 따라서 분산환경 하에서 엔지니어링 서비스 및 협업에 대한 요구가 급격히 증가하고 있다. 이러한 개발환경변화에 따라 가상 공동작업 환경하에서 설계, 가공 및 생산과 관련된 일들을 동시에 처리할 수 있는 방법론 개발에 많은 연구가 이루어져 왔다. 여기에는 가상기업간의 공동작업을 위한 인프라구조, 분산환경 하에서의 설계 및 생산결합, 분산환경 하에서 설계 및 프로세스 통합 등의 연구 결과들이 있었다^{11,12}.

또한, 분산환경하에서 협업제품개발을 효과적으로 수행하기 위해서는 인터넷 상에서 어떻게 제품정보를 공유하고 교환할 것인지에 대한 연구가 필요하다¹³. Hardwick 등은 가상기업 환경에서 제조정보의 공유를 위해 STEP표준과 CORBA(Common Object Request Broker Architecture) 표준을 이용한 응용시스템 간의 상호운용성 방안을 제시하였다¹⁴. 하지만 기업간 제품정보의 공유를 위해서는 제품의 라이프사이클을 고려하는 정보 스키마가 요구되며, 느슨하게 결합된 구조에서의 시스템간 상호운용성이 요구된다.

복잡한 제품설계 및 서비스를 효과적으로 수행하기 위해서는 협업프로세스를 정의하고 공유하고 실행하는 과정이 필요하다. Kim 등은 기업간 제품정보 공유 프레임워크를 제시했다¹⁵. Lee 등은 분산환경하에서 파트 및 어셈블리 모델링을 지원하는 가상모형 개발 방법론을 제시하였다^{16,17}. Hansen 등은 개방적이고 동적인 B2B를 가능하도록 대화모델(Conversational Model)을 제시하였다^{18,19}. 대화모델은 메시징 시스템 그리고 대화정책에 따라 대화를 수행할 수 있다. 대화정책은 웹서비스 기술 기반의 비즈니스 컴포넌트를 제공해 주는 파트너들 간의 유연한 협업에 이용될 수 있다.

또한, 시멘틱 웹 기술의 발전으로 지식의 재사용과 공유, 가상조직 간 조화와 협력 증대, 유비쿼터스 컴퓨팅환경하에서 정보 연결 용이성 증대가 가능해졌다^{15,18}. 하지만, 이들 연구는 분산·협업환경 하에서 대부분 기능 중심의 협업을 강조하였으며 표준화된 프로세스 중심의 협업을 통한 엔지니어링 혹은 설계 서비스를 제대로 지원하지는 못했다.

3. 엔지니어링 웹서비스 개요

3.1 엔지니어링 웹서비스 프레임워크

다양한 엔지니어링 서비스를 제공하기 위해서는 단

위시스템의 서비스뿐만 아니라 단위시스템을 프로세스로 연결하여 (즉, 프로세스 제어 및 연동) 새로운 서비스를 제공할 수 있어야 한다.

이러한 개념을 실현하기 위해서 본 연구에서는 Fig. 2에서 볼 수 있는 3-레이어에 기반을 둔 프레임워크를 제시한다: 1) 커뮤니케이션 레이어, 2) 프로세스 레이어, 3) 적용 레이어. 웹서비스 기반의 커뮤니케이션 레이어는 엔지니어링 웹서비스 요청자와 답변자 사이의 양방향 통신을 가능케 한다. 특히, 비동기적이고 장시간을 요하는 엔지니어링 트랜잭션을 지원한다. 프로세스 레이어는 서비스의 검색, 처리, 전송 및 결합을 위한 브로커 역할을 한다. 특히, 프로세스 템플릿과 병합 브로커의 도움으로 논리적인 엔지니어링 프로세스 결합 및 실행을 지원한다. 적용 레이어는 엔지니어링 서비스를 웹서비스로 표현 및 구현하고 이를 등록, 검색 할 수 있도록 지원한다. 등록된 서비스는 기존의 레거시(Legacy) 시스템과 연동된다.

예를 들면, Fig. 2에서 볼 수 있듯이, 사용자가 유비쿼터스 환경하에서 퍼블릭 인터페이스를 통해서 특정한 엔지니어링 서비스를 요청할 경우(예, 어셈블리 컴포넌트간의 간섭확인 혹은 솔리드 모델의 가시화 모델로의 변환), 그 요청은 우선 프로세스 브로커로 전송이 된다. 프로세스 브로커는 엔지니어링 서비스 저장소(Repository)를 검색하여 적합한 엔지니어링 서비스 템플릿을 검색한다. 검색이 이루어지면 브로커는 프로세스 상의 각 액티비티를 실행할 수 있는 웹서비스를 찾은 후 연동한다. 일반적으로 프로세스 템플릿은 하나 혹은 그 이상의 웹서비스로 구성되어 있다. 어떤 경우에는, 하나의 액티비티를 실행하기 위해서 여러 개의 서비스 제공자 중 사용자의 요구사항에 따라 가장 적합한 서비스를 선택해야 하는 경우도 있다. 최종적으로 적용 레이어에서 프로세스 액티비티에 해당되는 엔지니어링 서비스를 호출하게 된다.

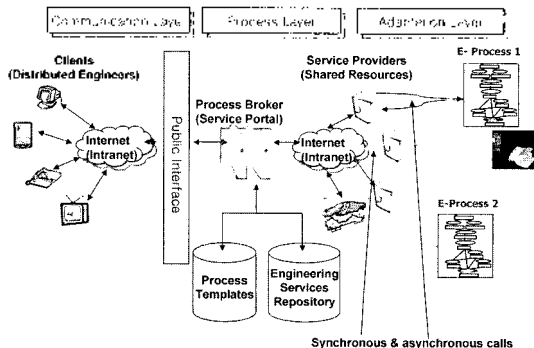


Fig. 2. Concept of Engineering Web Services.

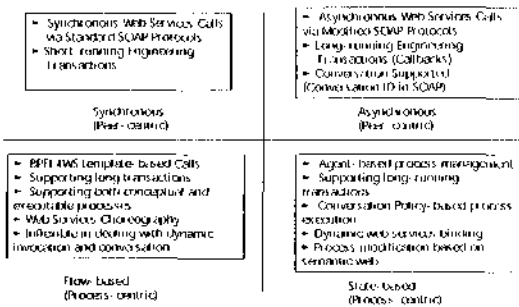


Fig. 3. Synchronization of Engineering Web Services.

3.2 엔지니어링 웹서비스 동기화

엔지니어링 웹서비스는 제품정보(Product), 프로세스(Process) 및 자원(Resources)을 통합할 수 있도록 지원하는 프레임워크 기술로써, 서비스 동기화 관점에서 다음의 문제점을 해결하기 위한 연구가 필요하다(Fig. 3 참조).

- 첫번째는 단위 중심의 엔지니어링 웹서비스 지원 문제다. 엔지니어링 서비스는 간단한 프로시저 혹은 웹 서비스 호출로는 복잡한 서비스를 제공할 수 없다. 웹서비스는 일반적으로 HTTP 프로토콜을 사용하기 때문에 동기화된 서비스 호출만을 허용한다. 따라서, 중요한 것은 비동기적이고 장시간이 걸리는 트랜잭션 지원이나 세션 유지, 대화를 어떻게 지원할 수 있을 것인가 하는 문제다.
- 두번째는 프로세스 중심의 엔지니어링 웹서비스 지원문제다. 단위 중심의 서비스를 결합하여 새로운 서비스를 창출할 수 있어야 한다. 이를 위해서는 어떠한 표준화된 프로세스 모델을 사용할 것인가 하는 문제다.
- 세번째는 프로세스 템플릿 지원문제다. 다양한 엔지니어링 서비스 프로세스가 정의된 후 실행시 어떻게 효과적으로 단위 서비스를 연결할 것인가 하는 문제다.
- 네번째는 프로세스의 유연성을 높이고 보다 다양한 협업을 지원하는 문제다. 피어 중심의 서비스가 자동화된 서비스 요청이 아니라 다양한 서비스 제공자 간의 협업 및 대화를 지원할 수 있어야 한다. 이를 어떻게 지원할 것인가 하는 문제다.
- 다섯번째는 제품정보 효율적인 공유 및 전송문제이다. 분산 협업 프로세스상에서 제품정보를 공유하고 효율적으로 전송 및 전송하기 위해서 어떻게 형상모델을 표현해야 하고 추상화시킬 것인가 하는 문제다.

동기화 관점에서는 크게 단위 중심의 서비스와 프로세스 중심의 서비스로 구분할 수 있다(Fig. 3). 또한, 단위 중심의 서비스는 서비스의 호출 방법에 따라 동기식과 비동기식으로 세분화될 수 있다. 또한 프로세스 중심의 서비스는 플로우 기반(Flow-based) 서비스와 상태 기반(State-based) 서비스로 세분화시킬 수 있다. 일반적으로 대부분의 서비스는 플로우 기반의 프로세스로 표현이 가능하다. 하지만, 라이프사이클 관리, 프로세스 브로커링, 대화 및 내용 통합(Content Aggregation) 등을 효율적으로 지원하기 위해서 플로우 기반의 프로세스만 사용할 경우 복잡도가 크게 증가하여 프로세스를 실행시킬 수 없는 경우가 자주 발생한다^[13]. 따라서, 프로세스 상에서 다양한 형태의 협업을 지원하기 위해서는 플로우 기반 및 상태기반 서비스 방식을 모두 지원해야 한다.

4. 엔지니어링 웹서비스 실행

4.1 비동기적 단위 중심의 서비스 실행

엔지니어링 서비스의 특성상 예측 불가능하고 아주 오랜 시간을 요하는 작업이 생길 수 있다. 또한 서비스를 수행하는 것이 사람일 수도 있기 때문에 엔지니어링 웹서비스는 이러한 문제점을 해결할 수 있어야 한다. 즉, 장시간이 걸리는 트랜잭션이나 대화를 지원하기 위해서는 비동기식 서비스를 반드시 지원해야 한다.

웹서비스는 HTTP 프로토콜의 Request-Response 프로토콜을 따르기 때문에 비동기식 서비스를 지원하기 위해서는 기존의 웹서비스 호출과 관련된 부분이 수정되어야 한다. 즉, SOAP 메시지가 수정되어야 하며 서버 및 클라이언트 측에서도 이를 수용할 수 있어야 한다. 본 연구에서는 Fig. 4에서 볼 수 있듯이 SOAP HEADER에서 <conversationID>와

```

The three SOAP headers related to conversations are
<<StartHeader>>: The client sends this header with the call to a method marked to "start" a conversation. The <StartHeader> contains <conversationID> and <callbackLocation> elements.
<<ContinueHeader>>: The client sends this with any subsequent calls to the web service's methods. It contains the conversation ID the client used to start the conversation.
<<CallbackHeader>>: The client receives this header when it receives a callback from the web service. The <CallbackHeader> includes the conversation ID the client used to start the conversation.

<SOAP:ENV:Header>
  <StartHeader xmlns="http://www.openurl.org/2002/04/soap/conversation/">
    <conversationID> someUniqueIdentifier /> <conversationID>
    <callbackLocation> http://www.mydomain.com/myClient /> <callbackLocation>
  /> <StartHeader>
</SOAP:ENV:Header>
<SOAP:ENV:Header>
  <ContinueHeader xmlns="http://www.openurl.org/2002/04/soap/conversation/">
    <conversationID> theUniqueIdentifierStartWithTheHeader /> <conversationID>
  /> <ContinueHeader>
</SOAP:ENV:Header>
  
```

Fig. 4. Modified SOAP message for supporting callbacks and conversations.

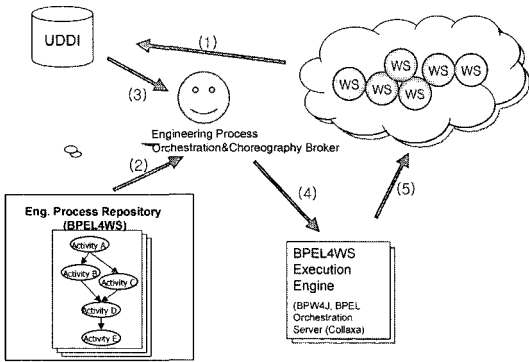


Fig. 5. Process templates and process broker.

<callbackLocation> 태그를 추가하여 대화와 콜백 (Callback)을 지원한다^[10]. 특히, 다양한 형태의 대화를 지원하기 위해서 <StartHeader> <ContinueHeader>, 그리고 <CallbackHeader>태크가 사용된다.

4.2 프로세스 중심의 서비스 동적 실행

프로세스 중심의 엔지니어링 웹서비스는 단위 중심의 서비스를 결합하여 새로운 서비스를 생성한다. 이를 올바르게 지원하기 위해서는 아래의 이슈를 고려해야 한다.

- 단위 중심의 엔지니어링 웹서비스 결합을 지원하는 표준 프로세스 채택
- 다양한 엔지니어링 서비스 지원을 위한 프로세스 템플릿 지원
- 동적바인딩과 대화 지원
- 웹서비스의 시맨틱 매칭

본 연구에서는 BP4LWS(Business Process Execution Language for Web Services)기반 엔지니어링 서비스 템플릿을 제시하여 엔지니어링 애플리케이션과 프로세스의 통합을 실현한다. Fig. 5는 제시된 프로세스 중심의 엔지니어링 웹서비스 개념을 보여준다. 새로운 엔지니어링 서비스가 생성된 경우, 우선 UDDI에 등록이 된다. 등록할 때 사용되는 정보는 WSDL이나 OWL-S 프로파일에 근거를 둔다. 또한 엔지니어링 서비스 제공자는 다양한 경험을 바탕으로 BP4LWS 기반의 엔지니어링 템플릿을 작성한 후 프로세스 저장소에 저장한다. 이를 기반으로 엔지니어링 서비스 요청이 있을 경우 프로세스 브로커는 등록된 서비스 프로파일을 기반으로 검색한 후 올바른 프로세스 템플릿을 찾게 된다. 찾은 후 현재 서비스가 가능한 웹서비스를 프로세스 액티비티에 연동한 후

```

<process name= TheProcess >
  <partnerLinks> for listing the external web services invoked
  from within the workflow
  <variables>, <assign>, <copy> and XPath queries for data storage
  and manipulation
  <correlationSets> for specifying bindings for a set of operations to a
  service instance
  <invoke> and <invoke> <receive> for consuming synchronous and
  asynchronous web services
  <scope> and <faultHandlers> for exception management
  <flow> and <sequence> for parallel execution paths
  <switch> for conditional branching
</process>

BP4LWS Configuration
  Conceptual process
    xxx bpel: a BP4LWS source file
    xxx wsdl: a WSDL file describing the interface
  Executable process
    bpel.xml: a deployment descriptor
    
```

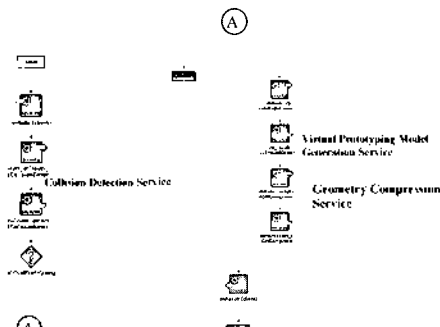
Fig. 6. BP4LWS-based process specifications.

서비스를 실행한다.

4.2.1 BP4LWS 기반의 프로세스 템플릿

프로세스 템플릿의 근간이 되는 BP4LWS는 SOA(Service-Oriented Architecture)를 기반 아키텍처로 삼고 있으며 이미 만들어져 있는 웹서비스 기반의 컴포넌트를 이용하여 내부의 새로운 비즈니스 프로세스를 창출하는 용도로 이용된다^[10]. BP4LWS 프로세스 모델은 플로우 기반의 구조가 표현되어 프로세스 구동시에 프로세스 모니터링이 가능하며, 느슨한 결합에 따른 프로세스 수행 시에 생길 수 있는 예외처리와 실패에 대한 보상(Compensation)을 지원한다(Fig. 6 참조). 또한 이미 만들어져 있는 비즈니스 내외부의 컴포넌트들을 조합하여 사용할 수 있는 기반 기술이므로 컴포넌트들의 재사용을 용이하게 하여 시스템의 라이프 사이클을 늘일 뿐만 아니라, 프로세스 모델링을 하는 것만으로도 비즈니스 프로세스를 곧바로 구동시킬 수 있는 강점을 지닌다. 비즈니스 프로세스를 웹기반 서비스 컴포넌트로 인식해 보면, 중심이 되는 BP4LWS 프로세스는 추상적인 모델이며, 비즈니스 프로세스를 구성하는 개별 액티비티는 실제로 다른 비즈니스 프로세스 컴포넌트를 호출함으로써 수행된다.

Fig. 6에서 볼 수 있듯이 BP4LWS기반의 프로세스는 개념 프로세스(Conceptual Process)와 실행 프로세스(Executable Process)로 분리시킬 수 있다. 개념 프로세스는 프로세스 정의, 인출력 데이터 표현 등을 담당하고 실행 프로세스에서는 실행시 프로세스상의 각 액티비티의 서비스 주소를 연동하는 역할을 담당한다. 따라서, 프로세스 템플릿은 개념 프로세스에 근간을 두고, 프로세스 구동시 프로세스 브로커의 도움으로 각 액티비티의 서비스를 동적으로 연결함으로써



The above two processes are linked together A and A points.

Fig. 7. A process template based on BPEL4WS representation.

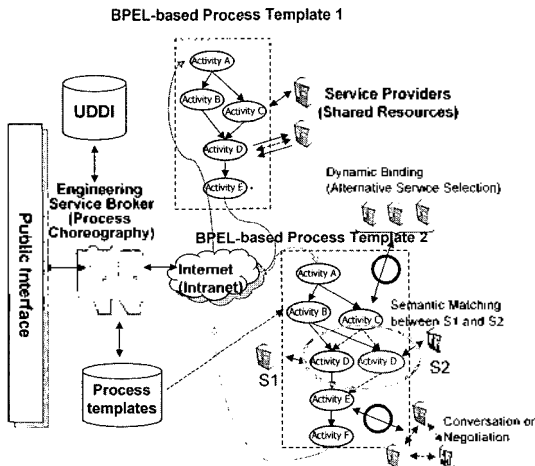


Fig. 8. Problems inherent in BPEL-based process templates.

다양한 서비스를 제공하게 된다. Fig. 7은 BPEL4WS에 기반을 둔 프로세스 템플릿 예를 보여준다. 프로세스 템플릿은 일련의 3개의 웹서비스를 호출함으로써 입력된 어셈블리 모델로부터 압축된 가시화 모델을 생성시킨다: 1) collision detection service, 2) virtual prototyping model generation service, 3) geometry compression service. 만약, 첫번째 서비스 호출에서 어셈블리 컴포넌트간에 간섭이 발생한 경우는 더 이상의 프로세스 실행을 진행시키지 않는다. 상세한 XML 표현은 Appendix A에 나와 있다.

하지만, 보다 다양한 엔지니어링 웹서비스를 지원하기 위해서는 대안 웹서비스(Alternative Web Services) 선택 및 연동, 협상 등을 지원해야 한다¹⁸⁾. Fig. 8에서 볼 수 있듯이 BPEL로 정의된 프로세스를 실행할 때 프로세스 엔진은 각 액티비티에 해당되는

서비스를 호출하게 되는데, 이때 서비스 호출에서 문제점이 발생할 수 있다. 이런 경우 BPEL에서는 예외로 처리한다. 하지만, 새로운 대안 서비스를 찾아 연결하거나 여러 가지 의사결정 과정을 거친 후 가장 적합한 대안 서비스를 선택해야 할 경우가 있다. 또한, 특정한 액티비티를 중심으로 다수의 참여자가 대화를 하고 그 결과에 따라 프로세스가 진행되는 경우도 있다. 첫번째를 동적바인딩이라고 하며 두번째를 대화지원(Conversation Support)라 한다. 하지만, BPEL은 이용될 웹서비스들의 순서를 구조화하는 핵심 비즈니스 프로세스 설계를 목적으로 한 것이지, 핵심 프로세스와 대안 웹서비스가 연결되기 이전의 과정을 표현하기 위한 것은 아니다. 본 연구에서는 조정자(Coordination Broker) 및 대화정책을 이용하여 이러한 문제점을 해결한다.

4.2.2 동적 바인딩과 대화지원

본 연구에서는 동적바인딩과 다사간의 협업을 지원하기 위해서 조정자(Coordination Broker)를 이용한다. 프로세스의 특정 액티비티에 웹서비스로 구현된 브로커를 연결시켜 이러한 문제점을 해결한다(Fig. 8과 Fig. 9 참조). 본 연구에서 사용되는 조정자에는 동적연동 조정자(Dynamic Binding Broker), 대화지원 조정자(Conversation Broker), 그리고 협상 조정자(Negotiation Broker)가 있다.

동적연동 조정자는 대안 웹서비스를 선택 및 연동이 가능하도록 도와준다. 대안 웹서비스는 우선 등록 과정을 통해서 동적연동 조정자에 등록을 한다. 등록된 서비스를 대상으로 다양한 기준을 근거로 동적바인딩을 하게 된다. 자세한 내용은 참고문헌 8과 9를 참조하기 바란다.

대화지원 조정자나 협상 조정자는 상태기반 프로세스 모델링 개념에 기반을 두어 서비스를 제공한다. 상

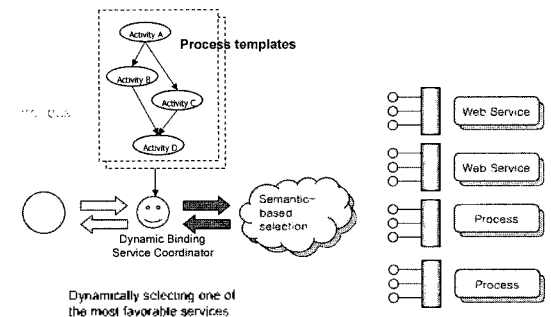


Fig. 9. Dynamic Binding.

- 1: Collision Testing
- 2: Collision Found
- 3: Done
- 4: Conflict Resolved
- 5: Unresolved

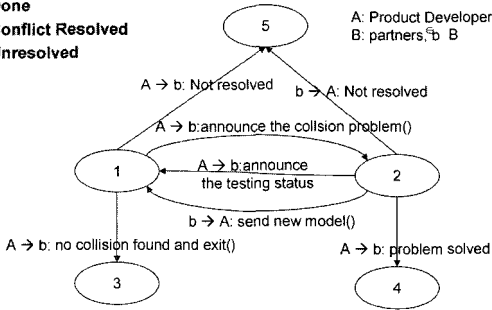


Fig. 10. Conversation Policy for Virtual Prototyping.

테기반 프로세스 모델링이란 상태를 중심으로 프로세스를 기술하는 모델링 기법이다. 이는 특정 상태에서 주어진 종류의 사건을 수행할 수 있고 이에 따라 다른 상태로 전환되는 것을 표현한 것으로 상태 머신(State-Machine)으로 표현할 수 있다. 본 연구에서는 이를 지원하기 위해 cpXML의 대화정책을 이용한다^[13]. Fig. 10은 대화지연 조정자를 중심으로 가상모형개발 서비스(Virtual Prototyping)에 이용되는 대화정책을 보여준다. 두 기업이 참여하여 간섭체크를 하는 상태 전이 다이어그램(State Transition Diagram)을 보여준다. 결국, 다양한 엔지니어링 웹서비스를 지원하기 위해서는 플로우 기반의 프로세스를 중심으로 상태기반의 대화정책을 함께 수용해야 한다. 조정자가 이러한 역할을 담당한다.

하지만, 프로세스 템플릿의 액티버티가 웹서비스 컴포넌트를 호출할 때 입력, 출력 등과 관련된 시맨틱 매칭(Semantic Matching)을 지원하지 못한다. Fig. 8에서 볼 수 있듯이 사용자의 요구를 반드시 만족시키는 프로세스 템플릿이 존재하지 않을 수 있다. 이러한 경우 필요에 따라서는 고객의 요구사항을 어느 정도 만족시키는 프로세스 템플릿을 찾고 이를 수행해서 엔지니어링 서비스를 제공해야 하는 경우도 발생할 수 있다^[20]. 이러한 경우를 지원하기 위해서는 액티버티의 시맨틱 매칭 과정이 필요하며 결과에 따라 프로세스를 수정해야 할 필요가 있다. 본 연구에서는 OWL-S의 프로파일 정보를 이용하여 온톨로지 매핑(Ontology Mapping) 과정을 통해서 시맨틱 매칭을 지원하려고 한다. 즉, 입력, 출력 파라미터에 사용되는 클래스를 온톨로지로 표현한 후 온톨로지 매핑을 수행한다. 뿐만 아니라, 프로세스 상에서 액티버티의 상속 개념을 도입하여 프로세스 액티버티와 관련된 시맨틱 매칭을 수행코자 한다. 하지만, 시맨틱 매칭에 관

한 연구가 현재 진행 중이며, 추후 연구과제이다.

4.3 제품형상 정보 공유 및 전송을 위한 형상 추상화

지금까지 설명된 단위 중심이나 프로세스 중심의 엔지니어링 서비스는 주로 동기화 관점이었다. 제품 개발과 분산 컴퓨팅에 근거한 협업 제품 개발 과정에서는 제품정보(특히, 형상정보)의 공유 및 전송이 올바르게 빠르게 이루어져야 한다^[9,21]. 특히, 협업을 위한 다양한 상호작용을 지원하기 위해서는 형상정보의 추상화와 일관된 형상명명 기법(Naming Consistency)을 지원해야 한다.

뿐만 아니라, 실행중인 엔지니어링 프로세스 상에서 서비스 제공자와 서비스 요청자 모두 분산환경에서 각자 필요한 정보를 지니고 있어야 한다. 따라서, 이들간의 원활한 커뮤니케이션을 하기 위해서는 각 형상요소는 제공자 및 요청자 사이에서 공통된 고

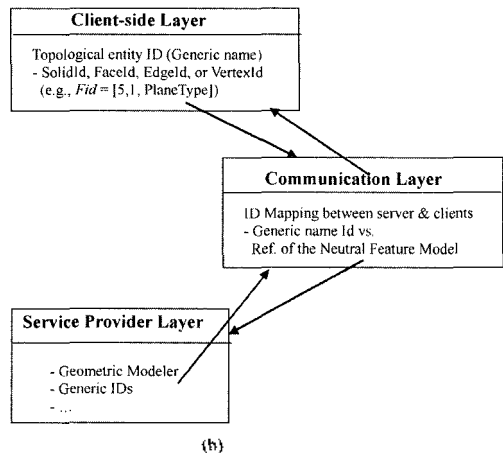
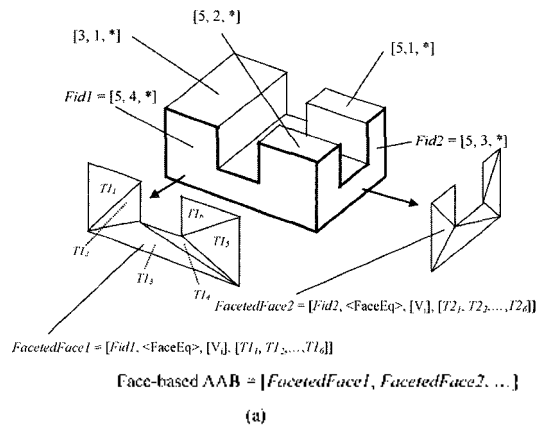
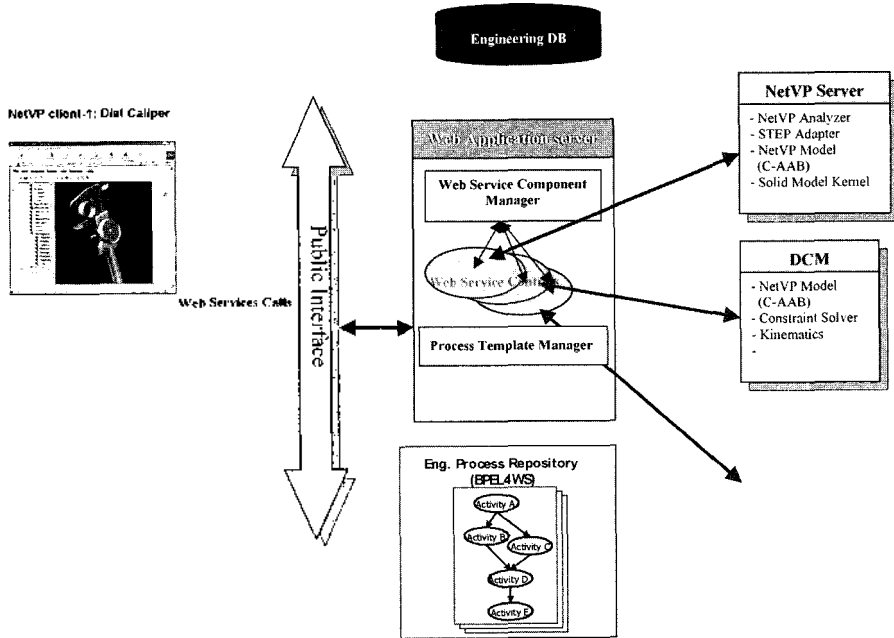


Fig. 11. ID mapping between service provider and client based on AAB.

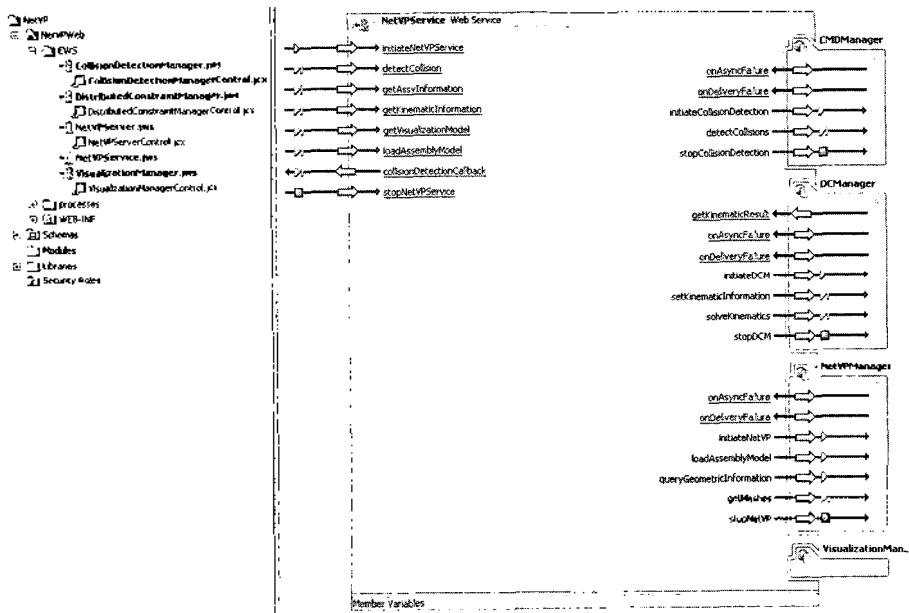
유한 이름을 지녀야 한다. 이를 위해 본 연구에서는 Attributed Abstract B-Rep의 개념을 제시한다(Fig. 11 참조)²²⁾. 결국, AAB은 추상화된 삼각망과 커뮤니티 케이스 ID역할을 하는 고유이름(Generic Name ID)으

로 구성되어 있다.

하지만, AAB 역시 인터넷 상에서 빠르게 공유 및 전송되기에는 상당한 크기를 차지한다. 이러한 분쇄 작업을 해결하기 위해서 Edgebreaker 알고리즘²³⁾ 이



(a) Network-enabled Virtual Prototyping Using Web Services



(b) Engineering Web Services Components implemented in WebLogic™

Fig. 12. Provided engineering web-services for network-enabled virtual prototyping.

용하여 AAB를 압축한 Compressed AAB(C-AAB)를 활용한다. 실험결과, C-AAB의 크기는 AAB에 비해 5%이하임을 볼 수 있었다. 궁극적으로, C-AAB를 이용하여 효과적인 제품정보 공유와 상호작용을 지원할 수 있으며, 다자간의 협업 또한 지원할 수 있다.

5. 시스템 구현

본 장에서는 제시된 프레임워크가 어떻게 엔지니어링 웹서비스를 효과적으로 지원할 수 있는지에 대한 구현결과를 제시한다. 우선 분산환경하에서 가상 모형 기술을 지원할 수 있는 NetVP(Network-centric Virtual Prototyping) 시스템을 설명한다. NetVP시스템은 Fig. 12에서 볼 수 있듯이 NetVP Server, Distributed Constraint Manager(DCM), Collision Detection Manager(DCM) 등의 서비스 컴포넌트로 구성되어 있다. NetVP Server는 가상모형 설계 및 분석에 필요한 다양한 기능을 제공한다. 특히, NetVP Analyzer는 NetVP Model을 분석할 수 있는 기능을 제공하고, NetVP Model은 C-AAB를 의미한다. ACIS™가 솔리드 모델링 커널로 이용되고 있다. DCM은 분산된 형상 제약조건을 해결할 수 있는 기능을 제공한다. 따라서 분산환경하에서 동적 시뮬레이션 및 어셈블리 모델링을 지원할 수 있다. CDM은 C-AAB를 기반으로 한 빠른 간섭확인(Rapid Collision Detection)을 지원한다. NetVP 서비스는 WebLogic™에서 구현되었으며, 각 서비스 제공자는 웹서비스 콘트롤러로 구현되었으며, 각 콘트롤러는 비동기적 엔지니어링 트랜잭션 및 대화를 지원한다.

또한, 프로세스 중심의 엔지니어링 웹서비스 프레임워크의 장점을 보여주기 위해서 Fig. 13에서 도시된 프로세스 템플릿 기반의 엔지니어링 서비스 구현결과를 설명한다. 제시된 템플릿은 일련의 4가지 엔지니어링 서비스 호출로 구성되어 있다. 서비스 A는 STEP 데이터를 ACIS 데이터로 변환시키고, 서비스 B는 ACIS 데이터를 기반으로 간섭확인을 수행하고, 서비스 C는 제조가능성(Manufacturability)을 분석하고, 서비스 D는 C-AAB를 생성한다. 특히, 본 템플릿을 수행하기 위해서 두개의 조정자가 이용되었다. 하나는 STEP 데이터 변환을 시키는 서비스에 대한 대안 서비스 선택에 이용되고, 또 다른 하나는 Fig. 10에서 사용된 대화정책을 기반으로 다수의 참여자가 어셈블리 분석을 하도록 지원한다(즉, 더 이상의 간섭이 없을 때까지 상호간의 협업이 이루어짐). 이를 지원하기 위해서 Collaxa™의 BPEL Server가 이용되었으며, 어셈블리 모델은 SolidWorks™에서 설계되었다. 디테일 모델의 크기는 1.987 KB였으며 최종 C-AAB의 크기는 116 KB였다.

6. 결론

가상기업 환경에서 가장 중요한 것은 협업이며 어떻게 협업을 효과적으로 지원할 것인가가 중요한 이슈로 떠오르고 있다. 이 때문에 협업프로세스 중심의 엔지니어링 서비스에 대한 요구가 증대하고 있다.

본 연구에서는 협업 제품개발 및 서비스를 효과적으로 지원하기 위해서 프로세스 중심의 엔지니어링 웹서비스 지원 방법을 제시하였다. 특히, 다양한 서비스를 효과적으로 지원하기 위해서 서비스 동기화 프로세스 중심의 서비스, 프로세스 템플릿, 동적마인딩 및 대안 서비스 선택, 그리고 대화정책 기반의 협업에 관한 구체적인 방법을 설명하였다.

추후 연구로는 다양한 조정자 개념을 도입하고 폭넓은 시맨틱 매칭 문제를 해결할 수 있는 방법을 개발하는 것이다. 또한, 다양한 프로세스 템플릿을 설계하고 이를 협업제품개발과정에 적용시키는 것이다.

감사의 글

이 논문은 2003년도 한국학술진흥재단의 지원에 의하여 연구되었음(KRF-2003-003-D00558).

참고문헌

1. 김 현, 김형선, 이주행, 정진미, 노남철, 이재열, "기

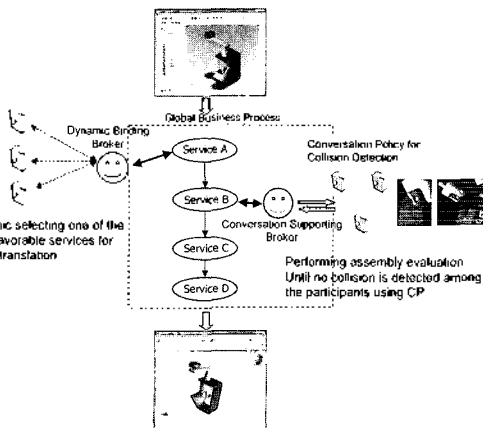


Fig. 13. An assembly evaluation and conversion process template and its execution.

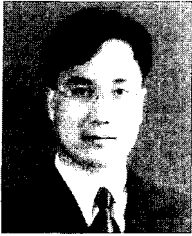
- 업간 제품정보 공유를 위한 협업적 제품거래 프레임워크,” 한국CAD/CAM학회 논문집, 제8권, 제4호, pp. 201-211, 2003.
2. Abrahamson, S., Wallace, D., Senin, N. and Sferro, P., “Integrated design in a service marketplace,” *Computer-Aided Design*, Vol. 32, No. 2, pp. 97-107, 2000.
 3. Hardwick, M., Spooner, D. L., Rando, T. and Morris, K. C., “Sharing manufacturing information in virtual enterprises,” *Communications of the ACM*, Vol. 39, No. 2, pp. 46-54, 1996.
 4. Kim, H., Kim, H.-S., Lee, J.-H., Jung, J.-M., Lee, J. Y. and Do, N. C., “Framework for sharing product information across enterprises,” *Proc. of 10th ISPE International Conf. on Concurrent Engineering: Research and Applications*, 2003.
 5. Lee, J. Y., Kim, H. and Kim, K., “A web-enabled approach to feature-based modeling in a distributed and collaborative design environment,” *Concurrent Engineering: Research and Applications*, Vol. 9, No. 1, pp. 74-87, 2001.
 6. Zhuang, Y., Chen, L. and Venter, R., “CyberEye: an internet-enabled environment to support collaborative design,” *Concurrent Engineering: Research and Applications*, Vol. 8, No. 3, pp. 213-229, 2000.
 7. CSC's Research Services, “The emergence of business process management,” CSC, 2002.
 8. 윤장혁, 이순재, 김광수, “상태기반 협상 모델을 이용한 동적 비즈니스 프로세스 통합,” *대한산업공학회 춘계학술대회*, pp. 40-45, 2003.
 9. 윤장혁, “브로커링 모델을 이용한 비즈니스 프로세스의 동적 통합,” 석사학위논문, 포항공과대학교, 2004.
 10. BPEL4WS, <http://www-106.ibm.com/developerworks/webservices/library/ws-bpel/>
 11. Web Services, <http://www.w3.org/2002/ws/>
 12. Lee, J. Y., “Shape representation and interoperability for virtual prototyping in a distributed design environment,” *International Journal of Advanced Manufacturing Technology*, Vol. 17, No. 6, pp. 425-434, 2001.
 13. cpXML, e-Commerce and Autonomic Computing Department, IBM T.J. Watson Center, <http://www.research.ibm.com/convsupport/papers/cpXML-v1.htm>
 14. Hansen, J. E., Nandi, P. and Kumaran, S., “Conversation support for business process integration,” *Proc. of 6th International Enterprise Distributed Object Computing Conf.*, pp. 65-74, 2002.
 15. OWL, <http://www.w3.org/TR/owl-guide/>, 2003.
 16. RDF, www.w3.org/RDF/, 2003.
 17. RDFS, <http://www.w3.org/TR/rdf-schema/>, 2003.
 18. The OWL Services Coalition, “OWL-S: Semantic Markup for Web Services”.
 19. WebLogic User's Guide, <http://www.bea.com/>, 2003.
 20. Laukkanen, M. and Helin, H., “Composing workflows of semantic web services,” *Proc. of Workshop on Web Services and Agent-based Engineering*, 2003.
 21. Rossignac, J., “Edgebreaker: connectivity compression for triangle meshes,” *IEEE Transactions on Visualization and Computer Graphics*, Vol. 5, No. 1, pp. 41-61, 1999.
 22. 이재열, 김현, 현성배, “네트워크 기반 특징형상 모델링,” 한국CAD/CAM학회 논문집, 제5권, 제1호, pp. 12-22, 2000.

Appendix A

```

<process name="EWS" targetNamespace=http://jaeyeol/ews suppressJoinFailure="yes" xmlns:tns="http://jaeyeol/ews"
xmlns="http://schemas.xmlsoap.org/ws/2003/03/business-process/" xmlns:bpelx="http://www.collaxa.com/bpel/extension"
xmlns:assembly="http://jaeyeol/assembly" xmlns:collision="http://jaeyeol/collision" xmlns:vp="http://jaeyeol/vp"
xmlns:compression="http://jaeyeol/compression">
  <partnerLinks>
    <partnerLink name="client" partnerLinkType="tns:EWS" myRole="EWSProvider"
      partnerRole="EWSRequester"/>
    <partnerLink name="CollisionDetection" partnerLinkType="collision:CollisionDetection"
      partnerRole="CollisionDetectionProvider" myRole="CollisionDetectionRequester"/>
    <partnerLink name="VPModelGeneration" partnerLinkType="vp:VP" partnerRole="VPProvider" myRole="VPRequester"/>
    <partnerLink name="GMCompression" partnerLinkType="compression:Compression" partnerRole="CompressionProvider"
      myRole="CompressionRequester"/>
  </partnerLinks>
  <variables>
    <variable name="input" messageType="tns:EWSRequestMessage"/>
    <variable name="output" messageType="tns:EWSResultMessage"/>
    <variable name="assemblyModel" messageType="assembly:AssemblyAnalysisRequestMessage"/>
    <variable name="assemblyAnalysisResult" messageType="assembly:AssemblyAnalysisResultMessage"/>
    <variable name="assemblyModel" messageType="collision:CollisionDetectionRequestMessage"/>
    <variable name="collisionDetectionResult" messageType="collision:EWSResultMessage"/>
    <variable name="assemblyModel" messageType="vp:VPRequestMessage"/>
    <variable name="vpModel" messageType="vp:VPResultMessage"/>
    <variable name="vpModel" messageType="compression:CompressionRequestMessage"/>
    <variable name="compressedModel" messageType="compression:CompressionResultMessage"/>
  </variables>
  <sequence name="main">
    <receive name="receiveInput" partnerLink="client" portType="tns:EWS" operation="initiate" variable="input"
      createInstance="yes"/>
    <invoke name="request_collision" partnerLink="CollisionDetection" portType="collision:CollisionDetection"
      operation="start_collision_detection" inputVariable="assemblyModel"/>
    <receive name="receive_collision_result"
      partnerLink="CollisionDetection" portType="collision:CollisionDetectionCallback"
      operation="collision_detection_result" variable="collisionDetectionResult"/>
    <switch name="virtualerotyping">
      <case condition="result">
        <terminate/>
      </case>
      <otherwise>
        <sequence>
          <invoke name="generate_vp" partnerLink="VPModelGeneration" portType="vp:VP" operation="initiate_vp"
            inputVariable="assemblyModel"/>
          <receive name="receive_vp" partnerLink="VPModelGeneration" portType="vp:VPCallback"
            operation="vp_result" variable="vpModel"/>
          <invoke name="compress" partnerLink="GMCompression" portType="compression:Compression"
            operation="initiate_compression" inputVariable="vpModel"/>
          <receive name="receive_compressed_model" partnerLink="GMCompression"
            portType="compression:CompressionCallback" operation="compression_result"
            variable="compressedModel"/>
        </sequence>
      </otherwise>
    </switch>
    <invoke name="callbackClient" partnerLink="client" portType="tns:EWSCallback" operation="compression_result"
      inputVariable="output"/>
  </sequence>
</process>

```



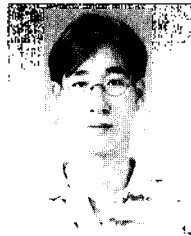
이재열

1992년 포항공과대학교 산업공학과 학사
1994년 포항공과대학교 산업공학과 석사
1998년 포항공과대학교 산업공학과 박사
1998년~2003년 한국전자통신연구원 선임연구원

2003년~현재 한국전자통신연구원 초빙연구원

2003년~현재 전남대학교 산업공학과 조교수

관심분야: Collaborative Virtual Engineering, Engineering Web Services, Business and Engineering Process Management, Ubiquitous and Context-Aware Computing



윤장혁

2002년 포항공과대학교 산업공학과 학사
2004년 포항공과대학교 산업공학과 석사
2004년~현재 LG CNS CALS&CIM팀 연구원

관심분야: Business Process Management, Web Services, Knowledge Management



이순재

1999년 포항공과대학교 산업공학과 학사
2001년 포항공과대학교 산업공학과 석사
2001년~현재 포항공과대학교 산업공학과 박사과정 재학중

관심분야: Business Process Management, Web Services Collaborations, Enterprise Architecture, Domain Specific Language



김현

1984년 한양대학교 기계설계학과 학사
1987년 한양대학교 기계설계학과 석사
1997년 한양대학교 기계설계학과 박사
1998년~1999년 한양대학교 산업공학과 겸임교수

1999년~현재 한국전자통신연구원 소프트웨어로봇연구팀장, 책임연구원

관심분야: Concurrent Engineering, Virtual Engineering, Distributed Collaborative Design, Engineering Knowledge Management, Intelligent System



김광수

1977년 서울공과대학교 산업공학과 학사
1979년 서울공과대학교 산업공학과 석사
1985년 University of Central Florida 산업공학과 박사

1985년~1988년 Rochester Institute of Technology 산업공학과 교수

1988년~현재 포항공과대학교 산업공학과 교수

관심분야: Enterprise Architecture Modeling, Business Process Management, Knowledge Management, Supply Chain Management