

다중병합을 지원하는 SOA기반의 BPIL워크플로우 시스템

SOA based BPIL workflow system to support multi integration

이기호*, 정창성*

Ki-ho Yi*, Chang-Sung Jeong*

e-mail : yikiho@korea.ac.kr*, csjeong@korea.ac.kr*

Abstract

Due to the development and proliferation of internet technology, cooperation and interaction between companies have become very important. However, different heterogeneous systems have been used as information systems for companies because of various types of pressure in the business environment. Therefore, business process integration and interaction are recognized as essential elements, not optional elements. In addition, it is important to make a business process which is valuable as an intellectual property in a storable form and share. So, many business process languages for service integration and GUI type's automatic systems using the same and making business process easily have been introducing. However, to integrate previously created web service[1] workflows, or to use a web service workflow for another workflow production like an activity, that is, research to support multi integration language and system is not ongoing except for research to make and connect adapter between workflow engines.[2] Also, research for current languages and systems that integrate the BPEL[3] process into a workflow in the industrial world does not exist. This means that integration between workflows being used at companies is difficult, and becomes the biggest obstacle to service integration, which is necessary for cooperation and interaction between companies. To solve this problem, in this paper, I defined BPIL based on XML[4], which can store business processes. Also, using this, BPIL workflow system based on SOA[5] which supports multiple integration and workflow profiling was designed and prototype was embodied.

* 高麗大學校 電氣電子電波工學部

(School of Electrical Engineering, Korea University)

★ 교신저자 (Corresponding author)

高麗大學校 電氣電子電波工學部(School of Electrical Engineering, Korea University)

※ 감사의 글 (Acknowledgment)

This research was supported by the MIC(Ministry of Information and Communication), Korea, under the ITRC(Information Technology Research Center) support program supervised by the IITA(Institute of Information Technology Assessment), Brain Korea 21 projects in 2007, Seoul Research and Business Development, Program, Seoul, Korea, and Seoul Forum for Industry-University-Research Cooperation.

接受日:2007年 12月 14日, 修正完了日: 2007年 12月 27日

I. 서론

서비스의 다중병합과 워크플로우 프로파일링을 지원하는 SOA기반의 P-BPIL 워크플로우 시스템은 P-BPIL을 통해 다중병합 형태의 웹 서비스 워크플로우

와 웹 서비스 워크플로우를 제작할 때 사전에 제작된 웹 서비스 워크플로우를 프로파일 형태로 저장한다. 이것을 하나의 액티비티 형태로 전환하고 새롭게 생성

하는 워크플로우에 삽입할 수 있으며, 현재 비즈니스 프로세스의 표준으로 자리잡은 BPEL로 제작된 비즈니스 프로세스 또한 하나의 액티비티 형태로 워크플로우에 삽입하여 사용할 수 있다. 이로써 웹 서비스와 프로파일 형태의 웹 서비스 워크플로우 간의 다중병합을 비롯해 성격이 다른 비즈니스 프로세스까지 하나의 워크플로우로 묶어 하나의 비즈니스 프로세스로 만들어 낼 수 있다.

II. 본론

2.1 시스템 구조

본 시스템은 크게 서비스에 대한 정보 저장 및 검색을 수행하는 부분(Service Providing Part)과 워크플로우의 제작 및 서비스 실행(Workflow and Processing Part) 부분으로 나뉘어 진다.

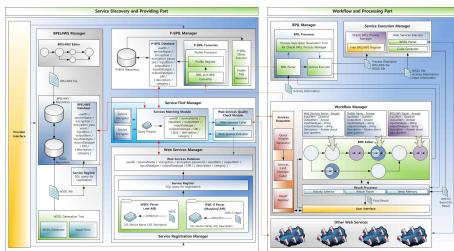


그림 1. 전체 시스템 구조도

2.1.1 System Discovery and Providing Part

첫 번째로 Service Providing Part는 BPELAWs Manager, P-BPIL Manager, Web Services Manager 그리고 이 세가지 매니저에서 서비스를 검색하는 Service Find Manager로 구성되어 있다.

BPELAWs Manager는 BPEL 프로세스를 제작하고 그와 관련된 WSDL[6] 파일을 제작해 저장해두고 관련된 정보를 추출하는 역할을 하는 부분이다. BPEL 프로세스를 제작할 수 있는 BPELAWs Editor와 BPEL 프로세스에서 필요한 WSDL을 생성해 주는 WSDL Generation Tool 그리고 BPELAWs 서비스에 대한 정보를 담고 있는 BPELAWs Database, 생성된 BPEL 파일을 저장하고 있는 BPELAWs Repository, 생성된 WSDL 파일을 저장하고 있는 WSDL Repository, P-BPIL만을 단독으로 실행시킬 수 있게 해주는 P-BPIL Direct Executor로 구성되어 있으며 BPELAWs Editor와 WSDL Generation Tool에서 만들어진 BPEL 파일과 WSDL 파일을 저장해두고 해당 서비스에 대한 정보를 BPELAWs Database에 저장해 두고 Service Find Manager의 서비스 검색요청에 응하게 된다

P-BPIL Manager는 BPIL로 정의된 워크플로우 프로파일을 관리하고 처리하는 부분으로 본 시스템에서 서비스 프로파일링을 처리하는 중요한 부분이다. P-BPIL 파일을 저장하고 있는 P-BPIL Repository와 P-BPIL에 대한 정보를 저장하고 있는 P-BPIL Database 그리고 프로파일에 대한 등록과 변환을 수행하는 P-BPIL Processor로 구성되어 있다. P-BPIL은 BPIL 워크플로우 에디터에서 생성한 워크플로우를 프로파일 형태로 저장하게 되면 P-BPIL로 전환되어 P-BPIL Repository에 저장되며 해당 프로파일에 대한 정보는 P-BPIL Database에 저장한다 그리고 이 정보를 바탕으로 Service Find Manager의 서비스 검색 요청에 응하게 된다.

Web Services Manager는 시스템 외부에 산재해 있는 웹 서비스들에 대한 정보를 저장해 두는 역할을 하는 부분이다. 서비스 제공자가 서비스를 등록할 수 있는 Provider Interface와 등록되는 서비스인 OWL-S[7], WSDL을 파싱해 검색에 필요한 정보를 추출하기 위한 OWL-S, WSDL Parser 그리고 추출한 정보를 Web Services Database에 저장하는 역할을 수행하는 Service Register, 추출한 정보를 저장하는 Web Services Database로 구성되어 있다. 서비스 제공자는 Provider Interface를 통해

OWL-S, WSDL파일을 시스템에 등록 요청을 하게 되고 이 파일은 OWL-S, WSDL Parser에 의해 파싱되어 Service Register를 통해 추출된 정보를 Web Services Database에 저장하게 된다. 그리고 이 정보를 바탕으로 Service Find Manager의 서비스 검색 요청에 응하게 된다.

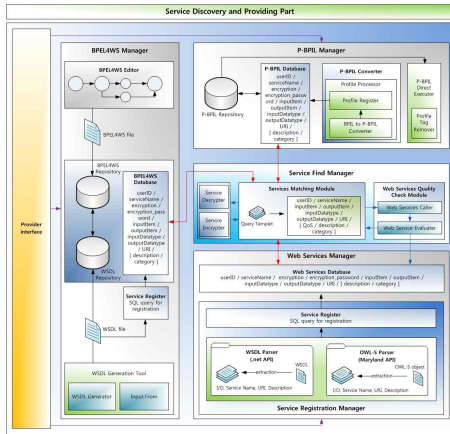


그림 2. Service Discovery and Providing Part
 Service Find Manager는 BPEL4WS Manager, P-BPIL Manager, Web Services Manager에서 제공하는 각 서비스에 대한 데이터베이스에서 사용자가 요청한 서비스와 부합하는 서비스를 찾아주고 서비스의 품질을 측정해주는 기능을 가진 부분이다. 사용자가 정의한 Query Templet을 이용해 사용자가 요청한 서비스를 찾아주는 역할을 하는 Service Matching Module과 웹 서비스의 품질을 측정해주는 Web Services Quality Check Module 그리고 각 서비스들에 대해 인증된 사용자만 사용할 수 있도록 부/복호화를 지원하는 Service En/Decrypter로 구성되어 있다. 사용자는 워크플로우 에디터의 User Interface를 이용 Query Templet을 정의해 Service Find Manager의 Service Matching Module로 넘기게 되고 Service Matching Module에서는 이 Query Templet을 기반으로 사용자가 요청한 서비스를 검색해 주게 된다. 서비스를 검색할 시에는 BPEL4WS, P-BPIL, Web Services 모두에서 검색을 실시하게 되며 조건이 맞으면 종류를 불문하고 검색된 결과를 모두 User Interface로 반환하게 된다. 이 때 서비스 제공자가 암호를

설정해 놓았다면 해당하는 서비스에 대한 암호를 알고 있어야 반환 과정을 완료할 수 있다. 그리고 이 서비스들 중 웹 서비스에 QoS가 정의되지 않은 경우 Web Services Quality Check Module에서 해당 웹 서비스를 호출해 QoS를 측정, Web Services Manager의 Web Services Database와 Service Matching Module로 QoS 값을 반환하게 된다.

2.1,2 Workflow and Processing Part

두 번째로 Workflow and Processing Part는 BPIL Manager, Service Execution Manager, Workflow Manager로 구성되어 있다.

BPIL Manager는 Workflow Manager의 BPIL Editor에서 제작된 BPIL과 P-BPIL을 파싱하고 Service Execution Manager에서 필요로 하는 Process Descriptor를 생성하며 서비스 실행을 요청하는 역할을 하는 부분이다. BPIL을 파싱하는 역할을 하는 BPIL Parser와 BPEL4WS를 실행하는 Oracle BPEL Process Manager[8]에서 필요로 하는 Process Descriptor[9]를 생성해 주는 Process Descriptor Generation Tool for Oracle BPEL Process Manager 그리고 서비스에 알맞게 Service Execution Manager를 호출해주는 Service Executor로 구성되어 있다. Workflow Manager의 BPIL Editor에서 BPIL과 P-BPIL을 생성하고 BPIL의 액티비티로 BPEL이 포함되어 있다면 Process Descriptor Generation Tool for Oracle BPEL Process Manager에서 Process Descriptor를 생성해 Service Execution Manager의 Oracle BPEL Process Manager로 Process Descriptor, BPEL4WS file, WSDL file을 전달하게 된다 그리고 일반 웹 서비스 액티비티의 경우 Web Services Executor로 WSDL file을 전달하게 된다.

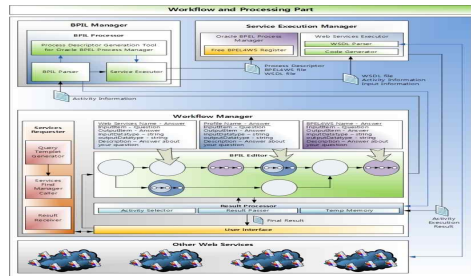


그림 3. Workflow and Processing Part

Service Execution Manager는 BPIL Processor에서 요청한 서비스 실행명령과 서비스 실행에 필요한 정보를 받아 수행하는 역할을 하는 부분이다. BPEL4WS를 실행하는 Oracle BPEL Process Manager와 웹 서비스를 실행하는 Web Services Executor로 구성되어 있으며 BPIL Processor에서 서비스를 분류해 해당하는 실행기로 요청과 실행에 필요한 정보를 보내면 각각의 서비스에 맞는 실행기에서 서비스의 수행을 실시하고 결과를 Workflow Manager의 Result Processor로 반환하게 된다.

Workflow Manager는 사용자에게 시스템 인터페이스를 제공하고 BPIL과 P-BPIL 제작 및 서비스의 탐색을 요청하는 역할을 하는 부분이다. BPIL과 P-BPIL을 제작할 수 있는 BPIL Editor, 서비스를 요청할 수 있는 Services Requester, 처리된 결과를 받아와 다음 액티비티로 넘겨주는 Result Processor, 그리고 사용자가 시스템을 이용할 수 있도록 해주는 User Interface로 구성되어 있다. 사용자는 BPIL Editor를 이용해 BPIL과 P-BPIL을 작성하게 되는데 이때 필요한 서비스들을 요청하기 위해 Services Requester를 이용한다. Services Requester는 사용자가 요구로 하는 조건들을 입력 받아 Query Templet으로 만들고 이것을 Service Find Manager로 넘기게 된다. Service Find Manager에서 검색된 결과를 받아와 BPIL Editor에서 BPIL과 P-BPIL 워크플로우를 제작하게 되며 이렇게 제작된 워크플로우를 실행하게 되면 해당 액티비티의 성격에 맞게 BPIL Manager와 Service Execution Manager를 통해 서비스가 수행되고 이 결과가 Result Processor로 반환되게 된다. Result Processor에서는 다음 액티비티를 선택하고 결과로 반환된 데이터를 선택한 액티비티의 인풋으로 전달하게 된다. 이렇게 액티비티 하나하나를 처리해 마지막 액티비티까지 처리가 되면 최종 결과를 User Interface를 통해 받아 볼 수 있다.

III. 구현 및 비교 분석

3.1 구현

프로토타입의 구현은 .NET과 J2EE 두 개의 플랫폼이 사용되었다. 우선 서버 쪽 웹 컨테이너와 웹 서비스 컨테이너로 .NET 플랫폼을 지원하기 위한 Microsoft IIS 6.0 그리고 J2EE의

지원하기 위한 톰캣과 Axis[10]가 Windows Server 2003 Standard 기반의 두 서버에 각기 나뉘어 설치되었다. BPEL4WS의 등록 및 실행을 위해서 Oracle BPEL Process Manager를 사용하였으며 OWL-S 파서는 매릴랜드 대학의 OWL-S API 모듈[11]을 WSDL은 .NET의 WSDL 파서를 이용하였다. 이러한 툴들을 기반으로 시스템의 구현은 Microsoft Visual Studio .NET 2005와 IBM Eclipse에서 수행되었다.

3.2 BPIL/P-BPIL(Business Process Integration Language / Profile-BPIL)

본 논문에서 정의한 BPIL은 서비스의 다중 병합을 지원하는 워크플로우를 생성하기 위한 XML 기반의 언어이다. XML을 기반으로 한 만큼 XML과 마찬가지로 데이터에 대한 호환성과 유동성을 가지며 복잡하지 않은 문법과 반드시 필요한 정보만을 간단하게 기술하도록 만들어져 있다. BPIL은 일반 웹 서비스, 웹 서비스 워크플로우, BPEL 프로세스 모두를 하나의 액티비티로 표현 가능하며 타입 속성을 통해 서비스의 종류를 파악하고 다중병합을 지원한다. BPIL로 정의된 문서는 BPIL Parser를 통해 처리되며 P-BPIL은 BPIL로 작성한 워크플로우에 프로파일로의 전환을 위한 관련 정보를 추가해 프로파일 형태로 전환한 것으로 역시 BPIL Parser에 의해 처리된다.

3.2.1 BPIL의 문법

표시자	정의
<sequence>	순차 서비스의 정의
<activity>	액티비티 정의
<activityID>	액티비티를 구분하기 위한 고유 식별자 정의
<type>	서비스의 형식 정의
<URI>	서비스의 위치 정의
<methodName>	서비스가 포함하고 있는 메소드 중 이용하고자 하는 메소드 정의
<inputType>	입력 자료형 정의
<outputType>	출력 자료형 정의
<save>	입력 또는 출력 값 중 원하는 값을 Temp Memory에 저장
<load>	Temp Memory에 저장한 값을 불러옴
<QoS>	웹 서비스의 품질 정의
<description>	서비스에 대한 설명 정의

>	
<branch>	액티비티의 분기 정의
<processing>	분기처리 정의
<if>	처리 조건 정의
<then>	처리 내용 정의
<else>	처리 조건이 아닐 경우에 대한 내용 정의
<end>	분기에서 종료 필요 시 사용
<profile>	프로파일 정의
<encryption>	프로파일의 암호화 여부 정의
<userID>	프로파일을 사용하는 식별자의 ID 정의
<profileID>	프로파일의 고유한 식별자 정의

그림 4. BPIL문법 정의표

3.2.2 BPIL 예제

아래의 어플리케이션 예제는 사원의 출장신청을 바탕으로 출장가능 여부를 판별하고 가능할 시 표 예약을 해주는 BPEL 워크플로우의 예이다.

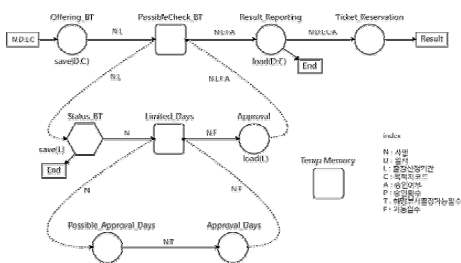


그림 5. 출장신청 BPIL 워크플로우

다음은 위의 워크플로우에 사용된 액티비티 속성의 몇 가지 예이다.

Activity	출장신청
Type	WS
activityID	Offering_BT
URI	http://spike.korea.ac.kr/WS/offer_businessTrip.wsdl?
methodName	offer
inputType	string
outputType	string
inputItem	사번:일자:기간:목적지코드
outputItem	사번:일자:기간:목적지코드
QoS	good
Description	사원의 출장 신청을 접수함

Activity	출장가능확인
Type	P-BPIL
activityID	PossibleCheck_BT
URI	http://spike.korea.ac.kr/PBPIL/possibleCheck_businessTrip.pbpil
methodName	null
inputType	string
outputType	string
inputItem	사번:기간
outputItem	사번:기간:승인여부:가능일수
QoS	null
Description	사원의 출장 가능 여부를 확인

그림 5. 액티비티 속성 정의

액티비티들의 속성은 type에 의해 구분되는데 WS는 웹 서비스, BPEL4WS는 BPEL, P-BPIL은 P-BPIL 서비스를 나타낸다. 위의 액티비티 속성과 그래프를 BPIL 코드로 표현하면 다음과 같다.

```

<?xml version="1.0" encoding="euc-kr"?>
<sequence>
  <activity>
    <type>WS</type>
    <activityID>Offering_BT</activityID>
    <URI>http://spike.korea.ac.kr/WS/offer_businessTrip.wsdl?</URI>
    <methodName>offer</methodName>
    <inputType>string</inputType>
    <outputType>string</outputType>
    <inputItem>N:D:L:C</inputItem>
    <outputItem>N:D:L:C</outputItem>
    <save>D:C</save>
    <QoS>good</QoS>
    <description>사원의 출장 신청을 접수함</description>
  </activity>
  .
  .
  <branch>
    <activity>
      <type>WS</type>
    <activityID>Result_Reporting</activityID>
  >
  <URI>http://spike.korea.ac.kr/WS/result_re
  
```

```

porting.wsdl?</URI>
  <methodName>report</methodName>
  <inputType>string</inputType>
  <outputType>string</outputType>
  <inputItem>N:L:F:A</inputItem>
  <outputItem>N:D:L:C:A</outputItem>
  <load>Offering_BT:D:C</load>
  <QoS>good</QoS>
  <description>신청한 출장에 대한
  결과</description>
</activity>
,
<process>
<if>
<Result_Reporting>outputItem(A==1)</Re
sult_Reporting>
  </if>
  <then>
<Ticket_Reservation>inputItem(N:D:L:C:A
)</Ticket_Reservation>
  </then>
  <else>
  <end>End</end>
  </else>
</process>
</branch>
<profile>
  <encryption>1</encryption>
  <userID>user1</userID>
</profileID>PossibleCheck_BT</profileID>
  <sequence>
  <branch>
  <activity>
  <type>BPEL4WS</type>
  <activityID>Status_BT</activityID>
</URI>http://spike.korea.ac.kr/BPEL4WS/s
tatus_businessTrip.bpel</URI>
<methodName>status</methodName>
  <inputType>string</inputType>
  <outputType>string</outputType>
  <inputItem>N</inputItem>

```

```

  <outputItem>N:S</outputItem>
  <save>L</save>
  <QoS>null</QoS>
  <description>출장을 신청한 사원이
  현재 출장중이거나 출장 날짜가
  없는지 확인</description>
</activity>
.
.
<processing>
  <if>
<Status_BT>outputItem(출장상태==0)</St
atus_BT>
  </if>
  <then>
<Limited_Days>inputItem(N:S)</Limited_
Days>
  </then>
  <else>
  <end>End</end>
  </else>
</processing>
</branch>
,
,
<profile>
  <encryption>0</encryption>
  <userID>user1</userID>
  <profileID>Limited_Days</profileID>
  <activity>
  <type>WS</type>
<activityID>Possible_Approval_Days</acti
vityID>
</URI>http://spike.korea.ac.kr/WS/possible
_AN.wsdl?</URI>
<methodName>possible_AN</methodNam
e>
  <inputType>string</inputType>
  <outputType>string</outputType>
  <inputItem>N</inputItem>
  <outputItem>N:T</outputItem>
  <QoS>good</QoS>
  <description>각 부서별 출장 가능
  횟수를 조회</description>
</activity>

```

```

<activity>
  <type>WS</type>

  <activityID>Approval_Days</activityID>

  <URI>http://spike.korea.ac.kr/WS/AN.wsdl
  ?</URI>
    <methodName>AN</methodName>
    <inputType>string</inputType>
    <outputType>string</outputType>
    <inputItem>N:T</inputItem>
    <outputItem>N:F</outputItem>
    <QoS>normal</QoS>
    <description>출장을 신청한 사원이
    지금까지 몇 번의 출장 신청
    받았는지 조회</description>
  </activity>
</profile>
</sequence>
</profile>
</sequence>

```

그림 6. BPIL예제 코드

3.3 Service Discovery and Providing Part

3.3.1 BPEL4WS Manager

BPEL4WS의 구현에서 주요한 부분을 차지하는 BPEL4WS 에디터는 openjgraph API[12]를 이용해 구현되었으며 여기에서 생성된 그래프를 바탕으로 BPEL4WS 파일과 WSDL파일은 openjgraph의 bpelWriter.writeBPEL(bpelFile);와 wsdlWriter.writeWSDL(wsdlFile);객체를 이용해 생성한다. BPEL4WS와 WSDL파일은 각각 BPEL4WS Repository와 WSDL Repository에 저장되며 WSDL Generation Tool은 wsdlWriter.writeWSDL(wsdlfile);을 기반으로 제작되었다. 이 BPEL4WS에 대한 정보는 BPEL4WS Database에 저장되는데 이 정보는 입출력 정보와 같은 필수 입력 항목을 제외하고 존재하는 추가 정보가 없다면 사용자가 입력하는 형식을 취하고 있으며 DBMS는 MS-SQL2000을 사용한다. 테이블의 컬럼값으로는 userID/serviceName/encryption/encryption_password/inputItem/outputItem/inputDatatype/outputDatatype/URI/QoS/description/category가 있다.

3.3.2 P-BPIL Manager

P-BPIL Manager의 구현은 BPIL Workflow

Editor에서 생성된 BPIL 파일에 프로파일과 관련된 정보를 추가해주는 형태를 띄고 있으며 프로파일에 관련된 정보는 사용자가 GUI 인터페이스를 통해 입력하는 형식을 취하고 있다. 그리고 전환된 P-BPIL에서 서비스의 검색과 관련된 데이터를 Profile Register를 통해 P-BPIL Database에 저장하는데 DBMS는 MS-SQL2000을 사용하며 테이블의 컬럼값은 BPEL4WS Manager와 동일하다.

3.3.3 Web Services Manager

Web Services Manager의 구현은 OWL-S와 WSDL형태로 제공되는 웹 서비스들을 Maryland UNIV.의 OWL-S 파서와 .NET에서 제공하는 WSDL 파서를 이용해 테이블의 컬럼값들을 추출하고 그 정보를 Service Register를 이용해 Web Services Database에 등록하는 형태를 취하고 있으며 DBMS는 역시 MS-SQL2000을 사용한다. 테이블의 속성값은 다른 서비스 매니저와 동일하다. 이 부분 역시 입출력 정보와 같은 필수 입력 항목을 제외하고 존재하는 추가 정보가 없다면 사용자가 입력하는 형식을 취하고 있다.

3.3.4 Service Find Manager

Service Find Manager의 구현은 각 매니저가 가지고 있는 DBMS에 쿼리를 하고 그 결과값을 취해 반환하는 형태로 구성되어 있다. 우선 Service Find Manager의 Service Matching Module 평선이 호출되고 Query Templet이 넘어오면 Services Matching Module에서 사용자에게서 넘겨받은 Query Templet을 바탕으로 select 쿼리를 생성하고 그 쿼리를 각 매니저의 DBMS로 수행요청을 하게 된다. 그 후 각 매니저의 DBMS에서 넘어온 결과 값을 Service Requester에게 반환하고 웹 서비스의 경우 QoS 값이 존재하는지 검사해 없을 경우 Web Services Quality Check Module로 해당 서비스의 URI를 넘겨 QoS 값을 구한다. 그리고 이들 서비스 중 암호가 걸려있는 서비스의 경우 Service En/Decrypter로 값을 넘겨 해당 서비스 제공자가 지정한 암호와 사용자에게 요청한 암호가 일치하는지 확인하여 일치할 경우에만 결과를 보여줄 수 있도록 한다.

3.4 Workflow and Processing Part

3.4.1 Workflow Manager

Workflow Manager은 BPIL Editor, Result Processor, Services Requester로 구성된다. BPIL Editor는 .NET C#의 GDI 객체를 이용해 구현되었다. 사용자가 검색한 서비스를 Drag&Drop형태로 에디터에 삽입할 수 있으며 다중 병합 모드를 선택하여 다중병합을 지원하는 워크플로우를 생성할 수 있다. P-BPIL이나 BPELAWLS같은 서비스의 경우 하나의 액티비티로 표현하되 색상과 모양을 다르게 표시한다. Result Processor는 BPIL Editor에서 전체 액티비티의 수행순서를 액티비티ID의 배열로 넘겨받은 후 수행된 액티비티의ID를 넘겨받아 다음으로 수행할 액티비티를 선택한 후 BPIL Editor로 해당 액티비티의ID를 넘겨준다. Result Processor의 Temp Memory는 BPIL 파서에서 <save>, <load>이벤트가 발생했을 경우 사용되는데 액티비티ID와 데이터가 동적 배열로 저장된다. Service Requester는 사용자가 GUI폼을 이용해 입력한 서비스 검색조건을 Query Templet이라는 텍스트 형태의 파일로 만들고 Service Find Manager를 호출하도록 구현되어 있다. Service Find Manager의 호출과 동시에 Query Templet을 Service Find Manager로 함께 넘긴다.

3.4.2 BPIL Manager

BPIL Manager는 BPIL Parser, Process Description Generation Tool for Oracle BPEL Process Manager, Service Executor로 구성된다. BPIL Parser는 Workflow Manager의 BPIL Editor에서 생성된 BPIL 파일을 분석하는 부분으로 Microsoft MSXML 4.0 Parser를 이용해 구현되었다. Process Description Generation Tool for Oracle BPEL Process Manager또한 Microsoft MSXML 4.0 Parser를 이용해 요구되는 값을 BPEL 파일에서 추출해 자동 생성하도록 구현되었다. BPIL 파서에서 BPELAWLS형태의 액티비티를 발견하였을 경우 호출되며 Process Description을 생성하는데 부족한 값이 있다면 사용자에게 요청하게 된다. Service Executor는 BPIL 파일에서 액티비티의 <type>속성을 파악해 BPELAWLS인 경우 Service Execution Manager의 Oracle BPEL Process Manager를, Web Services나 P-BPIL의 경우 Web Service Executor를 호출한다.

3.4.3 Service Execution Manager

Service Execution Manager는 Oracle BPEL Process Manager와 Web Services Executor로 구성되어 있는데 Oracle BPEL Process Manager는 오라클에서 제공하는 BPEL Process 엔진으로 컴파일과 등록 과정을 거쳐 BPEL 프로세스를 수행할 수 있다. 전달받은 Processor Descriptor, BPELAWLS 파일, WSDL 파일을 가지고 bpelc 명령을 통해 bpel 파일을 컴파일하고 이 파일을 obant 명령을 통해 이것을 등록하는 방식을 취하고 있어 이 일련의 과정을 batch 파일로 자동화 시키고 이것을 수행시키는 방식으로 구현되어 있다. Web Service Executor는 전달받은 WSDL파일, 액티비티 정보, 인풋 정보를 바탕으로 .NET의 WSDL 파서를 이용해 WSDL을 분석하고 이것을 수행하기 위한 C#코드를 생성하는 생성기를 구현하고 이것을 컴파일하고 실행해 결과를 얻는다. 이 과정의 자동화는 Oracle BPEL Process Manager와 같이 batch파일을 이용해 구현되었다.

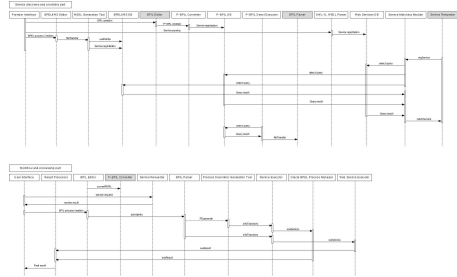


그림 7. 시퀀스 다이어그램

3.5 타 시스템과의 비교

3.5.1 BPWS4J

BPWS4J[13]는 IBM alphaWorks에서 이용 가능하며 최초로 개발된 BPEL엔진으로 대중적으로 가장 널리 알려져 있다. 텍스트 형태의 워크플로우 에디터를 제공하며, 프로세스 탭에서 트리 방식의 워크플로우를 작성하면 소스 탭에 BPEL 코드가 자동으로 생성된다. 하지만 에디터로 BPEL 코드는 자동으로 생성할 수 있으나 WSDL 파일을 수동으로 작성해야 한다. 그리고 웹 서비스의 등록이 엔진의 관리자를 통해서만 등록이 가능하므로 전과정의 자동화가 불가능하며 오픈 소스가 아니다.

3.5.2 Active BPEL

ActiveBPEL[14]은 상용 수준의 오픈 소스 BPEL 엔진이다. 고수준의 워크플로우 디자이너와 매뉴얼, 튜토리얼을 지원하고 있다. 그러나 BPWS4J와 마찬가지로 디자이너로 BPEL 파일까지만 자동으로 생성되며 WSDL 파일은 수동으로 작성해야 한다. 웹 서비스 등록은 BPEL 프로세스에 대한 자세한 정보를 가지고 있는 프로세스 서술자를 만든 후 jar 명령을 통해 bpr 파일을 만들어 등록하는 복잡한 과정을 가진다.

3.5.3 SDS

SDS[15]는 동적 웹 서비스 탐색을 위해 BPWS4J 엔진에 시멘틱 웹 기법을 도입한 BPEL 시스템이다. 수동으로 웹 서비스를 탐색하는 기존의 방법과는 다르게 BPWS4J의 수행 이전에 자동 탐색을 수행한다. 하지만 SDS는 자동화된 웹 서비스 조합 기능을 지원하지 않는다. 워크플로우 디자이너나 WSDL 자동 생성 또한 지원하지 않으며 웹 서비스 등록은 BPWS4J 방식을 따르고 있어 관리자 툴을 통해서만 가능하다. 현재 Stanford KSL 웹 사이트에서 데모 버전만 제공되며 오픈 소스 또한 아니다.

	분 시스템	BPWS4J	ActiveBPEL	SDS
Designer	GUI(Application)	Text(Eclipse Plugin)	GUI(Eclipse Plugin)	GUI(Applet)
Multi Integration	Yes	No	No	No
Workflow Management	Dynamic	Static	Static	None
Service Profiling	Yes	No	No	No
Automation	Auto	Manual	Manual	Auto
Service Discovery	Auto/Manual	Manual	Manual	Auto
완전성	Prototype	Normal	Good	Bad
Platform	.NET/J2EE	J2EE	J2EE	J2EE
Vender	Korea UNIV.	IBM	Active Endpoints	Stanford UNIV.

그림 8. 시스템 간 구현 기술 비교

3.5.1 Advantage

본 시스템과 비교할 수 있는 기존의 시스템들은 BPEL을 지원하는 시스템들뿐이며 이 시스템들과 비교해 본 시스템이 가지고 있는 장점은 크게 5가지로 요약해 볼 수 있다.

1. 다중병합 지원
2. 워크플로우 프로파일링 지원
3. GUI 형태의 워크플로우 에디터 지원
4. 서비스를 수행하기 위해 필요한 추가적 파일들을 자동으로 생성해 주고 등록 과정 또한 자동
5. 서비스의 통합 검색이 가능하며 검색을 사용자의 기호에 맞게 수동과 자동 중 선택해 사용할 수 있음

IV. 결론 & 추후연구

인터넷 관련 기술의 발달과 e-비즈니스의 대두로 인해 기업간 협력 및 상호작용이 매우 중요해 졌으나, 기업에서는 여러가지 유형의 압력으로 인해 서로 다른 이기종 시스템들이 사용되고 있는 현 상황에서 기업간 업무 프로세스의 통합과 상호 연동성 문제는 선택이 아닌 필수적 요소이다. 그러나, 현재 이러한 조건을 충족해 줄 수 있는 다중병합 시스템이 존재하지 않는 상황이며 이것은 기업간 협력 및 상호작용이 매우 어렵다는 것을 의미한다. 이에 본 연구에서 구현한 시스템은 다중병합을 비롯한 워크플로우 프로파일 시스템, 그리고 BPEL까지 포함하는 단일 워크플로우를 생성하고 실행하는 것을 가능하게 함으로써 기업에는 이전에 사용하던 BPEL 프로세스를 그대로 이용할 수 있게 하고 추가로 작성한 워크플로우는 프로파일로 만들어 두었다가 필요할 때 워크플로우의 액티비티로서의 사용은 물론 파일 형태의 전달 또한 가능하기에 업무 프로세스 통합과 상호연동성 문제를 효과적으로 해결할 수 있는 시스템이다. 고로 워크플로우의 재사용이 가능하다는 장점과 제작한 워크플로우를 파일형태로 저장해 간편히 공유할 수 있다는 장점을 가진다. 그러나 각 액티비티들이 성공적으로 서비스 되고 있는지 사전에 알 수 있는 방법이 없다. 웹 서비스 워크플로우에서는 수행 중인 액티비티에 문제가 있다면 그 액티비티를 문제

가 없는 액티비티로 변경해 주는 연구가 많이 진행되고 있으나 반드시 수행을 시킨 이후 문제가 발생할 때에만 정정을 시도하게된다. 이와 다르게 워크플로우를 구성하고 있는 액티비티들의 상태를 한 눈에 미리 보여주는 모니터링 기능을 가진 시스템은 존재하지 않는다. 이에 현 시스템에 액티비티 모니터링 기능을 추가하여 BPIL 워크플로우들을 수행하기 전 워크플로우의 액티비티들을 주기적으로 점검하여 수행될 워크플로우가 문제가 있는 액티비티를 가지고 있는지 사전에 모니터링이 가능하게 하는 시스템을 추가하고자 한다. 워크플로우는 한눈에 보여야 하며 문제가 있는 액티비티만 다르게 표현해 주어 어떠한 액티비티에 문제가 있는지 한 눈에 알 수 있도록 한다.

참고문헌

- [1] Web Services, "http://www.w3.org/2002/ws/", 2007.
- [2] National Information Society Agency in Korea "A research on B2B Workflow Integration and Interoperability based on XML", 2003.
- [3] BPEL, "http://www.ibm.com/developerworks/library/specification/ws-bpel/", 2007.
- [4] XML, "http://www.w3.org/XML/Core/#Publications/", 2007.
- [5] SOA terminology overview, Part 1: Service, architecture, governance, and business terms "http://www-128.ibm.com/developerworks/webservices/library/ws-soa-term1/", 2007.
- [6] W3C, Web Services Description Language (WSDL) Version 2.0 Part1 : Core Language, "http://www.w3.org/TR/2007/REC-wsdl20-20070626/", 2007.
- [7] OWL Services Coalition, "OWL-S: Semantic Markup for Web Services, OWL-S White Paper, http://www.daml.org/services/owl-s/1.2/overview/", 2006.
- [8] Oracle BPEL Process Manager, "http://www.oracle.com/technology/products/ias/bpel/index.html", 2007.
- [9] Deanna Bradshaw and Mark Kennedy, "Oracle BPEL Process Manager's Guide 10g(10.1.3.1.0)", 2007.
- [10] Apache Axis, "http://ws.apache.org/axis2/

1_3/userguide.html#intro"

- [11] Sirin E., Hendler J., and Parsia B., "Semi-automatic Composition of Web Services Using Semantic Description, Web Services:Modeling", Architecture and Infrastructure Workshop in Conjunction with ICEIS, 2003.
- [12] Openjgraph, "http://sourceforge.net/projects/opengraph"
- [13] IBM, BPWS4J, http://www.alphaworks.ibm.com/tech/bpws4j
- [14] Active Endpoints, ActiveBPEL, "http://www.active-endpoints.com"
- [15] SDS, Mandell D. and McIlraith S., "Adapting BPEL4WS for the Semantic Web: The Bottom-Up Approach to Web Service Interoperation", in Proceeding of the 2nd International Semantic Web Conference(ISWC2003), Sanibel Island, Florida, 2003.

저 자 소 개

이 기 호 (정회원)



2006년 : 건국대학교 컴퓨터학과
과 졸업 (공학사)
2008년 : 고려대학교 대학원 전
자전기공학과 (공학석사)
2006년 3월~현재 : 고려대학교
대학원 전자전기공학과 석사 졸업
예정
<주관심분야>

Web Services, Web Services Workflow, Project Management

정 창 성 (정회원)



1981년 : 서울대학교 전자공학과
졸업 (공학사)
1984~87년 : Northwestern UNIV.
컴퓨터 과학 (공학석사, 박사)
1987~1992년 : 포항공과대 조교수
1992~현재 : 고려대학교 전기전자
컴퓨터 공학과 정교수

<주관심분야>

Network Virtual Environment, Distributed & Parallel Supercomputing, Image Processing, Grid Computing, Ubiquitous Computing