

BPEL을 이용한 웹서비스 기반의 u-SCM 프로세스 모델링

박현실, 최은미
국민대학교 비즈니스IT전문대학원

Web Service-based u-SCM Process Modeling using BPEL

Hyoun-Shil Park , Eunmi Choi
Graduate School of Business IT, Kookmin University
E-mail : hyoun1110@hanmail.net, emchoi@kookmin.ac.kr

요 약

USN (Ubiquitous Sensor Network)의 인프라 기술은 유비쿼터스 환경에서의 다양한 서비스를 창출하고 실생활에 적용할 수 있는 기술로 정착하고 있다. 산업분야에서는 관계적 협업이 중시되는 비즈니스 환경에서 실시간 기업이 되기 위해 RFID, EPC 네트워크 등의 기술을 비즈니스 서비스 모델에 적용한 u-SCM으로 적용할 수 있다. 웹 서비스 환경에서 u-SCM을 적용하면서도 고차원의 공급사슬 통합이 이루어지기 위해서는, 여러 상호작용주체가 참여하는 글로벌 프로세스 수준에서 u-SCM프로세스를 관리하고 제어할 필요가 있으며, 본 논문에서는 유비쿼터스 컴퓨팅 기술을 SCM에 적용한 u-SCM 프로세스를, 비즈니스 프로세스 표준언어인 BPEL(Business Process Execution Language)을 적용하여, 각 업무프로세스를 웹서비스로 구성된 환경에서 효과적으로 통합하여 적용하도록 하는 방안을 제시한다. 이러한 접근법은 기업, 조직 간에 적체적소에 실시간 데이터를 수집하고 접근 및 분석하여, 업무 프로세스의 중단이 없도록 u-SCM 프로세스의 진행과정을 보다 효율적으로 관리, 모니터링 하는 것이 가능하도록 하며, 향후 u-SCM프로세스 수행 결과를 토대로 프로세스의 개선에 대한 방안을 모색할 수 있다.

1. 서론

최근 IT업계 화두는 ‘IT의 산업화를 넘어, 산업의 IT화’이다. 이에 적합하게 USN (Ubiquitous Sensor Network)의 인프라 기술은 유비쿼터스 환경에서의 다양한 서비스를 창출하고 실생활뿐만 아니라 산업분야에 적용할 수 있는 기술로 정착하고

있다. 특히 산업분야에서는 관계적 협업이 중시되는 비즈니스 환경에서 실시간 기업이 되기 위해 RFID, EPC 네트워크 등의 기술을 비즈니스 서비스 모델에 적용한 u-SCM으로 적용할 수 있다. 현재 이러한 u-SCM의 도입은 RFID, EPC 네트워크 기술을 통하여 제조공장, 유통업체, 우체국, 도서

관, 항공사, 해운·항만 등 다양한 분야로 확산되고 있으며, 기업간의 정보시스템 이질성을 인하여 이기종 시스템간의 상호작용 문제를 해결하기 위하여 등장한 웹 서비스 환경에서 u-SCM을 적용하면서도 고차원의 공급사슬 통합이 이루어지기 위해서는, 여러 상호작용주체가 참여하는 글로벌 프로세스 수준에서 u-SCM 프로세스를 관리하고 제어할 필요가 있다. 본 논문에서는 유비쿼터스 컴퓨팅 기술을 SCM에 적용한 u-SCM 프로세스를, 비즈니스 프로세스 표준언어인 BPEL(Business Process Execution Language)을 적용하여, 각 업무 프로세스를 웹서비스로 구성된 환경에서 효과적으로 통합하여 적용하도록 하는 방안을 제시한다. 이러한 접근법은 기업, 조직 간에 적재적소에 실시간 데이터를 수집하고 접근 및 분석하여, 업무 프로세스의 중단이 없도록 u-SCM 프로세스의 진행과정을 보다 효율적으로 관리, 모니터링 하는 것이 가능하도록 하며, 향후 u-SCM 프로세스 수행 결과를 토대로 프로세스의 개선에 대한 방안을 모색할 수 있다.

2장에서는 관련 연구로써 RFID 및 EPC 네트워크, u-SCM, BPEL, 웹 서비스 등에 대하여 개괄적인 소개를 한다. 3장에서는 BPEL을 이용한 u-SCM의 메타데이터 정의 및 프로세스 모델링을 제안하고, 4장에서 결론을 맺는다.

2. 관련연구

2.1 RFID 및 EPC 네트워크

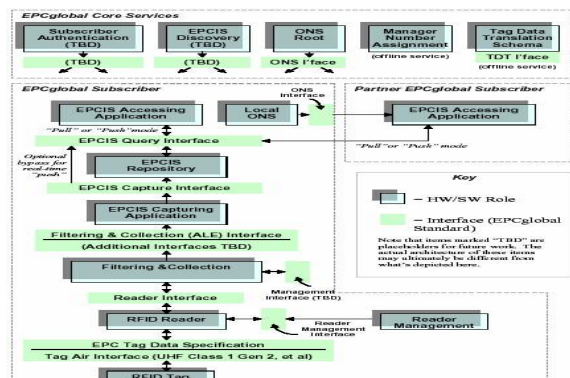
전자식별태그인 RFID(Radio Frequency Identification)는 일정주파수 대역에서 무선으로 데이터를 주고받는 무선주파수 인식시스템을 이용하여[1], 생산, 유통, 보관, 소비의 전 과정에 대한 정보 등 각종 정보를 담은 전자 칩으로써 자체 안테나를 갖추고 있으며, 리더기(reader)를 통하여 정보를 읽고, 이동통신망과 연계하여 정보시스템과 통합하여 사용된다[2].

EPC 센서 네트워크는 RFID의 인터넷아키텍처

에 무선 센서 네트워크를 결합한 인프라로 정의될 수 있으며, EPCglobal과 Auto-ID Labs에서는 인터넷에서 RFID식별 데이터를 실시간으로 검색하고 인식하여, 모니터링을 위한 논리적 컴포넌트 기능을 정의하였다[3].

전자 제품 코드인 EPC(Electronic Product Code)는 인터넷 상에서 유일하게 식별할 수 있는 코드로써, 모든 RFID 태그 데이터는 EPC를 이용하여 식별된다. 아이디 시스템(ID System)은 태그와 리더(Reader)를 포함하며, EPC미들웨어에 연결되어 있다. EPC미들웨어(Savant)는 태그로부터 데이터를 필터링(filtering), 집계(aggregation), 카운팅(counting) 등의 처리를 거쳐 데이터의 양을 줄여, 상위 레벨 컴포넌트인 EPCIS(EPC Information Service)에 이벤트로 제공하는 기능을 제공한다. EPCIS는 다양한 태그 정보를 상품에 대한 다양한 마스터 데이터(Master Data)와 함께 저장하고 역세상 어플리케이션(Accessing Application)에게 태그 데이터를 XML 포맷으로 만들어서 제공하기 위한 데이터베이스이다[3].

디스커버리 서비스인 ONS(Object Naming Service)와 EPCIS Discovery 서비스는 EPC 코드에 해당하는 정보를 가지는 EPCIS 서버에 대한 URL(Uniform Reference Location)로 변환하거나 EPCIS에 저장된 정보에 대한 검색 기능을 제공한다. <그림2>는 EPCglobal에서 정의하고 있는 논리적인 컴포넌트와 각 사이의 인터페이스를 나타낸다.



<그림2> EPC 아키텍처 프레임워크[3]

2.2 u-SCM

산업분야에서는 SCM과의 호환성을 통해 적용할 수 있다. 다양한 기관들과의 협력이 필요하고, 다양한 형태의 사용자가 물류 과정에서의 정보를 이용할 수 있으며 대량의 데이터에 대해서 실시간으로 처리할 수 있어야 하므로[3], RFID, EPC 네트워크 등의 기술과 함께 USN을 비즈니스 서비스 모델에 적용한 u-SCM을 통해 제품 또는 컨테이너, 팔레트 단위 등에 태그 혹은 센서 노드가 결합되어서 출하 시부터 각 단계를 포함하는 전 과정에서의 상태 정보를 실시간으로 제공해 줄 수 있다.

SCM(Supply Chain Management)은 자재 공급업체에서 소매에 이르는 모든 거래 파트너들 사이에 물리적 의미인 원료와 부품뿐만 아니라, 정보, 자금, 지식의 흐름 등을 통합적으로 관리 운영하여 불확실성을 줄이고 전체 최적화를 달성하여 궁극적으로 최소의 비용과 최고의 고객만족 달성을 목표로 하고 있는 경영패러다임이다[4].

SCM을 이용해서 기업들의 생산성 향상이 이루어지기는 했으나, 바코드나 수작업으로 인한 자원의 자동인식, 파악이 불가능해 생기는 정보 동기화의 문제와 물류망의 복잡화에 의한 정보 왜곡 현상 등의 문제로 기업의 생산성 극대화에는 한계를 갖는다.

따라서 SCM 전 과정에 걸쳐 자동화시킬 수 있는 방법으로 유비쿼터스 컴퓨팅을 SCM에 적용하여, 컴퓨팅과 네트워크 기반 기술인 RFID, 다양한 감지 센서를 이용한 태그, 임베디드 운영체제, 무선센서 네트워크 통신 기술 등을 통해 기업 내 모든 자원의 흐름을 관리할 수 있다[2].

국내외에서 IBM, Gillette, Gap, PRADA, Wal-Mart 등[5] 많은 단체, 회사들이 성과를 거둠으로써 SCM을 개선할 수 있는 해결책으로 각광받고 있다.

2.3 u-SCM에서 BPEL을 활용한 웹 서비스 통합의 적합성

웹 서비스는 XML기술과 인터넷 프로토콜을

이용하여 조직의 어플리케이션을 서비스화하여 제공하는 기술이다. 이기종 기업간 정보시스템의 상호운영성을 위해 웹 서비스는 네트워크상에 분산된 서비스 간의 상호 연동이 가능하도록 하기 때문에 현실적인 대안이 되고 있으며, 웹 서비스를 활용하여 기업간의 프로세스를 통합하려는 노력들이 이루어지면서 WSCI(Web Services Choreography Interface)나 BPEL4WS(Business Process Execution Language for Web Services)[6]과 같은 웹 서비스를 기반으로 한 비즈니스 프로세스 수행 언어들이 등장하고 있다. 특히, BPEL4WS 표준으로 기술되는 BPEL언어는 XML기반의 웹 서비스 정의 언어이다[7]

고객, 공급자, 파트너 등 다수의 기업이 상호 작용하는 공급사슬의 구성은 기본적으로 다양한 정보시스템들이 포함되어 있는 환경이며, 이들 정보시스템들이 얼마나 원활하게 상호작용하며 의사소통을 진행하는가에 성패가 달려있다. 그리고 효과적인 공급사슬 관리를 위해서는 공급 사슬 전반에 걸쳐 재고 및 최종 수요 정보의 실시간의 공유가 필요하며, 이에 입각한 실시간 의사결정과 통합적 운영이 필요하다[8].

BPEL을 활용하여 웹 서비스를 통합하는 방식은 재고에 대한 상호 공유와 같은 이기종 시스템 간의 공급사슬의 통합적 운영을 구현하는데 효과적으로 이용될 수 있다. BPEL을 이용하면 웹 서비스에 비즈니스 프로세스라 불리는 컴포지트 서비스를 쉽고 직관적으로 구현할 수 있다[9].

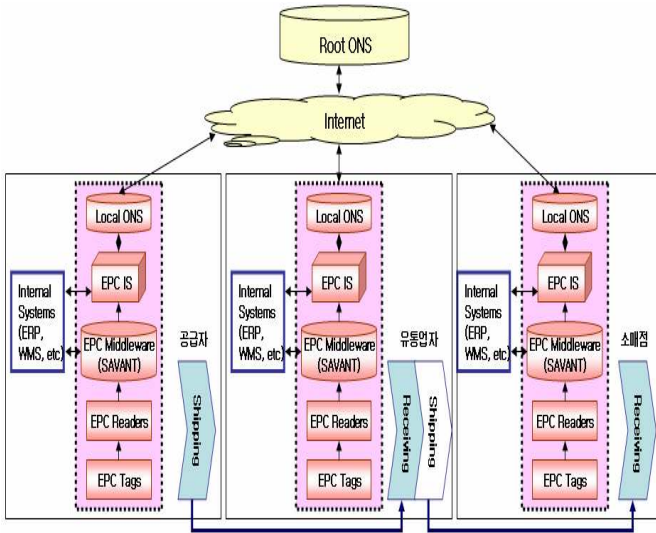
BPEL을 이용하여 u-SCM프로세스를 모델링함으로써 u-SCM프로세스를 효과적으로 관리하면서 의사결정에 필요한 요소를 모니터링, 분석할 수 있다.

3. BPEL을 이용한 u-SCM 모델링

3.1 u-SCM프로세스 모델링

본 연구에서는 <그림3>과 같이 RFID와 EPC네트워크를 이용하여 기존의 SCM프로세스와 다른 Shipping 프로세스와 Receiving 프로세스의 시나리

오를 살펴봄으로써 u-SCM프로세스를 고려한다. 공급 사슬의 각 액터들은 언제, 어디서든 local ONS를 통하여 remote ONS로부터 상품정보를 획득할 수 있다.



<그림3> RFID와 EPC네트워크를 활용한 “Shipping”과 “Receiving”프로세스

▪ Shipping Process

1. 주문량 검증

1.1 개별 제품 포장

1.2 RFID태그가 내재된 상자에 제품을 넣음

1.3 RFID태그가 내재된 팔레트(창고·공장 등의 화물 운반·저장하기 위한 받침대)에 상자를 실음

1.4 각 팔레트를 포장함

1.5 지게차를 이용해서 포장하는 구역으로부터 팔레트를 실음

1.6 RFID 리더기가 장착된 RFID GATE를 통과함

1.6.1 RFID 리더에 의해 상자와 팔레트에 부착된 태그가 자동으로 읽힘

1.6.1.1 SAVANT가 운반지역을 자동으로 확인함

1.6.1.2 잘못된 운반지역이면 자동으로 알람 서비스를 제공하고, 그렇지 않은 경우 운반지역으로 각 팔레트를 자동으로 연결함

1.6.1.3 SAVANT와 EPC-IS를 통해서 운반차량 관리자의 PDA, Local ONS(l-ONS), Internal Systems, remote ONS(r-ONS)에게 제품과 운반 정보를 자동으

로 전송함

1.7 운반차량에 팔레트를 적재함

1.8 운반차량이 RFID 리더기가 장착된 RFID GATE를 통과함

1.8.1 RFID 리더에 의해 운반차량 ID가 자동으로 읽힘

1.8.1.1 자동으로 Advanced Shipping Notice(ASN)를 보냄

1.8.1.2 운반차량 관리자의 PDA에 있는 Bill of Lading(BOL)을 자동으로 생성하고, 정보를 업데이트함

1.8.1.3 SAVANT와 EPC-IS를 통해서 운반차량 관리자의 PDA, Local ONS(l-ONS), Internal Systems, remote ONS(r-ONS)에게 BOL 복사본을 자동으로 보냄

1.8.1.4 Global Positioning System(GPS)의 운반추적 상태를 자동적으로 초기화시킴

1.9 운반지역으로 운반차량이 이동함

▪ Receiving Process

1. BOL을 받음

1.1 운반차량이 RFID 리더기가 장착된 RFID GATE를 통과함

1.1.1 RFID 리더에 의해 운반차량 ID가 자동으로 읽힘

1.1.1.1 BOL의 복사본이 인터넷을 통해서 r-ONS로부터 l-ONS에게 자동적으로 업로드됨

1.1.1.2 SAVANT와 EPC-IS를 통해서 ERP의 PO로 BOL을 자동으로 연결함

1.1.1.3 WMS의 재고량을 자동으로 생성함

1.1.1.4 지게차를 예약하기 위해 자동으로 관리자의 핸드폰이나 PDA로 알람 서비스를 제공함

1.1.1.5 자동으로 unloading을 초기화함

2. 제품 받음

2.1 지게차가 RFID 리더기가 장착된 RFID GATE를 통과함

2.2 운반차량 안에 지게차를 놓음

2.3 운반차량으로부터 팔레트를 짚음

2.4 창고내부로 지게차를 후진시킴

2.5 RFID 리더에 의해 상자와 팔레트에 부착된 태그가 자동으로 읽힘

2.5.1 SAVANT를 통해서 WMS, EPC-IS, I-ONS에 정보가 자동적으로 업데이트됨

2.5.2 자동으로 Shipping 프로세스를 초기화함

▪ u-SCM 메타데이터

* 개체에 관련된 메타데이터

이름	속성
원자재	원자재 번호/ID, 원자재명, 생산지, 생산일자, 위치정보(좌표정보)
부품	부품 번호/ID, 부품명, 제조업체명, 제조일자, 품질등급
중간자재	중간자재 번호/ID, 중간자재명, 제조업체명, 제조일자, 품질등급
완제품	제품 번호/ID, 입자재 및 부품 메타데이터 정보, 중간자재 메타데이터 정보, 제품명, 제품회사명, 품질등급
공정 컴포넌트(예: 팔레트)	공정 컴포넌트 번호/ID, 컴포넌트명, 제조업체명, 제조일자, 관리자
공장 내부	공장 내부 번호/ID, 공장 내부명, 건물주, 건설업체명, 건설일자
입고 정보 파일	날짜, 입고품목, 수량, 관리자명, 입고품목 공급업체명
출고 정보 파일	날짜, 출고품목, 수량, 관리자명, 출고품목 공급업체명
창고 내부	구역 번호/ID, 구역 명, 위치정보(좌표정보)
차량	차량 번호/ID, 위치정보(좌표정보) 입고/출고 정보 파일 메타데이터
제품상자/소형집합단위	제품상자/소형집합단위 번호/ID
컨테이너	컨테이너 번호/ID, 관리자, 제조업체명, 제조일자, 구경요소
완제품 납품 내역 파일	날짜, 납품 품목, 수량, 공급업체 명, 수요업체 명
제품 운반 카드	제품 운반 카드 번호/ID, 위치정보(좌표정보) 완제품 메타데이터
제품 선반	선반 구역 번호/ID, 구역 명, 위치정보(좌표정보)
쇼필 카드	쇼필 카드 번호/ID
고객 정보	고객명, 주민등록번호, 멤버쉽 유/무, 선호제품명, 필요한 제품명
센서 네트워크 ID	센서 네트워크 ID, 센서 네트워크 이름, 구축위치, 주요기능 설명, 동시 처리 가능한 질의 개수, 조건을 포함하는 질의를 처리할 수 있는지 여부, SUM/MAX/YMIN/YAVG(의 어그리게이션 질의 처리 수행 가능 여부, 기간 및 주기를 설정하여 반복적으로 질의를 처리할 수 있는 Continuous 질의 처리 수행 가능 여부, 관리자 정보, 구축시점, 센서네트워크 접속 가능 여부, 통신프로토콜 방식, 센서네트워크가 호스트에 접속을 요청하는 모드, 센서네트워크가 갖고 있는 센서노드 개수, 보안 레벨, 센서네트워크 모니터링 주기, 센서네트워크 모니터링 응답 제한시간[10]
센서노드 ID	센서 노드 ID, 주요기능, 위치, SW 버전, 센서노드의 IPv6주소[10]

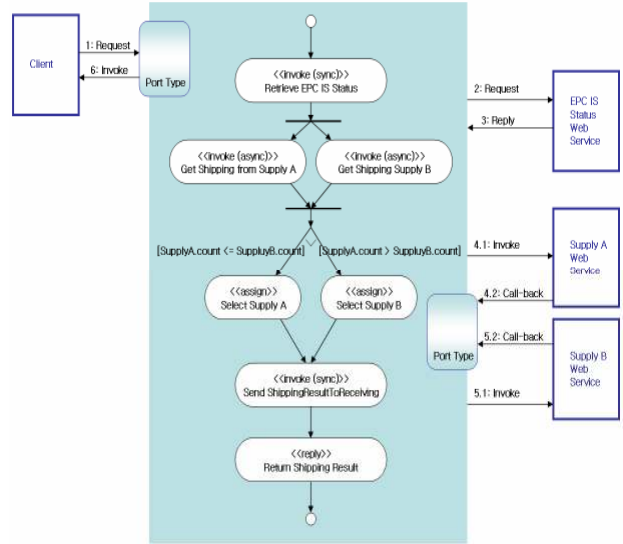
* 상황에 따라 달라지는 메타데이터

이름	속성
물리적 환경 상황	공장 내부, 창고 내부, 차량 내부, 건물 내부
신원 상황	ID, 비밀번호, 성명, Nick-name
공간 상황	위치, 방향, 속도, 량, 기속도
시간 상황	일자(연도, 월, 일), 시각, 요일, 계절
환경 상황	온도, 습도, 조도, 소음
컴퓨팅 시스템 상황	Power On/Off, On/Off-line, Idle/Active/Standby, Log-In/Out
센서 네트워크 상황	Power On/Off, Idle/Active/Standby, 평균 건원 건량,
센싱 타입 상황[10]	센싱 값의 단위, 센싱 데이터 종류, 데이터 크기, 센싱 주기[10]
가용 자원	센서노드 배터리, 디스플레이, 인터넷, 시스템
가용 상황	USN 자원, 인격자원, 컴포넌트, 차량, 시설
결관 상황	사용자, 허용정보, 인접성
이력 상황	
장애 상황	
(경고)사고 처리 상황	

3.2 BPEL모델링

BPEL에서 u-SCM 프로세스를 정의하는 과정에서, 기존 서비스를 조합하여 새로운 컴포지트(composite) 웹 서비스를 모델링 한다. BPEL 컴포지트 웹 서비스의 인터페이스는 다른 웹 서비스와 마찬가지로 일련의 포트 타입(port type)을 사

용하고, BPEL에 정의된 비즈니스 프로세스를 호출하려면, BPEL 정의 결과로 생성된 컴포지트 웹 서비스를 호출해야 한다. <그림4>는 u-SCM프로세스의 스키마틱 뷰(schematic view)를 보여준다.



<그림4> u-SCM을 위한 BPEL 프로세스

▪ 관련 웹 서비스

* EPC IS Status 웹 서비스: “ EPCIS StatusPT” 포트 타입(port type)을 통해 EPCIS Status작업(operation)을 수행하고 현재 주문량의 상태를 점검한다. 공급 가능한 원자재/부품/중간자재/완제품/공정컴포넌트/센서네트워크 등 개체관련 메타데이터 정보를 반환한다.

* Supply 웹 서비스: 비동기식으로 수행된다. 먼저, SupplyAvailabilityPT 포트 타입을 통해 SupplyAvailability 작업을 수행하고 공급 가능여부를 확인한다. 웹 서비스는 결과를 반환하기 위해 두 번째 포트 타입, SupplyCallbackPT를 사용한다. 이 포트를 통해서 SupplyShippingCallback 작업이 수행된다.

▪ WSDL 정의

BPEL 프로세스를 웹 서비스의 형태로 공개하는데 사용할 WSDL을 정의한다. 이 프로세스에서는 클라이언트로부터 메시지를 전달받고 결과를 반환하는 과정이 수반되므로, 입력 메시지를 위한 ShippingApprovalPT 포트 타입과 (callback을

사용한 비동기식) 결과 반환을 위한 ClientCallbackPT 포트 타입을 정의한다.

▪ **파트너 링크 타입(partner link type) 정의**

* shippingLT: BPEL 프로세스 클라이언트와 BPEL 프로세스 간의 상호작용을 기술하는데 사용된다(비동기식). shippingLT 파트너 링크 타입은 BPEL 프로세스의 WSDL에 정의된다.

```
<partnerLinkType name="shippingLT">
  <role name="shippingService">
    <portType name="tns:ShippingApprovalPT" />
  </role>
  <role name="shippingServiceCustomer">
    <portType name="tns:ClientCallbackPT" />
  </role>
</partnerLinkType>
```

* epcLT: BPEL 프로세스와 EPC IS Status 웹 서비스 간의 상호작용을 기술하는데 사용된다(동기식). epcLT 파트너 링크 타입은 EPC IS Status 웹 서비스의 WSDL에 정의된다.

```
<partnerLinkType name="epcLT">
  <role name="epcShippingStatusService">
    <portType name="tns:EPCShippingStatusPT" />
  </role>
</partnerLinkType>
```

* supplyLT: BPEL 프로세스와 Supply 웹 서비스 간의 상호 작용을 기술하는데 사용된다(동기식). Supply 웹 서비스는 BPEL 프로세스에 대하여 callback 호출을 수행한다. supplyLT 파트너 타입 링크는 Supply 웹 서비스의 WSDL에 정의된다.

```
<partnerLinkType name="supplyLT">
  <role name="supplyService">
    <portType name="tns:ShippingAvailabilityPT" />
  </role>
  <role name="supplyCustomer">
    <portType name="tns:ShippingCallbackPT" />
  </role>
</partnerLinkType>
```

▪ **BPEL 프로세스 개발**

파트너 링크를 정의하고, 변수를 선언하고, 프로세스 로직 정의를 작성한다. 본 연구에서는 디플트네임스페이스로 정의하고, 변수 타입은 WSDL message type을 사용한다.

```
<process name="BusinessShippingProcess"
  targetNamespace="http://packtpub.com/bpel/shipping/"
  xmlns="http://schemas.xmlsoap.org/ws/2003/03/business-process/"
  xmlns:shi="http://packtpub.com/bpel/shipping/"
  xmlns:epc="http://packtpub.com/service/epc/"
  xmlns:sup="http://packtpub.com/service/supply/" >
```

```
<partnerLinks>
  <partnerLink name="client"
    partnerLinkType="shi:shippingLT"
    myRole="shippingService"
    partnerRole="shippingServiceCustomer" />
  <partnerLink name="epcShippingStatus"
    partnerLinkType="epc:epcLT"
    partnerRole="epcShippingStatusService" />
  <partnerLink name="SupplyA"
    partnerLinkType="sup:shippingLT"
    myRole="supplyCustomer"
    partnerRole="supplyService" />
  <partnerLink name="SupplyB"
    partnerLinkType="sup:shippingLT"
    myRole="supplyCustomer"
    partnerRole="supplyService" />
</partnerLinks>
```

```
<variables>
  <!-- input for this process -->
  <variable name="ShippingRequest"
    messageType="shi:ShippingRequestMessage" />
  <!-- input for the Epc Shipping Status web service -->
  <variable name="EpcShippingStatusRequest"
    messageType="epc:EpcShippingStatusRequestMessage" />
  <!-- output from the Epc Shipping Status web service -->
  <variable name="EpcShippingStatusResponse"
    messageType="epc:EpcShippingStatusResponseMessage" />
  <!-- input for SupplyA and SupplyB web services -->
  <variable name="ShippingDetails"
    messageType="sup:ShippingRequestMessage" />
  <!-- output from SupplyA Supplies -->
  <variable name="ShippingResponseAA"
    messageType="sup:ShippingResponseMessage" />
  <!-- output from SupplyB Supplies -->
  <variable name="ShippingResponseBA"
    messageType="sup:ShippingResponseMessage" />
  <!-- output from BPEL process -->
  <variable name="ShippingResponse"
    messageType="sup:ShippingResponseMessage" />
</variables>
```

```
<sequence>
  <!-- Receive the initial request for business shipping from client -->
  <receive partnerLink="client"
    portType="shi:ShippingApprovalPT"
    operation="ShippingApproval"
    variable="ShippingRequest"
    createInstance="yes" />
  <!-- Prepare the input for the Epc Shipping Status Web Service -->
  <assign>
    <copy>
      <from variable="ShippingRequest" part="epc" />
      <to variable="EpcShippingStatusRequest" part="epc" />
    </copy>
  </assign>
  <!-- Synchronously invoke the Epc Shipping Status Web Service -->
  <invoke partnerLink="epcShippingStatus"
    portType="epc:EpcShippingStatusPT"
    operation="EpcShippingStatus"
    inputVariable="EpcShippingStatusRequest"
    outputVariable="EpcShippingStatusResponse" />
```

```

<!-- Prepare the input for AA and DA -->
<assign>
  <copy>
    <from variable="ShippingRequest" part="shippingData"/>
    <to variable="ShippingDetails" part="shippingData"/>
  </copy>
  <copy>
    <from variable="EpcShippingStatusResponse" part="shippingClass"/>
    <to variable="ShippingDetails" part="shippingClass"/>
  </copy>
</assign>
<!-- Make a concurrent invocation to AA in DA -->
<flow>
  <sequence>
    <!-- Async Invoke of the AA web service and wait for the callback -->
    <invoke partnerLink="SupplyASupplies"
      portType="sup:ShippingAvailabilityPT"
      operation="ShippingAvailability"
      input Variable="ShippingDetails" />
    <receive partnerLink="SupplyASupplies"
      portType="sup:ShippingCallbackPT"
      operation="ShippingCallback"
      variable="ShippingResponseAA" />
  </sequence>
  <sequence>
    <!-- Async Invoke of the DA web service and wait for the callback -->
    <invoke partnerLink="SupplyBSupplies"
      portType="sup:ShippingAvailabilityPT"
      operation="ShippingAvailability"
      input Variable="ShippingDetails" />
    <receive partnerLink="SupplyBSupplies"
      portType="sup:ShippingCallbackPT"
      operation="ShippingCallback"
      variable="ShippingResponseDA" />
  </sequence>
</flow>

```

```

<!-- Select the best offer and construct the ShippingResponse -->
<switch>
  <case condition="bpws:getVariableData('ShippingResponseAA',
    'confirmationData', '/confirmationData/Count')
    <=> bpws:getVariableData('ShippingResponseDA',
    'confirmationData', '/confirmationData/Count')">
    <!-- Select SupplyA Supplies -->
    <assign>
      <copy>
        <from variable="ShippingResponseAA" />
        <to variable="ShippingResponse" />
      </copy>
    </assign>
  </case>
  <otherwise>
    <!-- Select SupplyB Supplies -->
    <assign>
      <copy>
        <from variable="ShippingResponseDA" />
        <to variable="ShippingResponse" />
      </copy>
    </assign>
  </otherwise>
</switch>
<!-- Make a callback to the client -->
<invoke partnerLink="Client"
  portType="shi:ClientCallbackPT"
  operation="ClientCallback"
  input Variable="ShippingResponse" />
</sequence>
</process>

```

4. 결론

본 논문에서는 비즈니스 프로세스 표준언어인 BPEL을 이용하여, RFID, EPC 네트워크 기술을 이용한 u-SCM프로세스를 모델링 하였다. u-SCM프로세스를 위해서 Shipping과 Receiving프로세스로 구체화하고, EPC IS에 저장될 수 있는 메타데이터를 정의하였으며, BPEL에서 u-SCM

프로세스를 정의하는 과정에서, 기존 서비스를 조합하여 새로운 컴포지트(composite) 웹 서비스를 모델링 하였다. 본 논문에서 제안한 BPEL을 이용한 u-SCM프로세스는 유비쿼터스 환경에서도 기업 내 조직 간, 기업 간 정보공유를 통한 상호작용을 위한 업무프로세스 혁신에 도움이 될 초석이 될 것이다.

향후 실제 deploy 및 테스트를 거쳐서 프로토타입을 구현할 수 있도록 할 것이다. 설계 부분도 복잡한 프로세스 구조에 대해서 BPEL로 모델링하기 위한 방안에 대한 연구가 이루어져야 한다.

[참고문헌]

- [1] “국내외 물류부문의 RFID 도입에 따른 SWOT분석과 사례연구” 장명희, 노미진, 해운물류연구, 47권, 2005.12, p.151-179
- [2] “RFID 확산 추진현황 및 전망” 이은곤, 정보통신정책, 16권 6호, 2004, p.1-24
- [3] EPCglobal Architecture Framework. <http://www.epcglobalinc.org/>
- [4] scm.korcham.net: 대한상공회의소, KCCI, 2004
- [5] “공급사슬관리(SCM)를 활용한 기업의 물류 혁신” 이신규, 창업정보학회지, 9권 1호, 2006.3, p.231-251
- [6] Andrews. T. et al., Business Process Execution Language for Web Services BPEL4WS v.1.1. 2003
- [7] Snell, J., Automating business processes and transactions in Web Services, <http://www-106.ibm.com>, 2002
- [8] “BPM을 이용한 웹서비스 기반의 SCM프로세스 실행”, 배혜림외, 한국전자거래학회지 제9권 제4호, 2003, p65-83
- [9] “서비스 지향 아키텍처 기반의 자재관리시스템 설계”, 안민정, 이홍철, IE Interfaces Vol.20, No.3, p257-266, 2007.9
- [10] 정보통신단체표준, TTAS.KO-06.0168, 한국정보통신기술협회(TTA)