

프로덕트라인 아키텍처 기반 RFID 애플리케이션 개발을 위한 프레임워크

문 미 경[†] · 김 한 준^{**} · 염 근 혁^{***}

요 약

소프트웨어 프로덕트 라인은 소프트웨어 재사용에 대한 활동들을 미리 계획하고 개발 프로세스의 연속적인 부분으로 이루어지도록 지원하기 위한 방법이다. 이 방법에서 가장 중요한 기술은 관련된 시스템들, 즉 도메인에서 공통성과 가변성(commonality and variability: C&V)을 분석하고 관리하는 것이다. 본 논문에서는 소프트웨어 프로덕트 라인 방법과 RFID 적용 기술을 결합시키고자 한다. 즉, 각 기업 또는 도메인마다 RFID 기술을 도입하려고 할 때, 애플리케이션이 공통적으로 처리해야 하는 활동들을 식별하고 각 활동들이 가지는 가변요소들을 분석하여, 이를 재사용 가능한 소프트웨어 프로덕트 라인 아키텍처로 자산화 하는 방법을 제시한다. 또한 이를 실현하기 위하여 재사용 가능한 프로덕트 라인 아키텍처를 모델링 할 수 있고 식별된 공통된 기능들을 재사용 가능한 컴포넌트로 제공해주는 프레임워크를 개발한다. 공통된 아키텍처를 공유하고 반복적인 기능들을 자산으로 만들어 재사용하는 것은 경제적인 측면과 품질에서 상당한 이점을 가져다 준다.

키워드 : 소프트웨어 프로덕트 라인, 프로덕트 라인 아키텍처, 공통성과 가변성, RFID 애플리케이션, RFID 이벤트

Framework for Developing RFID Applications Based on Product Line Architecture

Mikyeong Moon[†] · Hanjun Kim^{**} · Keunhyuk Yeom^{***}

ABSTRACT

Software produce line engineering is a method that prepares for future reuse and supports seamless reuse in the application development process. Analyzing the commonality and variability between products in a product line is one of the essential concerns that must be considered when building a product line. In this paper, we suggest to combine a product line engineering approach with RFID applications technology. More concretely, common activities are identified among the RFID-enable applications and the variability in the common activities is analyzed in detail. Then, we suggest reusable product line architecture as a product line asset. In addition, as a tool for supporting this research, a framework which provides both reusable product line architecture for REID applications and the components that implement concrete realization of the architecture is developed. Sharing a common architecture and reusing assets to deploy recurrent services may be considered an advantage in terms of economic significance and overall quality.

Key Words : Software Product Line, Product Line Architecture, Commonality and Variability, RFID-enabled Application, RFID Event

1. 서 론

RFID(Radio Frequency IDentification)는 태그, 레이블, 카드 등에 저장된 데이터를 무선주파수를 이용하여 리더에서 자동 인식하는 기술이다[1]. 최근 RFID는 소형화, 저가화, 고기능화가 점차적으로 실현되면서 그 기술이 적용되는 분야도

더욱 다양해지고 있다. RFID 기술은 다른 업종간에 공동 이용이나 제품의 속성정보, 이력정보, 나아가 실시간 환경정보 등 각종 고도의 정보이용이 가능해져 기존의 애플리케이션과는 다른 새로운 비즈니스나 서비스를 형성시킨다. RFID 애플리케이션은 RFID 태그의 실시간 정보를 기반으로 하는 업무 프로세스를 의미하는 것으로, RFID 기반 정보를 이용해 비즈니스 프로세스를 자동화하고 변형시킬 수 있도록 요구된다 [2]. 이러한 요구사항을 위한 접근방법으로 도메인에서 공통적으로 재사용 될 수 있는 기능과 다소 상이하게 나타날 수 있는 기능들을 식별하고 이를 명시적으로 관리하는 소프트웨어 프로덕트 라인 방법이 적용될 수 있다.

RFID 애플리케이션이 공통으로 수행해야 하는 활동에는

* 이 논문은 2007년 정부(교육인적자원부)의 재원으로 한국학술진흥재단의 지원을 받아 수행된 연구임(지방연구중심대학육성사업/차세대물류IT기술연구사업단).

† 정 회 원 : 부산대학교 정보컴퓨터공학부 연구교수

** 준 회 원 : 부산대학교 컴퓨터공학과 석사과정

*** 정 회 원 : 부산대학교 컴퓨터공학과 교수

논문접수 : 2007년 6월 19일, 심사완료 : 2007년 11월 1일

RFID 미들웨어로부터 발생하는 저수준의 RFID 이벤트를 다양한 정보 서버들을 참조하여 고수준의 이벤트로 변환시키기 위한 일련의 활동(activity)들이 있다. 또한 이러한 공통된 활동들 사이에 도메인마다 다르게 적용되는 비즈니스 규칙을 가변적 활동으로 분리할 수 있다. 본 논문에서는 이러한 공통성과 가변성을 분석하여 이를 명시적으로 표현하고 체계적으로 구조화 시킴으로써 재사용 가능한 프로덕트 라인 아키텍처(product line architecture: PLA)로 자산화 하는 방법을 제시한다. 제시하는 아키텍처는 비즈니스 프로세스에서의 가변성을 식별하여 비즈니스 프로세스 모델(business process model: BPM)의 형태로 표현된다. 또한 본 논문에서 제시하는 방법을 지원하기 위한 도구로서 개발한 프레임워크를 제시한다. 프레임워크는 식별된 공통된 기능들을 재사용 가능한 컴포넌트로 제공해주며 프로덕트 라인 아키텍처를 개발 할 수 있도록 한다. 이러한 아키텍처는 XML로 변환되어 저장된다. 또한 하나의 RFID PLA는 다양한 사용자의 요구에 따라 결정(decision)과 가지치기(pruning) 과정을 거쳐 특정 RFID 애플리케이션을 위한 아키텍처로 만들어 준다. 이 아키텍처는 프레임워크에서 실제 실행됨으로써 RFID 관련 데이터가 프레임워크를 통해 자동으로 선처리되게 된다.

결론적으로 본 논문에서는 RFID 애플리케이션 개발을 위하여 소프트웨어 프로덕트 라인 기술을 접목시킴으로써 고품질의 RFID 애플리케이션을 빠르게 개발할 수 있게 해주며, RFID 기술을 도입하려는 각 기업들에게 각 기업의 비즈니스 규칙에 따른 유연한 옵션을 제공할 수 있게 해 준다.

2. 기반 연구

2.1 소프트웨어 프로덕트 라인 방법론

소프트웨어 프로덕트 라인은 일련의 관련된 시스템들, 즉 도메인 내에서 다시 재사용될 가능성이 높은 공통된 부분들을 식별하고, 시스템마다 상이하게 나타나는 가변적 요소들을 분석하는 것이다[3, 4]. 도메인의 주요 산출물들 -요구사항[5], 아키텍처[6] 등- 은 분석된 공통성과 가변성을 명시적으로 나타냄으로써 프로덕트 라인의 자산(asset)이 된다.

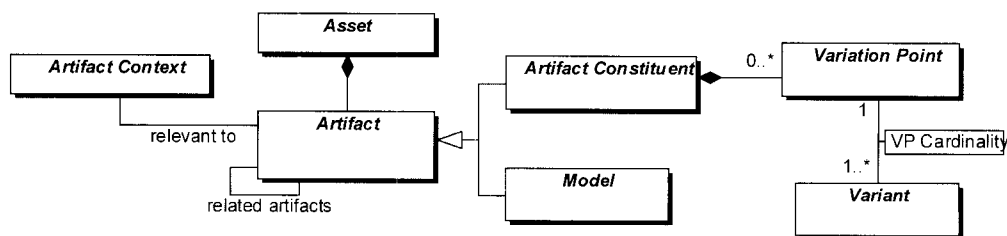
(그림 1)은 소프트웨어 프로덕트 라인 자산에 대한 메타모델이다. 자산은 하나 이상의 산출물(artifact)들로 구성되며, 다른 산출물과 관계를 가질 수도 있다. 하나의 산출물은 요구사항 개발, 설계 또는 런타임 문맥과 같은 특정 문맥(artifact context)과 관련된다. 산출물은 산출물 구성요소(artifact

constituent)와 모델(model)의 형태로 특수화(specialization)될 수 있다. 산출물 구성요소는 재사용 될 때, 수정될 수 있는 지점을 나타내는 가변점(variation point)을 가지기도 하며, 모델은 각 형태에 따라 산출물 구성요소가 내포하고 있는 가변점을 표현할 수 있어야 한다. 하나의 가변점에는 가변성 실현 시 바인딩 될 수 있는 구체적인 값인 가변치(variant)가 하나 이상 연결될 수 있다. 또한 실현 시 가변점에 바인딩 될 수 있는 가변치 개수를 대응값(vp cardinality)로 표현한다.

자산들은 소프트웨어 개발의 각기 다른 단계에서 생산되기 때문에 표현요소들의 추상화 수준이 다르며 이로 인해 각 핵심자산이 가지고 있는 가변성 또한 각기 다른 수준에서 각기 다른 유형으로 나타나게 된다. 또한 가변성의 상세화 정도에 따라서도 공통성을 식별하는 수준과 가변점을 상세화하는 수준으로 나누어 분석하여야 한다[7]. 자산의 공통성과 가변성 분석에 대한 기존 연구들에서는 핵심자산의 구분 없이, 추상화의 정도를 고려하지 않고 일관되게 가변성을 분석하였다. 본 논문에서의 초점은 비즈니스 프로세스 분석 문맥에서 가변성을 수준별로 구분하여 분석한 후, 이를 프로세스 모델 형태의 아키텍처로 표현하여 자산으로 만들고자 하는 것이다.

2.2 RFID 아키텍처

RFID는 라디오 신호에 따라 반응하는 태그를 이용하여 해당 사물을 인식하는 방법이다. RFID 아키텍처는 RFID 태그를 읽는 리더 (reader), RFID 미들웨어[8], EPC (Electronic Product Code) 에 대한 상세 정보를 보관하고 이용할 수 있는 EPCIS (EPC Information Service)[9]와 EPCIS의 위치를 찾기 위한 ONS (Object Name Service)[10] 등의 외부 지원 시스템들로 구성된다. RFID 미들웨어가 리더로부터 읽혀진 태그의 중복되는 이벤트를 제거하고 데이터의 흐름을 관리하는데 유용하지만, 이를 통해 발생하는 데이터도 단순한 태그 값의 형태이다. 애플리케이션에 전달되는 구체적인 한 예로, "Dock_A urn:epc:tag:sgtin-96:4.011562.0557083.19212150 up 10:12:00:06:05"에는 RFID 이벤트가 발생한 위치, 제품의 태그 값, 방향 및 시간의 정보만을 가지고 있다. RFID 애플리케이션 개발자는 이러한 정보로부터 "authenticatedUser (success): id 9034 name Hong Kildong type Manager"와 같이 의미 있는 정보를 끌어내야 한다. 이를 위해 RFID 애플리케이션 개발자들은 EPCIS, ONS 등의 여러 외부 정보 서버와의 통신을 통해 RFID 데이터에 의미(semantic) 정보를 부여하고



(그림 1) 소프트웨어 프로덕트 라인 자산 메타모델

비즈니스 규칙을 적용하는 과정을 수행해야 한다. 이러한 RFID 데이터 처리 과정은 애플리케이션 개발자들에게 어렵고 반복되고 작업이 된다. 또한 이는 RFID 이벤트 처리 과정과 비즈니스 로직 처리 과정을 혼합시킴으로써 향후 애플리케이션의 유지보수 및 수정도 어렵게 하는 요인이 된다.

그러므로 본 논문에서는 RFID 애플리케이션 개발을 위해 재사용될 수 있는 기능들을 추출하고 가변될 수 있는 요소들을 분석하여 프로덕트 라인 아키텍처로 만든 후, 이를 프레임워크 형태로 구현하고자 한다.

3. RFID PLA를 위한 전체적 구조 및 개념 정의

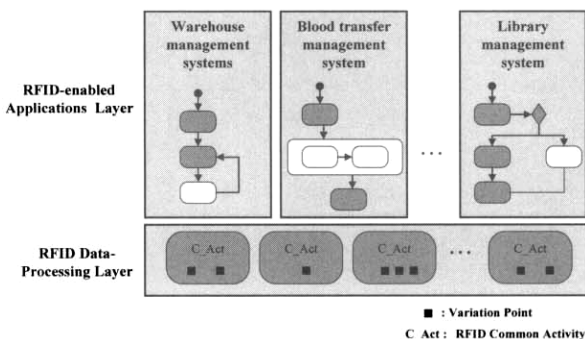
3.1 RFID 애플리케이션을 위한 2-계층 구조

RFID 애플리케이션은 RFID 태그의 실시간 정보를 처리하는 비즈니스 프로세스를 의미하며, 관련된 일련의 RFID 애플리케이션의 집합을 RFID 도메인으로 정의한다. RFID 도메인 아키텍처를 수준별로 구조화 시켜보면 (그림 2)와 같다.

먼저 하위 계층에는 RFID 시스템 도입 관련 RFID 데이터를 처리하기 위한 액티비티들이 위치한다. 이 계층의 액티비티들은 RFID 도메인에서 공통으로 사용되기 때문에 공통적 액티비티로 식별한다. 그러나 이러한 공통적 액티비티들도 내부적으로 가변되는 요소들을 가지고 있는데 이는 가변점을 통해 분석하여 표현한다. 이에 대한 설명은 4장에서 자세히 기술한다. (그림 2)의 상위 계층에는 하위 계층의 RFID 관련 액티비티들과 도메인 관련 - warehouse management system, blood transfer management system, 등 - 액티비티들이 일련의 흐름을 구성하여 비즈니스 프로세스 모델을 구축하게 된다. 이때, BPM에는 분석된 공통성과 가변성이 명시적으로 표현된다. 이에 대한 설명은 5장에서 자세히 기술한다.

3.2 RFID PLA를 위한 액티비티 분류

RFID PLA는 비즈니스 프로세스 모델(BPM)을 기본으로 한다. 비즈니스 프로세스 모델은 시작과 끝이 명확한 일련의 액티비티들의 흐름으로 표현된다. 액티비티는 업무의 의미를 가지는 최소 단위 업무활동으로 정의된다[11]. 여기에는 주문입력, 승인 등과 같이 사람이 수행하는 액티비티와 메일, 데이터 전송, 야간 daily batch job과 같이 자동으로



(그림 2) RFID 애플리케이션을 위한 2-계층 구조

수행되는 액티비티가 있다. 본 논문에서는 공통적으로 사용되는 이러한 액티비티 개념을 바탕으로 RFID PLA를 위하여 특수화 된 몇 가지 액티비티를 더 정의한다. (그림 3)에서 이러한 액티비티들 사이의 관계를 보여줌으로써 각 액티비티들의 개념을 명확히 하고자 한다.

- 도메인 액티비티 (domain activity) - 관련 BPM에서 같은 기능을 수행하지만 각 BPM마다 다소 상이한 차이점을 가진 액티비티들을 하나의 액티비티로 일반화(generalized)시킬 수 있다. 이 때, 일반화 된 액티비티를 도메인 액티비티로 정의한다. 또한 일반화 되기 전에 가지고 있던 차이점이 도메인 액티비티가 가지는 가변성이 된다. 도메인 액티비티는 RFID PLA의 기본 구성요소가 된다.
- RFID 액티비티 (RFID activity) - 액티비티 중, RFID 데이터 처리를 위해 활동하는 액티비티를 RFID 액티비티로 정의한다.
- RFID 공통적 액티비티 (RFID common activity) - 도메인 액티비티 중, RFID 데이터 처리를 위해 활동하는 액티비티를 RFID 공통적 액티비티로 정의한다. 이는 RFID 액티비티들의 일반화 형태이다.

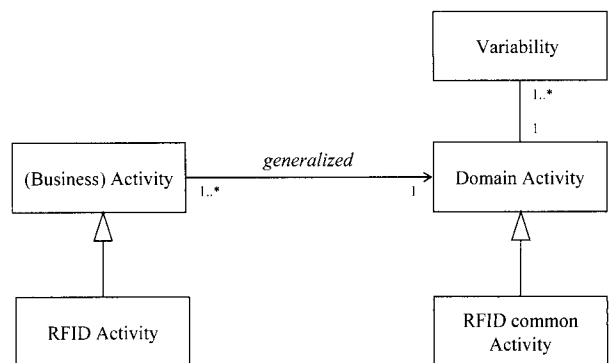
본 논문에서는 비즈니스 프로세스를 모델링하기 위하여 가장 많이 사용되는 표현기술언어 중 하나인 UML2 액티비티 다이어그램을 기본으로 하여 RFID PLA를 모델링 한다.

4. RFID 액티비티 식별 및 가변점 분석

RFID 액티비티는 RFID 애플리케이션 도메인의 공통적 액티비티로 추출할 수 있지만, 여전히 내부 가변성을 가지게 된다. RFID 액티비티들은 활동의 형태에 따라 다음의 세 가지로 분류하여 식별 할 수 있다.

4.1 RFID 이벤트 처리 액티비티

RFID 미들웨어로부터 RFID 데이터를 받기 위해서는 이벤트 주기와 주기에서 생성되는 하나 이상의 보고서를 설명하는 스펙(specification)을 정의(define)하고 등록(subscribe)해야 한다. 스펙에는 읽기 주기가 이벤트 주기에 포함되는



(그림 3) RFID PLA를 위한 액티비티 분류

<표 1> RFID 이벤트 트리거 액티비티

RFID 액티비티	설명	가변점
이벤트 트리거 액티비티	RFID 미들웨어에 스펙을 등록하고 그 결과로 넘어오는 보고서의 처리를 담당한다.	Event request method //이벤트 요청방법 [immediate(spec_ECSpec):ECReport poll(specName:string):ECReports subscribe(specName:string, notificationURI:string):void] LogicalReaders //논리 리더의 목록 BoundarySpec //이벤트 사이클의 시작과 끝을 결정하는 방법 ReportSpecs //이벤트 사이클이 실행된 후에 반환되어야 하는 보고서의 목록 Variables for receiving event //이벤트 사이클 동안 넘겨주는 태그 목록을 저장하기 위한 변수
시간 트리거 액티비티	특정 시각 또는 기간을 설정하여 RFID 데이터를 처리한다.	Setting of trigger time

<표 2> Information service 참조 액티비티

RFID 액티비티	설명	가변점
EPCIS 액티비티	EPCIS는 EPC와 관련된 정보를 획득하고, 관리하며 이를 공유하기 위한 외부 인터페이스를 정의한다. EPCIS 액티비티는 EPCIS에 대한 접근을 처리한다.	Query statements [getEPCCSchema getEPCAttribute getEPCClassSchema getEPCClassAttribute] Parameters required in a query statement [EPC / Schema / Xpath / EPCClass]
ONS 액티비티	ONS는 EPC와 관련한 EPCIS 위치 정보 검색서비스를 제공한다. Ons 액티비티는 ONS에 대한 접근을 처리한다.	
EPCIS Discovery Service (DS) 액티비티	EPCIS DS는 이동하는 EPC의 전체 보관자(custodian) 목록 (EPCIS URL)을 제공한다. EPCIS DS 액티비티는 EPCIS DS에 대한 접근을 처리한다.	Query statements [getEPCLocations getEPCMovingLocations getEPCMovingLocationsTime] Parameters required in a query statement [EPC / epcisURI / inTime / outTime]

논리적 관독기 목록, 이벤트 주기의 경계 결정 방법 사양 및 이 이벤트 주기에서 생성되는 보고서를 각각 설명하는 사양 목록이 포함되어 있다[12]. 이러한 요소들을 설정하고 이에 따라 올라오는 RFID 데이터를 처리하기 위한 액티비티가 이벤트 트리거 액티비티이다. 또한 RFID 시스템은 실시간 데이터를 기본으로 처리하지만, 특정 시간에 일괄(batch) 처리도 가능하게 할 수 있다. 이를 위해 시간 트리거 (time trigger) 액티비티를 정의한다. 이벤트 트리거 액티비티의 가변점으로 이벤트를 요청하는 방법이 있다. 이벤트 요청 방법에는 미들웨어에 스펙을 등록하지 않고 이벤트를 pull 방식으로 요청하는 immediate, 미들웨어에 등록된 이벤트를 pull 방식으로 요청하는 poll, 미들웨어에 등록되어 있는 이벤트 스펙에 따라 push 방식으로 이벤트를 요청하는 subscribe 방식으로 구분할 수 있기 때문에 이는 각 애플리케이션의 요구사항에 맞게 결정된다. 또한 이벤트 요청 스펙에 기술되어야 하는 항목들이 가변점으로 정의된다. 여기에는 이벤트 사이클 동안 하나 이상의 리더로부터 데이터를 가져와야 하기 때문에 논리 리더의 목록(LogicalReaders)과 이벤트 사이클의 시작과 끝을 결정하는 방법을 기술하는 부분(BoundarySpec), 이벤트 사이클이 실행된 후 반환되어야 하는 보고서의 목록(ReportSpecs)이 기술된다. 또한 이벤트 사이클 동안 RFID 미들웨어로부터 넘겨받은 태그목록을 저장하기 위한 변수들이 가변점으로 정의된다.

4.2 Information service 참조 액티비티

RFID 데이터의 관련 정보를 조회하기 위하여 EPCIS (Electronic Product Code Information Service), ONS (Object Naming Service), EPCIS DS (Discovery Service) 등¹⁾을 참조할 수 있다. 이러한 정보 서비스를 사용하기 위하여 <표 2>에 나열되어 있는 액티비티들을 정의한다.

4.3 비즈니스 서비스 액티비티

RFID 애플리케이션은 비즈니스 규칙을 설정하고 이 규칙에 따라 RFID 이벤트를 검사한 후, 규칙 조건에 해당하는 비즈니스 서비스를 수행한다. <표 3>에는 RFID 비즈니스 규칙에 따라 호출되는 비즈니스 서비스들을 처리할 수 있도록 해주는 액티비티들이 나열되어 있다. 규칙 검사를 통한 서비스 호출은 클라이언트 파트너 링크의 WSDL(web service description language)과 그 서비스가 배치된 엔드 포인트 주소를 이용하게 된다. 파트너 링크의 WSDL에는 데이터 타입정의, 통신 데이터의 추상 메시지, 서비스 동작에 대한 추상 오퍼레이션 등이 정의되어 있다 [13]. 서비스 연결 액티비티는 규칙 검사를 통해 의미적으로 필터링 된 RFID 데이터를 비즈니스 서비스의 클라이언트 파트너 WSDL에 정의된 호출 인

1) 현재는 EPC에 대한 속성정보 서버와 이력정보 서버에 접근하는 액티비티만 고려할 것이지만, 추후 많은 센서에 대한 실시간 환경정보를 위해서 이를 처리하기 위한 'context ontology 액티비티' 등이 추가될 수 있다.

〈표 3〉 비즈니스 서비스 액티비티

RFID 액티비티	설명	가변점
규칙 검사 (rule checker) 액티비티	RFID 데이터에 대하여 비즈니스 규칙조건을 적용한다. 의미적(semantic) 필터링 역할을 한다.	Rule condition
서비스 연결 (service mapper) 액티비티	비즈니스 프로세스의 파트너 서비스를 호출한다. 서비스 이름, 배치된 엔드 포인터, 초기화 오퍼레이션, 그리고 초기화 오퍼레이션의 인자 메시지를 속성으로 정의한다.	ServiceName endpoint targetNamespace operation message

자에 바인딩 시킨다. 이 때, 비즈니스 서비스 중 자주 발생 되는 몇 가지 액티비티들 - 이메일 서비스 액티비티, 저장 서비스 액티비티, 모니터링 서비스 액티비티- 은 특화된 비즈니스 서비스로 식별해 놓고 이를 프레임워크에서 직접 처리 가능하도록 구현해 놓을 수 있다.

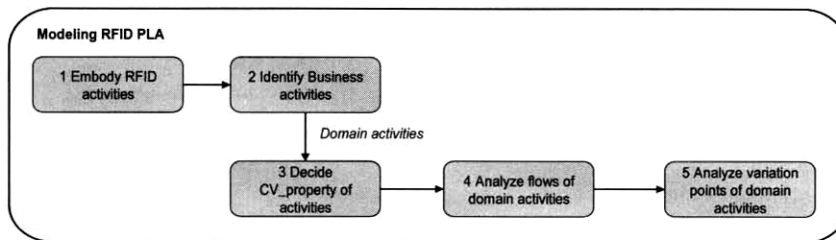
5. RFID PLA 개발

4장에서 식별된 RFID 액티비티를 바탕으로 본 장에서는 RFID PLA를 모델링하는 방법을 설명한다. (그림 4)는 RFID PLA 개발 프로세스를 단계별로 도식화하여 나타낸 것이다. 각 단계별 설명은 LIT(logistic information technology) 연구센터[14]에서 개발한 RFID 시스템 (RFID 리더 인터페이스, RFID 미들웨어, EPCIS, EPCIS DS 등)을 기반으로 구현한 스마트 선반(smart shelf) 시스템의 예를 사용한다. 스마트 선반 시스템은 진열된 개별 상품에 부착된 RFID 태그와 선반의 RFID 리더를 이용해, 직원들이 해당 상품의 정확한 위치를 모니터링 할 수 있게 해 주는 시스템이다. 또한 RFID 리더가 내장된 선반에 진열된 상품을 소비자가 선반에서 들었을 때 그 정보가 실시간으로 재고관리 시스템과 연계되어

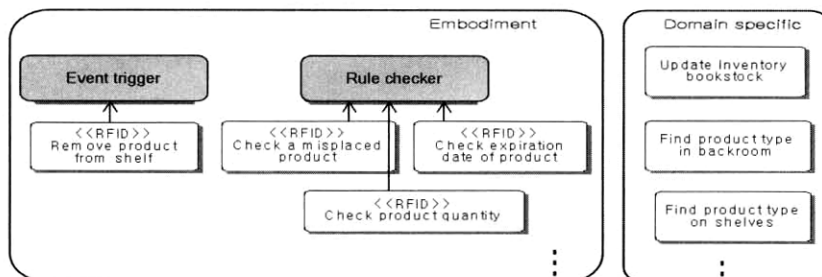
실시간 재고실사가 가능하도록 하며, 잘못된 위치에 진열된 품목을 탐지하여 운영자에게 알릴 수도 있도록 한다.

5.1 도메인 액티비티 구성

도메인 액티비티 도출을 위한 첫 번째 방법은 식별된 RFID 액티비티를 구체화(embodiment) 시키는 것이다. RFID 액티비티의 구체화란 특정 RFID 도메인에 맞게 해당 액티비티의 가변점들을 채우는 것을 의미한다. 예를 들어, 이벤트 트리거 액티비티를 구체화시켜서 스마트 선반에 있는 상품을 읽도록 한다. 이 때, 이벤트 트리거 액티비티는 이벤트 요청 방법, 이벤트 스펙 항목 등을 채워 미들웨어에 등록할 수 있는 형태로 만든다. 이 때, 가변치가 가변점에 바인딩되지 못하는 경우, 이러한 가변점들은 특정 애플리케이션 개발 시점으로 그 결정을 연기시키게 된다. 도메인 액티비티 도출의 두 번째 방법은 RFID 액티비티 외의 비즈니스 액티비티들을 추출하는 것이다. 이는 도메인의 요구사항 문서 또는 기존의 BPM을 분석하여 추출할 수 있다. (그림 5)은 스마트 선반 도메인을 위해 도출된 도메인 액티비티들을 보여준다. 왼쪽 박스는 RFID 액티비티로 추출된 이벤트 트리거(Event trigger) 액티비티와 규칙 검사(Rule checker) 액티비티로부터 구체화되어 추출된 도메인 액티비티들을 보여준다. 그 외 비즈니스 액티비티들이 오른쪽 박스에 포함되어 있다.



(그림 4) RFID PLA개발을 위한 프로세스


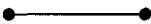


(그림 5) 도메인 액티비티 도출과정

<표 4> 도메인 액티비티의 CV_property

Property type	설명	표현
Common	대부분의 애플리케이션에서 실현되어야 하는 도메인 액티비티	스테레오타입 <<common>> 사용 생략할 수 있음.
Optional	특정 애플리케이션에서만 실현될 가능성이 있는 도메인 액티비티	스테레오타입 <<optional>> 사용

<표 5> 가변점 확장 매커니즘을 위한 표현요소

가변성 표현 요소	설명	표현
가변점 (variation point: vp)	도메인 액티비티가 가변되는 요소를 가지고 있음을 표현할 때 가변점을 붙인다.	
가변치 (variant)	가변되는 형태의 도메인 액티비티 또는 일련의 도메인 액티비티의 흐름을 나타낸다. 도메인 액티비티의 가변점에 연결된다.	
가변점 유형 (vpType)	가변점에 바인더 되는 가변치를 결정하는 유형을 의미한다. True/False의 결정으로 가변치를 결정할 수 있는 유형인 Boolean decision vpType과 여러 개의 가변치 중, 몇 개를 선택해서 결정할 수 있는 유형인 Select decision vpType로 구분한다.	vp property 항목으로 표현
가변점 대응 (vp Binding) 관계	가변점과 가변치 사이의 연결관계를 나타낸다.	
가변치 대응값 (vp Cardinality)	애플리케이션 개발 시 가변점에 연결될 수 있는 가변치의 개수를 의미한다.	vp property 항목으로 표현

5.2 가변성 식별

식별된 도메인 액티비티에 가변성을 분석하여 이를 RFID PLA에 표현한다. RFID PLA에서는 다음과 같이 두 가지 수준으로 구분하여 가변성을 분석한다.

5.2.1 1- 수준 가변성 분석 -

공통성과 선택적 속성 (CV_property)을 이용한 가변성 표현

하나의 도메인 액티비티에서 나타날 수 있는 가변성의 유형이다. 하나의 도메인 액티비티는 애플리케이션 개발 시 기능적 측면에 따라 반드시 실현(realization)되거나 또는 선택적으로 실현될 수 있는 가변성을 가진다. 이러한 가변적 성질은 공통성과 선택적 속성으로 표현될 수 있으며, 이를 도메인 액티비티의 CV_property로 정의한다. 이러한 성질은 비즈니스 프로세스를 구성하는 흐름에서 비즈니스 액티비티의 추가 또는 삭제를 발생시킨다. <표 4>는 도메인 액티비티가 가질 수 있는 CV_property에 대한 설명과 PLA에서 표현할 수 있는 표현법에 대한 설명이다.

5.2.12 2- 수준 가변성 분석 -

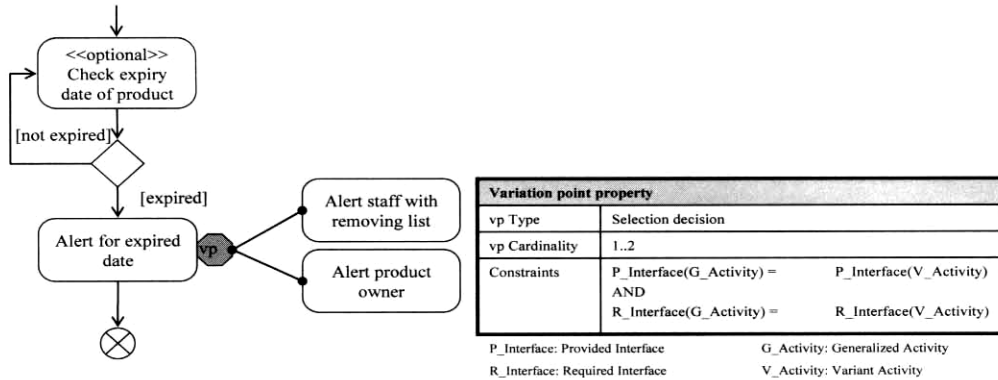
가변점 확장 매커니즘을 이용한 가변성 표현

하나 이상의 도메인 액티비티들에서 발생할 수 있는 가변성 유형이다. 일련의 도메인 액티비티들 또는 도메인 액티비티의 흐름을 일반화(generalization)시켜 이를 가변점을 이용하여 표현하는 것이다. 이러한 성질은 비즈니스 프로세스에서 비즈니스 액티비티의 대체(replacement) 또는 확장(expansion)을 발생시킨다. 가변점 확장 매커니즘을 이용하기 위하여 필요한 가변성 개념들을 <표 5>에서 설명하고 있다.

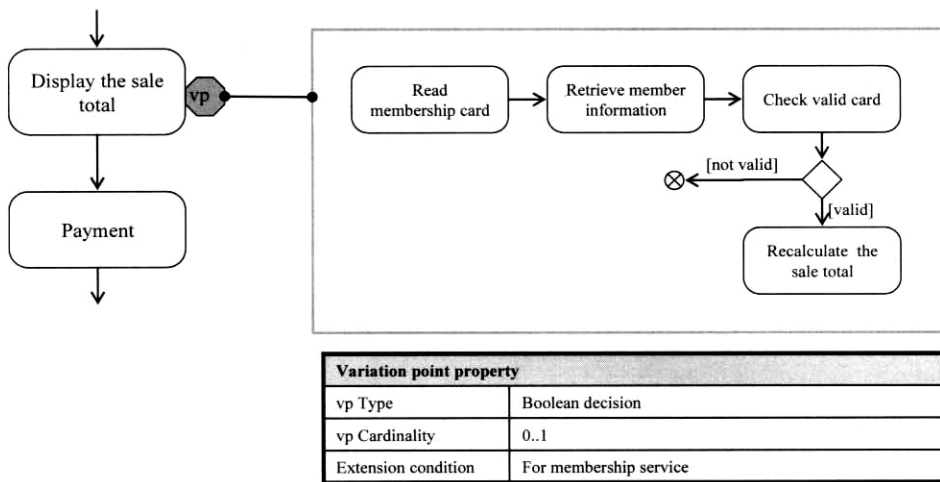
가변점 확장 매커니즘을 이용하여 표현하는 가변성의 유형은 다음과 같다.

① 일반화 관계를 가진 가변점 - 일련의 유사한 기능을 수행하는 두 개 이상의 액티비티들은 하나의 추상화된 도메인 액티비티로 일반화시킬 수 있다. 일반화된 도메인 액티비티는 가변점을 가지게 되고 이러한 가변점에 구체화(specialization)된 도메인 액티비티가 가변치로 연결된다. (그림 6)은 스마트 선반 사례연구의 일부분인, 상품의 유효기간을 확인하고 유효기간이 적절하지 못한 경우 경고 메시지를 알리는 비즈니스 프로세스 모델이다. 이때, 구현하는 애플리케이션마다 유효기간에 대한 오류 메시지를 매장직원에게 알리는 경우도 있고, 상품 판매자에게 바로 알리는 경우도 있다. 이러한 가변적 오류 알림 기능들을 나타내는 도메인 액티비티들을 하나의 일반화된 도메인 액티비티(*Alert for expired date*)로 나타내고 구체적 도메인 액티비티들 (*Alert staff with removing list*, *Alert product owner*)은 가변점과 연결하여 가변치로 나타내고 있다. 또한 가변점 속성으로 vpType는 selection vpType, vpCardinality는 1,2, 그리고 삭제되었을 경우, 인터페이스의 일치를 검사하기 위한 Constraints를 나타낸다.

② 확장 관계를 가진 가변점 - 일련의 도메인 액티비티의 흐름이 애플리케이션마다 특정 조건에 따라 가변적으로 수행되는 경우, 이를 분리하여 가변치로 만들고 애플리케이션 개발 시 결정할 수 있도록 가변점에 연결시킨다. (그림 7)은 계산대에서 발생하는 비즈니스 프로세스 모델의 일부분을 보여준다. 고객이 구입한 상품을 계산대 리더에서 모두 읽은 후, 총합을 보여주는 도메인 액티비티(*Display the sale total*)가 있다. 이때, 회원제를 적용하여 회원에게는 할인혜택을 주는 제도를 채택하는 경우, 고객의 회원카드를 읽은 후, 상품 총액을 다시 계산하는 프로세스가 진행된다.



(그림 6) 일반화 관계를 가진 가변점 예



(그림 7) 확장 관계를 가진 가변점의 예

이러한 회원제 서비스를 적용하는 확장 조건에 따라 추가 발생하는 프로세스를 가변치로 표현하였으며, 이를 가변점에 연결하여 표현하고 있다.

비터 다이어그램의 표현요소들을 모두 사용하며, <표 5>에 기술한 요소들이 추가되었다. 본 프레임워크의 주요 기능들은 다음과 같다.

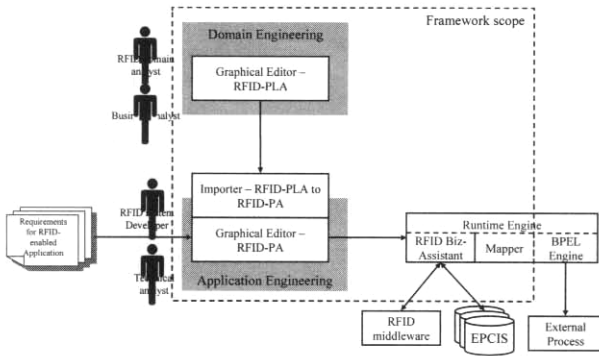
6. 지원 도구 - PLA기반 RFID 애플리케이션 구축을 위한 프레임워크 개발

본 논문에서 연구한 내용을 바탕으로 RFID 애플리케이션 개발을 지원해 줄 수 있는 프레임워크 (RFID PLA 프레임워크)를 개발한다. 프레임워크는 재사용성을 구현하기 위한 기법 중 하나로서 아키텍처의 재사용 가능한 요소들을 프레임워크 내부에서 구현해 놓고 다양한 애플리케이션에서 이를 재사용할 수 있도록 제공해 주는 기능을 가진다 [15]. 그림 8은 애플리케이션 개발 환경에서 RFID PLA 프레임워크가 지원해 줄 수 있는 주요 기능들을 중심으로 구현 범위를 보여주는 그림이다.

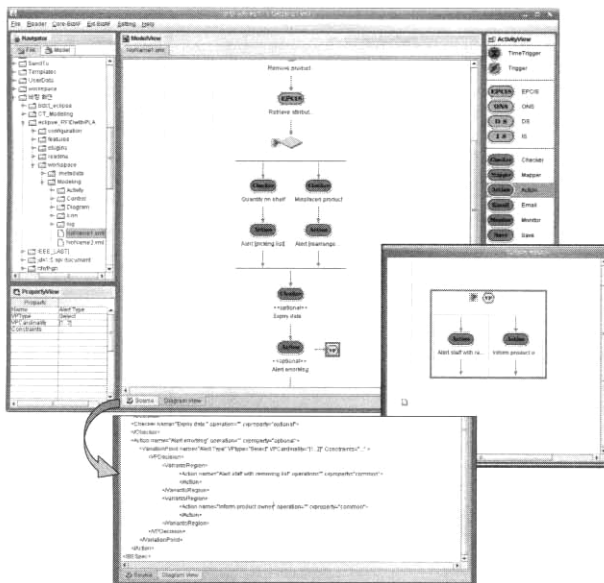
본 프레임워크에서 재사용을 위해 구현하고 있는 요소는 4장에서 설명한 RFID 액티비티들이다. 이들의 가변점은 액티비티의 속성창을 통해 파라미터로 설정할 수 있도록 한다. RFID PLA를 모델링하기 위한 표현요소들은 UML 액티

6.1 RFID PLA를 모델링 하기 위한 graphical editor

(그림 9)는 프레임워크에서 RFID PLA를 모델링하기 위한 사용자 인터페이스를 보여준다. 오른쪽 판넬에 RFID PLA를 위한 모델링 요소들인 RFID 액티비티와 UML 액티비티 다이어그램 모델링 요소들이 정의되어 있다. 이러한 모델링 요소들을 드래그 앤 드롭하여 가운데에디터 창에 갖다 놓음으로써 간단히 모델링 할 수 있다. 또한 각 액티비티의 가변점과 속성들에 대한 정의를 왼쪽 아래 판넬(PropertyView)에서 설정할 수 있다. 도메인 액티비티가 가변점을 가지는 것을 표현하기 위해 팝업창을 띄워 가변점을 가지는 속성을 클릭함으로써 자동 가변점 모델요소가 붙게 된다. 또한 가변점에 붙게 되는 하나 이상의 가변치들은 메인 흐름과 분리하여 가변치 창(variants window)에서 표현한다. 이는 PLA 모델이 복잡해지는 것을 줄이기 위한 목적을 가진다. 각 모델링 요소들은 XML 기반의 모델링 언어로 바뀌어 저장된다. 그러므로 모델링 요소들을 에디터 창에 드래그 앤 드롭하게 되면 이와 동시에 이에 해당하는 소스코드가 자동 생



(그림 8) PLA기반 RFID 애플리케이션 개발을 위한 프레임워크



(그림 9) RFID PLA 프레임워크 사용자 인터페이스

성된다. 또한 반대로 코드의 수정 시에 PLA 모델의 변경도 자동 관리된다. 소스 코드는 에디터 창의 source탭을 클릭함으로써 사용자가 직접 볼 수 있고 수정할 수도 있다. 프레임워크는 이 소스 코드를 분석하여 RFID 액티비티에 따른 RFID 데이터를 처리할 수 있게 된다.

(그림 9)에서 보여주고 있는 PLA 모델은 스마트 선반에서 RFID 리더가 상품의 태그를 읽어 오류 상품을 검사하는 프로세스를 표현한 것이다. 처음 처리되는 시간 트리거 액티비티와 이벤트 트리거 액티비티에 대한 변환 코드가 부록A에서 보여준다. 시간 트리거 액티비티는 RFID 애플리케이션마다 선택적으로 나타날 수 있는 활동이기 때문에 CV_property가 optional로 설정이 되었으며, 이벤트 트리거 액티비티에 대한 가변점들은 속성창을 통해 가변치들이 바인딩되었다. 또한 (그림 6)에 설명되었던 예가 프레임워크를 통해 모델링 되어 있는 것을 확인할 수 있다. "Alert for expired date" 도메인 액티비티에 붙어있는 가변점을 클릭하면 RFID 애플리케이션 개발 시 대체될 수 있는 가변치들이 나타난다. 그리고 이때 가변점에 대한 속성이 propertyView 판넬에서 기술되었다.

6.2 RFID PLA Importer

RFID PLA 모델로부터 특정 애플리케이션을 위한 RFID product architecture(PA)를 만들기 위해 RFID PLA를 가져올 수 있다. 이 때, 가변성이 표현되어 있는 PLA는 사용자로부터 결정(decision)-과정을 거쳐 가지치기(pruning)를 하게 된다. 사용자로부터 결정-과정은 도메인 액티비티의 CV_property 결정과 가변치에 대한 Boolean decision과 selection decision으로 이루어진다. 사용자의 결정에 의해 RFID PLA에서 제거되는 선택적 도메인 액티비티는 앞뒤 인터페이스의 일치를 제약사항에 기술된 내용을 기반으로 검사를 하게 된다. 만약 일치하는 경우, 도메인 액티비티 사이의 흐름은 자동 연결되며, 그렇지 못한 경우는 흐름이 연결되지 않는다. 이는 연결을 위해 애플리케이션 개발자가 중간 어댑터 또는 변환을 해 줘야 함을 의미한다. 사용자 결정에 의해 가변치가 Boolean decision으로 결정되는 경우, 이는 확장 흐름이 추가됨을 의미하게 되며, 선택된 확장 흐름이 메인 흐름에 자동 추가된다. 사용자 결정에 의해 가변치가 selection decision으로 결정되는 경우, 사용자는 가변점 속성의 기술 요소들을 바탕으로 가변치를 선택하게 된다. 가변점 속성의 기술 요소와 다른 결정은 프레임워크에서 검사를 하게 된다. 가변치들의 선택 후, 이들의 흐름 연결은 사용자에 의해서 이루어지게 된다.

6.3 RFID PA 모델링 및 실행

애플리케이션 개발자는 RFID PA를 모델링 하기 위한 에디터 창이 제공된다. 애플리케이션 개발자는 위에서 설명한 방법과 같이 RFID PLA를 import 하여 PL를 모델링 할 수도 있고, 직접 모델링 할 수도 있다. 가변성이 모두 제거된 PA는 RFID 데이터를 얻기 위해서 RFID 미들웨어에 이벤트 명세를 정의하고 등록할 수 있는 기능을 제공한다. (그림 8)의 RFID Biz-Assistant와 Mapper는 RFID PA 활동들의 흐름에 따라 외부의 다양한 시스템과 연동을 통해 RFID 데이터를 처리해주는 실행엔진이다. 이것은 크게 활동들에 대한 일련의 순서를 제어하는 컴포넌트와 정의된 각 활동들을 독립적으로 수행할 수 있도록 구현하는 각각의 컴포넌트로 구성되어 있다. 특히, RFID 미들웨어, EPCIS, ONS 등의 여러 시스템과 연동하는 부분을 컴포넌트로 구현하여 프레임워크에서 제공해 줌으로써 RFID 애플리케이션 개발자는 RFID 시스템의 구성 요소들과의 직접적인 통신부를 개발하지 않아도 되게 된다.

또한 RFID 미들웨어와 RFID 리더의 가용성에 독립적으로 시뮬레이터 해 볼 수 있기 위해 가상의 RFID 리더와 가상의 RFID 미들웨어에 연결이 가능하도록 하였다. 이는 프레임워크 내부에서 RFID 사용 환경을 구축하여 개발자가 가상의 RFID 이벤트, EPC 상세정보, EPCIS IP주소를 설정하여 Logical한 RFID지원 환경을 제공해 줌으로써 RFID 애플리케이션 개발 테스트를 용이하게 해 준다. 마지막으로, RFID 미들웨어로부터 올라오는 RFID 태그를 모니터링 할 수 있는 기능이 제공된다. 모니터링 유형에는 콘솔타입과 파일타입이 있다. 콘솔타입은 이벤트 스펙을 subscribe한 이후부터 실시간으로 올라오는 RFID 데이터를 확인할 수 있다. 파일타입은 로그의 형태로 시간기록과 함께 파일에 기록한다.

<표 6> RFID 액티비티 재사용 빈도수

액티비티	RFID 도메인	스마트선반 도메인	창고관리 도메인	약국통합관리 도메인	u-도서관리 도메인
RFID 이벤트 처리 액티비티	이벤트트리거	1	3	2	4
	시간트리거	0	1	0	1
Information service 참조 액티비티	EPCIS	1	3	1	3
	ONS	1	3	1	3
	EPCIS DS	0	1	1	0
비즈니스 서비스 액티비티	규칙검사	3	7	5	12
	서비스연결	2	5	3	4

7. 평 가

지금까지 많은 산업체에서 소프트웨어 프로젝트 라인을 적용했을 시, 비용을 절감하고 개발시간을 단축할 수 있었으며, 또한 품질을 향상할 수 있었다고 보고하고 있다[3, 7, 16]. 이를 통해 본 논문에서 제안하는 RFID 프로젝트 라인 아키텍처를 이용하여 특정 RFID 애플리케이션을 위한 아키텍처를 빠르고 쉽게 만들어 낼 수 있음을 예측할 수 있다.

뿐만 아니라, 본 논문에서는 제안한 프레임워크를 이용하여 다음의 두 가지 측면에서 효과가 있음을 평가해 보았다. 이는 실제 다양한 RFID 애플리케이션 - 스마트 선반 응용 시스템, 창고 관리 시스템, 약국 통합관리 시스템, u 도서관리 시스템을 개발해 본 결과를 바탕으로 한 것이다.

7.1 다양한 RFID 도메인에서의 RFID 액티비티 재사용 빈도수

본 논문에서 분석한 RFID 액티비티들이 다양한 RFID 도메인에 얼마나 공통적으로 사용되었는지를 평가하기 위하여 <표 6>과 같이 RFID 액티비티의 재사용 빈도수를 측정하였다. 이를 통해 대부분의 식별된 RFID 액티비티가 모든 도메인에서 반드시 사용되어야 하는 활동임을 확인할 수 있었다. 특히, 창고관리 도메인의 경우, 창고 입고, 창고 출고, 창고선반 각각을 위한 RFID 데이터를 처리하기 위해 이벤트 트리거가 3개가 필요하였다. 또한, 규칙검사 액티비티는 선택성을 가진 액티비티들이 가장 많이 생성되었기 때문에 사용빈도수가 높게 나타났으며, 각 기업의 비즈니스 규칙을 적용하는 요소이기 때문에 재사용 빈도수의 변동이 가장 많이 나타났다. 예를 들어, u-도서관리 도메인 중, 대출관련 RFID 이벤트 트리거에 대하여 사용자 인식불가, 대출권수 초과, 미납 연체료 등의 규칙검사가 사용되었다.

<표 7> 프로그램 코드수 비교

항목	RFID 액티비티 이용	Non RFID 액티비티
RFID 미들웨어 접근	0	249
ECSpec 작성	0	90
EPCIS 접근	0	233
ONS 접근	0	100
프로세스 처리	0	348
수신 쓰레드 처리	0	100
스펙 작성	219	0
총계	219	1120

7.2 RFID 데이터 자동 선처리를 통한 설계 및 코드 수 비교

<표 7>은 u-도서관리 시스템을 통해 실질적으로 프로그램 코드 수를 비교한 것이다. 비록 프로그램 코드 수는 개발자의 코딩 습관에 따라서 많은 차이가 나지만, RFID 액티비티의 기능에 따른 필요량을 작성한 프로그램 코드 수로 판단하여 비교해 본 것이다. 이를 통해 프레임워크 내에 구현된 RFID 액티비티들을 사용하면 RFID 미들웨어, EPCIS, ONS 등의 외부 서버와의 통신을 위해 개발자가 노력을 하지 않아도 됨을 알 수 있다. 특히, RFID 액티비티를 이용하기 위해 작성해야 하는 스펙 (219 라인)이 개발자가 직접 수동으로 적는 것이 아니라 아키텍처 모델링을 통해 자동 생성(0 라인)된다는 점을 고려한다면, 코드수의 비교는 더 커질 것이다.

8. 결론 및 향후 연구

RFID는 많은 양의 데이터와 이벤트로 구성된다. RFID는 유연한 인프라스트럭처(infrastructure)를 기반으로 신뢰성 있고 안전한 데이터를 생성해야 한다. 이 인프라스트럭처는 기존의 데이터, 애플리케이션 그리고 비즈니스 프로세스를 통합할 수 있어야 하고, 여기에 새로운 애플리케이션과 비즈니스 프로세스를 쉽게 추가할 수 있어야 한다. 본 논문에서는 RFID 애플리케이션의 효율적인 개발을 위하여 프로젝트 라인 아키텍처 기술을 접목시켜 RFID 관련 활동들을 선처리해 줄 수 있는 프레임워크를 개발하였다. 프레임워크는 RFID 데이터를 처리하기 위한 공통된 활동들을 정의하고 있으며, 애플리케이션 및 프로세스의 통합 기능을 포함하고 있다. 또한 이벤트 드리븐(event driven) 아키텍처와 애플리케이션마다 가변되는 비즈니스 룰을 가변점으로 관리할 수 있는 확장된 프로세스 모델링 기능들을 포함하고 있다. 개발된 프레임워크를 이용하여 다양한 RFID 애플리케이션을 개발해 보았으며, 그 결과 RFID 시스템의 도입 시 처리해야 하는 활동들의 개발 양을 현저히 줄일 수 있음을 확인하였다.

본 연구는 현재 RFID 태그의 속성정보와 이력정보만을 활용하는 범위에서 확장하여 실시간 변경되는 문맥정보를 처리하기 위한 활동들을 정의하고 있다. 문맥정보 중, 첫 번째 활용정보는 RTLS(real time location system)를 이용한 위치정보와 가장 간단한 몇 가지 센서 (온도센서) 정보를 처리하기 위해 필요한 활동들을 분석 중이다.

참 고 문 헌

[1] LARAN RFID, "A Basic Introduction to RFID technology and Its use in the supplychain," URL: http://www.primtronix.com/uploadedFiles/Laran_WhitePaper_RFID.pdf, January 2004.

[2] One Network Enterprises, "RFID Business Process Enablement, The most flexible approach to RFID-enabled supply chain business processes," URL: <http://www.onenetwork.com/solutions/rfid/>

[3] Clements, P., and Northrop, L., *Software Product Lines: Practices and Patterns*, Addison Wesley, 2001.

[4] H. Gomma, *Designing Software Product Lines with UML, From Use Cases to Pattern-Based Software Architectures*, Addison-Wesley, 2004.

[5] Moon, M.K., Yeom, K.H, and Chae, H.S., "An Approach to Developing Domain Requirements as a Core Asset Based on Commonality and Variability in a Product Line," *IEEE Transactions on Software Engineering*, vol.31, no.7, pp.551 - 569, Jul. 2005.

[6] Moon, M.K., Chae, H.S., and Yeom, K.H, "A Metamodel Approach to Architecture Variability in a Product Line," *Proceedings of 9th International Conference ICSR 2006, Lecture Notes in Computer Science Vol.4039*, pp.115-126, Jun 2006.

[7] 문미경, 채홍석, 엄근혁, "도메인 핵심자산의 가변성 분석을 위한 2차원적 접근방법," *정보과학회논문지: 소프트웨어 및 응용*, 제33권 제6호, pp.550-565, 6월 2006년

[8] EPCglobal Inc., URL: <http://www.epcglobalinc.org>

[9] EPCglobal, "EPC Information Services (EPCIS) Version 1.0 Specification," *EPCglobal Working Draft*, June 2005.

[10] M. Mealling, "EPCglobal Object Name Service (ONS) 1.0," *EPCglobal Working Draft*, December 2004.

[11] Cover, R. "Standards for Business Modeling, collaboration, and Choreography", (last modified October 15, 2004) URL: [Cover Pages http://coverpages.xml.ocverpages.org/bpm.html](http://coverpages.xml.ocverpages.org/bpm.html)

[12] K. Traub, S. Bent, T. Osinski, S. N. Peretz, S. Rehling, S. Rosenthal, B. Tracey, "The Application Level Events (ALE) Specification, Version 1.0," *EPCglobal Proposed Standard*, February 2005.

[13] Web Services Description Language (WSDL) 1.1, <http://www.w3.org/TR/ws>, March 2001.

[14] 부산대학교 차세대분류 IT기술연구사업단, 제1 총괄과제, URL: <http://lit.pusan.ac.kr>

[15] Schmidt D.C., Buschmann F., "Pattern, Frameworks, and Middleware: Their Synergistic Relationships," *Proceedings of the 25th International Conference on Software Engineering (ICSE'03)*, 2003.

[16] K. Pohl, G. Böckle, F. van der Linden, *Software Product line Engineering - Foundations, Principles, and Techniques*, Springer, Berlin, Heidelberg, New York, 2005.

문 미 경



e-mail : mkmoon@pusan.ac.kr

1990년 2월 이화여자대학교 전자계산학과 (학사)

1992년 2월 이화여자대학교 전자계산학과 (이학석사)

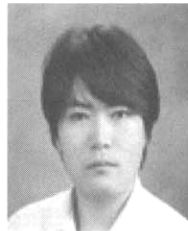
2005년 2월 부산대학교 컴퓨터공학과 (공학박사)

2005년 3월~2005년 8월 부산대학교 차세대분류IT기술사업단 박사후 연구원

2005년 9월~2006년 8월 부산대학교 컴퓨터 및 정보통신 연구소 기금교수

2006년 9월~현재 부산대학교 정보컴퓨터공학부 연구교수
 관심분야: 소프트웨어 프로덕트 라인 공학, 적응형 소프트웨어 개발, RFID 미들웨어 개발 및 RFID 기반 애플리케이션 개발 등

김 한 준



e-mail : khjplus@korea.com

2007년 부산대학교 정보컴퓨터공학과 (학사)

2007년~현재 부산대학교 대학원 컴퓨터공학과(석사과정)

관심분야: SOA, Software Product Line, RFID 등

엄 근 혁



e-mail : yeom@pusan.ac.kr

1985년 2월 서울대학교 계산통계학과 (학사)

1992년 8월 Univ. of Florida 컴퓨터공학과 (석사)

1995년 8월 Univ. of Florida 컴퓨터공학과 (박사)

1985년 1월~1988년 2월 금성반도체 컴퓨터연구실 연구원

1988년 3월~1990년 6월 금성사 정보기술연구소 주임연구원

1995년 9월~1996년 8월 삼성SDS 정보기술연구소 책임연구원

1996년 8월~현재 부산대학교 컴퓨터공학과 부교수

관심분야: 소프트웨어 재사용, 프로덕트라인 공학, 소프트웨어 아키텍처, 컴포넌트 기반 소프트웨어 개발, 적응형 소프트웨어 개발, RFID기반 미들웨어 등