

RFID 도메인에서 소프트웨어 프로덕트 라인

구축을 위한 도구 개발

김한준^o 문미경 염근혁
부산대학교 컴퓨터공학과
{khjplus^o, mkmoon, yeom}@pusan.ac.kr

A Tool Development for Software Product Line in RFID Domain

Hanjun Kim^o Mikyeong Moon Keunhyuk Yeom
Dept. of Computer Eng., Pusan National Univ.

소프트웨어 프로덕트 라인 공학은 소프트웨어의 재사용을 향상시키기 위해 소프트웨어 프로덕트의 개발 활동을 미리 계획하고 체계적으로 개발 프로세스의 진행이 가능하도록 지원하기 위한 방법이다. 소프트웨어 프로덕트 라인 공학은 소프트웨어 플랫폼과 매스 커스터마이제이션(Mass Customisation)에 사용되는 소프트웨어 어플리케이션을 개발하는 패러다임이다[1]. 소프트웨어 프로덕트 라인 공학은 그림 2에서와 같이 크게 도메인 엔지니어링 단계와 어플리케이션 엔지니어링의 두 부분으로 구성된다[2]. 먼저 도메인 엔지니어링 단계는 해당 도메인에 대한 공통성(Commonality)을 찾아내고, 또한 가변성(Variability)을 찾아내는 단계이다. 어플리케이션 엔지니어링 단계는 앞서 진행된 도메인 엔지니어링 단계의 산출물과 새로운 요구사항에 따라 소프트웨어를 개발하는 단계이다. 도메인 엔지니어링 단계의 공통성은 모든 어플리케이션 엔지니어링 단계에서 재사용되고, 가변성은 어플리케이션 엔지니어링의 요구 사항에 따라 사용여부가 결정된다. RFID 아키텍처는 태그를 인식하기 위한 리더, 필터링 및 이벤트 처리를 하는 RFID 미들웨어, EPC(Electronic Product Code)에 관한 정보를 보관하는 EPCIS(EPC Information Service)[3]와 EPCIS에 관한 정보를 보관하는 ONS(Object Name Service)[4], EPCIS DS(EPCIS Discovery Service) 등의 시스템으로 구성된다. RFID 이벤트는 RFID 미들웨어로부터 전달되는 이벤트로서 Logical Reader Name, Tag Value, Direction, Time 등의 정보를 포함하게 되는데, RFID 어플리케이션에서는 Reference Data와 비즈니스 로직을 사용하여 고수준의 이벤트를 생성하게 된다[5]. RFID 기술은 반복적이고 복잡한 과정을 거치게 되어, RFID 어플리케이션 엔지니어에게 많은 부담을 주고 유지보수에 어려움을 준다[6]. 이 논문에서는 RFID 기술과 소프트웨어 프로덕트 라인 공학을 접목시킨 RFID 도메인에서 소프트웨어 프로덕트 라인 구축을 위한 도구를 제시한다. 소프트웨어 프로덕트 라인 공학을 이용한 RFID 소프트웨어의 개발은 반복적으로 발생하는 RFID 이벤트에 대한 처리를 공통요소의 재사용으로 효율적으로 진행시킨다. 또한 도메인에 따라 가변적으로 발생하는 요구사항에 빠르게 대처하여 생산성을 향상시키는 효과를 가져다준다.

RFID도메인에서 소프트웨어 프로덕트라인 적용하기 위해서는 먼저 공통적인 요소와 가변적인 요소를 분석하는 과정이 필요하다. 공통적인 요소로 RFID 액티비티는 크게 세 가지의 요소로 구분이 된다. 먼저 RFID 미들웨어로 부터 받는 RFID 데이터에 대한 명세 정보를 처리하는 액티비티인 TimeTrigger와 Trigger가 존재한다. 이 액티비티는 RFID 데이터의 정보로서 이벤트의 주기 정보나 보고서의 사양 목록 등의 정보를 처리한다. 두 번째로 RFID 데이터의 참조 정보를 조회하기 위한 EPCIS, ONS, EPCIS DS 액티비티가 존재한다. 그리고 마지막으로 비즈니스 규칙을 정하고 규칙에 따라 서비스를 수행하는 Checker, Mapper, Action 액티비티가 존재한다. 그리고 가변적인 요소로 하나의 액티비티에서 발생하는 가변성과 하나 이상의 액티비티에서 발생하는 가변성이 존재한다. 먼저 하나의 액티비티에서 발생하는 가변성의 유형이다. RFID 어플리케이션 개발 시 반드시 구현 되어야 하는 부분과 상황에 따라 구현되어야 하는 부분의 가변성을 나타낸다. 이를 CV_property로 정의하고, 각 도메인 액티비티는 common 또는 optional의 속성을 가진다. CV_property의 속성값을 common으로 가진 액티비티는 어플리케이션 엔지

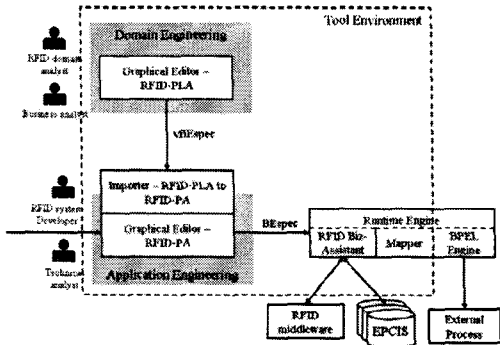


그림 1 RFID 소프트웨어 프로젝트 라인 도구 환경

본된다.

RFID 소프트웨어 프로젝트 라인 도구는 그림 1에서와 같이 도메인 엔지니어링과 어플리케이션 엔지니어링 과정을 수행하는 그래픽 에디터와 Importer, 그리고 RFID Runtime Engine에서의 실행의 두 가지 부분으로 구성된다. 그래픽 에디터에서 도메인 엔지니어는 가변성이 포함이 된 비즈니스 이벤트 명세 (vBESpec : variable Business Event specification)를 생성한다. vBESpec의 경우, 분석에서의 공통적인 요소와 가변적인 요소를 모두 포함하게 된다. 그리고 어플리케이션 엔지니어는 생성된 vBESpec에서 가변성을 제거한 비즈니스 이벤트 명세(BESpec : Business Event specification)를 생성한다. 도메인 엔지니어에 의해 생성된 vBESpec내에서 CV_property와 VariationPoint를 요구사항에 따라 Decision&Pruning을 거쳐 가변적인 요소가 포함되지 않은 BESpec를 생성하게 된다. 생성된 BESpec은 RFID Runtime Engine에서 비즈니스 로직으로 사용되어진다. 그림 1에서와 같이 Runtime Engine의 Biz-Assistant와 Mapper는 외부의 다양한 시스템과의 연동을 통해 RFID 데이터 처리하게 된다.

RFID 기술을 물류, 교통, 금융 등 많은 분야에서 이용된다. 이러한 RFID 기술은 급변하는 비즈니스에 맞추어 유연하게 변화가 가능해야하고 또한 빠르게 RFID 소프트웨어를 생산 할 수 있어야 한다. 본 논문에서는 RFID 기술 중 비즈니스 로직을 생성하는데 있어서 소프트웨어 프로젝트 라인 기법을 적용하여 소프트웨어 재사용성을 높이는 도구 환경을 제시하였다. 도구 환경에서는 RFID 이벤트를 처리하는데 있어서 요구되는 공통된 활동을 정의한다. 또한 RFID 이벤트에서 가변적으로 발생하는 비즈니스 로직에 대해 가변점으로 관리하는 확장된 모델링 기능을 제공한다. 이는 공통 요소를 관리함으로써 재사용성을 높이며, 가변 요소를 관리함으로써 빠른 시간 내에 다양하고 유동적인 RFID 어플리케이션 개발을 가능하게 한다. 향후 RFID 기술이 발전함에 따라 더 복잡한 가변요소를 가지거나 추가되는 액티비티의 존재가 발생함에 따라 도구 환경의 확장이 필요할 것으로 예측된다.

참고문헌

- [1] Pohl, Klaus, Böckle, Günter, Linden, Frank J. van der, *Software Product Line Engineering*, Springer, 2005.
- [2] Clements, P./ and Northrop, L., *Software Product lines: Practices and Patterns*, Addison Wesley, 2001.
- [3] EPCglobal, "EPC Information Services (EPCIS) Version 1.0 Specification", EPCglobal Working Draft, June 2005.
- [4] M. Mealling, "EPCglobal Object Name Services (ONS) 1.0", EPCglobal Working Draft, December 2004.
- [5] M. Moon, Y. Kim, and K.Yeom, "Contextual Events Framework in RFID System", In the Proceeding of the 3rd ITNG, pp.586-587, 2006.
- [6] M. Moon, K. Yeom, "Product Line Architecture for RFID-enabled Applications", Proceedings of the 10th International Conference on Business Information Systems(BIS), LNCS, Vol.4439, pp.638-651, 2007.