RFID 미들웨어를 위한 EMS 컴포넌트의 분석 및 설계*

안성우° 박재관 석수욱 홍봉희
부산대학교 컴퓨터공학과
{swan°, jkpack, seok1968, bhhong}@pusan.ac.kr

The Analysis and Design of EMS Component for RFID Middleware

Sungwoo Ahn° Jaekwan Park Suwook Seok Bonghee Hong
Department of Computer Science and Engineering, Pusan National University

요 약

많은 글로벌 기업들이 비용절감 및 효율성 증대를 위하여 RFID 시스템을 도입하거나 도입을 적극 검토 중에 있다. RFID 시스템으로부터 획득된 수많은 정보들이 기업의 업무효율을 증대시킬 수 있도록 하기 위해서는 상위 응용 서비스에서 RFID 데이터를 효율적으로 사용할 수 있도록 리더와 응용 서비스 간의 중계 역할을 할 수 있는 RFID 미들웨어의 역할이 매우 중요하다. 이러한 요구사항에 맞춰 최근 EPCglobal에서는 RFID 미들웨어인 ALE(Application Level Event)를 제시하고 있다. ALE는 RFID 리더에 의해 읽혀진 EPC 이벤트 데이터를 실시간으로 수집하여 원본 데이터의 오류를 보정한 후에 사용자와 응용 서비스의 요구에 따라 특정 이벤트 데이터를 필터링하여 보고하는 역할을 한다.

본 논문에서는 ALE의 실시간 EPC 이벤트 데이터 처리를 위한 EMS(Event Management System) 컴포넌트를 제시하며 EMS에 대한 요구사항을 분석하고 설계하였다. EMS 컴포넌트는 RFID 리더를 통해서 끊임없이 들어오는 스트림 형태의 EPC 이벤트 데이터를 블록킹 없이 수집하는 역할을 한다. 또한, RFID 리더에서 수집한 데이터의 보정 및 필요 데이터 추출을 위한 다양한 필터링 기능을 제공함으로써 수집된 데이터의 정확성을 높이며 신속한 데이터 제공을 가능하게 한다.

1. 서 론

유비쿼터스 컴퓨팅을 선도하는 차세대 핵심요소로 RFID 기술이 주목을 받고 있다. 각종 태그 정보를 SCM, ERP 등의 다양한 엔터프라이즈 애플리케이션에서 활용하기 위해서는 정확한 장소에 실시간으로 인증된 정보를 전달할 수 있는 RFID 미들웨어가 필요하다. 최근 RFID 기반의 미들웨어 제품 및 솔루션은 EPC 코드 등과 같은 간단한 형식의 데이터를 처리할 수 있으나 대용량의 데이터를 실시간으로 필터링 및 수집하기 위한 고려가 미진한 상태이다[1].

EPCglobal은 RFID 미들웨어의 표준화를 주도하고 있으며, 소프트웨어 관련 산업체에서는 이를 참조 모델로 솔루션을 개발하고 있다. EPCglobal은 RFID 미들웨어와 관련하여 2002년에 구현 스펙 중심의 Savant Version 0.1[2] 제안하였으며, 2003년에 Savant Version 1.0[3]을 제안하였으나, 2004년에는 인터페이스 중심의 ALE(Application Level Event)를 제안하고 있다. ALE는 EPC 정보의 필터링 및 수집 역할을 담당하는 필터링 미들웨어이다.

ALE는 RFID 리더에서 끊임없이 들어오는 스트림 형태의 EPC 이벤트 데이터를 블록킹 없이 수집하여 용도에 맞게 분류하고 해당 데이터를 원하는 곳에 전달하는 역할을 수행해야 한다. 따라서 ALE는 성능 최적화와 EPC 데이터의 실시간 처리 효율성을 위한 다양한 요구사항을 만족해야 한다.

ALE는 TMS(Task Management System), RIED(Realtime In Memory Event Database), EMS(Event Management System)의 모듈로 구성될 수 있다[2]. TMS는 애플리케이션으로부터 EMS가 처리해야할 태스크를 관리하는 모듈이며, RIED는 EPC 정보를 EPC-IS(EPC Information Service)로 전달하기 전에 보관하는 메모리기반 DB이다. 특히, EMS는 Smoothing, Coordination, Forwarding 필터(Filter)와 Memory, Database Event, Network Event 로거(Logger)를 지원함으로써 ALE의 핵심 역할을 담당한다.

본 논문에서는 ALE의 EMS와 관련된 구현 및 인터페이스 스펙을 만족하면서 동적 운영을 지원할 수 있는 EMS 컴포넌트를 설계하였다.

본 논문의 구성은 다음과 같다. 2장에서는 EPCglobal의 미들웨어 스펙에 대하여 기술한다. 3장에서는 EMS의 분석 및 설계 내용을 기술하며, 4장에서 EMS의 핵심 구현 기술 및 방향을 제시한다. 마지막으로 5장에서 결론을 맺는다.

2. EPCglobal의 미들웨어 표준 스펙

RFID 미들웨어의 표준화는 EPCglobal의 SAG(Software Action Group)에서 추진하고 있으며, 현재 그림 1의 EPC Network Architecture를 제시하고 있다. 이 구조는 이전에 제시된 Savant Specification[2],[3]과 달리 각 부분의 역할과 외부 인터페이스만을 정의하고 있다.

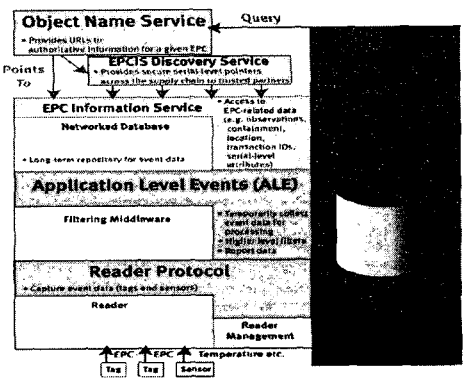

그림 1 EPC Network Architecture

* 이 논문은 교육인적자원부 지방연구중심대학육성사업 "차세대물류IT기술연구사업단"의 지원에 의하여 연구되었음

Reader Protocol[4]은 다양한 센서나 RFID 태그를 읽기 위한 리더와 각종 태그 간의 Air Interface Protocol을 정의하고 있다. ALE는 읽혀진 태그 정보들을 필터링 및 수집하는 미들웨어로써, 데이터의 정제가 가장 중요한 작업이다. EPC-IS는 EPC와 관련된 정보를 저장하기위한 저장소를 제공하며, 이를 엑세스하기 위한 서비스를 제공한다. 수많은 EPC-IS는 각각 Object Naming Service(ONS), EPC Discovery Sevice(EPC-DS)에 메타 정보를 등록하여, 데이터를 P2P형태로 공유하는 분산된 네트워크 데이터베이스 환경을 구축한다. ONS는 EPC 정보에 대한 EPC-IS URL을 제공하는 Naming Service를 제공한다. EPC-DS는 Supply Chain에서 특정 EPC 정보가 거쳐간 모든 EPC-IS에 대한 엑세스 정보를 제공한다.

특히, 필터링 미들웨어인 ALE는 리더로부터 읽혀지는 수많은 스트림 데이터를 실시간으로 처리할 수 있어야 하며, 다양한 리더로부터 수신되는 여러 가지 다른 프로토콜들을 지원해야 한다. 또한 이벤트 정보들을 보정 하거나 특정 이벤트 정보만을 걸러낼 수 있는 다양한 필터(Filter)를 지원해야 하며 EPC-IS나 외부 응용 서비스로 데이터를 전달할 수 있는 각종 로거(Logger)를 제공해야 한다[2],[4].

3. ALE의 EMS 컴포넌트 분석 및 설계

3.1 요구사항 분석

2장에서 살펴본 바와 같이 ALE는 RFID 리더에서 끊임없이 들어오는 스트림 형태의 EPC 이벤트 데이터를 블록킹 없이 수집하여 용도에 맞게 분류하고 해당 데이터를 원하는 곳에 전달하는 역할을 수행해야 한다. ALE 시스템의 최적화와 EPC 데이터의 실시간 처리 효율성을 위해서 아래와 같은 요구사항이 만족되어야 한다.

- 각 리더에서 초당 100개 이상 보내는 이벤트 데이터 스트림에 대해서 실시간으로 처리가 가능해야 한다.
- 리더에서 보내는 이벤트 데이터들을 보정하거나 특정 이벤트 데이터만을 걸러낼 수 있는 필터를 지원해야 한다.
- 데이터베이스나 메모리 데이터 구조 등에 처리 결과를 저장할 수 있는 이벤트 로거를 지원해야 한다.
- 웹서비스와 연동을 하기 위한 네트워크 이벤트 로거를 지원해야 한다.
- 이벤트들을 끊임없이 처리하기 위해 각 프로세싱 유닛(Processing Unit)들은 각각의 스레드(Thread)에서 동작해야 하며 유닛과 유닛 사이에는 버퍼링 기능이 있어야 한다.
- 프로세싱 유닛을 생성하고 특정 설정을 통해 DAGs(Directed Acyclic Graphs)를 생성할 수 있어야 한다.
- DAGs에 등록된 리스너(Listener)를 ALE 시스템의 재구동 없이 동적으로 추가가 가능하게 함으로써 시스템이 멈추는 경우로 인해 실시간 데이터 처리가 지연되어서는 안된다.

본 논문에서는 이러한 요구사항을 충족시키기 위하여 EPC 이벤트 데이터의 수집, 분류 및 수집정보 보고의 역할을 할 수 있는 EMS 컴포넌트를 설계하였다. 리더에서 읽혀진 EPC 이벤트 데이터를 처리하기 위한 EMS 컴포넌트는 그림 2와 같이 나타낼 수 있다.

리더에서 읽은 EPC 이벤트 데이터를 처리하기 위해서 EMS에는 큐(Queue), 필터(Filter), 로거(Logger)로 구성된 프로세싱 유닛들이 있으며 상위 응용 서비스에서 요구하는 이벤트 데이터를 얻어내기 위해서 프로세싱 유닛들이 조합되어 하나의 리스너로 등록이 되어 동작한다. EPC 이벤트 데이터의 수집 및 제공을 위한 EMS 컴포넌트의 역할 및 동작 방법에 대해서 3.2절에서 설명하고 있다.

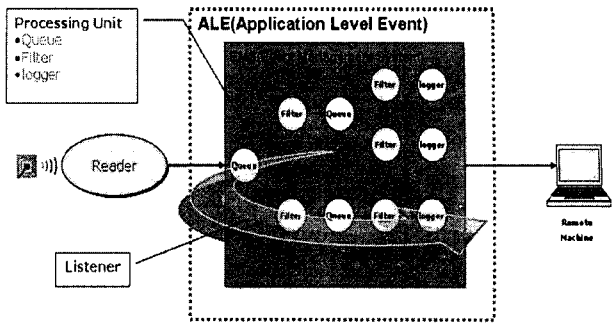

그림 2 ALE의 EPC 이벤트 데이터 처리 구성도

3.2 EMS 컴포넌트의 구조

EMS는 리더 인터페이스를 통해 들어오는 EPC 이벤트 데이터의 필터링, 로깅 기능을 수행한다. 그림 3은 EMS 컴포넌트의 구조를 나타내고 있다.

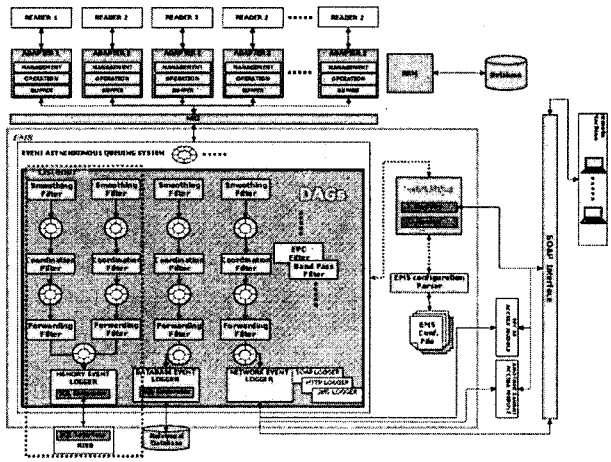

그림 3 EMS 컴포넌트의 구조도

EMS는 필터와 로거들을 이용하여 DAGs 형태의 구조를 만들어 이벤트 데이터 들을 처리하는데 하나의 DAGs를 이루는 구조를 리스너라고 부른다.

각 필터와 로거는 프로세싱 유닛으로 일반화되며 프로세싱 유닛들은 각기 다른 스레드로 병렬적으로 동작하게 된다. 병렬로 동작하는 각 프로세싱 유닛들 사이에는 큐가 존재해서 데이터의 블록킹이 발생하지 않도록 버퍼링 기능을 담당한다.

프로세싱 유닛 중 필터는 잘못 읽혀진 EPC 데이터의 Smoothing(에러처리) 및 Cleaning과 Coordination(중복처리), Forwarding(전달) 등을 수행한다. Forwarding 필터로는 EPC 코드별로 필터링을 수행하는 EPC 필터와 리더별로 필터링을 해주는 Band 필터, 그리고 RFID 태그가 설치된 아이템이 리더의 범위 내에 들어오거나 나가는 이벤트들을 필터링 해 주는 Delta 필터 등이 기본적으로 제공된다.

프로세싱 유닛 간의 연결은 큐, 필터, 로거가 레고형식(Lego-like fashion)으로 구성될 수 있으며 그림 4와 같이 맨 앞에 큐가 위치하며 맨 마지막에 로거가 위치하게 된다. 이와 같이 프로세싱 유닛을 구성함으로써 상위 응용 서비스의 요구에 따라 새로운 필터를 등록하여 바로 적용할 수가 있다.

각 리스너의 끝에는 주로 로거들이 존재한다. 로거는 해당 EPC 데이터를 원하는 타겟 모듈로 저장하는 역할을 수행한다. 기본적으로 제공되는 로거로는 네트워크상의 원격 컴퓨터에 쓰기가 가능한 NetworkEventLogger와 일반 RDB에 쓰기가 가능한 DBEventLogger 그리고 ALE 내의 메모리 구조체에 쓰기가 가능한 MemoryEventLogger가 있다. 로거 역시 필요에 따라 새로운 로거를 개발하여 적용 가능하다.

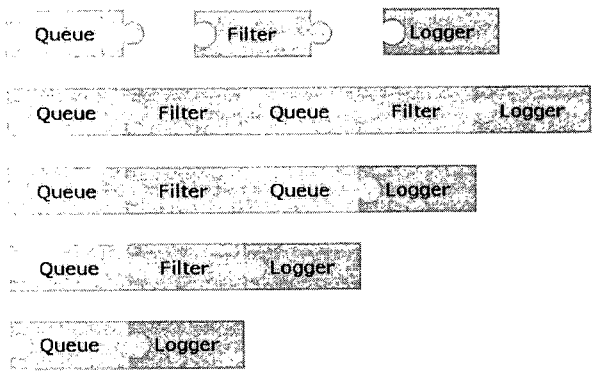

그림 4 프로세싱 유닛간의 연결 방법

상위 응용 서비스는 해당 모듈로 원하는 EPC 데이터를 가져올 수 있도록 EMS에 리스너를 등록할 수 있는데 여기서 리스너를 등록, 해지하고 모니터링 하는 기능을 ControlStation이 담당하게 된다. 본 논문에서 제시하는 ControlStation은 EPCglobal의 Savant Version 0.1[2]에서 리스너의 등록과 해지시에 재구동을 필요로 하는 것과는 달리 재구동할 필요 없이 동적으로 가능하도록 하는 특징을 가진다.

3.3 EMS 컴포넌트의 노출 인터페이스 명세

RFID 리더는 실시간으로 읽혀진 EPC 이벤트 데이터를 EMS 컴포넌트의 큐를 통하여 저장을 요청한다. 또한, 외부 응용 서비스에서는 EMS의 가동/중지를 요청할 수 있으며 필요한 EPC 이벤트 데이터를 획득하기 위해 리스너를 등록하여 큐, 필터, 로거가 DAGs를 구성하여 동작할 수 있도록 한다. 이러한 기능을 제공하기 위하여 본 논문에서 제시하고 있는 EMS 컴포넌트는 표 1과 같은 외부노출 인터페이스를 제시하고 있다.

표 1 EMS의 외부노출 인터페이스

public static String startup()
기능 | EMS 시스템을 구동
public static String shutdown()
기능 | EMS 시스템을 중지
public static Vector listPublicListeners()
기능 | public listener들의 name list를 반환
public static String addPublicListener(String listenerName, String queueName, String loggerClass, String loggerArgs, String filterClass, String filterArgs)
기능 | EMS queue에 listener를 추가
public static String removePublicListener(String listenerName)
기능 | queue에서 listener를 제거
void logEpcEvent(String eventType, long timeStamp, String epc, String readerEpc)
기능 | EMS queue에 EPC data를 저장
void logNonEpcEvent(String eventType, long timeStamp, String readingType, String value, String readerEpc)
기능 | EMS queue에 non EPC data를 저장

4. EMS 컴포넌트의 핵심 구현기술 및 구현방향

본 논문에서 제시하고 있는 EMS는 RFID 리더에 의해 수집된 EPC 이벤트 데이터를 용도에 맞게 빠르게 분류하고 해당 데이터를 원하는 곳에 적절히 배치하는 역할을 수행해야 한다. 또한, RFID 리더는 적어도 초당 수백 개의 EPC 이벤트 데이터들을 ALE내의 EMS로 보내기 때문에 이러한 환경에서 이벤트 스트림을 적절하게 버퍼링하고 각 프로세싱 유닛의 부하 상태를 분석하여 블록킹 없이 EMS의 역할을 수행하도록 해야 한다. 이를 위해 프로세싱 유닛들이 스레드로 동작할 때 각 스레드의 우선순위 메소드를 통해 우선순위를 조정하여 부하를 분산시키는 최적화된 스케줄링 기술을 구현하는 것이 필요하다.

RFID 기반 미들웨어가 정확한 데이터를 전달한다는 것은 RFID 장치(리더)로부터 수집된 정보 중 응용 서비스가 관심 있는 데이터만을 필터링하여 전달하는 것을 의미한다. 데이터 필터링 기능은 데이터의 형식 및 응용 환경에 따라 요구되는 기능이 달라진다. 단순한 EPC 코드를 활용하는 응용에서 필요한 정보를 얻는 방법과 훨씬 더 복잡한 구조를 갖는 데이터를 이용하는 응용에서 정보를 필터링하는 방법은 다르다. 또한 처리되어야 할 데이터의 양, 동시에 처리되는 필터링 조건의 수 등을 고려하여 데이터 처리 방법을 채택하여야만 데이터의 손실 없이 실시간 처리가 가능하다. 이러한 실시간 이벤트 처리를 가능하게 하기 위해서 다수의 리더로부터 수신되는 데이터를 임시 저장하는 큐잉 기술, 오류 데이터를 처리하거나 특정 대상으로 전달하는 Smoothing 및 Forwarding 필터링 기술, DB로 저장하는 로깅 기술 등에 대한 효율적인 알고리즘을 개발하여 구현하는 것이 필요하다.

5. 결론 및 향후연구

본 연구에서는 RFID 리더에서 읽은 수많은 이벤트 데이터들을 필터링을 통해 원본 데이터의 오류를 보정하고 상위 응용 서비스의 요구에 따라 특정한 이벤트 데이터들만을 특정 목표로 포워딩 해주기 위해 RFID 미들웨어인 ALE의 EMS 컴포넌트를 설계하였다. 향후 연구로 EPCglobal의 ALE에서 노출하고 있는 외부 인터페이스와의 연동을 통하여 구현 및 테스트를 하는 것이 필요하며 본 논문에서 제시하는 다양한 필터링 방법에 대한 효율적인 알고리즘을 개발하여 컴포넌트 구현에 적용하는 것이 필요하다.

6. 참고 문헌

[1] 원종호, 이미영, 김명준, "유비쿼터스 컴퓨팅 환경을 위한 RFID 기반 센서 데이터 처리 미들웨어 기술 동향", 전자통신동향분석, 제 19권 제 5호 pp 21-30, 2004.

[2] Oat Systems & MIT Auto-ID Center,"The Savant Version 0.1(Alpha)", Auto-ID Center, 2002.

[3] Sean Clark, Ken Traub, Dipan Anarkat, Ted Osinski, "Auto-ID Savant Specification 1.0", Auto-ID Center, 2003.

[4] John Price, Ed Jones, Howard Kapustein, Ravi Pappu, Darrell Pinson, Richard Swan, Ken Traub, "Auto-ID Reader Protocol 1.0", Auto-ID Center, 2003.