

동적 이벤트 처리 기반의 RFID 시스템을 위한 애스펙트 모듈 설계 (Aspect module design for dynamic event based RFID system)

박세승*, 황희정**, 최진탁***

(Seiseung Park*, HeeJoung Hwang**, Jintak Choi***)

요 약

현재 RFID 미들웨어의 표준은 RFID 표준을 주도하고 있는 EPCglobal의 ALE (Application Level Event) 이며 시스템 인프라스트럭처에 독립적으로 운영 될 수 있도록 최소한의 인터페이스와 확장점 만을 제시하고 있다. 그러나 확장점을 통한 기능의 확장에 대한 부분은 표준화 되어 있지 않으며 운영중에 있는 시스템에 적용하기 위해서는 새로운 코드를 추가 하거나 변경해야 하는 문제점이 있다. 본 논문에서는 이러한 문제점 해결을 위해 AOP(Aspect Oriented Programming) 기법을 도입한 동적 ALE 미들웨어 프레임워크를 제시하고 이를 구현하기 위한 애스펙트를 설계했다. 설계된 미들웨어 프레임워크 및 애스펙트는 동적인 기능의 확장을 제공해 기존 ALE 기반 미들웨어의 기능확장에 따르는 유지보수의 어려움을 해결하고 표준화된 방법으로 새로운 기능을 추가할 수 있다.

Abstract

The current RFID middleware standard is ALE (Application Level Event) of EPCglobal that takes the lead in RFID standardization. ALE suggests the minimal interface and extension points so that it can be operated independently from system infrastructure, and actual implementation is left to vendors. But there is no specific mention with regard to functional extension through extension points, and It is very hard to extend or modify the function on running system. To solve this problem, we suggest AOP (Aspect Oriented Programming)-based ALE middleware framework for developing RFID middleware. In addition, we designed aspect for RFID middleware for implementing and extending functions according to ALE rules and for the convenience of maintenance.

* 조선대학교 전자공학과 교수, ** 가천의과학대학교 정보기술학과 조교수, *** 인천대학교 컴퓨터공학과 교수

Key words : RFID, 미들웨어, AOP

© THE KOREAN SOCIETY FOR INDUSTRIAL AND APPLIED MATHEMATICS, 2006

I. 서 론

최근 컴퓨팅 기술의 발달은 과거 사용하는 컴퓨터에서 생활 속의 컴퓨팅 환경으로 그 개념이 변화되고 있다. 유비쿼터스 컴퓨팅은 기존 사무공간에서 활용되던 컴퓨터 및 네트워크 환경을 일상생활의 모든 공간에서 사용 할 수 있도록 하는 방식으로의 변화를 의미한다[1]. RFID 기술은 이러한 유비쿼터스 컴퓨팅 실현의 핵심기술로 주목받고 있으며 우리나라에서는 2004년 2월 차세대 국가정보사업의 핵심 전략으로 IT839 정책을 수립한 이후 2006년에는 u-IT839 정책으로 확대 추진하고 있다. RFID 관련 서비스 및 기술은 8대 신규 서비스 및 9대 신 성장 동력에 포함 되어 있을 정도로 그 중요성에 부각되고 있으며 특히 RFID 태그로부터 수집된 데이터를 기업정보 시스템에 전달하는 RFID 미들웨어 기술은 서비스 응용 분야가 복잡해짐에 따라 그 중요성이 증대되고 있다. Frost & Sullivan 의 World RFID Middleware Markets 보고서에 따르면, RFID 미들웨어 시장은 2011년 2억 2천만 달러에 이를 것으로 전망하고 있다[2]. 특히 변화하는 환경에 능동적으로 대응할 수 있도록 동적 기능 확장을 지원하는 RFID 미들웨어에 대한 요구가 점점 높아지고 있다[2,10].

현재 RFID 미들웨어의 표준은 RFID 표준을 주도하고 있는 EPCglobal의 ALE (Application Level Event) 이다. ALE는 시스템 구조에 독립적으로 운영 될 수 있도록 최소한의 인터페이스와 확장점 만을 제시하고 있으며 실제적인 구현부분은 벤더영역으로 남겨 두고 있다[3]. 이러한 ALE 의 구조는 다양한 기업환경에 따라 확장 가능한 개방형 구조로 볼 수 있지만 많은 부분이 벤더 자체의 규약으로 개발될 가능성이 크기 때문에 미들웨어 규모가 커졌을 경우 표준의 의미가 희석되는 문제점이 있으며 특히 확장점 을 통한 기능의 확장에 대해서는 별다른 언급이 없기 때문에 이에 대한 표준화 방안이 요구되고 있다. 현재 IBM, Sun, Oracle, SAP 등 많은 회사에서 ALE 기반의 RFID 미들웨어를 출시하였으나 이들 제품은 각 벤더에서 제공하는 소프트웨어플랫폼 에 종속적이고 기능의 확장 역시 벤더별 개발 기준을 따라야 하는 문제점이 있다[4,10]. 이에 대한 대안으로 오픈소스 프로젝트 인 Singularity[7]가 있으나 인터페이스 에 의한 설계를 따르지 않고 직접적인 클래스를 구현한 단순 미들웨어 구현으로 표준화된 RFID 미들웨어의 개발 지원을 위한 플랫폼으로 사용하기에는 한계가 있다.

기존 RFID 미들웨어 구현상의 문제점은 컴포넌트 혹은 객체지향 기법으로 시스템을 구축했다 하더라도 프로그램 코드 내 횡 단적으로 존재하는 횡단 관심사(cross-cutting concern) 들로 인해 기능의 변경 및 추가 등의 작업에 많은 시간과 노력이 필요하다는 점이다[5,8]. 본 논문 에서는 이와 같은 문제점 해결을 위해 Aspect-Oriented Programming (AOP) 기술을 RFID 미들웨어 구현에 도입하고자 한다. 애스펙트 란 최근 주목받고 있는 새로운 프로그래밍 패러다임인 AOP 의 핵심 개념으로 시스템 전반에 흩어져 유지보수를 어렵게 하고 동적 확장을 방해하는 하는 횡단관심사(cross cutting concern)를 해결하기 위해 디자인된 모듈을 말한다[5]. 일반적으로 애스펙트는 특정 기능을 구현하고 있는 코드들과 그 코드들이 삽입되어야 할 위치들로 기술되어 있다. AOP는 2006년 1월 저명한 IEEE 소프트웨어 저널의 표지를 장식할 정도로 주목받고 있는 차세대 프로그래밍 기술이다[5].

이 연구에서는 기존 RFID 미들웨어의 문제점 해결을 위해 AOP(Aspect Oriented Programming) 기반 ALE 미들웨어 프레임워크 설계 및 ALE 규약에 따른 기능구현과 확

장, 그리고 유지보수의 편리성을 위해 RFID 미들웨어 전용 애스펙트를 설계한다.

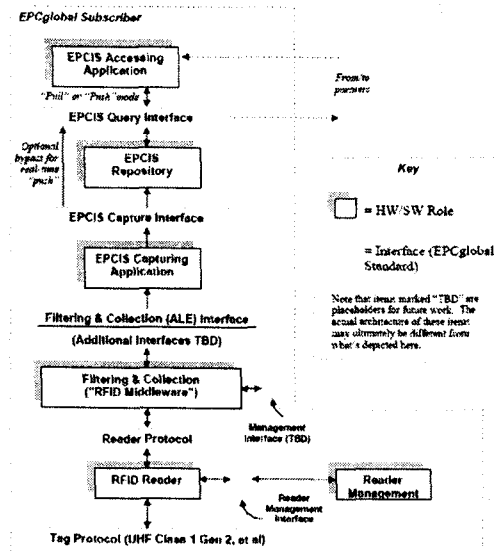
II. RFID 시스템 아키텍처

1. RFID 시스템 개요

일반적으로 RFID 시스템은 사물에 부착되어 이를 유일하게 구분할 수 있는 정보를 가지고 있는 RFID태그, 그리고 태그에 담겨진 정보를 인식하기 위한 리더기, 리더기와 연결되어 태그로부터 수집된 정보를 처리하기 위한 호스트 시스템으로 구성 되어 진다[2]. RFID 미들웨어는 다수의 리더기로부터 태그정보를 수집하고 수집된 데이터를 정제 하고 응용프로그램에서 필요로 하는 형태로 데이터를 가공해 전달하는 소프트웨어로 기존 응용 시스템과 RFID 하드웨어의 연동을 지원하는 역할을 수행한다.

현재 RFID 미들웨어와 관련한 국제표준화는 EPCglobal 을 중심으로 이루어지고 있으며 최근에는 EPC class1 Gen.2가 RFID 코드 분야의 ISO 표준으로 승인되는 등 EPCglobal 은 RFID 분야의 실질적인 표준화 기관으로 발전 하였다[4].

EPCglobal에서 제시하는 RFID 응용 시스템의 구조는 그림 1과 같다.



(그림 1) Epcglobal 표준 시스템 구성
(Fig. 1) Standard Epcglobal System

일반적인 RFID 시스템의 기본적인 요구사항과 EPCglobal 의 시스템 아키텍처 컴포넌트와의 관계는 표 1과 같다.

(표 1) RFID 시스템 요구사항과 구성요소
 (Table 1) RFID System requirement and Component

기능	H/W 구성요소	S/W 구성요소
RFID 태그 인코딩	RFID 리더 및 안테나	애플리케이션
인코딩 된 RFID 태그 부착	RFID 태그 라벨 인쇄기	애플리케이션
FID 태그가 부착된 아이템의 트래킹	RFID 리더, 안테나, 태그	미들웨어
RFID 정보를 비즈니스 애플리케이션과 통합	RFID 리더, 안테나, 태그	미들웨어, Information Service
비즈니스 간 공유될 수 있는 정보의 제공	RFID 리더, 안테나, 태그	미들웨어, Information Service
지능형 장치의 self organization 지원	RFID 리더, 안테나, 태그	미들웨어

2. ALE(Application Level Event)

EPCglobal의 미들웨어 규격은 초기 MIT Auto-ID 센터의 Savant 기반의 참조 모델을 제안 했으나 2004년부터는 인터페이스 중심의 ALE(Application Level Events)를 제안 해 2005년 7월 EPCglobal 의 공식 표준으로 승인 되었다[4]. 아직까지 Savant 기반의 미들웨어도 일부 사용되고 있으나 최근 도입되는 RFID 시스템 들에서는 대부분 ALE 기반의 미들웨어를 사용하고 있다. 일반적으로 RFID 미들웨어는 역할에 따라 다음과 같은 세가지 컴포넌트 블록으로 구분한다.

- Edge 미들웨어
- 이기종 의 다수 RFID 리더를 통합 관리하고 제어
- Filtering and Collection 미들웨어(ALE)

3. AOP(Aspect Oriented Programming)

Aspect Oriented Programming 은 객체지향 프로그래밍의 문제점을 보완해 소프트웨어 생산성 및 유지보수를 향상시킬 수 있는 새로운 대안으로 2006년 1월 저명한 IEEE 소프트웨어 저널 의 표지를 장식할 정도로 주목받고 있는 차세대 프로그래밍 기술이다[7].

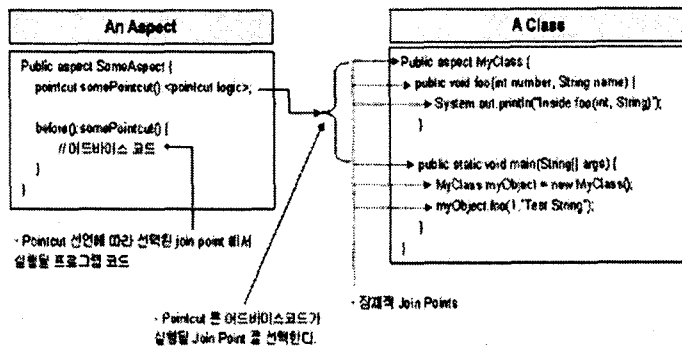
현재 애스펙트(Aspect)의 우리말 표현이 명확하지 않지만 기본적인 개념은 프로그래밍의 본래 관심사(즉, 비즈니스로직)와 프로그램 모듈 전반에 흩어져 핵심 관심사 구현에 횡단적으로 필요한 횡단 관심사(가장 간단한 예로 로깅)를 분리 하고자 하는 프로그래밍 기법을 말한다.

일반적으로 다양한 소프트웨어 환경에 유기적으로 대응할 수 있는 소프트웨어 시스템을 만들기 위해서는 설계가 가장 중요한 것으로 여겨지고 있다. 또한 객체지향 설계 및 프로그래밍

기술 등을 통해 소프트웨어의 유지보수와 생산성을 향상시키기 위한 노력도 필요하다. 그러나 아무리 완벽한 설계라 할지라도 문제점이 있기 마련이며 시스템을 운영하다보면 처음에는 생각지 못했던 다양한 문제점이 발생하게 된다. 이러한 문제는 주어진 프로그래밍 모델에서 일반적인 책임 구분을 넘어서기 때문이다[5]. 예를 들어, 객체 지향 프로그래밍에서 모듈화의 기본 단위는 클래스이며, 횡단적 관심사는 여러 클래스에 걸쳐 있는 관심사를 말한다. 전형적인 횡단적 관심사에는 로깅, 문맥 의존형 에러 처리, 성능 최적화 및 설계 패턴 등이 포함된다[15]. 애스펙트의 정의는 몇몇 코드 혹은 결합점 에서 실행 되어져야 할 행위인 어드바이스, 그리고 어디에서, 언제, 어떻게 실행되는지에 대한 설명이 포함될 수 있다.

ALE 미들웨어를 위한 애스펙트의 설계를 위해서는 먼저 ALE 스펙 분석을 통해 기본적인 횡단 관심사를 도출해 내는 것이 필요하다. 그 다음 확장점에 대한 표준화된 구현과 횡단 관심사의 효율적인 처리 방안을 찾아내야 한다. 또한 적절한 결합점(join point)과 교차점(pointcut)을 파악하고 필요한 동작을 기술하는 어드바이스(advice)코드를 설계해 나가야 한다.

결합점 이란 프로그램 실행 코드 중 애스펙트의 삽입이 가능한 영역을 말하는 것으로 프로그램의 실행 플로우 혹은 구조에서 추가적인 행위가 부가될 수 있는 잘 정의된 장소를 말한다. 대표적으로 필드정의 및 접근/수정자, 예외, 실행 이벤트, 상태를 포함한 다양한 상황으로 정의 될 수 있지만 보편적으로는 메서드 호출위치가 기본이 된다. 교차점은 결합점 중 실제 어드바이스가 적용되는 지점을 말하는 것으로 어드바이스 조각을 실행하기 위한 결합점에 대한 프로그램적 관심을 선언하기 위한 메커니즘으로 볼 수 있다. 다른 해석으로 결합점 선정을 위한 규칙으로 이해 할 수도 있다. 그림 2는 AOP 에 있어 애스펙트, 결합점, 어드바이스의 관계를 도식화 한 것이다.



(그림 2) AOP 기본 구조
(Fig. 2) Basic AOP structure

III. ALE 미들웨어프레임워크 설계

1. ALE 미들웨어 요구사항

ALE 미들웨어 프레임워크는 ALE 표준 스펙의 요구사항을 기본적으로 제공 하면서 이를 바탕으로 확장된 미들웨어의 구현을 가능하게 하는 API제공, 그리고 AOP 와의 연동을 지원 하기 위한 모듈로 구성된다.

가장 기본적인 ALE 미들웨어 기능은 Edge 영역의 여러 RFID 리더와 소프트웨어 시스템 으로부터 센싱 되고 전달된 RFID 태그 정보를 필터링 하고 요약(collection), 그룹핑 하여 ALE 미들웨어로부터 데이터를 중계 받고자 하는 여러 엔터프라이즈 애플리케이션에 이벤트 정보를 실시간으로 전달하는 것이다. 애플리케이션과의 메시지 교환에는 표준 웹 서비스 API 를 제공해 XML/SOAP 메시지를 자동으로 생성하고 전달한다.

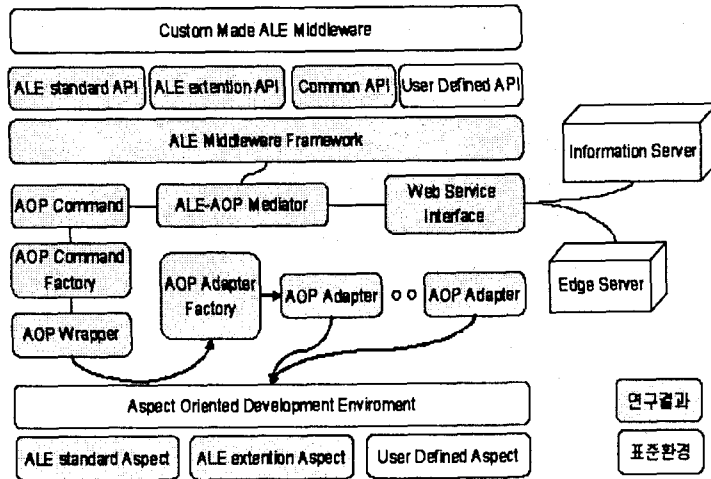
엔터프라이즈 애플리케이션과의 연동에는 메시지 요청즉시 결과를 전달하는 동기 방식과 요청후 응답을 기다리지 않고 계속해서 다른 작업을 병행 할 수 있는 비동기 연동방식을 모두 지원해야 한다.

이러한 웹서비스 API를 통해 태그 센싱 정보를 얻고자 하는 RFID 리더를 쉽게 선택하고 이벤트 사이클을 직접 제어하며, 여러 보고 형식을 직접 정의할 수 있어야 한다.

제안하는 ALE 미들웨어 프레임워크는 AOP 에 대한 캡슐화를 위해 Mediator Pattern 을 적용한 ALE-AOP Mediator 를 배치 하였으며 이를 통해 ALE 기능을 AOP로 연결 처 리 할 수 있다. 또한 AOP 관련 기능의 수행 처리를 위한 AOP Command Factory 및 AOP Command 객체를 설계하고 다양한 AOP 프레임워크를 지원할 수 있도록 AOP Adapter 를 두어 설계된 RFID 미들웨어 프레임워크가 여러 AOP 환경에서 운영될 수 있도록 하였다.

사용자 정의 RFID 미들웨어의 개발을 위해서는 표준 ALE API를 제공하고 AOP 확장을 위한 확장 API 를 함께 제공한다. 또한 공개기반의 표준 라이브러리 및 사용자 정의 API를 추가 할 수 있도록 해 개발의 유연성을 가질 수 있도록 고려되어 있다. 또한 외부 시스템과의 인터페이스를 위해 표준화 된 웹 서비스 인터페이스를 제공하므로 웹 서비스 표준을 제공하는 모든 WAS에서 운영 될 수 있다. AOP 레벨 에서는 ALE 표준 기능 지원을 위한 애스펙 트 및 확장 애스펙트, 그리고 사용자 정의 애스펙트를 추가 할 수 있도록 가이드 라인을 제공 한다.

그림 3은 본 논문에서 구현한 ALE 미들웨어 프레임워크의 전체 구성을 보여준다.



(그림 3) 구현시스템 개요

(Fig. 3) System architecture overview

2. ALE 메인 API 스펙구현

2.1 ALE 표준 API

ALE 표준 API는 Epcglobal 표준에 따른 미들웨어와의 호환성을 유지하기 위해 구현하는 것으로 기본적으로 RFID 이벤트 라이프사이클을 관장하는 ECSpec 관리에 주된 목적이 있다.

(표 2) ALE 표준 API

(Table 2) Standard ALE API

```

define(specName:string, spec:ECSpec) : void
undefine(specName:string) : void
getECSpec(specName:string) : ECSpec
getECSpecNames() : List // returns a list of specNames
as strings
subscribe(specName:string, notificationURI:string) : void
unsubscribe(specName:string, notificationURI:string) : void
poll(specName:string) : ECRports
immediate(spec:ECSpec) : ECRports
getSubscribers(specName:String) : List // of notification
URIs
getStandardVersion() : string
getVendorVersion() : string
<<extension point>>
    
```

표 2에 따른 ECSpec을 정의하기 위한 XML 스키마는 다음과 같이 정의 할 수 있다.

[표 3] ECSpec XML 스키마
[Table 3] ECSpec XML Schema

```

<xsd:complexType name="ECSpec">
  <xsd:complexContent>
    <xsd:extension base="epcglobal:Document">
      <xsd:sequence>
        <xsd:element name="logicalReaders"
type="ale:ECLogicalReaders"/>
        <xsd:element name="boundarySpec"
type="ale:ECBoundarySpec"/>

        <xsd:element name="reportSpecs"
type="ale:ECReportSpecs"/>
        <xsd:element name="extension"
type="ale:ECSpecExtension" minOccurs="0"/>
      </xsd:sequence>
      <xsd:attribute name="includeSpecInReports"
type="xsd:boolean" default="false"/>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

```

2.2. ALE 웹서비스 구현

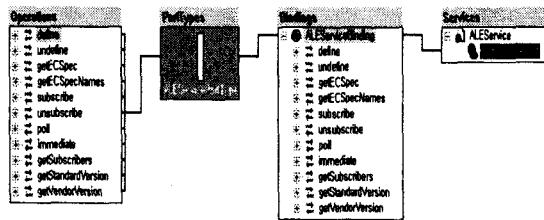
ALE 미들웨어는 기본 API로 XML/SOAP 기반 웹 서비스를 ALE 인터페이스 API로 제공한다. 이벤트 사이클(ECSpec) 정의 및 해제를 위해서는 define, undefine 서비스를 제공한다. ALE 이벤트에 대한 응용시스템의 구독 등록 및 해제를 위해서는 subscribe, unsubscribe 서비스를 제공한다. subscribe 서비스를 이용해서 이벤트 구독 등록을 하는 경우에는 ALE 통지 기능을 이용해 비동기 방식으로 응용시스템에 ALE 이벤트 정보를 전송한다. 만일 동기방식으로 ALE로부터 직접 이벤트 정보를 받고자 한다면 응용시스템은 poll 이나 immediate 서비스를 이용 할 수 있다.

poll 서비스는 기존에 define 서비스를 이용해 이미 정의되어 있는 ECSpec에 대해 동기 방식으로 ALE 이벤트 정보를 ECReports로 제공한다. immediate 서비스는 임시적으로 ECSpec을 정의하고 동기 방식으로 ALE 이벤트 정보를 ECReports로 제공한다. immediate에 의해 정의된 ECSpec은 재사용되지 않는다. 표 4 는 ECSpec 지원을 위한 XML 설정 파일 목록 이다.

[표 4] XML 설정화일
(Table 4) XML configuration file

파일명	내용	비고
ALE.wsdl	ALE Web Service Description Language	EPC 표준
AleMessage.xsd	ALE API의 SOAP 메시지 Envelop 타입 정의 XML Schema	자체 정의
AleException.xsd	ALE API SOAP Fault Exception 타입 정의 XML Schema	자체 정의

그림 4는 ECSpec의 XML 웹서비스 구성도를 보여주고 있으며 ECSpec 이외의 서비스들도 유사한 방식으로 정의한다.



(그림 4) ECSpec XML 웹서비스 구성도
(Fig. 4) ECSpec XML Webservices configuration

3. ALE 미들웨어 확장 애스펙트 설계

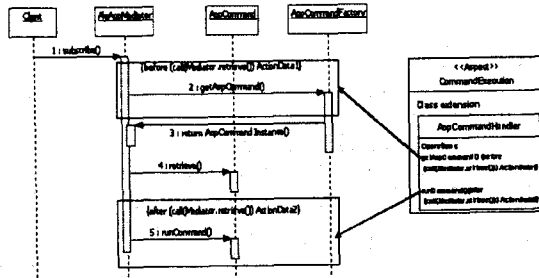
3.1. 설계 패턴 및 설계 기법

일반적으로 애플리케이션의 기능 확장이나 유지보수를 위해 애스펙트 를 사용하기 위해서는 먼저 적절한 결합점을 분석 및 교차점을 기술하고 필요한 어드바이스 코드를 기술하는 과정이 필요하다.

그러나 본 논문에서는 표준 ALE 미들웨어 스펙에서 AOP를 지원하는 것이므로 개발자가 AOP에 대한 추가적인 학습 없이 AOP기능을 이용할 수 있도록 ALE 미들웨어 프레임워크를 설계 했다. 설계된 미들웨어 프레임워크는 AleAppMediator 객체를 통해 필요한 기능을 command 형태로 전달하고 AppCommnad 객체가 AppCommandFactory 를 통해 해당 애스펙트 를 참조하는 형식으로 되어 있다. 따라서 AOP 기반의 기능의 확장이나 수정을 원하는 경우 단지 AOP 에 대한 구현 부분을 캡슐화 하기 위해 만들어진 애스펙트 클래스를 상속받아 원하는 기능이나 수정이 필요한 부분을 일반적인 자바 코드로 구현하면 된다.

그림 7은 UML 기반의 애스펙트 설계를 보여주고 있다.

ALE 미들웨어를 위한 애스펙트의 설계에는 UML 기반의 표기법을 사용하고 있어 기존 UML을 사용하던 개발자들이 보다 손쉽게 ALE 미들웨어 애스펙트를 이해하고 개발 할 수 있도록 지원한다.

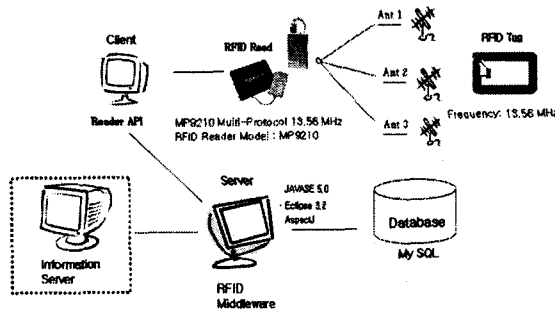


(그림 5) UML 기반의 애스펙트 설계
 (Fig. 5) UML based aspect design

3.2. 실험환경

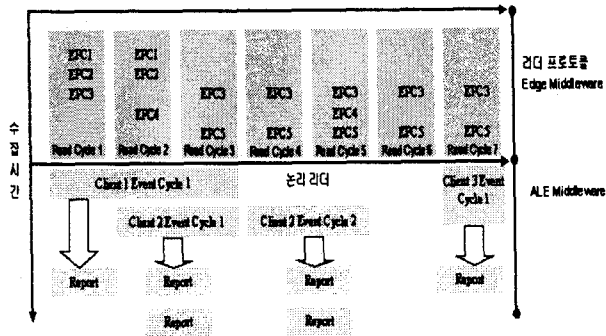
실험의 목적은 설계된 ALE 미들웨어 프레임워크가 기본 ALE 스펙을 정상적으로 구현하고 있는지에 대한 여부와 제공된 프레임워크를 통해 AOP에 대한 사전 지식 없이도 기능을 확장 하거나 기존 기능을 수정 및 반영 여부를 확인하고 타 RFID 미들웨어와의 성능을 비교 하는 것이다. 이를 위해 다음과 같은 실험 환경을 준비 하였다.

실험은 하나의 표준 13.56Mhz RFID 태그/리더를 사용하며 리더와의 물리적인 인터페이스 부분은 리더 벤더에서 제공하는 기본 API를 이용하며 미들웨어에서 기업정보 시스템으로 메시지를 분기하기 위한 Information Server는 미들웨어 기본 구현에 대한 실험과는 무관 하므로 별도로 설치하지 않았다. 그림 8은 실험환경을 도식화 한 것이다.



(그림 6) 실험환경
 (Fig. 6) Testing environment

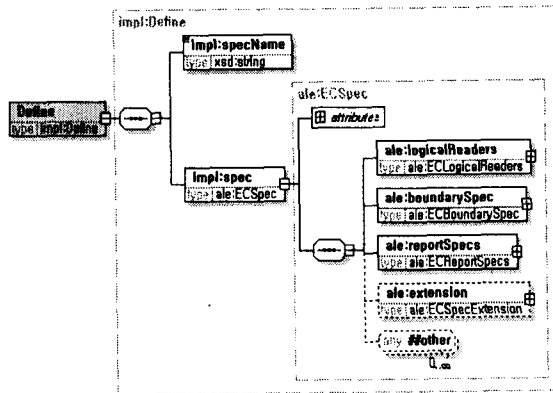
구축된 실험환경에서 RFID 미들웨어 운영을 위한 이벤트 사이클은 그림 7과 같이 정의 한다. Edge 미들웨어 영역인 리더 프로토콜은 리더 제조사에서 제공하는 전용 API를 사용하고 ALE 미들웨어 영역이 실제 실험에 포함된 부분이다.



(그림 7) 실험을 위한 이벤트사이클 정의
(Fig. 7) Event cycle definition

3.3 ALE 미들웨어 확장

ALE 미들웨어 확장을 실험하기 위해 표준 ALE 메소드 중 가장 기본이 되는 define 메소드를 선정하고 실험을 위해 그림9와 같이 Define 메소드에 대한 XML 스키마에 따라 수정된 define 모듈을 정의 하고 해당 XML 웹 서비스를 구현 한다.



(그림 8) define() 메서드 구현을 위한 XML 스키마
(Fig. 8) XML schema for define() method

그림 8은 실험에 사용된 ALE 이벤트 사이클 중 이벤트 사이클 정의를 위한 수정된 define 서비스 요청에 대한 웹서비스 request SOAP 메시지이다.

[표 5] 실험에 사용된 XML SOAP 메시지
 (Table 5) XML SOAP message for testing

```

<SOAP ENV:Envelope
xmlns:SOAP ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:SOAP ENC="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:m0="urn:epcglobal:ale:xsd:1">
<SOAP ENV:Body>

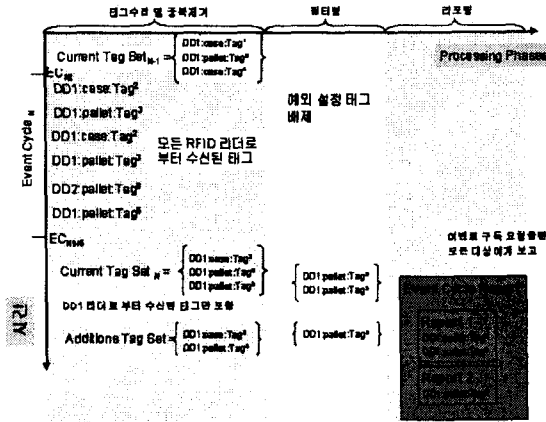
<m0:reportSpecs>
<m0:reportSpec ale:reportOnlyOnChange="false"
ale:reportIfEmpty="false" ale:reportName="입고검수이벤트">
  <m0:reportSet ale:set="CURRENT"/>
  <m0:groupSpec>
    <m0:pattern>urn:epc:pat:sgtin 96:X.*.*</m0:pattern>
  </m0:groupSpec>
  <m0:output ale:includeCount="true" ale:includeTag="true"
ale:includeRawDecimal="false" ale:includeEPC="false"
ale:includeRawHex="true"
ale:includeGroupListMemberEx="true">
    </m0:output>
  </m0:reportSpec>
</m0:reportSpecs>
<m0:extension/>
  </m:spec>
</m:Define>
</SOAP ENV:Body>
</SOAP ENV:Envelope>
  
```

4. Tag 데이터 필터링 및 추적

실험과정은 태그수집 및 중복제거 -> 필터링 -> 리포팅 순서로 RFID 리더로부터 수신된 데이터에 대한 흐름을 추적하는 것으로 진행 되었으며 DD1, DD2 는 수식 (1)과 같이 구성 되었다.

설정된 태그 구성에 따라 이벤트 사이클에 따른 태그 데이터를 처리한 결과는 시간에 따른 이벤트 사이클의 동작 환경에서 절차에 따라 요구된 리포트가 생성되었는지를 확인하는 것으로 진행되었으며 그림 9와 같은 결과를 얻었다.

$$\begin{aligned}
 \text{Current Tagset } N &= \left\{ \begin{array}{l} DD1:\text{case:Tag}^1 \\ DD1:\text{pallet:Tag}^2 \\ DD1:\text{case:Tag}^k \end{array} \right\} \\
 \text{Current Tagset } N &= \left\{ \begin{array}{l} DD1:\text{case:Tag}^2 \\ DD1:\text{pallet:Tag}^3 \\ DD1:\text{pallet:Tag}^3 \end{array} \right\} \\
 \text{Current Tagset} &= \left\{ \begin{array}{l} DD1:\text{case:Tag}^2 \\ DD1:\text{pallet:Tag}^3 \end{array} \right\} \quad (1)
 \end{aligned}$$



(그림 9) 실험결과 데이터 리포트
 (Fig. 9) Testing result report

실험 결과에 따르면 수정된 define 메소드에 따라 예외설정 태그 데이터를 배제 시키면서 수정된 define 메소드에 따른 ECRReport 가 생성 되었고 생성된 결과는 이벤트 구독을 요청한 모든 대상 클라이언트에 전달됨을 확인할 수 있었다. 실험 결과를 정리하면 제안하고 구현한 ALE 미들웨어 프레임워크의 지원 범위는 표 6과 같다.

(표 6) 제안모델과 기존모델 비교
 (Table 6)

순번	항목	제안 모델	기존 모델
1	표준 ALE 지원 RFID 미들웨어	지원	지원
2	기능확장을 위한 API 제공	지원	지원
3	AOP 캡슐화 및 프레임워크 제공	지원	미지원(개별 구현가능)
4	ALE 지원을 위한 표준 애스펙트 제공	지원	미지원
5	통합 프레임워크 기반 개발 지원	지원	미지원
7	통합개발환경 지원(IDE)	미지원	부분적 지원

IV. 결 론

제안된 ALE 미들웨어 프레임워크 및 애스펙트는 AOP에 대한 전문적인 지식이 없어도 AOP환경을 지원하는 RFID 미들웨어를 개발 할 수 있도록 설계 되었다. 이와 같은 설계상의 특징으로 인해 개발자들은 보다 손쉽게 표준에 기반한 RFID 미들웨어를 개발 할 수 있으며 기존 ALE 기반 미들웨어의 유지보수 어려움을 해결하고 표준화 된 방법으로 기능의 확장이 가능하다. 실험을 통해 표준 ALE 스펙의 RFID 미들웨어 로서 동작 하는데 아무런

문제가 없었으며 제공된 API를 통해 AOP에 대한 사전 지식 없이 AOP 기반의 기능 확장이 가능 했음을 보였다. 또한 이벤트 사이클에 따른 메시지 추적을 통해 구현된 프레임워크가 전체적으로 통합운영 되는 것을 확인할 수 있었다.

본 논문의 연구 결과 RFID 미들웨어 시스템의 동적 확장과 생산성 및 기능의 확장에 애스펙트를 적용 하는것에 대한 가능성을 확인 했으며 국내 에서는 아직 시작단계인 AOSD(Asspect Oriented Software Development) 분야에서 유비쿼터스 응용 분야와의 접목을 시도한 것에 의미를 부여하고 싶다.

향후 연구과제로는 다양한 AOP 개발 프레임워크 지원 및 옛지 미들웨어에 대한 통합 구현, 그리고 기존 RFID 미들웨어와의 보다 손쉬운 인터페이스 모델을 연구하는 것이다.

참 고 문 헌

1. Frost & Sullivan, "World RFID Middleware Markets," Frost & Sullivan Research Report, 2005.
2. Christian Floerkemeier, Mattias Lampe, "RFID middleware design - addressing application requirements and RFID constraints," In Proceedings of sOc-EUSAI conference, 2005.
3. EPCglobal, <http://www.epcglobalinc.com>.
4. "Extending the EPC Network - The Potential of RFID in Anti-Counterfeiting," In Proceedings of ACM Symposium on Applied Computing, 2005.
5. William Griswold, Macneil Shonle, Kevin Sullivan, Yuanyuan Song, Nishit Tewari, Yuanfang Cai, Hridayesh Rajan, "Modular Software Design with Crosscutting Interfaces," Journal of IEEE SOFTWARE, 2006.
6. Erich Gamma, Richard Helm, Ralph Johnson, John Vissides, "Design Patterns-Elements of Reusable Object-Oriented Software," Addison Wesley, 1995.
7. Singularity Project, <http://singularity.firstopen.org>
8. Ivar Jacobson, Pan-Wei NG, "Aspect-Oriented Software Development with Use Cases," Addison Wesley, 2005.
9. 장병준, 안선일, 이윤덕, "RFID/USN 기술개발 동향," 정보과학회지, 2005.
10. 황희정, 강운구 "피어 네트워크에서의 효율적 객체공유를 위한 공유객체모델", 정보과학회지, 2004.



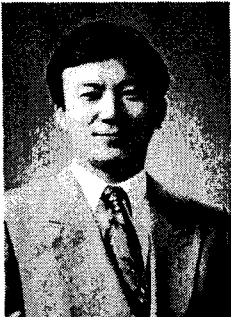
朴世承(Seiseung Park)(정회원)

- 1975. 조선대학교 전자공학과 학사
- 1980. 조선대학교 전자공학과 석사
- 1990. 경희대학교 전자공학과 박사
- 1980. ~ 현재 조선대학교 전자공학과 교수
- 관심분야 : 로보틱스 및 응용, 이동로봇



黃喜正(HeeJoung Hwang)(會員)

- 1997. 경기대 경영학 학사
- 2000. 인하대 전자계산학과 석사
- 2005. ~ 현재 인천대학교
컴퓨터공학과 박사과정중
- 2000. ~ 현재 가천의과학대학교
정보기술학과 조교수
- 관심분야 : 소프트웨어공학, 분산객체, 의료정보



崔鎭卓(Jintak Choi)(正會員)

- 1977. 동국대 수학과 학사
- 1982. 동국대 전자계산학과 석사
- 1991. 경희대 전자공학과 박사
- 1987. ~ 현재 인천대학교
컴퓨터공학과 교수
- 관심분야 : 데이터베이스, 정보보호, 암호학