

# RFID 애플리케이션 개발을 위한 비즈니스 이벤트 프레임워크 설계

정회원 손 충 범\*

## Design of Business Event Framework for Developing RFID Application

Chung Beom Son\* *Regular Member*

요 약

RFID는 자동인식 및 데이터 획득 기술의 하나로서 사물이 가지고 있는 정보를 자동적으로 취득, 온라인으로 관련 정보를 처리하는 자동처리 시스템 구현의 핵심요소 기술이다. 그러나 해당 사물에 대한 간단한 정보만을 알 수 있고 부가적인 상세 정보를 알기에는 매우 제한적이다. 또한, RFID 애플리케이션의 비즈니스 이벤트를 처리하기 위해서는 RFID에 대한 지식과 통신 방법을 익혀 개발하는데 어려움이 있다.

본 논문에서는 RFID 애플리케이션의 개발에 도움을 주는 비즈니스 이벤트를 정의하고 비즈니스 이벤트를 효율적으로 관리할 수 있는 비즈니스 이벤트 프레임워크를 설계한다.

**Key Words** : RFID, Business Event, Framework

ABSTRACT

RFID as one of automatic identification and data acquisition technique is a important technology of automatic processing system implementation to automatically get the information which the thing is having and to process related information with on-line. However, we can know the brief information about the thing and to know additional detail of the thing is very limited. In addition, it is difficult to learn and develop a knowledge and a communication method about RFID to process business event of RFID applications.

In this paper, we define the business events which can help the development of RFID applications and design a business events framework to efficiently manage the business events.

### 1. 서 론

RFID (Radio Frequency Identification)는 식별자 정보를 갖고 있는 태그와 이 정보를 무선 주파수를 통해서 이용해서 읽을 수 있는 리더로 구성된다. RFID는 비접촉으로 사물을 인식할 수 있기 때문에 보안, 물류, 물품 판매 등에 널리 활용되고 있다. 짧은 시간 내에 빠르게 생성되는 RFID 이벤트의 형태는 단순하

지만 총 데이터의 크기는 아주 크기 때문에 RFID 이벤트를 효율적으로 처리해 주는 RFID 미들웨어 시스템들이 여러 업체들에 의해 개발되고 있다.

이런 RFID 시스템을 응용에 적용하기 위해서는 RFID 특성을 고려하여 시스템을 설계 및 개발해야 한다. RFID 시스템 환경에서는 대량의 태그 데이터가 수집되어 연속적으로 발생되는데 이런 데이터는 중복되거나 필요없는 데이터가 포함된다. 이런 이유로

\* 본 논문은 2008학년도 인하공업전문대학 교내연구비지원에 의하여 연구되었습니다.

\* 인하공업전문대학 정보통신과(cbson@inhac.ac.kr)

논문번호 : 10002-0121, 접수일자 : 2010년 1월 21일

RFID을 적용한 응용에서는 RFID 데이터에서 필요한 정보를 분류하고 실시간으로 효율적으로 처리해야 한다. 또한 RFID 시스템과 통신을 하기 위해서는 다양한 RFID 프로토콜이 필요하다. 하지만 이런 다양한 시스템을 활용하기에 좀 더 편리한 방법 있다면 RFID 어플리케이션 개발에 도움이 될 것이다.

이에 본 논문에서는 RFID 애플리케이션에서 편리하게 이용할 수 있는 비즈니스 이벤트를 정의하고 이런 비즈니스 이벤트를 생성하여 애플리케이션에 전달할 있는 비즈니스 이벤트 프레임워크(Business Event Framework)를 설계한다.

본 논문의 구성은 다음과 같다. 2장에서는 RFID에 대한 관련 연구를 살펴보고, 3장에서는 비즈니스 이벤트와 비즈니스 이벤트 프레임워크에 대한 설계 내용과 사례 및 평가 내용을 언급한다. 마지막으로 결론 및 향후 연구과제에 대해 제시한다.

## II. 관련연구

EPCglobal 네트워크는 RFID 기술을 이용하여 상품을 자동으로 식별하고 식별된 상품 정보를 인터넷을 통하여 거래업체들과 공유함으로써 상품 이동 현황을 실시간 추적 조회할 수 있는 시스템으로 EPCglobal에서 제안한 구조가 업계에서 사실상의 표준으로 간주되고 있다. EPCglobal 구조의 핵심은 RFID 기술과 EPC로 RFID 기술은 무선으로 동작 가능한 태그, 센서에 대한 기술이고 EPC는 제품에 대한 전자적 일련번호에 대한 표준이다.

RFID의 특성상 범위 내의 태그를 모두 읽기 때문에 같은 EPC(Electronic Product Code)정보가 여러 번 읽히거나 원하지 않는 EPC 정보가 다른 데이터와 함께 리더를 통해 읽혀질 수 있다. 이러한 RFID의 특성으로 인하여 애플리케이션 개발자는 스무싱(smoothing), 필터링과 같은 데이터 가공 방법과 많은 양의 데이터를 실시간으로 효율적으로 처리할 수 있는 방법을 고려해야 한다. 이 외에도 읽혀진 EPC에 대한 제품의 상세정보가 필요하거나 해당 물품의 이동경로에 대한 정보 등이 필요할 수 있는데 이러한 것들을 지원하기 위해서 RFID 시스템 아키텍처에 대한 연구가 진행 중에 있다.

그림 1에서 처럼 EPCglobal의 구조에서는 미들웨어으로써 ALE(Application Level Events)엔진이 존재하고 이 미들웨어의 상위에 애플리케이션이 연결된다<sup>1,2</sup>.

EPCglobal 네트워크에서는 EPCIS나 ONS와 같이 상황에 따라 사용될 수도 있고 사용되지 않을 수 있는

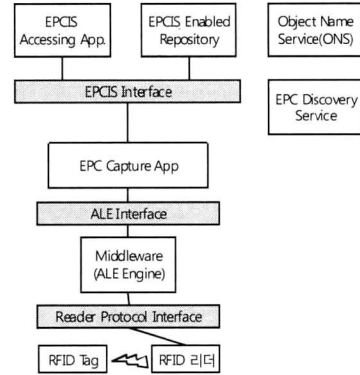


그림 1. EPC 네트워크 구조

별도의 구성 요소도 도입하고 있다. EPC 관련 데이터를 데이터베이스에 저장 및 관리하고, 그 데이터에 대한 상세한 정보를 제공하는 EPCIS(EPC Information Service), 특정 EPC와 관련된 상세 정보가 저장된 EPCIS의 위치를 알려주는 ONS(Object Name Service), 특정 EPC에 대한 관련 데이터의 위치 정보를 제공하는 검색 서비스 등이 외부 시스템에 존재하고 애플리케이션은 이러한 시스템과도 통신하여 정보를 주고 받는다<sup>3</sup>[4]. 이처럼 EPCglobal의 구조에서는 애플리케이션이 다양한 시스템과 정보를 주고받아야 한다. 이렇게 여러 시스템을 직접 제어하여 특정 애플리케이션을 개발할 수 있지만 개발 비용이 많이 들기 때문에 이 시스템들을 통합하여 좀더 편리하게 사용할 수 있도록 지원하는 플랫폼이 있다면 더 편리할 것이다. 이러한 이유로 애플리케이션에서 RFID 이벤트를 편리하고 효과적으로 이용할 수 있도록 지원하는 방법에 대한 연구가 필요하다.

## III. 비즈니스 이벤트 프레임워크 설계

여기서는 소단원에 관한 내용을 간단히 살펴본다. 여기서는 소단원에 관한 내용을 간단히 살펴본다.

### 3.1 비즈니스 이벤트 정의

#### 3.1.1 이벤트

RFID 이벤트는 EPCglobal에서 제시한 미들웨어에서 응용에 전달하는 이벤트로써 숫자로 되어 있는 EPC와 읽혀진 시간, 리더 등에 대한 데이터가 포함되어 있다. 그러나 이러한 정보만으로는 EPC의 정확한 의미를 알 수 가 없다. 응용시스템에서 사용하기 위해서는 수집된 EPC를 가지고 ONS에 질의하여 EPC를

생성한 제조사의 EPCIS 주소를 획득하여 다시 EPCIS로부터 EPC와 연관된 고유한 속성정보를 획득해야 한다. 이처럼 다른 구성요소와 통신을 해야 하므로 응용시스템이 처리해야 하는 본연의 업무 수행에 있어 어려움이 있다. 이에 반해 RFID 비즈니스 이벤트는 RFID 이벤트와 달리 비즈니스 로직에서 바로 사용될 수 있는 이벤트를 의미하며, 예를 들어, 학생에 대한 소속을 조회하기 위해 일단 RFID 미들웨어로부터 EPC를 읽어 들어 응용에 전달하여 처리되는데 이 EPC 정보만 가지고는 소속을 조회할 수 없으므로 소속 정보가 저장된 EPCIS에 질의해야 한다. 이러한 과정을 거친 후에 비즈니스 로직을 수행할 수 있다. 이때, 비즈니스 이벤트 규칙을 적용하여 RFID 비즈니스 이벤트를 생성하여 바로 비즈니스 로직에 사용된다.

### 3.1.2 BEDef

비즈니스 이벤트 정의는 크게 두 부분으로 구성되는데 필드 선언 부분과 여러 처리 과정을 정의하는 메소드 부분이다. 필드 선언 부분에서는 비즈니스 이벤트와 관련된 데이터 전달이나 중간 처리과정에서 사용되는 필드를 선언한다. 필드에는 일반적인 데이터형 이외에 RFID 고유의 데이터형을 지원된다. 메소드 부분에서는 미리 정의된 여러 개의 메소드를 비즈니스 이벤트 생성에 필요한 순서에 따라 정의한다.

### 3.1.3 태그

태그정보를 읽었을 때는 BEDef의 정의에 따라 여러 처리과정을 거치게 된다. 각각의 처리과정은 독립적으로 수행되는데 각 처리과정 사이에 필요한 정보를 넘겨주기 위한 매개체로서 필드를 사용한다. 각 처리과정에서는 처리결과를 필드에 저장하거나 입력으로 필드를 사용한다. 필드는 int 등의 일반적인 데이터형뿐만 아니라 RFID 시스템에 특화된 데이터형도 지원하여 BEDef의 사용 편의성을 높인다. 예를 들면 ALE 엔진에서 한 RFID 이벤트 사이클 동안 넘겨주는 태그 목록들을 한 번에 저장할 수 있는 태그리스트 같은 데이터형을 제공하여 BEDef 작성 시 편의성과 가독성을 높일 수 있다. 필드를 선언하기 위해 사용되는 XML 스키마<sup>5)</sup> 문법은 그림 2와 같다.

XML 스키마에서 보듯이 fields 엘리먼트 내에 자식으로 field 엘리먼트가 여러번 올 수 있다. 각 필드는 field 엘리먼트를 이용하여 선언하여 field의 이름을 나타내고, 필드의 타입이나 리스트 타입여부, 초기 값은 속성정보를 이용하여 표현한다. 이 중 데이터형은 항상 정의해야 하고 리스트 속성과 초기 값은 명시

```
<schema>
  <element name="fields">
    <complexType>
      <sequence>
        <element ref="field"/>
      </sequence>
    </complexType>
  </element>
  <element name="field" type="string"
    maxOccurs="unbounded">
    <complexType>
      <attribute name="type" type="string"
        use="required"/>
      <attribute name="list" type="boolean"
        use="optional"/>
      <attribute name="intValue" type="string"
        use="optional"/>
    </complexType>
  </element>
</scheman>
```

그림 2. 필드 선언 XML 스키마

```
<fields>
  <field type="EPC" list="true">vEPCList</field>
  <field type="int" intValue="0">vInt</field>
  <field type="string" >vString</field>
</fields>
```

그림 3. 필드 선언 예제

적으로 표현하거나 생략이 가능한데 생략했을 때에는 리스트 타입이 아닌 일반 타입이 된다. 그림 3은 EPC 목록을 저장할 EPC형 리스트 필드, int형 필드와 문자열 필드를 선언한 실제 예제이다.

### 3.1.4 메소드

비즈니스 이벤트를 생성하기 위해서는 ALE에서 리더로부터 읽은 정보를 가져오고 EPCIS나 ONS와도 정보를 주고 받아야 한다. 이러한 행동들은 서로 간섭 없이 독립적으로 수행할 수 있다. 이러한 독립적인 처리에 대한 정의를 할 수 있도록 메소드를 정의하였다. 메소드는 기본적으로 ALE, EPCIS나 ONS 등과의 상호작용에 대한 정의를 하는 것이다. 외부 시스템과의 상호작용에 대한 정의 외에도 내부적인 처리 및 비즈니스 이벤트 전달에 대한 정의를 하기 위한 추가적인 메소드가 있다. 이러한 메소드를 일련의 처리 순서대로 배치하여 비즈니스 이벤트를 생성한다. 메소드의 종류는 표 1과 같다.

메소드는 종류에 따라서 다른 메소드를 내부에 가

표 1. 메소드

메소드명	설명
ale	ALE로부터 태그에 대한 정보를 획득
epcis	EPC에 대한 상세정보를 획득
ons	EPC에 대한 상세정보를 저장한 EPCIS의 주소 획득
event	조건에 따라 비즈니스 이벤트 생성
list	list 타입에 대한 반복 처리
compute	산술 연산 처리
epcis_ds	EPC의 이동에 대한 상세정보를 획득

질 수 있는 메소드와 가질 수 없는 메소드로 구분된다.

### 3.2 시스템 구조

본 연구에서 제안하는 비즈니스 이벤트 프레임워크의 구조는 그림 4와 같다. 제시한 구조는 애플리케이션과 다른 시스템과의 사이에 비즈니스 이벤트 프레임워크가 존재하여 다른 시스템과의 상호작용을 전달하게 된다.

애플리케이션은 이전의 구조와 달리 비즈니스 이벤트 프레임워크를 이용하여 RFID 관련 처리를 할 수 있다. 비즈니스 이벤트 프레임워크는 애플리케이션이 수행할 작업 중, RFID 이벤트와 관련된 작업을 대신 하여 수행한다.

이벤트 관리기는 RFID 미들웨어에서 받은 이벤트에 대해 비즈니스 이벤트의 생성을 위해서 서로 통신하여 이벤트를 실시간으로 수집하고 RFID 미들웨어의 상태, 리더의 상태 또는 구성 정보를 수집한다. BEReport 생성기는 요구한 이벤트에 대해 XML 기반의 BERDef을 작성한다. BEF 관리기는 기존에 생성

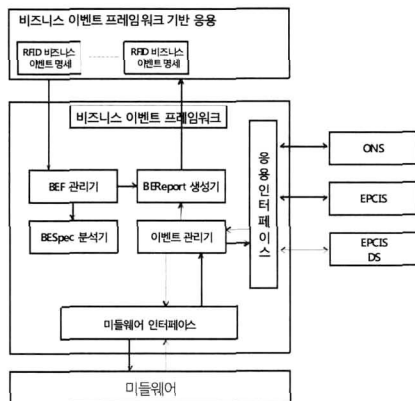


그림 4. 비즈니스 이벤트 프레임워크 기반 구조

된 비즈니스 이벤트 명세들을 관리한다. BESpec 분석기는 비즈니스 이벤트 명세를 분석하는 기능을 담당한다.

### 3.3 클래스 설계

그림 5는 클래스 다이어그램을 나타낸다. Event-Manager 클래스는 RFID 미들웨어에서 받은 이벤트에 대해 비즈니스 이벤트의 생성을 위해서 서로 통신하여 이벤트를 실시간으로 수집하고 RFID 미들웨어의 상태, 리더의 상태 또는 구성 정보를 수집한다. BEReportMaker 클래스는 요구한 이벤트에 대해 XML 기반의 BERDef을 작성한다. BEFManager는 기존에 생성된 비즈니스 이벤트 명세들을 관리한다. BESpecAnalyzer는 비즈니스 이벤트 명세를 분석하는 기능을 담당한다. MIDInterface 클래스는 RFID 미들웨어와 BEF 사이의 상호 접속을 지원한다. AppInterface 클래스는 ONS, EPCIS, EPCIS DS와 BEF 사이의 상호 접속을 지원한다.

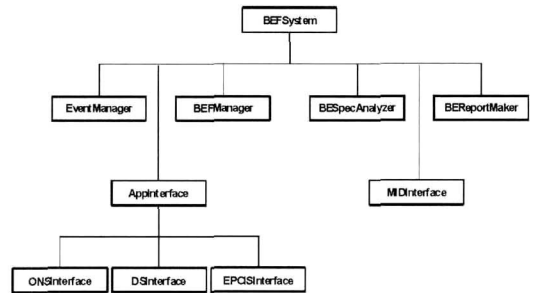


그림 5. 클래스 다이어그램

### 3.4 사례 및 평가

#### 3.4.1 출결 관리 사례 개요

대학에서 해당 과목에 대한 학생들의 강의 출결을 관리함에 있어 시간 낭비, 대리 출석 및 수업 중 이탈 등 여러 가지의 애로사항들이 발생한다. 이런 사항들을 해결하기 위해 RFID를 적용하여 출결 관리 시스템을 구축하고자 한다. 출결 관리 시스템은 고유한 학생의 위치, 특정 강의실에서의 특정 과목의 학생 출결 정보를 제공하고, 해당 교수에게 출결 정보 및 에러메시지를 발송하도록 한다.

#### 3.4.2 사나리오

출결 관리 시스템은 학생들의 입실, 퇴실에 관한 상세정보를 전달 받아서, 학번, 학생명, 학생 위치 정보를 업데이트하여 해당 교수가 이를 모니터링 한다. 또한 학생이 수업시간에 밖으로 퇴실하여 입실을 하지

않을 경우 해당 교수에게 여러 메시지를 발송한다.

### 3.4.3 RFID 비즈니스 이벤트 정의

시나리오에 따라서 출결 관리 시스템이 사용할 수 있는 RFID 비즈니스 이벤트가 다음과 같이 정의되었다.

표 2. 출결 관리 RFID 비즈니스 이벤트 정의

비즈니스 이벤트명	조건	포함정보
출석	입구리더에서 발생한 RFID 이벤트	EPC, 학번, 학생명, 현재위치
학생 위치 에러	(입출구리더에서 발생한 RFID 이벤트) AND (해당 강의실 != 현재 위치)	EPC, 학번, 학생명, 현재위치, 강의실

### 3.4.4 BEDef 기술

분석된 비즈니스 이벤트에 따라서 참조명세와 BEDef을 작성한다. providerSpec은 ALE Specification 1.1<sup>[6]</sup>을 참조하여 정의하였다. 먼저 입구 리더와 출구 리더에 관한 ProviderSpec인 그림 6과 7을 각각 작성하고 참조 이름을 각각 Classroom\_A\_In, Classroom\_A\_Out으로 정의한다.

다음으로 비즈니스 이벤트의 생성을 위해서 EPCIS에 저장되어 있는 고유정보인 강의실 정보와 학생명을 질의하기 위해 그림 8, 9와 같은 QuerySpec을 작성하고 참조이름을 각각 Classroom\_Query, Student-Name\_Query로 정의한다. EPCIS를 이용하여 EPC와 관련된 정보를 질의하기 위해서 EPCIS Specification 1.0의 EPCIS Query Control 인터페이스를 준수하였다.

마지막으로 BEDef은 표 2에서 출석 비즈니스 이벤트와 학생 위치에러 비즈니스 이벤트에 대해 각각 그

```

<providerSpec>
<ale:ECSpec xmlns:ale="urn:epcglobal:ale:xsd:1" ..>
<logicalRedader>A_Reader_In</logicalReader>
</logicalRedaders>
<boundarySpec>
<repeatPeriod unit="MS">1000</repeatPeriod>
<duration unit="MS">1000</duration>
</boundarySpec>
<reportSpec reportName="report">
<reportSet set="CURRENT" />
<output includeEPC="true" includeCount="true"/>
</reportSpec>
</ale:ECSpec>
</providerSpec>
    
```

그림 6. 강의실 A 입실 ProviderSpec

```

<providerSpec>
<ale:ECSpec xmlns:ale="urn:epcglobal:ale:xsd:1" ..>
<logicalRedader>A_Reader_Out</logicalReader>
</logicalRedaders>
<boundarySpec>
<repeatPeriod unit="MS">1000</repeatPeriod>
<duration unit="MS">1000</duration>
</boundarySpec>
<reportSpec reportName="report">
<reportSet set="CURRENT" />
<output includeEPC="true" includeCount="true" />
</reportSpec>
</ale:ECSpec>
</providerSpec>
    
```

그림 7. 강의실 A 퇴실 ProviderSpec

림 10와 그림 11으로 표현 가능하다.

```

<querySpec>
<queryname>SimpleStaticDataQuery</queryName>
<ns3:queryParams xmlns:ns4="urn:epcglobal:epcis:xsd:1"
xmlns:ns3="urn:epcglobal:epcis=query:xsd:1" ...>
<param>
<name>EQ_dataType</name>
<value><string>static</string></value>
</param>
<param>
<name>EQ_epc</name>
<value><string>#epc</string></value>
</param>
<param>
<name>EQ_schemaName</name>
<value><string>StudentSchema</string></value>
</param>
<param>
<name>EQ_attr1</name>
<value><tag>productAttr/NumberOfStudent</tag>
</value>
</param>
<param>
<name>EQ_attr2</name>
<value><tag>productAttr/NameOfStudent</tag>
</value>
</param>
</querySpec>
    
```

그림 8. 학생명을 질의하는 QuerySpec

```

<querySpec>
<queryname>SimpleStaticDataQuery</queryName>
<ns3:queryParams
xmlns:ns4="urn:epcglobal:epcis:xsd:1"
xmlns:ns3="urn:epcglobal:epcis=query:xsd:1" ...>
<param>
<name>EQ_dataType</name>
    
```

```

<value><string>static</string></value>
</param>
<param>
  <name>EQ_epc</name>
  <value><string>#epc</string></value>
</param>
<param>
  <name>EQ_schemaName</name>
  <value><string>ClassroomSchema</string></value>
</param>
<param>
  <name>EQ_attr</name>
  <value><tag>/productAttr/NameOfClassroom</tag>
</value>
</param>
</querySpec>

```

그림 9. 강의실 정보를 질의하는 QuerySpec

```

<?xml version="1.0" encoding="KSC5601"?>
<BEDef>
  <fields>
    <field type="EPC" list="false" initialValue="">
      vEPC</field>
    <field type="string" list="false"
      initialValue="강의실A">vCurPlace</field>
    <field type="string" list="false" initialValue="">
      vClassroomName</field>
    <field type="string" list="false" initialValue="">
      vStudentName</field>
    <field type="string" list="false" initialValue="">
      vStudentNumber</field>
  </fields>
  <providers>
    <ALE>ClassRoom_A_In</ALE>
  </providers>
  <list source="#ClassRoom_A_In.report.epcList"
    assign="vEPC">
    <EPCIS address="http://localhost:8080
      /EPCISService">
      <getStaticData query="ClassroomName_Query"
        epc="vEPC1">vClassroomName
      </getStaticData>
      <getStaticData query="StudentName_Query"
        epc="vEPC1">vStudentName
      </getStaticData>
    </EPCIS>
    <event name="출석" terminate="false">
      <condition>vClassroomName == vCurPlace
      </condition>
      <action>
      </action>
      <dataSet name="StudentInfo">
        <data name="epc" type="string">vEPC</data>
        <data name="학번" type="string">
          vStudentNumber</data>
        <data name="학생명" type="string">
          vStudentName</data>

```

```

<data name="현재위치" type="string">
  vCurPlace</data>
  <data name="강의실" type="string">
    vClassroomName</data>
  </dataSet>
</event>
</list>
</BEDef>

```

그림 10. 출석 BEDef

```

<?xml version="1.0" encoding="KSC5601"?>
<BEDef>
  <fields>
    <field type="EPC" list="false" initialValue="">
      vEPC</field>
    <field type="string" list="false"
      initialValue="강의실A">vCurPlace</field>
    <field type="string" list="false" initialValue="">
      vClassroomName</field>
    <field type="string" list="false" initialValue="">
      vStudentName</field>
    <field type="string" list="false" initialValue="">
      vStudentNumber</field>
  </fields>
  <providers>
    <ALE>ClassRoom_A_In</ALE>
  </providers>
  <list source="#ClassRoom_A_In.report.epcList"
    assign="vEPC">
    <EPCIS address="http://localhost:8080
      /EPCISService">
      <getStaticData query="ClassroomName_Query"
        epc="vEPC1">vClassroomName
      </getStaticData>
      <getStaticData query="StudentName_Query"
        epc="vEPC1">vStudentName
      </getStaticData>
    </EPCIS>
    <event name="학생위치에러" terminate="false">
      <condition>vClassroomName != vCurPlace
      </condition>
      <action>
      </action>
      <dataSet name="StudentInfo">
        <data name="epc" type="string">vEPC</data>
        <data name="학번" type="string">
          vStudentNumber</data>
        <data name="학생명" type="string">
          vStudentName</data>
        <data name="현재위치" type="string">
          vCurPlace</data>
        <data name="강의실" type="string">
          vClassroomName</data>
      </dataSet>
    </event>
  </list>
</BEDef>

```

그림 11. 학생 위치 에러 BEDef

3.4.5 평가

본 논문에서 제시하고 있는 비즈니스 이벤트 프레임워크를 출결 관리 사례에 적용해 보았다. 이 비즈니스 이벤트 프레임워크에 적용된 출결관리 사례에서 비즈니스 이벤트 프레임워크를 사용하지 않고 애플리케이션을 개발함으로써 RFID 비즈니스 응용시스템의 개발에 필요한 작업량이 표 3과 같이 분석된다.

비즈니스 이벤트 프레임워크를 사용할 경우에는 EPC 네트워크 구성요소들과의 통신모듈을 직접 개발하지 않으므로 비즈니스 이벤트 프레임워크를 사용하지 않고 개발했을 때보다 비즈니스 이벤트 프레임워크 기반으로 개발했을 때의 요구작업 시간이 47%로 감소된 것을 알 수 있다.

작업시간 이외에도 추가로 RFID 애플리케이션에서 필요한 모듈의 코드라인 수로 비교가 가능하다. 표 4는 표 3의 요구작업 항목 중에서 개발해야 할 모듈에 대한 코드라인 수를 비교한 것이다.

비즈니스 이벤트 프레임워크를 사용하지 않을 때 필요한 모듈로 RFID 미들웨어와의 통신 모듈, EPCIS와의 통신모듈, ONS와의 통신모듈, EPCIS 이벤트 생성 모듈이 있고 이 모듈을 위한 코드라인 수는 1065여 라인 정도이다. 비즈니스 이벤트 프레임워크를 사용할 경우에는 비즈니스 이벤트 프레임워크와의 통신모듈이 필요하고, 이는 350라인 정도로 비즈니스 이벤트 프레임워크를 사용하지 않은 경우와 비교해 봤

표 3. 작업일수 비교

요구 작업 항목	비즈니스 이벤트 프레임워크	
	사용 (일)	미사용 (일)
ALE인터페이스(ECSpec, ECReport) 습득	3	3
RFID 미들웨어와의 통신기술 및 모듈제작	0	10
EPCIS 인터페이스와 통신기술 및 모듈제작	6	15
ONS 인터페이스와 통신기술 및 모듈제작	1	5
EPCIS DS 인터페이스와 통신기술 및 모듈제작	1	5
비즈니스 이벤트 프레임워크 인터페이스 습득	4	0
비즈니스 이벤트 프레임워크와의 통신기술 및 통신모듈 제작	6	0
EPCIS 캡처링을 위한 다양한 EPCIS 이벤트 생성모듈 제작	0	4
총 계	21	42

표 4. 코드라인 수 비교

요구 모듈 항목	비즈니스 이벤트 프레임워크	
	사용 (라인)	미사용 (라인)
RFID 미들웨어와의 통신모듈	0	641
EPCIS와의 통신모듈	0	210
ONS와의 통신모듈	0	80
EPCIS DS와의 통신 모듈	0	134
비즈니스 이벤트 프레임워크와의 통신모듈	350	0
EPCIS 캡처링을 위한 다양한 EPCIS 이벤트 생성 모듈	0	352
총 계	350	1,065

을 때 32%로 코드라인 수가 감소된다.

IV. 결 론

본 논문에서는 RFID 응용의 개발에 있어 편리하게 사용될 수 있는 비즈니스 이벤트 프레임워크를 설계하여 제시하였다. 이 프레임워크를 사용할 경우 기존 RFID 응용 개발자의 개발비용과 습득해야 하는 지식을 증가하여 개발에 어려움이 있었다. 이러한 어려움을 줄이고자 본 논문에서는 비즈니스 로직을 바로 수행시킬 수 있는 비즈니스 이벤트를 정의하였다. 그리고 이러한 비즈니스 이벤트를 생성하여 응용에 전달할 수 있는 비즈니스 이벤트 프레임워크를 제시하였다. 이 비즈니스 이벤트 프레임워크를 적용할 경우 요구 작업일수 47%와 코드 라인수 32% 감소에 대한 비용 절감 효과가 있음을 알 수 있다. 향후 연구 과제로는 비즈니스 이벤트 프레임워크의 기능 확장과 외부 시스템에 대한 인터페이스의 추상화를 통해 여러 시스템에서 동일한 형태로 적용될 수 있도록 지원하는 것이다.

참 고 문 헌

- [1] 유승화, “유비쿼터스 사회의 RFID,” 전자신문사, 2005.
- [2] 안재명 외 2명, “EPCglobal Network 기반의 RFID 기술 및 활용,” 글로벌, 2007.02.
- [3] EPCglobal Inc, <http://www.epcglobalinc.org>
- [4] EPCglobal, “The EPCglobal Architecture Framework EPCglobal Final Version 1.2”, [http://www.epcglobalinc.org/standards/architecture/architecture\\_1\\_2-framework-20070910.pdf](http://www.epcglobalinc.org/standards/architecture/architecture_1_2-framework-20070910.pdf),

September 2007.

- [5] XML Schema Part 1: Structures Second Edition, <http://www.w3.org/TR/2004/REC-xmlschema-1-20041028/>
- [6] EPCglobal, "The Application Level Events Specification, Version 1.1 Part 1: Core Specification", EPC global Last Call Working Draft, May 2007.
- [7] EPCglobal, "EPC Information Services Version 1.0 Specification", Ratified Standard, April 2007.
- [8] EPCglobal, "EPCglobal Object Name Service 1.0", EPCglobal Ratified Specification, October 2005.
- [9] XQuery 1.0: An XML Query Language, <http://www.w3.org/TR/2007/REC-xquery-20070123/>

손 총 범 (Chung-Beom Sonim)

정회원



1997년 2월 충북대학교 정보통신공학과

1999년 2월 충북대학교 정보통신공학과 석사

2002년 8월 충북대학교 정보통신공학과 박사

2003년 3월~현재 인하공업전문대학 정보통신과 부교수

<관심분야> 무선인터넷, XML, 센서네트워크