(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2008-0060893
(43) 공개일자 2008년07월02일

(51) Int. Cl.
G06F 9/06 (2006.01) *G06K 17/00* (2006.01)
(21) 출원번호 10-2006-0135509
(22) 출원일자 2006년12월27일
심사청구일자 2006년12월27일

(71) 출원인
부산대학교 산학협력단
부산 금정구 장전동 산30 부산대학교 내
(72) 발명자
염근혁
부산 연제구 연산9동 망미주공 APT 117동 1403호
문미경
부산 해운대구 좌동 대우2차아파트 206동 801호
(뒷면에 계속)
(74) 대리인
문춘오, 오위환

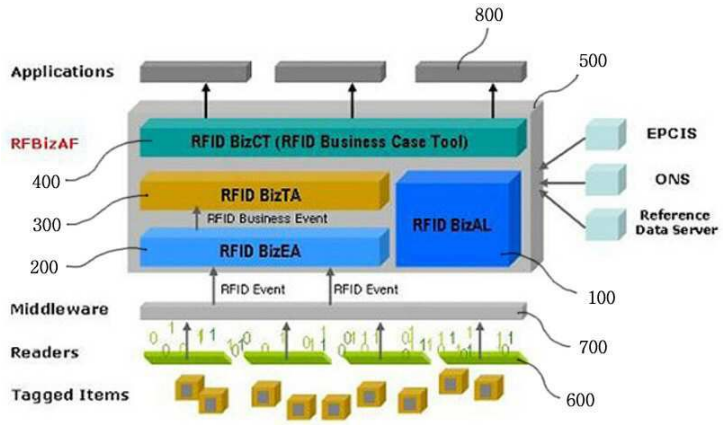
전체 청구항 수 : 총 15 항

(54) ＲＦＩＤ 어플리케이션 지원을 위한 미들웨어 시스템 및 그의 운영 방법

(57) 요 약

본 발명은 RFID 고유의 어플리케이션 개발을 지원하고 RFID 비즈니스 이벤트와 RFID 비즈니스 태스크의 용이한 모델링 프로세스를 지원하여 RFID 어플리케이션의 생산 및 관리의 효율성을 높인 RFID 어플리케이션 지원을 위한 미들웨어 플랫폼에 관한 것으로, RFID 미들웨어 계층을 갖는 미들웨어 시스템에 있어서,어플리케이션 계층(Application Layer)과 RFID 미들웨어 계층(RFID middleware Layer) 사이에 구성되어,상기 RFID 미들웨어 계층을 통하여 전달받는 RFID 이벤트를 참조 데이터 및 비즈니스 룰과 결합하여 RFID 응용 어플리케이션이 활용할 수 있는 RFID 비즈니스 이벤트를 정의하고, 다양한 속성의 입력을 지원하는 비주얼한 환경을 갖고 RFID 응용 어플리케이션이 필요로 하는 RFID 비즈니스 이벤트와 RFID 비즈니스 태스크의 모델링을 지원하는 RFID 비즈니스 어웨어 프레임워크(RFID BizAF)를 포함한다.

대표도 - 도1



(72) 발명자

김영봉

부산 해운대구 반여1동 송보자유APT 710호

유선미

부산 진구 양정동 현대아파트 201동 2503호

이찬영

부산 해운대구 반여1동 현대그린맨션 302동 209호

정민선

부산 금정구 장전동 383-19 2층

김성훈

부산 영도구 영선동 4가 반도보라 104동 2002호

김성진

부산 동래구 복천동 우성베스토피아 108동 702호

특허청구의 범위

청구항 1

RFID 미들웨어 계층을 갖는 미들웨어 시스템에 있어서,

어플리케이션 계층(Application Layer)과 RFID 미들웨어 계층(RFID middleware Layer) 사이에 구성되어,

상기 RFID 미들웨어 계층을 통하여 전달받는 RFID 이벤트를 참조 데이터 및 비즈니스 룰과 결합하여 RFID 응용 어플리케이션이 활용할 수 있는 RFID 비즈니스 이벤트를 정의하고, 다양한 속성의 입력을 지원하는 비주얼한 환경을 갖고 RFID 응용 어플리케이션이 필요로 하는 RFID 비즈니스 이벤트와 RFID 비즈니스 태스크의 모델링을 지원하는 RFID 비즈니스 어웨어 프레임워크(RFID BizAF)를 포함하는 것을 특징으로 하는 RFID 어플리케이션 지원을 위한 미들웨어 플랫폼.

청구항 2

RFID 미들웨어 계층을 갖는 미들웨어 시스템에 있어서,

어플리케이션 계층(Application Layer)과 RFID 미들웨어 계층(RFID middleware Layer) 사이에,

RFID 미들웨어 계층을 통하여 전달받는 RFID 이벤트를 참조 데이터 및 비즈니스 룰과 결합하여 RFID 응용 어플리케이션이 활용할 수 있는 RFID 비즈니스 이벤트를 정의할 수 있도록 지원하는 언어로 이루어진 RFID 비즈니스 어웨어 랭귀지(RFID BizAL) 블록;

정의된 RFID 비즈니스 이벤트의 생성을 에이전트 기반으로 관리하고, RFID 미들웨어 및 참조 데이터 서버와의 통신 기능을 제공하는 RFID 비즈니스 이벤트 어시스턴트(RFID BizEA) 블록;

미리 정의된 RFID 비즈니스 태스크의 실행을 에이전트 기반으로 관리하고 메일 전송 및 이벤트 저장 기능을 수행하는 RFID 비즈니스 태스크 어시스턴트(RFID BizTA) 블록;

비주얼한 환경을 갖고 상기 RFID 응용 어플리케이션이 필요로 하는 RFID 비즈니스 이벤트와 RFID 비즈니스 태스크를 쉽게 모델링 하도록 지원하는 RFID 비즈니스 케이스 툴(RFID BizCT) 블록;

을 포함하는 RFID 비즈니스 어웨어 프레임워크(RFID BizAF)가 구성되는 것을 특징으로 하는 RFID 어플리케이션 지원을 위한 미들웨어 플랫폼.

청구항 3

제 2 항에 있어서, RFID 비즈니스 어웨어 랭귀지(RFID BizAL) 블록은,

상기 RFID 비즈니스 이벤트와 RFID 비즈니스 태스크의 연결을 정의하고, RFID 기술의 특성을 반영하는 비즈니스 이벤트 모델을 제공하는 것을 특징으로 하는 RFID 어플리케이션 지원을 위한 미들웨어 플랫폼.

청구항 4

제 1 항 또는 제 2 항에 있어서, RFID 비즈니스 어웨어 프레임워크(RFID BizAF)는,

비즈니스 이벤트 해석 및 데이터 변환을 위하여 RFID BizAL 해석 모듈과,

RFID BizAL와 EC Spec 정의 간의 변환을 위해 필요한 번역 모듈과,

RFID BizAL에서 정의된 필요한 부분만을 추출하여 처리 과정에 전달하기 위한 EC Report 분석 모듈을 포함하는 것을 특징으로 하는 RFID 어플리케이션 지원을 위한 미들웨어 플랫폼.

청구항 5

제 2 항에 있어서, RFID 비즈니스 어웨어 랭귀지(RFID BizAL)는,

태그정보를 읽었을 때는 이미 정의된 비즈니스 이벤트를 정의하기 위한 스펙(BESpec)에 따라 독립적으로 수행되는 각 처리과정 사이에 필요한 정보를 넘겨주기 위한 매개체로써 사용되는 변수와,

상기 RFID 이벤트로부터 RFID 비즈니스 이벤트를 생성하기 위한 세부 액티비티 부분으로 구성되는 것을 특징으

로 하는 RFID 어플리케이션 지원을 위한 미들웨어 플랫폼.

청구항 6

제 5 항에 있어서, RFID 비즈니스 어웨어 랭귀지의 변수는,

일반 프로그래밍 언어에서 지원하는 데이터 타입으로 계산과정에 사용되거나 조건문 처리 또는 최종 이벤트 생성시의 데이터 저장에 사용되는 기본 타입과,

EPC, Tag, RawHex, RawDecimal과 같이 RFID 시스템에서 고유하게 사용되는 RFID 특화 타입과,

리더를 이용해 태그 정보를 읽을 때 범위 내에 있는 여러 태그들의 정보가 한번에 읽혀지는 경우 목록에 대한 처리 부하를 줄이기 위한 리스트 타입으로 구성되는 것을 특징으로 하는 RFID 어플리케이션 지원을 위한 미들웨어 플랫폼.

청구항 7

제 5 항에 있어서, RFID 비즈니스 어웨어 랭귀지의 액티비티들은,

ALE(Application Level Event)로부터 태그에 대한 정보를 획득하기 위한 ALE 액티비티와,

EPC에 대한 상세 정보를 획득하기 위한 EPCIS 액티비티와,

EPC에 대한 상세 정보를 저장한 EPCIS의 주소 획득하기 위한 ONS 액티비티와,

list 타입에 대한 반복 처리를 위한 List 액티비티와,

조건에 따른 이벤트를 생성하기 위한 Event 액티비티와,

산술 연산 처리를 위한 Compute 액티비티와,

프로세스 처리 중지를 위한 Terminate 액티티비와,

EPC의 이동에 대한 상세 정보를 획득하기 위한 EPCIS DS 액티비티를 포함하는 것을 특징으로 하는 RFID 어플리케이션 지원을 위한 미들웨어 플랫폼.

청구항 8

제 7 항에 있어서, ALE 액티비티는,

어플리케이션이 필요한 이벤트를 등록할 수 있도록 ECSpec을 제공하고 등록된 ECSpec에 대한 결과로 ECReport를 생성하여 전달하는 ALE 엔진에 어떠한 ECSpec을 생성하여 등록할지와 그 결과로 넘어온 ECReport를 어떻게 처리할지에 대하여 정의하고,

ECSpec은 내부에 태그정보를 읽을 리더의 목록, 태그를 읽는 시간을 설정하는 boundary, 어떤 태그를 읽을지를 설정하는 reportSpec을 포함하는 것을 특징으로 하는 RFID 어플리케이션 지원을 위한 미들웨어 플랫폼.

청구항 9

제 7 항에 있어서, EPCIS 액티비티는,

EPC에 있는 해당하는 제품의 상세 정보를 다른 시스템에서 이용 가능하도록 인터페이스를 제공하는 것을 특징으로 하는 RFID 어플리케이션 지원을 위한 미들웨어 플랫폼.

청구항 10

RFID 미들웨어 계층을 갖는 미들웨어 시스템에 있어서,

어플리케이션 계층(Application Layer)과 RFID 미들웨어 계층(RFID middleware Layer) 사이에,

현재 사용자의 컴퓨터에 있는 각 폴더의 파일들의 정보를 보여주기 위한 네비게이터(Navigator), BESpec을 XML의 형태로 보여주거나, GUI의 형태로 화면에 보여주는 것으로 각 형태에서 편집이 가능하도록 지원하는 메인뷰(MainView), 사용가능한 액티비티와 태스크의 목록을 보여주는 액티비티 뷰(ActivityView), 각 액티비티의 세부 속성값을 표현하는 프로퍼티 뷰(PropertyView), 에러 메시지와 알림 메시지를 표현하는 메시지 뷰(MessageVie

w)와 같은 창들을 구비하고, RFID 비즈니스 이벤트와 RFID 비즈니스 태스크의 모델링을 지원하는 RFID 비즈니스 케이스 툴(RFID BizCT);

을 포함하는 RFID 비즈니스 어웨어 프레임워크(RFID BizAF)가 구성되는 것을 특징으로 하는 RFID 어플리케이션 지원을 위한 미들웨어 플랫폼.

청구항 11

제 10 항에 있어서, RFID 비즈니스 케이스 툴(RFID BizCT)은,

리더 등록, 수정, 삭제를 위한 화면과, 새 파일을 생성하기 위한 화면과,

변수의 추가, 수정, 삭제를 위한 화면과, 액티비티의 생성, 액티비티의 이동, 액티비티의 삭제를 위한 화면과, 액티비티 속성을 설정하기 위한 화면과, 액티비티 복사를 위한 화면과, 뷰/소스 변환을 위한 화면을 제공하는 것을 특징으로 하는 RFID 어플리케이션 지원을 위한 미들웨어 플랫폼.

청구항 12

어플리케이션 계층과 RFID 미들웨어 계층 사이에 구성되는 RFID 비즈니스 어웨어 프레임워크(RFID BizAF)에 의해,

개발될 RFID 어플리케이션이 사용할 해당 리더의 비즈니스 문맥에 관계된 객체들을 추출하는 단계;

상기 추출한 객체들과 연관된 참조정보를 판별하는 단계;

상기 추출된 태그 부착 객체 및 이들과 연관된 참조 정보를 이용하여 어플리케이션의 동작에 필요한 RFID 이벤트와 관련된 비즈니스 규칙을 추출하여 기술하는 단계;

인식된 태그 부착 객체들간의 처리 흐름을 정의하고 결정된 논리적 리더들을 물리적 리더로 기술하는 단계와,

도출된 태그 부착 객체들의 비즈니스 규칙과 처리순서에 따라 BESpec으로 맵핑하여 기술하는 단계;를 수행하여 RFID 비즈니스 이벤트를 생성하는 것을 특징으로 하는 RFID 어플리케이션 지원을 위한 미들웨어 플랫폼.

청구항 13

제 12 항에 있어서, 해당 리더의 비즈니스 문맥에 관계된 객체들을 추출하는 단계를 수행하기 이전에,

개발될 RFID 어플리케이션이 사용할 논리적 리더의 개수와 배치 위치를 결정하는 단계;

추출된 리더의 비즈니스 문맥을 정의하는 단계;를 수행하는 것을 특징으로 하는 RFID 어플리케이션 지원을 위한 미들웨어 플랫폼.

청구항 14

제 12 항에 있어서, 도출된 태그 부착 객체들의 비즈니스 규칙과 처리순서에 따라 BESpec으로 맵핑하여 기술하는 단계에서,

BESpec을 RFID 비즈니스 어웨어 프레임워크(RFID BizAF)에 등록하면 RFID 비즈니스 어웨어 프레임워크(RFID BizAF)는 기술된 비즈니스 규칙조건에 부합할 때에 비즈니스 이벤트를 발생시켜 RFID 어플리케이션에 전달하는 것을 특징으로 하는 RFID 어플리케이션 지원을 위한 미들웨어 플랫폼.

청구항 15

제 12 항에 있어서, 논리적 리더들을 물리적 리더로 기술하는 단계는,

RFID 어플리케이션에서 리더에서 발생한 관심 있는 RFID 이벤트를 전달받기 위하여 리더가 동작하는데 필요한 물리적인 요소를 설정하고 결과로 받을 값에 대한 태그를 읽는 시간 간격, 태그의 비트수, 적용할 필터를 포함하는 정보를 알려주는 것임을 특징으로 하는 RFID 어플리케이션 지원을 위한 미들웨어 플랫폼.

명 세 서

발명의 상세한 설명

발명의 목적

발명이 속하는 기술 및 그 분야의 종래기술

<15> 본 발명은 RFID 미들웨어 플랫폼에 관한 것으로, 구체적으로 RFID 고유의 어플리케이션 개발을 지원하고 RFID 비즈니스 이벤트와 RFID 비즈니스 태스크의 용이한 모델링 프로세스를 지원하여 RFID 어플리케이션의 생산 및 관리의 효율성을 높인 RFID 어플리케이션 지원을 위한 미들웨어 플랫폼에 관한 것이다.

<16> 일반적으로 RFID 시스템은, 기본적으로 사물에 부착되고 이를 유일하게 구분할 수 있는 정보를 담고 있는 RFID 태그, 태그에 담겨진 정보를 인식하는 RFID 리더기, 리더기로부터 인식된 태그정보를 처리하는 호스트 시스템으로 구성된다.

<17> 호스트 시스템이란 RFID 리더기로부터 인식된 태그인식정보를 처리하고 응용 프로그램이 이를 활용하는 전반적인 소프트웨어 시스템을 지칭하며, 이중에서 RFID 미들웨어는 기존 응용 시스템과 RFID 하드웨어간의 연동을 지원하는 역할을 담당한다.

<18> RFID 미들웨어는 그 특성상 지속적인 서비스가 어려운 환경에서 수행된다.

<19> RFID 미들웨어는 데이터 센터와 같은 안전한 장소에서 수행되고 숙련된 운영자가 존재하지 않는다. 즉, RFID 미들웨어는 공장, 창고, 매장 등과 같이 컴퓨터 시스템을 원활하게 수행할 수 있는 별도의 환경이 아닌 일반 사용자와 각종 주변 기계에 노출된 환경에서 수행된다. 따라서 RFID 미들웨어 시스템을 구성하는 리더(Reader)와 엣지매니저(EdgeManager)에 고장이 발생할 가능성이 높다.

<20> 먼저, 리더의 경우는 태그에서의 정보를 직접적으로 읽는 시스템이므로 현장에서 외부 환경에 완전히 노출되어 있을 수 있다. 예를 들어 짐을 운반하는 도중, 다른 차량이나 컨테이너의 사고로 리더 시스템에 치명적인 손상을 줄 수도 있다.

<21> 리더가 장애를 일으킨 사실은 여러 방법을 통해서 체크 할 수 있다.

<22> 장애를 일으킨 리더가 발견이 된 경우 해당 리더의 일을 대신해 줄 수 있는 방법이 없으므로, 시스템 자체에 큰 문제가 없다면 시스템을 다시 시작하는 방법도 있을 수 있다.

<23> 만약, 시스템 재부팅을 통해서도 해결을 할 수가 없다면 담당 기술자나 관리자에게 빠른 시간 내에 알려주어서 시스템을 복구할 수 있도록 해야 한다.

<24> 그리고 예를 들어, 창고에 있는 엣지매니저를 지게차가 실수로 충격을 주어서 엣지매니저에 고장이 발생할 수도 있는데, 이때 RFID 미들웨어 시스템이 지속적으로 서비스를 하기 위해서는 다른 엣지매니저가 고장난 엣지매니저가 관리하던 리더에 대한 처리를 대신하는 것이 요구된다.

<25> 또한, RFID 미들웨어가 수행되는 환경에서는 미들웨어를 전문적으로 관리할 운영자가 존재하지 않을 가능성이 있다. 따라서, 엣지매니저의 고장시에 해당 리더들에 대한 처리를 다른 엣지매니저로 위임하는 작업을 최대한 운영자의 개입이 없이 시스템에서 자동으로 제공하는 것이 바람직하다.

<26> 그러나 종래 기술의 RFID 미들웨어 시스템서는 이러한 문제 발생에 대한 효율적인 대처가 이루어지지 않는다.

<27> 그리고 RFID 기술의 개발 및 표준화 작업은 사용자들에게 좀더 편리하고 유익한 방향으로 나아가는 것을 궁극의 목표로 하고 있다. 기술 개발과 더불어 이 기술이 비즈니스 영역에 어떻게 접목될지, 적용 단계에서 발생하는 여러 문제점들은 무엇이며, 이를 어떻게 해결해 나갈지에 대한 실질적인 방안이 중요하다.

<28> 이를 위해 다양한 분야에서 RFID를 이용한 각종 어플리케이션의 실증실험이나 실용화가 진행되고 있다.

<29> 이러한 어플리케이션을 위한 RFID 미들웨어는 리더에서 발생하는 RFID 데이터에 대하여 필터링, 집합화, 이벤트 생성 등의 기능을 통하여 어플리케이션에서 관심을 가지는 데이터 형태로의 선처리 기능을 수행한다.

<30> 그러나 RFID 미들웨어가 항상 어플리케이션이 필요로 하는 형태로 이벤트를 전달하는 것은 아니다. 이를 해결하기 위해, RFID 미들웨어로부터 인식된 RFID 이벤트를 가공하여 어플리케이션이 필요로 하는 고수준 이벤트의 형태로 변환하는 기능이 필요하다.

<31> 현재 대부분의 RFID 미들웨어는 인식된 RFID 이벤트를 가공하지 않고 어플리케이션으로 전달하는 기본적인 기능만을 수행하고 있고, RFID 이벤트 외에도 센서 데이터를 인식하여 어플리케이션으로 전달하는 기능을 부가적으

로 수행하는 RFID 미들웨어도 있다.

<32> 그러나 이벤트를 조합하여 응용 프로그램이 활용할 수 있는 고수준의 형태로 표현하는 기초적인 기능을 제공하는 RFID 미들웨어도 일부 있지만, 대부분의 RFID 미들웨어의 경우 어플리케이션 개발자가 직접 RFID 미들웨어로부터 수집된 이벤트를 이용하는 로직을 개발해야하는 문제가 있다.

<33> 이와 같은 RFID 어플리케이션 개발에 관하여 더 설명한다.

<34> RFID 기술을 시스템에 적용하기 위해서는 RFID의 특성을 고려하여 어플리케이션을 개발해야 한다.

<35> RFID 환경에서는 대량의 태그 데이터가 수집되어 끊임없이 연속적으로 들어오게 되는데 RFID 어플리케이션은 이러한 데이터에서 필요한 정보를 분류하고 이를 실시간으로 효율적으로 처리해야 한다. 짧은 시간 내에 발생하는 대량의 RFID 이벤트들을 효율적으로 처리하기 위해 RFID 미들웨어들이 개발되고 있다.

<36> 그러나 RFID 미들웨어를 통해 나오는 RFID 이벤트의 값은 아직도 단순한 태그 값이기 때문에 RFID 어플리케이션은 이 값들을 받아 그들의 어플리케이션에서 의미 있는 값으로 변환시켜야 한다. 이를 위해 RFID 이벤트를 분석하고 각각의 태그 값에 대하여 EPCIS(EPC Information Service), ONS(Object Name Service) 등의 외부 정보서버들과 통신을 하여 이들에 해당하는 속성 값들을 가져와야 한다.

<37> RFID 어플리케이션 개발자는 이러한 처리를 위해 여러 외부 시스템들에 대한 인터페이스를 모두 알고 있어야 하며, 통신 프로토콜, 멀티 쓰레드 처리를 위한 프로그래밍 관련 지식을 익혀야 한다.

<38> 따라서, 이전의 RFID 미들웨어에서는 효과적인 RFID 응용 어플리케이션 개발 환경을 제공이 되지 못하고, RFID 미들웨어 시스템에서 요구되는 효율적인 문제 대처 능력과 관리가 어렵다.

발명이 이루고자 하는 기술적 과제

<39> 본 발명은 이와 같은 문제를 해결하여 RFID 미들웨어 시스템에서 요구되는 효율적인 문제 대처 능력과 관리 기능 그리고 효과적인 RFID 응용 어플리케이션 개발 환경을 제공하기 위한 것으로, RFID 미들웨어 플랫폼을 새롭게 구현하여 RFID 고유의 어플리케이션 개발을 지원하고 RFID 비즈니스 이벤트와 RFID 비즈니스 태스크의 용이한 모델링 프로세스를 지원하여 RFID 어플리케이션의 생산 및 관리의 효율성을 높인 RFID 어플리케이션 지원을 위한 미들웨어 플랫폼을 제공하는데 그 목적이 있다.

<40> 본 발명은 RFIID BizAL로 기술된 RFID 비즈니스 명세를 처리하여 정의된 RFID 비즈니스 이벤트가 발생하면 이를 요청한 RFID 응용 애플리케이션에게 전달하거나, 또는 사전에 정의된 태스크를 수행하는 프레임워크(RFID BizAF)를 제공하는데 그 목적이 있다.

<41> 본 발명은 RFID BizAF를 활용하여 RFID 미들웨어를 통해 제공되는 EPC를 가공한 보다 상위 수준의 RFID 비즈니스 이벤트 이용이 가능하게 하여 보다 효율적인 비즈니스 로직의 정의가 가능하도록한 RFID 어플리케이션 지원을 위한 미들웨어 플랫폼을 제공하는데 그 목적이 있다.

<42> 본 발명은 RFID 비즈니스 어웨어 프레임워크 블록(RFID BizAF)을 이용하여 어플리케이션 개발을 위한 비즈니스 이벤트를 추출이 효율적으로 이루어지도록 하기 위한 새로운 RFID 비즈니스 이벤트 생성 모델링 프로세스를 제공하는데 그 목적이 있다.

발명의 구성 및 작용

<43> 이와 같은 목적을 달성하기 위한 본 발명에 따른 RFID 어플리케이션 지원을 위한 미들웨어 플랫폼은 RFID 미들웨어 계층을 갖는 미들웨어 시스템에 있어서, 어플리케이션 계층(Application Layer)과 RFID 미들웨어 계층(RFID middleware Layer) 사이에 구성되어, 상기 RFID 미들웨어 계층을 통하여 전달받는 RFID 이벤트를 참조 데이터 및 비즈니스 룰과 결합하여 RFID 응용 어플리케이션이 활용할 수 있는 RFID 비즈니스 이벤트를 정의하고, 다양한 속성의 입력을 지원하는 비주얼한 환경을 갖고 RFID 응용 어플리케이션이 필요로 하는 RFID 비즈니스 이벤트와 RFID 비즈니스 태스크의 모델링을 지원하는 RFID 비즈니스 어웨어 프레임워크(RFID BizAF)를 포함하는 것을 특징으로 한다.

<44> 다른 목적을 달성하기 위한 본 발명에 따른 RFID 어플리케이션 지원을 위한 미들웨어 플랫폼은 RFID 미들웨어 계층을 갖는 미들웨어 시스템에 있어서, 어플리케이션 계층(Application Layer)과 RFID 미들웨어 계층(RFID middleware Layer) 사이에, RFID 미들웨어 계층을 통하여 전달받는 RFID 이벤트를 참조 데이터 및 비즈니스 룰과 결합하여 RFID 응용 어플리케이션이 활용할 수 있는 RFID 비즈니스 이벤트를 정의할 수 있도록 지원하는 언

어로 이루어진 RFID 비즈니스 어웨어 랭귀지(RFID BizAL) 블록;정의된 RFID 비즈니스 이벤트의 생성을 에이전트 기반으로 관리하고, RFID 미들웨어 및 참조 데이터 서버와의 통신 기능을 제공하는 RFID 비즈니스 이벤트 어시스턴트(RFID BizEA) 블록;미리 정의된 RFID 비즈니스 태스크의 실행을 에이전트 기반으로 관리하고 메일 전송 및 이벤트 저장 기능을 수행하는 RFID 비즈니스 태스크 어시스턴트(RFID BizTA) 블록;비주얼한 환경을 갖고 상기 RFID 응용 어플리케이션이 필요로 하는 RFID 비즈니스 이벤트와 RFID 비즈니스 태스크를 쉽게 모델링 하도록 지원하는 RFID 비즈니스 케이스 툴(RFID BizCT) 블록;을 포함하는 RFID 비즈니스 어웨어 프레임워크(RFID BizAF)가 구성되는 것을 특징으로 한다.

<45> 또 다른 목적을 달성하기 위한 본 발명에 따른 RFID 어플리케이션 지원을 위한 미들웨어 플랫폼은 RFID 미들웨어 계층을 갖는 미들웨어 시스템에 있어서, 어플리케이션 계층(Application Layer)과 RFID 미들웨어 계층(RFID middleware Layer) 사이에, 현재 사용자의 컴퓨터에 있는 각 폴더의 파일들의 정보를 보여주기 위한 네비게이터(Navigator), BESpec을 XML의 형태로 보여주거나, GUI의 형태로 화면에 보여주는 것으로 각 형태에서 편집이 가능하도록 지원하는 메인뷰(MainView), 사용가능한 액티비티와 태스크의 목록을 보여주는 액티비티 뷰(ActivityView), 각 액티비티의 세부 속성값을 표현하는 프로퍼티 뷰(PropertyView), 에러 메시지와 알림 메시지를 표현하는 메시지 뷰(MessageView)와 같은 창들을 구비하고, RFID 비즈니스 이벤트와 RFID 비즈니스 태스크의 모델링을 지원하는 RFID 비즈니스 케이스 툴(RFID BizCT);을 포함하는 RFID 비즈니스 어웨어 프레임워크(RFID BizAF)가 구성되는 것을 특징으로 한다.

<46> 또 다른 목적을 달성하기 위한 본 발명에 따른 RFID 어플리케이션 지원을 위한 미들웨어 플랫폼은 어플리케이션 계층과 RFID 미들웨어 계층 사이에 구성되는 RFID 비즈니스 어웨어 프레임워크(RFID BizAF)에 의해, 개발될 RFID 어플리케이션이 사용할 해당 리더의 비즈니스 문맥에 관계된 객체들을 추출하는 단계;상기 추출한 객체들과 연관된 참조정보를 판별하는 단계;상기 추출된 태그 부착 객체 및 이들과 연관된 참조 정보를 이용하여 어플리케이션의 동작에 필요한 RFID 이벤트와 관련된 비즈니스 규칙을 추출하여 기술하는 단계;인식된 태그 부착 객체들간의 처리 흐름을 정의하고 결정된 논리적 리더들을 물리적 리더로 기술하는 단계와, 도출된 태그 부착 객체들의 비즈니스 규칙과 처리순서에 따라 BESpec으로 맵핑하여 기술하는 단계;를 수행하여 RFID 비즈니스 이벤트를 생성하는 것을 특징으로 한다.

<47> 이하, 본 발명에 따른 RFID 어플리케이션 지원을 위한 미들웨어 플랫폼의 바람직한 실시예에 관하여 상세히 설명하면 다음과 같다.

<48> 본 발명에 따른 RFID 어플리케이션 지원을 위한 미들웨어 플랫폼의 특징 및 이점들은 이하에서의 각 실시예에 대한 상세한 설명을 통해 명백해질 것이다.

<49> 도 1은 본 발명에 따른 다양한 어플리케이션 지원을 위한 RFID 미들웨어 플랫폼의 구성도이다.

<50> 본 발명은 RFID 비즈니스 어웨어 프레임워크(RFID Business-Aware Framework;RFID BizAF)를 통하여 RFID 이벤트뿐만 아니라 외부 정보 서버로부터 공급되는 다양한 참조 데이터를 함께 조합하여 가공함으로써 보다 풍부한 고수준 비즈니스 이벤트를 생성할 수 있도록한 것이다.

<51> 그리고 어플리케이션의 유지보수 비용과 RFID 응용 어플리케이션을 개발하는 시간을 감소시키기 위해서는 RFID 기술에 대한 깊은 지식과 의존을 필요로 하지 않고도 RFID 응용 어플리케이션을 개발할 수 있는 효과적인 환경이 필요하다.

<52> 본 발명에 따르면, 어플리케이션 개발자는 기존에는 직접 코드로 작성하던 여러 처리 과정을 본 발명에 따른 RFID 비즈니스 어웨어 랭귀지(RFID Business-Aware Language;RFID BizAL)를 이용하여 보다 편리하게 정의할 수 있다.

<53> 정의된 비즈니스 이벤트는 RFID 비즈니스 어웨어 프레임워크를 통해 처리되어 결과적으로 개발자는 어플리케이션에 필요한 정보만을 미들웨어로부터 전달받아 이를 이용한 어플리케이션을 개발할 수 있도록 한다.

<54> 이와 같이, 본 발명에 따른 RFID 어플리케이션 지원을 위한 미들웨어 플랫폼을 구성하는 RFID BizAL과 RFID BizAF는 효과적인 RFID 응용 어플리케이션 개발 환경을 제공할 수 있다.

<55> 예를 들어, 특정 도메인에 속하는 서로 다른 어플리케이션 간에는 공통되는 규칙이 존재하지만, 각 어플리케이션에서 규칙을 정의하는 방법이 서로 다르기 때문에 공통된 규칙의 중복 개발이 존재하게 되고, 어플리케이션 간의 규칙 교환을 위해서는 각 어플리케이션에 적합한 형태로 규칙을 커스터마이징 과정이 추가적으로 필요하다. 이를 해결하기 위하여 본 발명에 따른 RFID 어플리케이션 지원을 위한 미들웨어 플랫폼에서는 단일의

RFID BizAL을 사용함으로써 특정 도메인에 속하는 서로 다른 어플리케이션 간 규칙의 재사용성을 제공한다.

<56> 본 발명에 따른 RFID 어플리케이션 지원을 위한 미들웨어 플랫폼은 도 1에서와 같이, 어플리케이션 계층(Application Layer)(800)과 RFID 미들웨어 계층(RFID middleware Layer)(700) 사이에 RFID 비즈니스 어웨어 프레임워크(RFID Business-Aware Framework;RFID BizAF)(500)를 구성하여 RFID 이벤트뿐만 아니라 외부 정보 서버로부터 공급되는 다양한 참조 데이터를 함께 조합하여 가공함으로써 보다 풍부한 고수준의 RFID 비즈니스 이벤트를 생성할 수 있도록한 것이다.

<57> RFID 비즈니스 어웨어 프레임워크(RFID Business-Aware Framework;RFID BizAF)(500)는 어플리케이션에 라이브러리 형태로 추가된다.

<58> RFID 비즈니스 어웨어 프레임워크(RFID Business-Aware Framework;RFID BizAF)(500)의 구성은 다음과 같다.

<59> 상기 RFID 미들웨어 계층(700)을 통하여 리더스 계층(600)으로부터 전달받는 저수준의 RFID 이벤트를 참조 데이터 및 비즈니스 룰과 결합하여 RFID 응용 어플리케이션이 활용할 수 있는 고수준의 RFID 비즈니스 이벤트를 정의할 수 있도록 지원하는 언어로 이루어져 RFID 비즈니스 이벤트와 RFID 비즈니스 태스크의 연결을 정의하고, RFID 기술의 특성을 반영하는 비즈니스 이벤트 모델을 제공하는 RFID 비즈니스 어웨어 랭귀지(RFID Business-Aware Language;RFID BizAL) 블록(100)과, 사전에 정의된 RFID 비즈니스 이벤트의 생성을 에이전트 기반으로 관리하고, RFID 미들웨어 및 참조 데이터 서버와의 통신 기능을 제공함으로써 RFID 응용 어플리케이션을 편리하게 개발할 수 있도록 하는 RFID 비즈니스 이벤트 어시스턴트(RFID Business Event Assistant;RFID BizEA) 블록(200)과, RFID 비즈니스 이벤트 어시스턴트 블록(200)의 RFID 비즈니스 이벤트를 받아 미리 정의된 RFID 비즈니스 태스크의 실행을 에이전트 기반으로 관리하고 메일 전송 및 이벤트 저장 기능을 수행하는 RFID 비즈니스 태스크 어시스턴트(RFID Business Task Assistant; RFID BizTA) 블록(300)과, 다양한 속성의 입력을 지원하는 비주얼한 환경을 갖고 RFID 응용 어플리케이션이 필요로 하는 RFID 비즈니스 이벤트와 RFID 비즈니스 태스크를 쉽게 모델링 하도록 지원하는 RFID 비즈니스 케이스 툴(RFID Business Case Tool;RFID BizCT) 블록(500)으로 구성된다.

<60> 여기서, RFID 미들웨어 계층(RFID middleware Layer)(700)에서 어플리케이션 계층(Application Layer)(800)으로 전달하는 RFID 이벤트는 숫자로 이루어진 EPC 등의 형태를 갖는 것으로 단순히 헤더 정보, 업체 코드, 상품 코드, 일련 번호로만 이루어져 있기 때문에 코드와 관련된 상세 정보는 알 수 없다.

<61> 그리고 RFID 비즈니스 이벤트 어시스턴트(RFID Business Event Assistant;RFID BizEA) 블록(200)에 의해 생성 관리되는 RFID 비즈니스 이벤트는 비즈니스 로직에 바로 사용될 수 있는 것으로 이벤트를 전달할 때 관련 데이터를 같이 전달하므로 어플리케이션이 바로 어떤 의미인지를 알 수 있다.

<62> 이상에서 설명한 본 발명에 따른 RFID 비즈니스 어웨어 프레임워크(RFID Business-Aware Framework;RFID BizAF)(500)는 비즈니스 이벤트 해석 및 데이터 변환을 위하여 RFID BizAL 해석 모듈과, RFID BizAL와 EC Spec 정의 간의 변환을 위해 필요한 번역 모듈과, EC Report 분석 모듈을 포함한다.

<63> RFID 비즈니스 이벤트(Business Event)를 생성하기 위한 로직을 RFID BizAL을 이용하여 RFID 비즈니스 어웨어 프레임워크(RFID Business-Aware Framework;RFID BizAF)(500)에 전달하면 이 작성 내용을 읽어서 어떤 내용인지 분석을 할 필요가 있다.

<64> 즉, RFID BizAL내에는 많은 요소가 있기 때문에 각각의 요소를 분류하고 요소별 처리 과정을 연결할 필요가 있는데, 이를 위해서 RFID BizAL을 해석하는 모듈을 구비한다.

<65> RFID BizAL 해석 모듈은 가장 기본적인 ALE Spec을 생성하기 위한 정보를 비즈니스 이벤트로부터 식별하고, 이후 추가 처리 과정이 어떤 것이 있는지 따로 분류하여 하나의 비즈니스 이벤트 처리 프로세스가 활성화되면 해당 정보를 이용할 수 있도록 한다.

<66> 그리고 다음과 같은 기능을 수행하기 위하여 ALE 매니저와의 정보 교환을 위한 번역 모듈을 구비한다.

<67> RFID BizAF(RFID Business-Aware Framework)와 연동되어 기능을 수행하는 하위 계층은 ALE 매니저(Application Level Event Manager)이다.

<68> ALE 매니저는 EPC Global에서 표준화한 인터페이스를 이용하여 정보를 주고받는데 RFID BizAF에서 필요한 정보를 받기 위해서는 이러한 표준화된 인터페이스를 이용해야 한다. 해당 인터페이스 표준에서는 ECSpec(Event Cycle Specification)과 ECReport가 있어서 ECSpec은 필요한 정보를 정의하는 것이고 ECReport에는 실제 리더

로부터 읽혀진 정보가 있다.

<69> RFID BizAF에서는 BED에 정의된 내용을 해석하여 필요한 정보를 EC Spec형태로 변경하여 ALE 매니저에 전달해야 하는데, 이를 위하여 RFID BizAL와 ECSpec정의 간의 변환을 위해 번역 모듈을 구비한다.

<70> 또한, 결과로 넘어온 ECReport도 RFID BizAL(RFID Business-Aware Language)에서 정의된 필요한 부분만을 추출하여 처리 과정에 전달하기 위한 EC Report 분석 모듈을 구비한다.

<71> 그리고 본 발명에 따른 RFID 비즈니스 어웨어 프레임워크(RFID Business-Aware Framework;RFID BizAF)(500)는 외부 어플리케이션과의 통신을 위하여 통신 모듈을 구비한다. 즉, ECSpec과 ECReport는 ALE 매니저의 정보 형태이기 때문에 실제 통신을 위해서는 해당 통신 프로토콜에 맞는 ALE 매니저와의 통신 모듈을 구비한다.

<72> 또한, EPCIS의 데이터를 이용하기 위해서는 해당 정보 공급처에서 사용하는 통신 프로토콜에 따라서 통신을 해야 하고 정보도 해당 인터페이스를 이용하여 주고받게 된다. 이를 위해 외부 요소의 정보 형태와 프로토콜에 따른 액세스 모듈을 구비하고, 이 액세스 모듈은 비즈니스 이벤트 처리 과정의 특정 처리 부분과 연동되어서 기능을 수행하게 된다.

<73> 먼저, 본 발명에 따른 RFID 어플리케이션 지원을 위한 미들웨어 플랫폼을 구성하는 RFID 비즈니스 어웨어 랭귀지(RFID Business-Aware Language;RFID BizAL) 블록(100)에 관하여 상세히 설명한다.

<74> RFID 비즈니스 어웨어 랭귀지(RFID Business-Aware Language;RFID BizAL) 블록(100)(BESpec)은 변수 부분과 처리 과정을 정의하는 액티비티 부분으로 구성된다.

<75> 변수 부분에서는 이벤트 관련 데이터 전달 또는 중간 처리과정에서 사용되는 변수를 정의하는 것으로, 변수에는 일반적인 타입 이외에 RFID 고유의 타입이 지원된다.

<76> 액티비티 부분에서는 RFID 이벤트로부터 RFID 비즈니스 이벤트를 생성하기 위한 세부 액티비티들을 정의한다.

<77> 먼저, RFID 비즈니스 어웨어 랭귀지의 변수 부분에 관하여 설명한다.

<78> 태그정보를 읽었을 때는 이미 정의된 비즈니스 이벤트를 정의하기 위한 스펙(Business Event Specification;BESpec)에 따라 여러 처리 과정을 거친다.

<79> 각각의 처리과정은 독립적으로 수행되는데 각 처리과정 사이에 필요한 정보를 넘겨주기 위한 매개체로써 변수를 사용하는데, 각 처리과정에서는 처리 결과를 변수에 저장하거나 입력으로 변수를 사용한다.

<80> 변수는 int등의 일반적인 타입뿐만 아니라 RFID 시스템에 특화된 타입도 지원하여 BESpec의 사용 편이성을 높인다. 예를 들면, ALE 엔진에서 한 이벤트 사이클 동안 넘겨주는 태그 목록들을 한 번에 저장할 수 있는 태그리스트 같은 데이터 타입을 제공하여 BESpec 작성 시 편의성과 가독성을 높일 수 있다.

<81> RFID 비즈니스 어웨어 랭귀지의 변수는 기본 타입과 RFID 특화 타입 그리고 리스트 타입을 구분한다.

<82> 먼저, 기본 타입은 일반적인 프로그래밍 언어에서 지원하는 데이터 타입을 지원하는 것으로, 이러한 데이터 타입은 계산과정에 사용되거나 조건문 처리 또는 최종 이벤트 생성 시 데이터 저장에 사용된다. 지원되는 데이터 타입은 int, string형이 있고 데이터 타입간의 연산은 타입별로 다르게 처리된다.

<83> 그리고 RFID 특화 타입은 RFID 시스템에만 사용되는 고유의 타입으로, RFID 시스템에서는 EPC, Tag, RawHex, RawDecimal과 같은 고유하게 사용되는 데이터들이 있는데 이러한 데이터들을 string 타입 등으로 처리하지 않고 고유의 타입을 지원함으로써 가독성과 사용편의성 등을 향상시킨다.

<84> EPC는 코드 전체를 사용할 경우도 있고 코드 중 일부분만을 사용할 경우도 있다. 이러한 특성을 지원하고자 EPC 데이터 타입은 코드 전체를 참조하는 방법과 내부의 부분요소를 사용할 수 있는 방법도 제공한다.

<85> 개발자는 필요에 따라서 EPC데이터 타입을 바로 사용하거나 부분 접근 방법을 사용하여 BESpec을 쉽게 정의할 수 있다. 예를 들어 개별 상품에 대한 생산 년월일을 알고자할 때는 EPC 전체가 필요하므로 변수를 바로 사용할 수 있고, 특정 상품 종류의 이름, 스펙 등에 대해 알고자할 때는 해당 EPC의 클래스 코드만 필요하므로 이때는 'EPC.class' 와 같은 방법을 이용하여 코드중의 일부분에 쉽게 접근할 수 있다.

<86> 그리고 리스트 타입은 RFID의 특성상 리더를 이용해 태그 정보를 읽을 때 범위 내에 있는 여러 태그들이 한꺼번에 읽혀서 목록으로 넘어오기 때문에 데이터 처리 시 단일 태그나 코드에 대한 처리보다는 목록에 대한 처리가 더 많이 발생하는데, 이러한 특성에 따라 데이터 처리의 편의성을 위한 것이다.

<87> 이와 같은 리스트 타입은 정의된 모든 기본 타입과 RFID 특화 타입에 대하여 확장 형태로 사용한다. 다른 타입을 선언할 때 리스트 옵션을 설정하면 해당 타입의 리스트가 생성되어 사용할 수 있다.

<88> 그리고 RFID 비즈니스 어웨어 랭귀지의 액티비티 부분에 관하여 설명한다.

<89> 비즈니스 이벤트를 생성하기 위해서는 ALE에서 리더로부터 읽은 정보를 가져오고 EPCIS나 ONS와도 정보를 주고받아야 한다. 이러한 행동들은 서로 간섭 없이 독립적으로 수행할 수 있는데, 본 발명은 이러한 독립적인 처리에 대한 정의를 할 수 있도록 액티비티(Activity)를 다음과 같이 정의한다.

<90> 액티비티는 기본적으로 ALE, EPCIS나 ONS 등과의 상호작용에 대한 정의를 하는 것이다. 외부 시스템과의 상호작용에 대한 정의 외에도 내부적인 처리 및 이벤트 전달에 대한 정의를 하기 위한 추가적인 액티비티가 있다.

<91> 이러한 액티비티를 일련의 처리 순서대로 배치하여 비즈니스 이벤트를 생성한다.

<92> RFID 비즈니스 어웨어 랭귀지 이벤트 발생과 관련된 액티비티의 종류는 표 1과 같다. 물론, 이외에 이벤트 발생 결과 수행해야 하는 태스크의 액티비티도 존재한다.

표 1

<93>

종류	이름	설 명
Primitive	ALE	ALE로부터 태그에 대한 정보를 획득
Primitive	EPCIS	EPC에 대한 상세 정보를 획득
Primitive	ONS	EPC에 대한 상세 정보를 저장한 EPCIS의 주소 획득
Contain	List	list 타입에 대한 반복처리
Contain	Event	조건에 따라 이벤트 생성
Primitive	Compute	산술 연산 처리
Primitive	Terminate	프로세스 처리 중지

<94> 액티비티는 종류에 따라서 다른 액티비티를 내부에 가질 수 있는 액티비티(Contain Activity)와 가질 수 없는 액티비티(Primitive Activity)로 구분된다.

<95> 예를 들면, 도 2에서와 같이 list 액티비티는 내부에 compute 액티비티나 EPCIS 액티비티 등의 다른 액티비티를 포함할 수 있다.

<96> 도 2는 본 발명에 따른 RFID 비즈니스 어웨어 랭귀지의 액티비티들간의 관계를 나타낸 구성도이다.

<97> 이와 같은 RFID 비즈니스 어웨어 랭귀지의 액티비티 부분에 관하여 구체적으로 설명한다.

<98> 도 3a내지 도 3g는 본 발명에 따른 RFID 비즈니스 어웨어 랭귀지의 액티비티들의 DTD를 나타낸 구성도이다.

<99> 첫째, ALE 액티비티는 ALE 엔진과의 통신에 대하여 정의하는 것으로, ALE 엔진은 어플리케이션이 필요한 이벤트를 등록할 수 있도록 ECSpec을 제공하고 등록된 ECSpec에 대한 결과로 ECReport를 생성하여 전달한다.

<100> ALE 액티비티에서는 ALE 엔진에 어떠한 ECSpec을 생성하여 등록할지와 그 결과로 넘어온 ECReport를 어떻게 처리할지에 대하여 정의한다. ECSpec은 내부에 태그정보를 읽을 리더의 목록, 태그를 읽는 시간을 설정하는 boundary, 어떤 태그를 읽을지를 설정하는 reportSpec등으로 구성되는데, ECSpec의 전체 구성은 도 3a에서와 같다.

<101> ECSpec의 정의는 개발자가 필요로 하는 것들을 정의한 것이므로 BESpec에서도 해당 기능을 직접 조절할 수 있도록 생략하지 않고 제공한다. 하지만 약간의 편의와 결과에 대한 처리를 위하여 약간의 수정을 하는데 공통적으로 쓰일 수 있는 부분은 생략이 가능하고 나머지 부분을 개발자가 설정한다.

<102> 그리고 ECSpec의 결과로 ECReport가 넘어오게 되면 이것을 처리하기 위하여 output부분을 확장한다. 각각의 그룹별로 리포트에 포함할 데이터를 설정하게 되는데 이렇게 넘어온 결과를 리스트 타입의 변수에 설정할 수 있도록 한다. 이 후의 액티비티에서는 ECReport에 있는 데이터를 확인하고자 할 때 이 변수에 설정된 데이터를 이용하면 된다.

<103> 세부적인 문법은 EPCglobal의 ALE 인터페이스에 대한 표준 문서를 따르고 확장된 부분은 ALE 액티비티의 DTD를 나타낸 도 3b에서와 같다.

<104> 둘째, EPCIS 액티비티는 EPC에 해당하는 제품의 상세 정보를 저장하고 있는 것으로, EPCIS는 이러한 정보를 다른 시스템에서 이용 가능하도록 인터페이스를 제공한다.

<105> BEF는 비즈니스 이벤트를 생성하여 넘겨주는데 이러한 과정에서는 태그로부터 읽은 코드에 대한 상세정보를 얻어야할 경우가 있다. 하지만 처리 후의 저장과정은 비즈니스 로직을 수행한 후에 하므로 EPCIS에서 제공하는 여러 기능 중에서 저장이나 수정과 관련된 기능은 지원하지 않고 조회와 관련된 기능만을 이용 가능하도록 한다.

<106> 그리고 이벤트를 생성하기 위해서는 EPC나 EPC 클래스 둘 다 이용할 수 있으므로 모두 지원한다. 이에 따라 EPCIS 액티비티에서는 표 2와 같은 인터페이스를 이용할 수 있도록 제공하고 EPCIS 액티비티의 XML 문법을 위한 DTD는 도 3c에서와 같다.

표 2

<107>

Operation	설 명
getAttributeDataEPC	EPC에 대한 속성 정보 조회
getSchemasOfEPC	EPCdp 대한 스키마 정보 조회
getAttributeDataOfEPCClassCode	EPC Class에 대한 속성 정보 조회
getSchemasOfEPCClassCode	EPC Class에 대한 스키마 정보 조회

<108> 셋째, ONS(Object Name Service)는 EPC에 해당하는 제품의 상세 정보가 어떤 EPCIS(EPC Information Service)에 있는지 알려주는 시스템으로, ONS에서 제공하는 인터페이스는 표 3과 같다.

<109> ONS 액티비티에서는 이벤트에 해당하는 제품의 상세 정보가 저장된 EPCIS의 주소에 대한 조회만이 필요하므로 searchEPCIS만을 정의에 제공하고, 정의된 XML 문법을 DTD로 표현하면 도 3d에서와 같다.

표 3

<110>

Operation	설 명
serchEPCIS	EPC정보를 가지고 있는 EPCIS의 주소를 반환한다.
insertEPCIS	EPC 정보를 가지고 있는 EPCIS의 주소를 저장한다.
deleteEPCIS	EPC 정보를 가지고 있는 EPCIS의 주소를 삭제한다.

<111> 넷째, List 액티비티는 다음과 같은 작업을 수행하기 위하여 제공된다.

<112> RFID 시스템의 특성상 리더로부터 읽혀진 결과가 목록으로 넘어오는데, 이러한 목록을 편리하게 처리할 수 있도록 BEF에서는 리스트 타입을 제공한다.

<113> 하지만 리스트에 저장된 데이터에 대하여 개별적인 작업을 수행해야할 경우가 있다. 이러한 경우에 리스트에 저장된 각각의 항목에 대하여 다른 처리를 할 수 있도록 List 액티비티가 제공된다.

<114> List 액티비티는 입력으로 List 타입의 변수를 받게 되고 내부에는 다른 액티비티의 목록이 포함된다. 실행 시에는 입력 List 변수에 설정된 데이터의 수만큼 반복해서 작업을 수행한다.

<115> List 액티비티의 XML 문법을 위한 DTD는 도 3e에서와 같다.

<116> List 액티비티는 list 태그로 시작하고 속성으로 두 가지 변수를 가진다.

<117> source에는 리스트 타입의 변수가 정의되어 해당 변수에 있는 데이터의 수만큼 반복해서 처리한다. assign에는 리스트 타입의 데이터 중에 특정 값이 할당되어 내부의 액티비티에서 사용할 수 있다. 내부에는 다른 액티비티들을 정의할 수 있다.

<118> 다섯째, Compute 액티비티는, 비즈니스 이벤트를 생성하기 위해서는 수신된 RFID 이벤트를 가공해야 하는데, 이 경우 산술처리나 로직처리가 필요하게 되면 이러한 처리 작업을 지원할 수 있도록 제공된다.

<119> Compute 액티비티는 일반적인 프로그래밍 언어에서 지원하는 변수간의 산술연산 및 로직 연산을 가능하게 한다. 이러한 연산을 이용해 개발자는 비즈니스 이벤트 생성에 필요한 중간 처리과정을 정의할 수 있다.

<120> Compute 액티비티의 XML 문법을 위한 DTD는 도 3f에서와 같다.

<121> 여섯째, Event 액티비티에서는 조건에 따른 이벤트 생성에 대한 정의를 한다.

<122> Event 액티비티는 일반적인 이벤트의 형식에 따라 크게 3부분으로 구성되는데 첫 번째 부분은 condition문이다. condition문에서는 해당 이벤트를 발생시킬지 말지를 수식을 이용하여 판단하여 조건이 만족될 때에만 해당 Event 액티비티를 수행한다.

<123> Event 액티비티의 두 번째 부분은 action문이다. action문에서는 이벤트를 생성하는데 필요한 추가적인 처리를 정의한다. 예를 들면 출입구에 리더가 설치되어 사람이 출입할 때 인증된 사용자에 한해서만 문을 열어주면서 해당 사람의 이름을 표시할 때 인증된 경우에만 출입구 리더에 읽혀진 태그에 해당하는 이름을 알아올 필요가 생기게 된다. 이러한 경우를 위하여 action절에서는 다른 액티비티들을 사용할 수 있다.

<124> 그리고 마지막으로 세 번째 부분은 generate문이다. generate문에서는 실제로 이벤트를 생성하게 된다. 생성된 이벤트는 정의된 프로토콜에 따라서 어플리케이션에 전달된다. 이때 이벤트에는 이벤트 이름 이외에도 이벤트와 관련된 데이터를 같이 전달하여 어플리케이션이 이용할 수 있도록 한다. Event 액티비티의 XML 문법을 위한 DTD는 도 3g에서와 같다.

<125> 그리고 이하에서 도 1의 RFID 비즈니스 케이스 툴(RFID Business Case Tool;RFID BizCT) 블록(500)에 관하여 상세히 설명한다.

<126> 도 4a내지 도 4l은 본 발명에 따른 RFID 비즈니스 케이스 툴의 각 단계에서의 화면 구성도이다.

<127> RFID 비즈니스 케이스 툴(RFID Business Case Tool;RFID BizCT) 블록(500)은 RFID 비즈니스 이벤트와 RFID 비즈니스 태스크를 쉽게 모델링 하도록 지원하는 도구로써, 편리한 모델링과 다양한 속성의 입력을 지원하는 비주얼한 환경을 갖는다.

<128> RFID 비즈니스 케이스 툴(RFID Business Case Tool;RFID BizCT) 블록(500)은 비즈니스 어웨어 랭귀지(RFID BizCT)로 표현되는 액티비티를 GUI 형식으로 정의하는 도구로써, 네비게이터(Navigator), 메인뷰(MainView), 액티비티 뷰(ActivityView), 프로퍼티 뷰(PropertyView), 메시지 뷰(MessageView)와 같은 창들을 구비한다.

<129> 먼저, 네비게이터(Navigator)를 구성하는 File 창은 현재 사용자의 컴퓨터에 있는 각 폴더의 파일들의 정보를 보여주는 것으로 각 폴더에 있는 BE Spec의 파일을 볼 수 있다. 그리고 Model 창은 BE Spec을 구성하는 액티비티들을 보여준다.

<130> 사용자는 모델창에 표현된 액티비티를 드래그하여 메인뷰에 드롭시킬 수 있다.

<131> 그리고 메인뷰는 BE Spec을 XML의 형태로 보여주거나, 또는 GUI의 형태로 화면에 보여주는 것으로 각 형태에서 편집이 가능하다.

<132> 그리고 액티비티뷰는 사용가능한 액티비티와 태스크의 목록을 보여주는 것으로 각각의 액티비티를 드래그하여 MainView의 Diagram View에 위치시킴으로써 BE Spec을 작성할 수 있다.

<133> 그리고 프로퍼티뷰는 각 액티비티의 세부 속성값을 표현하고, 메시지뷰는 에러 메시지와 알림 메시지를 표현한다.

<134> 이와 같은 RFID 비즈니스 케이스 툴(RFID Business Case Tool;RFID BizCT)의 각 단계에서의 화면 구성 및 특성을 설명한다.

<135> 먼저, 도 4a는 어플리케이션이 실행된 초기 상태의 화면을 나타낸 것으로, 네비게이터(Navigator), 메인뷰(MainView), 액티비티 뷰(ActivityView), 프로퍼티 뷰(PropertyView), 메시지 뷰(MessageView)와 같은 창들이 표시된다.

<136> 그리고 도 4b는 리더 등록, 수정, 삭제 과정을 나타낸 것으로, 리더 리스트 다이얼로그가 뜨고 레지스터를 선택하면 레지스터 리더(Register Reader) 다이얼로그가 뜬다. 등록할 리더명을 입력한 후 OK를 누르고 리더 리스트에서 OK를 누르면 등록이 완료된다. 수정과 삭제는 해당 리더를 리더 리스트에서 선택한 후 마찬가지로 수행하면 된다.

<137> 그리고 도 4c는 새파일을 생성하는 과정을 나타낸 것으로 File 메뉴에서 New를 선택하면 새파일을 생성하기 위한 화면이 메인뷰에 표시된다.

<138> 그리고 도 4d는 변수의 추가, 수정, 삭제 과정을 나타낸 것으로, 메인뷰의 문서 아이콘을 클릭하면 Variables

다이얼로그 창이 나타난다. 변수 추가를 위해서 Add 버튼을 누르면 AddVariable 다이얼로그 창이 나타나며, 추가할 변수의 속성을 입력한 후 OK 를 누르면 변수가 추가된다. 변수의 수정과 삭제 시에는 Variables 다이얼로그창의 테이블에서 해당 변수를 선택한 후 수정 또는 삭제를 수행할 수 있다.

<139> 그리고 도 4e는 액티비티의 생성, 액티비티의 이동, 액티비티의 삭제 과정을 나타낸 것으로, 액티비티뷰(ActivityView)의 각 액티비티를 마우스로 드래그하여 메인뷰의 다이어그램 뷰(Diagram View) 창에서 "Drop Activity Here" 라고 쓰인 곳에 위치시키면, 해당 액티비티가 생성되어 BE Spec에 추가된다.

<140> 액티비티를 클릭하고 다른 위치에 놓음으로써 액티비티를 이동시킬 수 있다. 액티비티를 클릭하고 Delete 키를 누르면 선택된 액티비티가 삭제된다.

<141> 그리고 태스크 액티비티를 드래그하여 이벤트 액티비티 위에 드롭하면 이벤트 액티비티에 태스크가 추가된다.

<142> 그리고 도 4f 내지 도 4j는 액티비티 속성을 설정하는 과정을 나타낸 것이다.

<143> 먼저, 도 4f는 ALE 속성 입력 과정을 나타낸 것으로 메인뷰(MainView)의 ALE 다이어그램을 마우스로 더블 클릭하면 ALE 다이얼로그가 나타나며 LogicalReaders와 BoundarySpec과 ReportSpec이 설정 가능하다.

<144> 그리고 도 4g는 이벤트 속성 설정 과정을 나타낸 것으로, 메인뷰의 이벤트 다이어그램을 마우스로 더블 클릭하면 이벤트 다이얼로그가 나타나며 Condition과 Generate을 설정할 수 있다.

<145> 그리고 도 4h는 List 속성 설정을 나타낸 것으로 메인뷰의 리스트 다이어그램을 마우스로 더블 클릭하면 아래와 같은 리스트 다이얼로그가 나타나며 Attribute를 설정할 수 있다.

<146> 그리고 도 4i는 EPCIS 속성 설정 과정을 나타낸 것으로, 메인뷰의 EPCIS 다이어그램을 마우스로 더블 클릭하면 아래와 같은 EPCIS 다이얼로그가 나타나며 Expression을 설정할 수 있다.

<147> 그리고 도 4j는 메일 태스크 속성 설정 과정을 나타낸 것으로, 메일 태스크 다이어그램을 마우스로 더블 클릭하면 아래와 같은 MailTask 다이얼로그가 나타나며 각 속성값을 설정할 수 있다.

<148> 그리고 도 4k는 액티비티 복사 과정을 나타낸 것으로, Navigator 창의 모델뷰에서 복사할 액티비티를 드래그한 후 메인뷰의 복사될 위치에 놓으면 해당 액티비티가 복사되며 이때, 액티비티의 속성값들도 그대로 복사된다.

<149> 그리고 도 4l은 뷰/소스 변환 과정을 나타낸 것으로, 메인뷰의 다이어그램뷰에서 액티비티를 추가하고 값을 설정한 후 Source 탭을 누르면 해당 BESpec이 생성된다. 또한 Source 창에서 액티비티를 추가, 수정, 삭제하고 값을 새로 설정하고 다이어그램뷰 탭을 누르면 수정된 다이어그램이 생성된다.

<150> 이하에서 본 발명에 따른 RFID 어플리케이션 지원을 위한 미들웨어 플랫폼을 이용한 RFID 비즈니스 이벤트 생성 모델링에 관하여 설명한다.

<151> 도 5는 본 발명에 따른 RFID 비즈니스 이벤트 생성 모델링 프로세스를 나타낸 플로우 차트이다.

<152> RFID Business-aware Framework(RFID BizAF)은 RFID 미들웨어와 어플리케이션 사이 층에 위치하며, RFID 어플리케이션의 개발을 지원하기 위한 것이다.

<153> 이는 RFID 미들웨어에서 생성된 RFID 이벤트를 처리하고 다른 시스템과의 상호작용을 담당하며, 어플리케이션의 조건에 맞는 결과 값만을 RFID 비즈니스 이벤트로 생성하여 전달하는 기능을 담당한다.

<154> RFID 비즈니스 어웨어 프레임워크 블록(RFID BizAF)을 이용한 어플리케이션 개발을 위해서는 효과적으로 비즈니스 이벤트를 추출하는 것이 필요하다.

<155> 본 발명에서는 다음과 같은 단계에 의해 RFID 비즈니스 이벤트를 생성한다.

<156> 먼저, 개발될 RFID 어플리케이션이 사용할 논리적 리더의 개수와 배치 위치를 결정한다.(S501)

<157> RFID 기술에 있어 하드웨어인 리더와 태그는 중요 요소이다. 특히, 리더는 어플리케이션의 많은 부분에 영향을 주는 요소이므로 명확한 결정을 필요로 한다.

<158> 그리고 추출된 리더의 비즈니스 문맥을 정의한다.(S502)

<159> 리더가 놓인 위치에 따라 리더가 수행하는 주요 역할이 달라지는데, 해당 리더가 수행하는 주요 역할을 비즈니스 문맥이라 정의한다. 여기서, 역할이 비슷한 리더는 그룹으로 묶는데, 이는 반복되는 작업 수행을 줄여 일의 효율성을 높이기 위함이다.

<160> 이어, RFID 애플리케이션은 태그가 부착된 객체가 인식될 때 자동화된 처리를 수행하는데, 이를 위하여 해당 리더의 비즈니스 문맥에 관계된 객체들을 추출한다.(S503)

<161> 그리고 추출한 객체들과 연관된 참조정보가 무엇인지 판별한다.(S504)

<162> RFID 태그가 부착된 객체는 유일한 식별 정보를 갖는데, 단순한 숫자 형태의 정보는 미들웨어를 거치면서 이벤트가 발생한 장소, 시간, EPC를 포함한 형태의 정보로 변환된다.

<163> 여기서, EPC 값만으로는 인식된 물체가 무엇인지 판별할 수 없으므로, 의미 있는 정보가 되기 위해 EPC 상세정보가 저장된 EPCIS를 참조하여 추출한 객체들과 연관된 참조정보가 무엇인지 판별하는 것이다.

<164> 이어, 추출된 태그 부착 객체 및 이들과 연관된 참조 정보를 이용하여 어플리케이션의 동작에 필요한 RFID 이벤트와 관련된 비즈니스 규칙을 추출하여 기술한다.(S505)

<165> 비즈니스의 특정 부분을 정의하거나 제약하는 선언을 비즈니스 규칙이라 하는데, 비즈니스에 대해 진실인 내용을 선언하는 사실, 사용자나 시스템의 수행 행위를 제약하는 제약조건, 어떤 조건의 진실 여부에 따라 새 지식을 만드는 규칙인 추론이 비즈니스 규칙에 해당한다.

<166> 그리고 여러 개의 태그를 동시에 인식하는 것은 RFID 기술의 특징 중 하나인데, 동시에 다른 종류의 여러 개의 태그 부착 객체가 인식되었을 때 객체 인식의 효율성을 위하여 이들 간의 처리 흐름을 정의한다.(S506)

<167> 사람이 물품을 운반할 경우 사람의 접근 권한에 따라 물품의 처리 여부가 결정되므로 먼저 처리되어야 할 객체가 사람이 된다. 이 단계는 태그 처리 순서를 정의를 통해 불필요한 태그 처리 동작이 발생하지 않도록 한다.

<168> 또한, 태그 부착 객체가 인식되었을 때 이를 처리하는 흐름을 정의하는 것도 필요하다. 객체가 인식되면 ONS, EPCIS 질의를 통해 객체에 대한 상세 정보를 취득하고 관련된 비즈니스 규칙들을 적용해야 한다.

<169> 이어, 개발될 RFID 어플리케이션이 사용할 논리적 리더들을 물리적 리더로 자세히 기술한다.(S507)

<170> 이는 RFID 어플리케이션은 리더에서 발생한 관심 있는 RFID 이벤트를 전달받아야 하는데, 이를 위해 리더가 동작하는데 필요한 물리적인 요소를 설정하고 결과로 받을 값에 대한 정보를 알려주는 것이다.

<171> 해당 정보는 태그를 읽는 시간 간격, 태그의 비트수, 적용할 필터 등이 설정 요소가 된다. 설정은 S501 단계에서 결정된 논리적 리더 각각에 대해 이루어지며, 이는 XML 형태인 ECSpec으로 작성하여 미들웨어에 등록된다.

<172> 어플리케이션이 BizAF를 이용할 경우, BizAL의 Trigger 액티비티로 BESpec에 기술되며, BizAF가 이를 미들웨어에 등록한다.

<173> 그리고 S506 단계에서 도출된 태그 부착 객체들의 비즈니스 규칙과 처리순서에 따라 BESpec으로 맵핑하여 기술한다.(S508)

<174> BESpec을 BizAF에 등록하면 BizAF는 기술된 비즈니스 규칙조건이 참이 될 때 비즈니스 이벤트를 발생시켜 RFID 어플리케이션에 전달한다.

<175> 이와 같은 본 발명은 RFID 미들웨어 플랫폼을 새롭게 구현하여 RFID 고유의 어플리케이션 개발을 지원하고 RFID 비즈니스 이벤트와 RFID 비즈니스 태스크의 용이한 모델링 프로세스를 지원하여 RFID 어플리케이션의 생산 및 관리의 효율성을 높일 수 있도록 한다.

<176> 이상 설명한 내용을 통해 당업자라면 본 발명의 기술 사상을 일탈하지 아니하는 범위에서 다양한 변경 및 수정이 가능함을 알 수 있을 것이다.

<177> 따라서, 본 발명의 기술적 범위는 실시예에 기재된 내용으로 한정되는 것이 아니라 특허 청구의 범위에 의하여 정해져야 한다.

발명의 효과

<178> 이와 같은 본 발명에 따른 RFID 어플리케이션 지원을 위한 미들웨어 플랫폼은 다음과 같은 효과가 있다.

<179> 첫째, 본 발명은 RFID 미들웨어 플랫폼을 새롭게 구현하여 RFID 고유의 어플리케이션 개발을 지원하고 RFID 비즈니스 이벤트와 RFID 비즈니스 태스크의 용이한 모델링 프로세스를 지원하여 RFID 어플리케이션의 생산 및 관리의 효율성을 높이는 효과가 있다.

<180> 둘째, ALE 기반 RFID 미들웨어를 이용하는 어플리케이션 개발자가 RFID BizAL과 RFID BizAF를 이용하여 RFID 데이터 중심의 어플리케이션 로직을 설계할 수 있다.

<181> 셋째, 본 발명에 따른 RFID BizAL과 RFID BizAF는 매우 지속적이고 방대한 RFID 원시 이벤트를 직접 코드로 작성할 때 발생할 수 있는 복잡성 및 어려움을 해결함으로써, 개발자에게 필요한 어플리케이션 로직을 용이하게 개발할 수 있는 환경을 제공한다.

<182> 넷째, 본 발명에 따른 RFID BizAL과 RFID BizAF는 ALE(Application Level Event) 표준 스펙을 준수하는 다양한 RFID 미들웨어의 연동을 지원한다.

<183> 따라서, 타 RFID 미들웨어가 ALE 표준 스펙을 준수할 경우 RFID BizAL과 RFID BizAF와 연동을 통해 효과적으로 RFID 원시 이벤트를 고수준의 비즈니스 이벤트로 변환할 수 있다.

<184> 이는 RFID 미들웨어 개발자가 자신의 RFID 미들웨어에 RFID BizAL과 RFID BizAF를 도입하여 보다 확장된 기능을 가지는 RFID 미들웨어를 개발할 수 있도록 하는 효과를 갖는다.

<185> 다섯째, 단일의 RFID Business-Aware Language를 개발하고, 이를 기반으로 어플리케이션동작 시 활용되는 기업의 규칙을 비즈니스 이벤트로 기술함으로써, 서로 다른 어플리케이션 간 규칙의 공유나 재사용성을 높일 수 있다.

<186> 즉, 어플리케이션 간 규칙을 공유하거나 비즈니스 파트너 간의 지식 교환이 가능하게 된다.

<187> 이는 특정 도메인 내의 여러 어플리케이션이 공통적으로 필요로 하는 비즈니스 이벤트를 패키지 형태로 개발하고, 이를 해당 도메인의 어플리케이션 개발 시에 재사용함으로써 어플리케이션 개발의 효율성을 높이는 효과를 갖는다.

도면의 간단한 설명

<1> 도 1은 본 발명에 따른 다양한 어플리케이션 지원을 위한 RFID 미들웨어 플랫폼의 구성도

<2> 도 2는 본 발명에 따른 RFID 비즈니스 어웨어 랭귀지의 액티비티들간의 관계를 나타낸 구성도

<3> 도 3a내지 도 3g는 본 발명에 따른 RFID 비즈니스 어웨어 랭귀지의 액티비티들의 DTD를 나타낸 구성도

<4> 도 4a내지 도 4l은 본 발명에 따른 RFID 비즈니스 케이스 툴의 각 단계에서의 화면 구성도

<5> 도 5는 본 발명에 따른 RFID 비즈니스 이벤트 생성 모델링 프로세스를 나타낸 플로우 차트

<6> 도면의 주요 부분에 대한 부호의 설명

<7> 100. RFID 비즈니스 어웨어 랭귀지 블록(RFID BizAL)

<8> 200. RFID 비즈니스 이벤트 어시스턴트 블록(RFID BizEA)

<9> 300. RFID 비즈니스 태스크 어시스턴트 블록(RFID BizTA)

<10> 400. RFID 비즈니스 케이스 툴 블록(RFID BizCT)

<11> 500. RFID 비즈니스 어웨어 프레임워크 블록(RFID BizAF)

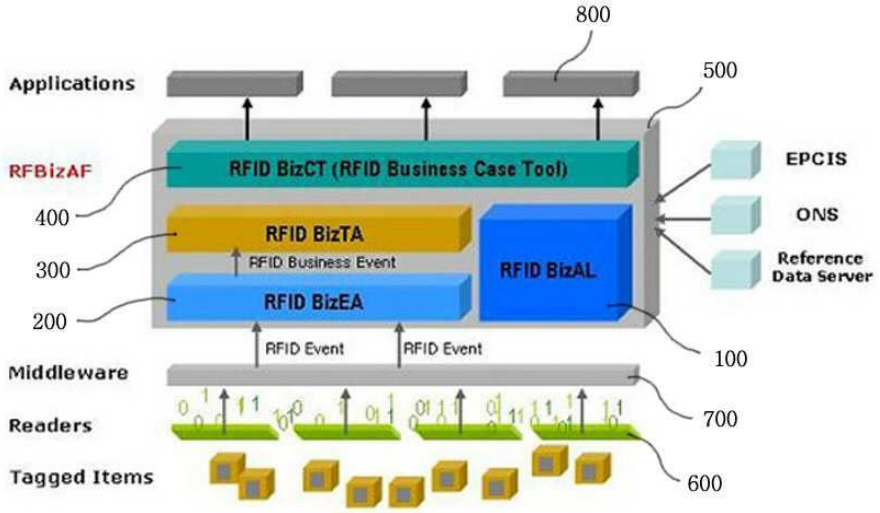
<12> 600. 리더스 계층

<13> 700. RFID 미들웨어 계층

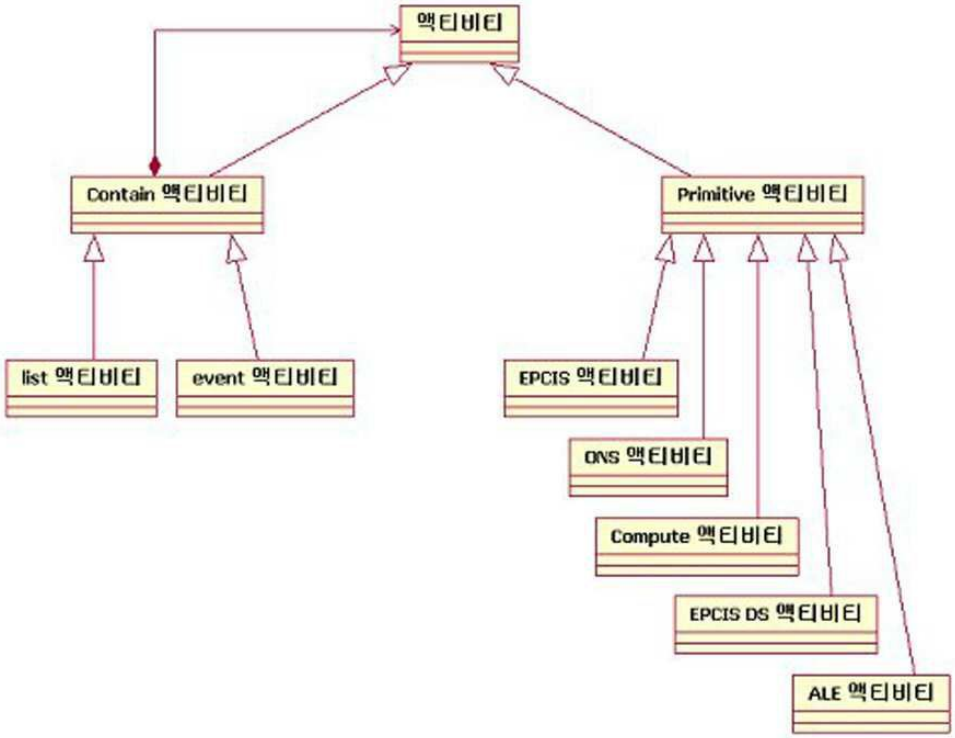
<14> 800. 미들웨어 계층

도면

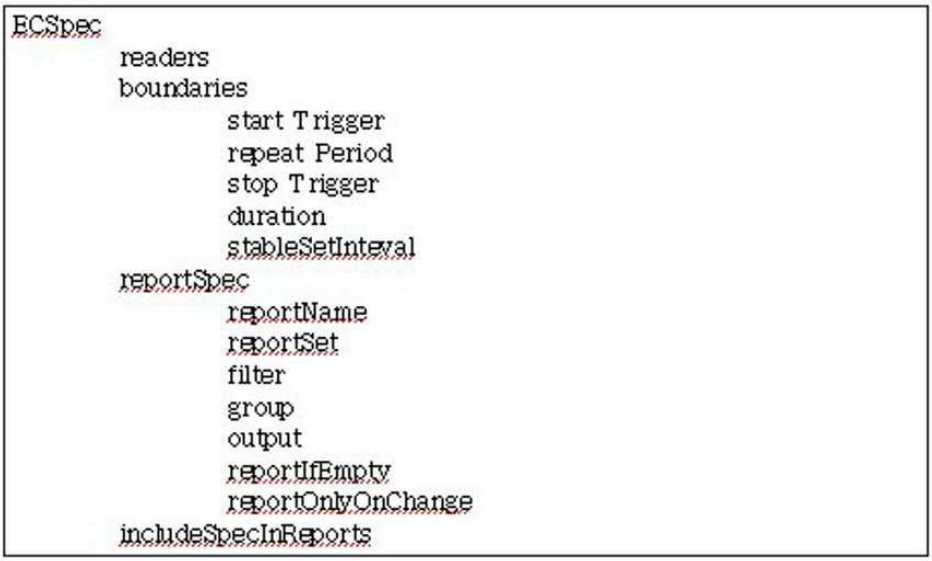
도면1


800
Applications
500
RFBizAF
RFID BizCT (RFID Business Case Tool)
EPCIS
400
RFID BizTA
ONS
RFID Business Event
300
RFID BizAL
Reference Data Server
RFID BizEA
200
RFID Event
RFID Event
100
Middleware
700
Readers
Tagged Items
600

도면2


액티비티
Contain 액티비티
Primitive 액티비티
list 액티비티
event 액티비티
EPCIS 액티비티
ONS 액티비티
Compute 액티비티
EPCIS DS 액티비티
ALE 액티비티

도면3a



도면3b

```
<!ELEMENT reportSpec (reportName, reportSet, filter?, group?,
output, result, reportIfEmpty?, reportOnlyOnChange?)>
  <!ELEMENT result EMPTY>
  <!ATTLIST result EPC CDATA #IMPLIED>
  <!ATTLIST result Tag CDATA #IMPLIED>
  <!ATTLIST result RawHex CDATA #IMPLIED>
  <!ATTLIST result RawDecimal CDATA #IMPLIED>
  <!ATTLIST result Count CDATA #IMPLIED>
```

도면3c

```
<!ELEMENT EPCIS (getEPCAttribute | getEPCSchema | getEPCClassAttribute | getEPCClassSchema)+ >
<!ATTLIST EPCIS address CDATA>
  <!ELEMENT getEPCAttribute #PCDATA>
  <!ATTLIST getEPCAttribute epc CDATA #REQUIRED>
  <!ATTLIST getEPCAttribute schema CDATA #REQUIRED>
  <!ATTLIST getEPCAttribute xpath CDATA #REQUIRED>
  <!ELEMENT getEPCSchema #PCDATA>
  <!ATTLIST getEPCSchema epc CDATA #REQUIRED>
  <!ELEMENT getEPCClassAttribute #PCDATA>
  <!ATTLIST getEPCClassAttribute epcClass CDATA #REQUIRED>
  <!ATTLIST getEPCClassAttribute schema CDATA #REQUIRED>
  <!ATTLIST getEPCClassAttribute xpath CDATA #REQUIRED>
  <!ELEMENT getEPCClassSchema #PCDATA>
  <!ATTLIST getEPCClassSchema epcClass CDATA #REQUIRED>
```

도면3d

```
<!ELEMENT ONS (query)+ >
<!ATTLIST EPCIS address CDATA #IMPLIED>
  <!ELEMENT query #PCDATA>
  <!ATTLIST query epc CDATA #REQUIRED>
```

도면3e

```
<!ELEMENT list (activity)*>
<!ATTLIST list source CDATA #REQUIRED>
<!ATTLIST list assign CDATA #REQUIRED>
```

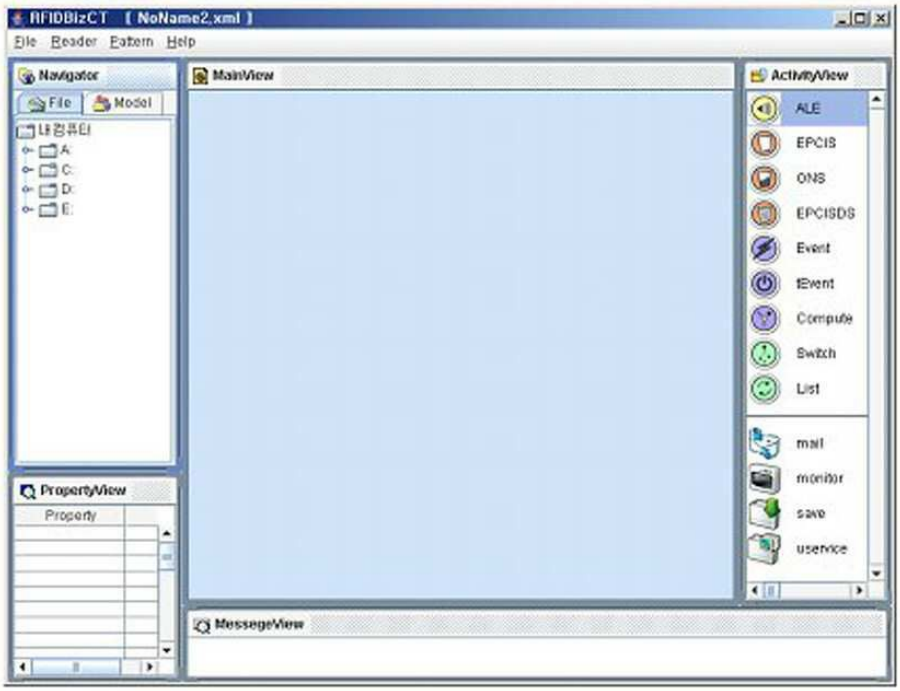
도면3f

```
<!ELEMENT compute (expression)*>
  <!ELEMENT expression #PCDATA>
```

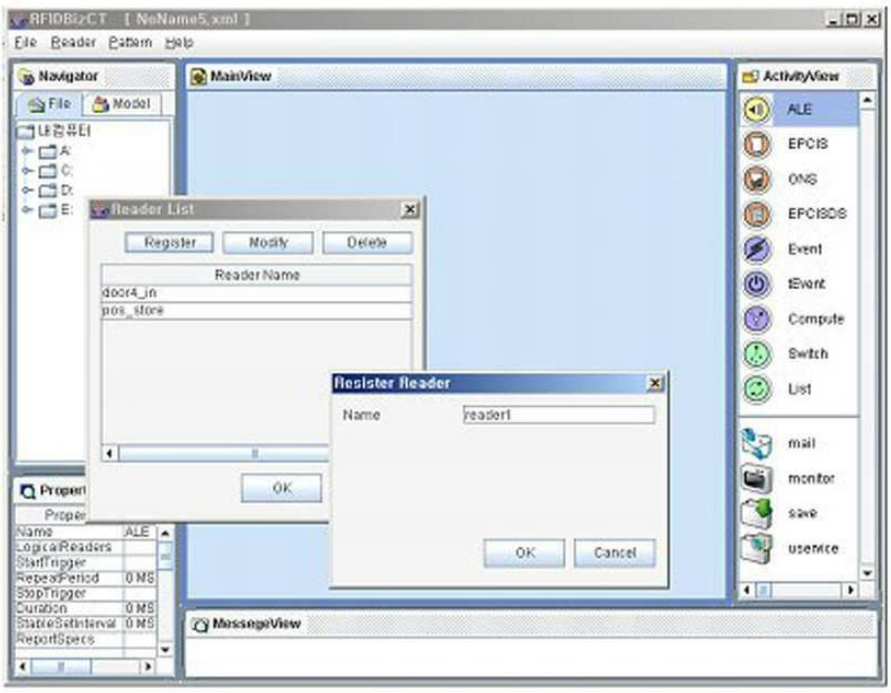
도면3g

```
<!ELEMENT event (condition, action?, generate, action?)>
  <!ELEMENT condition #PCDATA>
  <!ELEMENT action (activity)*>
  <!ELEMENT generate (data)*>
  <!ATTLIST generate name CDATA #REQUIRED>
    <!ELEMENT data #PCDATA>
    <!ATTLIST data name CDATA #REQUIRED>
```

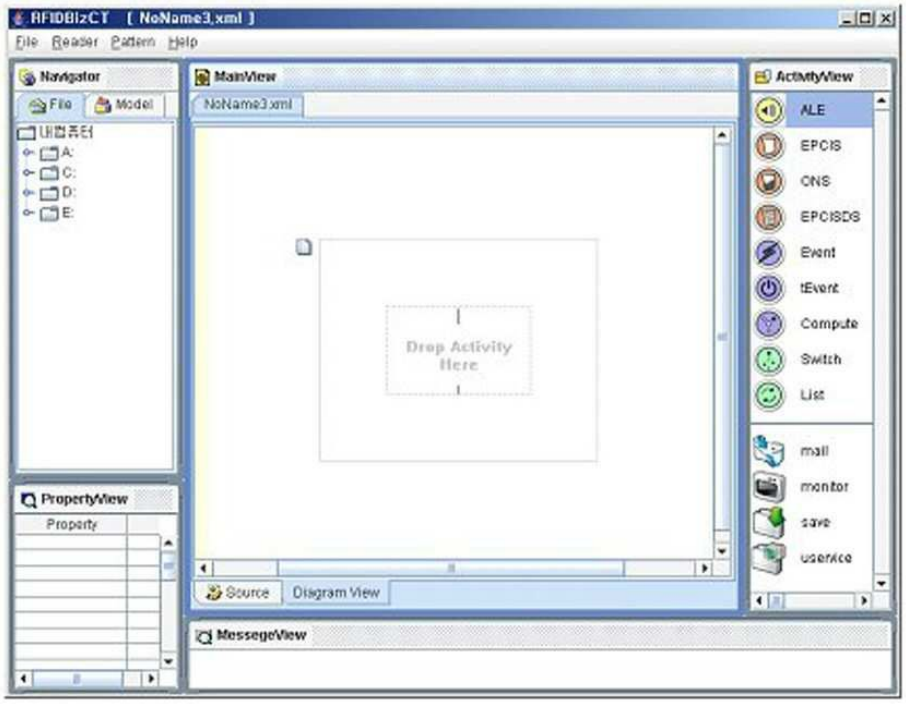
도면4a



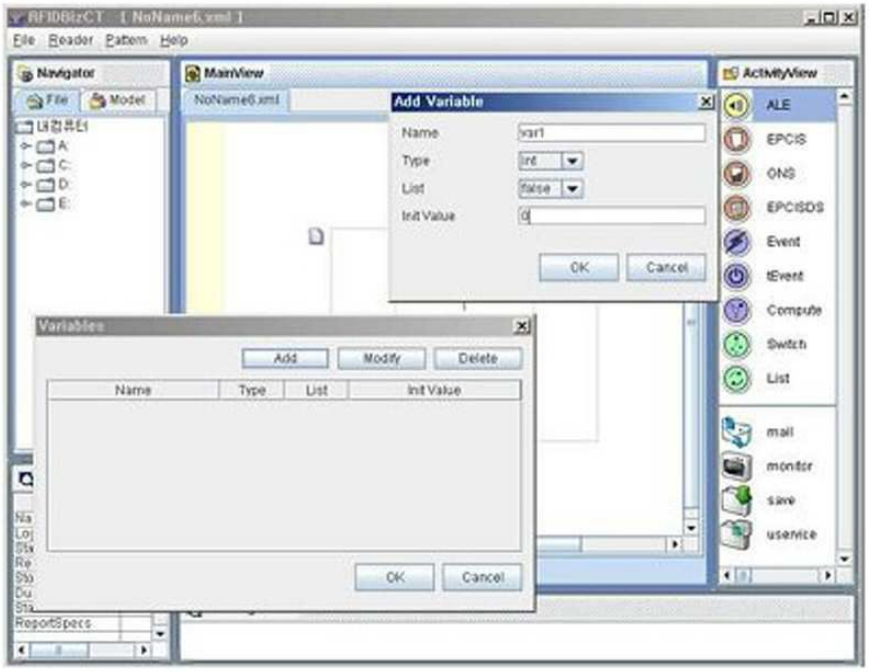
도면4b


RFIDBizCT [NoName5.xml]
File Reader Pattern Help
Navigator
File Model
MainView
ActivityView
ALE
EPCIS
ONS
EPCISDS
Event
tEvent
Compute
Switch
List
mail
monitor
save
userexe
Reader List
Register
Modify
Delete
Reader Name
door4_in
pos_store
OK
Register Reader
Name
reader1
OK
Cancel
MessegeView

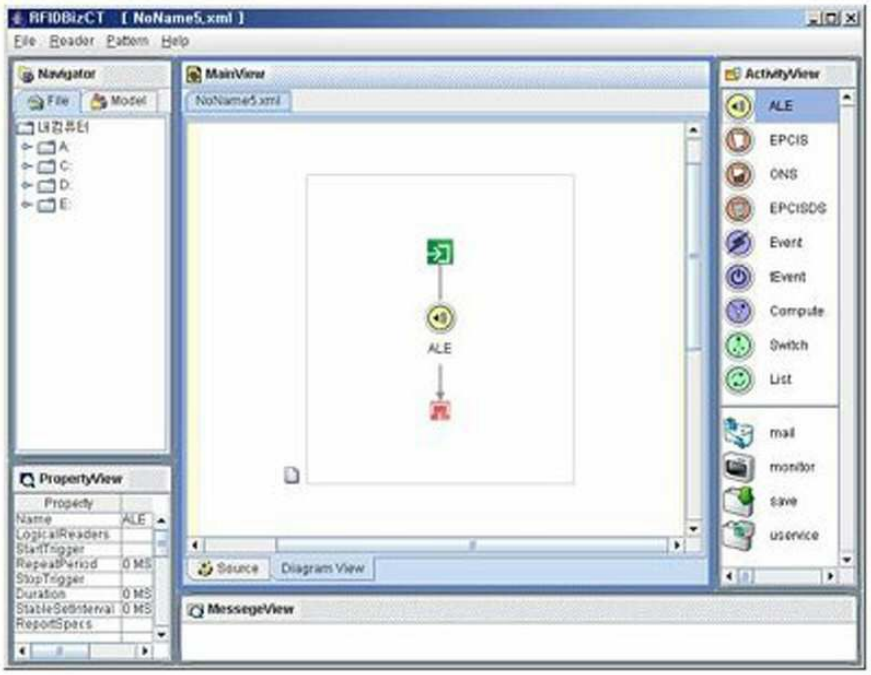
도면4c


RFIDBizCT [NoName3.xml]
File Reader Pattern Help
Navigator
File Model
MainView
NoName3.xml
ActivityView
ALE
EPCIS
ONS
EPCISDS
Event
tEvent
Compute
Switch
List
mail
monitor
save
userexe
Drop Activity Here
PropertyView
Property
Source
Diagram View
MessegeView

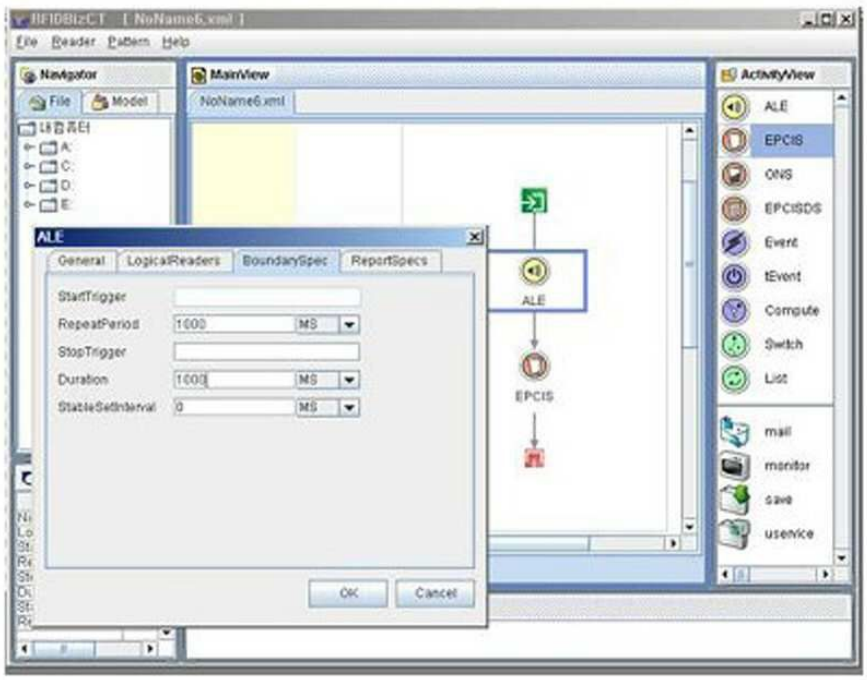
도면4d


File Reader Pattern Help
Navigator
File Model
MainView
NoName6.xml
Add Variable
Name var1
Type int
List false
Init Value 0
OK Cancel
Variables
Add Modify Delete
Name Type List Init Value
OK Cancel
ActivityView
ALE
EPCIS
ONS
EPCISDS
Event
tEvent
Compute
Switch
List
mail
monitor
save
uservice
ReportSpecs

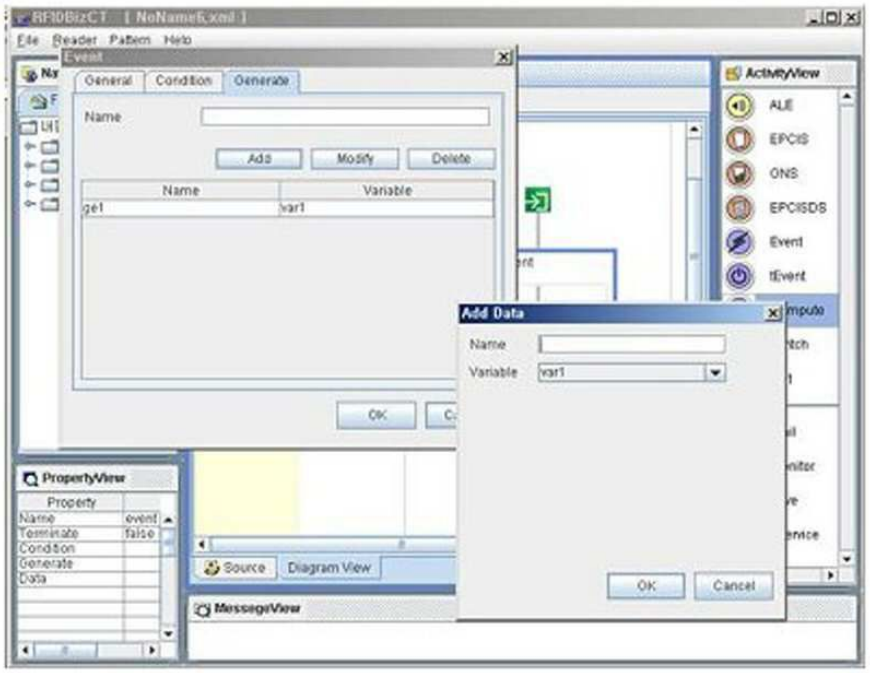
도면4e


RFIDBizCT [NoName5.xml]
File Reader Pattern Help
Navigator
File Model
MainView
NoName5.xml
ALE
Source Diagram View
ActivityView
ALE
EPCIS
ONS
EPCISDS
Event
tEvent
Compute
Switch
List
mail
monitor
save
uservice
PropertyView
Property
Name ALE
LogicalReaders
StartTrigger
RepeatPeriod 0 MS
StopTrigger
Duration 0 MS
StableSetInterval 0 MS
ReportSpecs
MessegeView

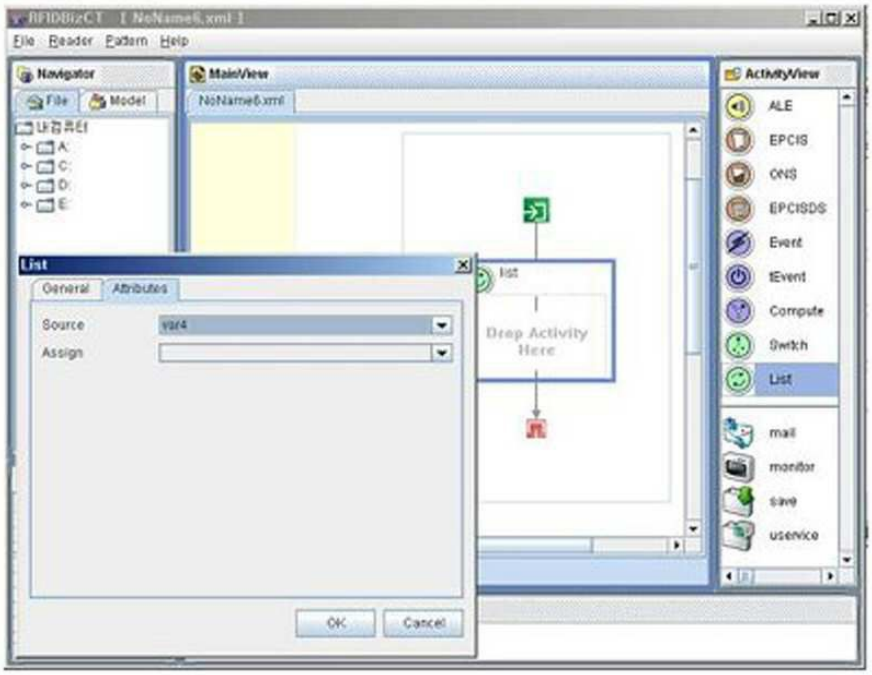
도면4f


RFIDBizCT [NoName6.xml]
File Reader Pattern Help
Navigator
File Model
MainView
NoName6.xml
ActivityView
ALE
EPCIS
ONS
EPCISDS
Event
tEvent
Compute
Switch
List
mail
monitor
save
uservice
ALE
General LogicalReaders BoundarySpec ReportSpecs
StartTrigger
RepeatPeriod 1000 MS
StopTrigger
Duration 1000 MS
StableSetInterval 0 MS
OK Cancel

도면4g


RFIDBizCT [NoName6.xml]
File Reader Pattern Help
Event
General Condition Generate
Name
Add Modify Delete
Name Variable
ge1 var1
ActivityView
ALE
EPCIS
ONS
EPCISDS
Event
tEvent
Add Data
Name
Variable var1
OK Cancel
PropertyView
Property
Name event
Terminate false
Condition
Generate
Data
Source Diagram View
MessageView

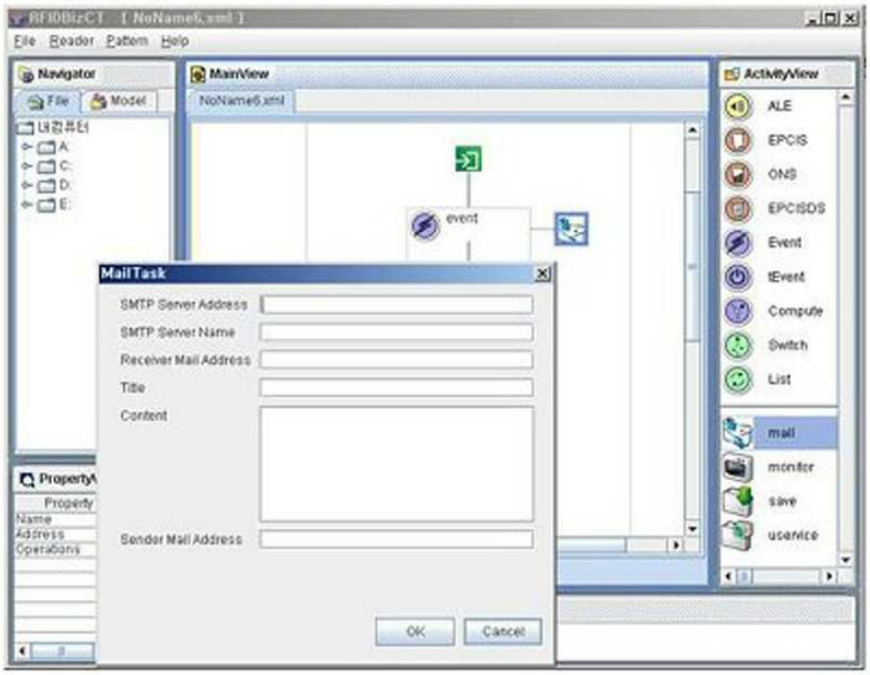
도면4h


File Reader Pattern Help
Navigator
File
Model
MainView
NoName6.xml
ActivityView
ALE
EPCIS
ONS
EPCISDS
Event
tEvent
Compute
Switch
List
mail
monitor
save
uservice
List
General
Attributes
Source
var4
Assign
list
Drop Activity Here
OK
Cancel

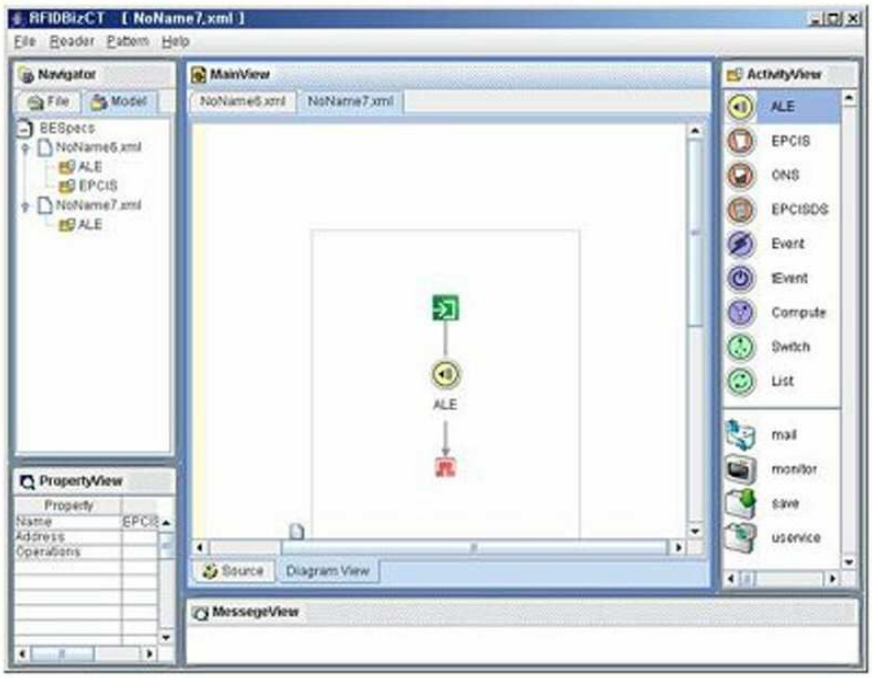
도면4i


File Reader Pattern Help
EPCIS
General
Operations
Add
Modify
Delete
Operation
EPC
Schema
Xpath
Result
getEPCAttribute
var4
var1
Add Operation
Operation
getEPCClassAttribute
EPCClass
var4
Schema
Xpath
Result
var1
OK
Cancel
ActivityView
ALE
EPCIS
ONS
EPCISDS
Source
Diagram View
MessageView

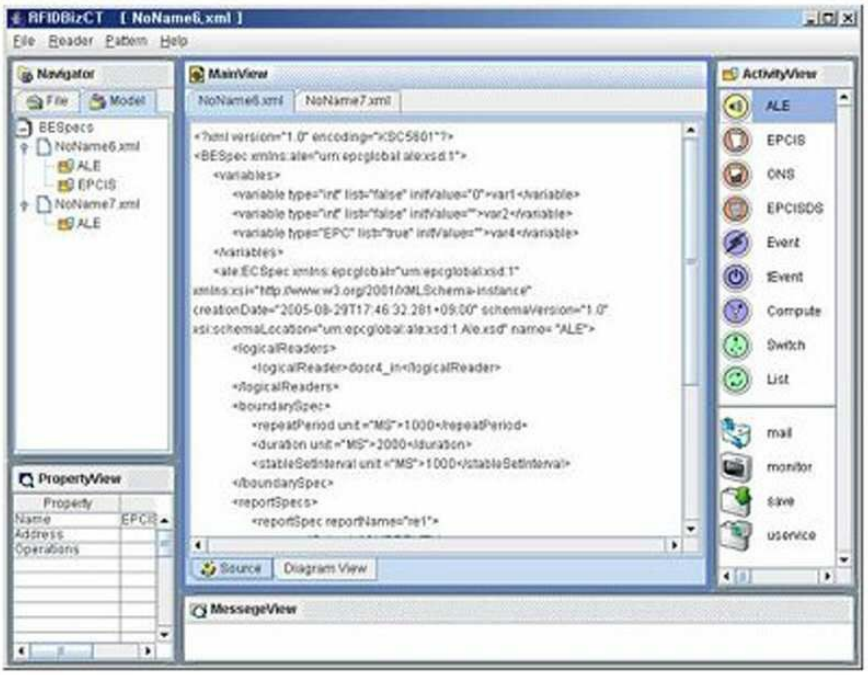
도면4j


File Reader Pattern Help
Navigator
File Model
MainView
NoName6.xml
ActivityView
ALE
EPCIS
ONS
EPCISDS
Event
tEvent
Compute
Switch
List
mail
monitor
save
uservice
event
MailTask
SMTP Server Address
SMTP Server Name
Receiver Mail Address
Title
Content
Sender Mail Address
OK
Cancel

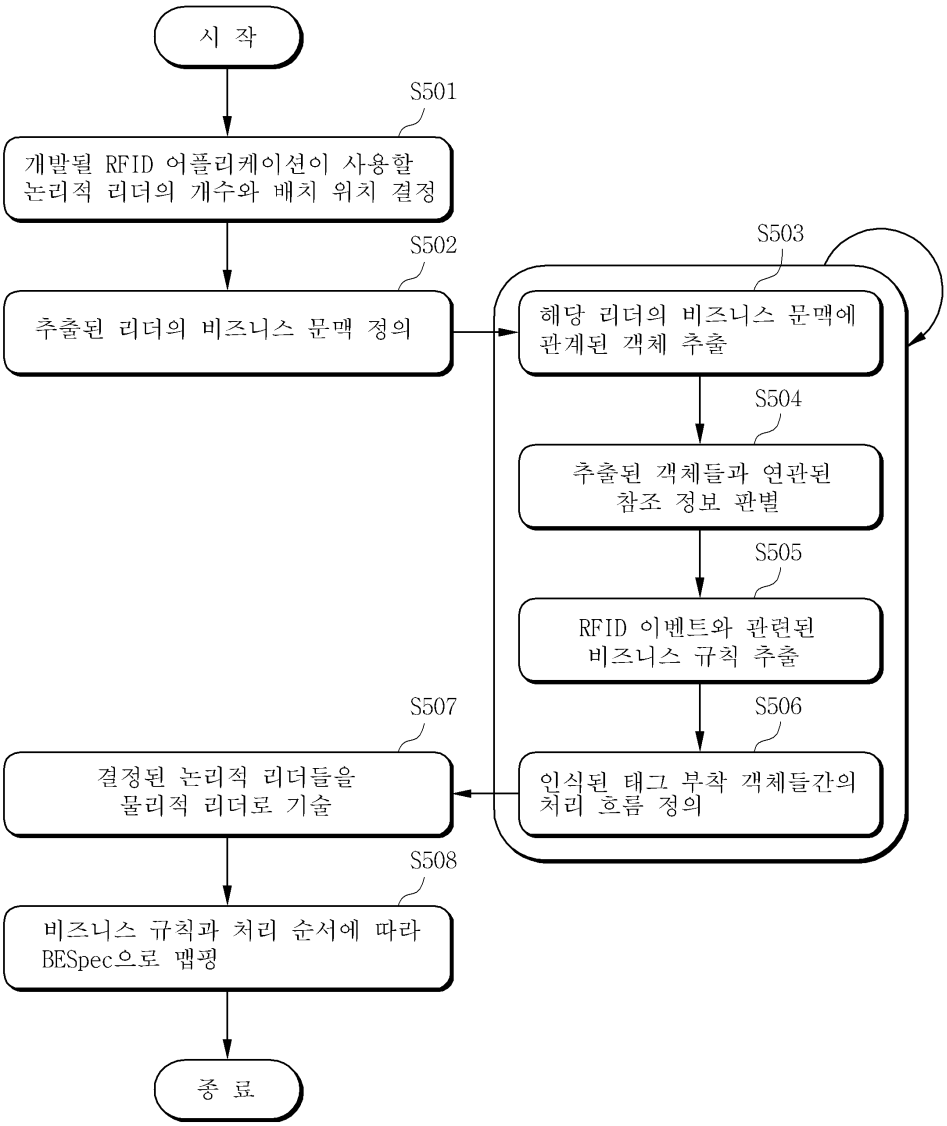
도면4k


RFIDBizCT [NoName7.xml]
File Reader Pattern Help
Navigator
File Model
BESpecs
NoName6.xml
ALE
EPCIS
NoName7.xml
ALE
MainView
NoName6.xml
NoName7.xml
ALE
Source
Diagram View
MessageView
ActivityView
ALE
EPCIS
ONS
EPCISDS
Event
tEvent
Compute
Switch
List
mail
monitor
save
uservice
PropertyView
Property
Name
Address
Operations

도면41


RFIDBizCT [NoName6.xml]
File Reader Pattern Help
Navigator
File Model
BESpecs
NoName6.xml
ALE
EPCIS
NoName7.xml
ALE
PropertyView
Property
Name EPCIS
Address
Operations
MainView
NoName6.xml NoName7.xml
<?xml version="1.0" encoding="KSC5601"?>
<BESpec xmlns:ale="urn:epcglobal:ale:xsd:1">
<variables>
<variable type="int" list="false" initValue="0">var1</variable>
<variable type="int" list="false" initValue="">var2</variable>
<variable type="EPC" list="true" initValues="">var4</variable>
</variables>
<ale:ECSpec xmlns:epcglobal="urn:epcglobal:xsd:1"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
creationDate="2005-09-29T17:46:32.281+09:00" schemaVersion="1.0"
xsi:schemaLocation="urn:epcglobal:ale:xsd:1 Ale.xsd" name= "ALE">
<logicalReaders>
<logicalReader>door4_in</logicalReader>
</logicalReaders>
<boundarySpec>
<repeatPeriod unit ="MS">1000</repeatPeriod>
<duration unit ="MS">2000</duration>
<stableSetInterval unit ="MS">1000</stableSetInterval>
</boundarySpec>
<reportSpecs>
<reportSpec reportName="re1">
Source Diagram View
MessageView
ActivityView
ALE
EPCIS
ONS
EPCISDS
Event
tEvent
Compute
Switch
List
mail
monitor
save
uservice

도면5


시 작
S501
개발될 RFID 어플리케이션이 사용할 논리적 리더의 개수와 배치 위치 결정
S502
추출된 리더의 비즈니스 문맥 정의
S503
해당 리더의 비즈니스 문맥에 관계된 객체 추출
S504
추출된 객체들과 연관된 참조 정보 판별
S505
RFID 이벤트와 관련된 비즈니스 규칙 추출
S506
인식된 태그 부착 객체들간의 처리 흐름 정의
S507
결정된 논리적 리더들을 물리적 리더로 기술
S508
비즈니스 규칙과 처리 순서에 따라 BESpec으로 맵핑
종 료