(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2008-0098267
(43) 공개일자 2008년11월07일

(51) Int. Cl.
G06F 15/16 (2006.01) *G06F 17/00* (2006.01)
(21) 출원번호 10-2007-0043736
(22) 출원일자 2007년05월04일
심사청구일자 2007년05월04일

(71) 출원인
박병섭
서울 서초구 서초4동 삼풍아파트 9동 602호
(72) 발명자
박병섭
서울 서초구 서초4동 삼풍아파트 9동 602호
(74) 대리인
특허법인엔트리

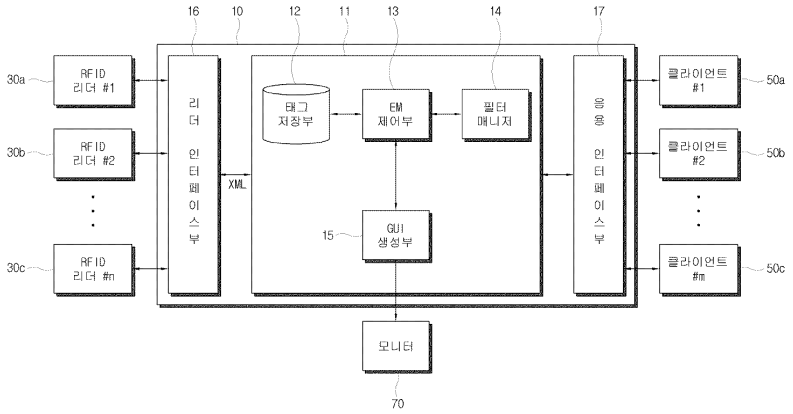
전체 청구항 수 : 총 10 항

(54) ＲＦＩＤ 미들웨어 시스템

(57) 요 약

본 발명은 상호 상이한 기종의 복수의 RFID 리더에 네트워크를 통해 각각 연결된 RFID 어댑터를 가지며, 상기 복수의 RFID 리더에 의해 읽힌 스트림 태그 데이터를 상기 RFID 어댑터를 통해 수집하는 리더 인터페이스부와; 상호 상이한 프로토콜을 사용하는 복수의 클라이언트와 상기 각 프로토콜에 따라 상기 복수의 클라이언트와 데이터를 교환하는 응용 인터페이스부와; 상기 리더 인터페이스부를 통해 수집된 상기 스트림 태그 데이터를 관리하고, 상기 응용 인터페이스부를 통해 상기 클라이언트로부터 수신되는 필터링 조건에 따라 상기 스트림 태그 데이터를 가공 및 필터링하여 응용 데이터를 생성하며, 상기 응용 인터페이스부를 통해 상기 복수의 클라이언트 중 적어도 하나 이상에 상기 응용 데이터를 전송하는 이벤트 매니저를 포함하는 것을 특징으로 한다. 이에 따라, 상호 상이한 복수의 프로토콜에 따른 플랫폼의 접근이 가능하며, 실시간으로 대용량의 데이터 처리가 가능하게 된다.

대 표 도



특허청구의 범위

청구항 1

상호 상이한 기종의 복수의 RFID 리더에 네트워크를 통해 각각 연결된 RFID 어댑터를 가지며, 상기 복수의 RFID 리더에 의해 읽힌 스트림 태그 데이터를 상기 RFID 어댑터를 통해 수집하는 리더 인터페이스부와;

상호 상이한 프로토콜을 사용하는 복수의 클라이언트와 상기 각 프로토콜에 따라 상기 복수의 클라이언트와 데이터를 교환하는 응용 인터페이스부와;

상기 리더 인터페이스부를 통해 수집된 상기 스트림 태그 데이터를 관리하고, 상기 응용 인터페이스부를 통해 상기 클라이언트로부터 수신되는 필터링 조건에 따라 상기 스트림 태그 데이터를 가공 및 필터링하여 응용 데이터를 생성하며, 상기 응용 인터페이스부를 통해 상기 복수의 클라이언트 중 적어도 하나 이상에 상기 응용 데이터를 전송하는 이벤트 매니저를 포함하는 것을 특징으로 하는 RFID 미들웨어 시스템.

청구항 2

제1항에 있어서,

상기 응용 인터페이스부는 JMS(Java Message Service) 프로토콜, SOAP(Simple Object Access Protocol), 및 HTTP(HiperText Transfer Protocol) 중 적어도 어느 하나로 상기 클라이언트와 통신하는 것을 특징으로 하는 RFID 미들웨어 시스템.

청구항 3

제2항에 있어서,

상기 이벤트 매니저는

상기 리더 인터페이스부를 통해 수집된 상기 스트림 태그 데이터가 저장되는 태그 저장부와,

상기 필터링 조건에 따라 상기 스트림 태그 데이터를 가공 및 필터링하여 응용 데이터를 생성하는 필터 매니저와,

상기 필터 매니저에 의해 생성된 상기 응용 데이터를 상기 응용 인터페이스부를 통해 해당 클라이언트로 전송하는 EM 제어부를 포함하는 것을 특징으로 하는 RFID 미들웨어 시스템.

청구항 4

제3항에 있어서,

상기 필터 매니저와 상기 각 클라이언트 간의 필터링 클래스 구조는,

상기 각 프로토콜을 이용하여 필터를 관리하는 복수의 개별 프로토콜 클래스와;

상기 각 필터에 대한 필터정보가 기록된 FilterSet 클래스와, 상기 필터정보에 대한 연산요청을 위한 API(Application Program Interface)를 갖는 FilterMgr 클래스와, 상기 각 개별 프로토콜 클래스가 상기 FilterMgr 클래스의 API(Application Program Interface)를 통해 상기 FilterSet 클래스에 동시 접속할 때 소정의 임계영역을 설정하여 접근을 제어하는 FlterServer 클래스로 구성된 공통 클래스를 포함하는 것을 특징으로 하는 RFID 미들웨어 시스템.

청구항 5

제5항에 있어서,

상기 FilterSet 클래스는 매소드가 존재하지 않는 복수의 맴버로 구성되며;

상기 FilterSet 클래스를 구성하는 상기 멤버는 상기 각 필터에 대한 필터규칙을 포함하는 것을 특징으로 하는 RFID 미들웨어 시스템.

청구항 6

제5항에 있어서,

상기 FlterServer 클래스는 상기 각 개별 프로토콜 클래스가 상기 FilterMgr 클래스의 상기 API를 통해 상기 FilterSet 클래스에 접속하는 경우 쓰레드(Thread)를 생성하며;

상기 쓰레드는 상기 FilterSet 클래스의 상기 필터정보가 기록된 파일에 록(Lock)을 걸고 상기 FilterMgr 클래스를 통해 수신된 상기 연산요청을 수행하는 것을 특징으로 하는 RFID 미들웨어 시스템.

청구항 7

제4항 내지 제6항 중 어느 한 항에 있어서,

상기 개별 프로토콜 클래스는

상기 JMS(Java Message Service) 프로토콜을 이용하여 상기 필터를 관리하는 JMS 클래스와,

AXIS 상에서 상기 SOAP(Simple Object Access Protocol)를 이용하여 상기 필터를 관리하는 SOAP 클래스를 포함하는 것을 특징으로 하는 RFID 미들웨어 시스템.

청구항 8

제7항에 있어서,

상기 필터링 클래스 구조는 상기 JMS 클래스에 대한 상기 JMS(Java Message Service) 프로토콜을 지원하는 JMS 서비스 제공자를 더 포함하며;

상기 JMS 클래스는

상기 JMS(Java Message Service) 프로토콜을 위한 API(Application Program Interface)를 제공하며, 상기 JMS 서비스 제공자에 접속하여 상기 필터정보에 대한 연산요청을 상기 JMS 서비스 제공자에 저장하는 JMSController 클래스와,

상기 JMS 서비스 제공자에 저장된 상기 필터정보에 대한 연산요청을 독출하여 상기 공통 클래스의 상기 FilterMgr 클래스를 통해 상기 FilterSet 클래스에 전달하는 ListenerBody 클래스와,

상기 JMS 서비스 제공자에 연결되며, 상기 ListenerBody 클래스를 생성하여 비동기적으로 상기 필터정보에 대한 연산요청을 처리하는 Listener 클래스를 포함하는 것을 특징으로 하는 RFID 미들웨어 시스템.

청구항 9

제7항에 있어서,

상기 SOAP 클래스는

상기 SOAP(Simple Object Access Protocol)를 위한 API(Application Program Interface)를 제공하며, 상기 AXIS에 접속해서 상기 필터정보에 대한 연산요청을 수행하는 SOAPController 클래스와,

상기 SOAPController 클래스로부터의 상기 연산요청을 상기 공통 클래스의 상기 FilterMgr 클래스를 통해 상기 FilterSet 클래스에 전달하는 FilterMgr.jws 클래스를 포함하는 것을 특징으로 하는 RFID 미들웨어 시스템.

청구항 10

제3항에 있어서,

상기 RFID 어댑터는 상기 각 RFID 리더와 TCP/IP를 통해 연결되며;

상기 리더 인터페이스부를 통해 수집된 상기 스트림 태그 데이터는 XML 파일 형태로 상기 이벤트 매니저에 전달되어 상기 태그 저장부에 저장되는 것을 특징으로 하는 RFID 미들웨어 시스템.

명 세 서

발명의 상세한 설명

발명의 목적

발명이 속하는 기술 및 그 분야의 종래기술

<16> 본 발명은 RFID 미들웨어 시스템에 관한 것으로서, 보다 상세하게는 상호 상이한 복수의 프로토콜에 따른 플랫폼의 접근이 가능하며, 실시간으로 대용량의 데이터 처리가 가능한 RFID 미들웨어 시스템에 관한 것이다.

<17> RTE(Real Time Enterprise)란 기업 내부, 외부를 포함하는 전체적인 관리점에서 지속적인 프로세스 개선과 정보의 실시간 전달을 통해 업무 지연 요소를 최소화하고 의사결정의 속도를 높여 경쟁력을 향상시킨 기업을 의미한다. 최근 많은 기업들이 이런 RTE 구축을 위해 RFID(Radio Frequency Identification) 시스템을 토입하거나 도입을 추진 중에 있다.

<18> 이러한 RFID 시스템은 사물에 RFID 태그를 부착하고 리더를 통해 태그의 정보를 무선으로 읽음으로써 사물의 정보 및 주변 환경의 정보를 자동으로 수십하여 활용하는 것을 의미한다.

<19> RFID 시스템이 다양한 응용 서비스에 사용되기 위해서는 RFID 장치에 저장되어 있는 데이터를 적절한 장소와 적절한 시간에 응용 서비스로 전달하는 RFID 미들웨어 시스템이 요구된다.

<20> 이러한 RFID 미들웨어 시스템이 정확한 데이터를 전달한다는 것은 리더로부터 수집된 정보 중 응용 서비스가 관심있는 데이터만을 필터링하여 전달하는 것을 위미하며, 데이터 필터링 기능은 데이터의 형식 및 응용 환경에 따라 요구되는 기능이 달라진다.

<21> 기존에 단순한 EPC(Electronic Product Code)를 활용하는 응용에서 필요한 정보를 얻는 방법과 훨씬 더 복잡한 구조를 갖는 데이터를 이용하는 응용에서 정보를 필터링하는 방법은 상이하다. 또한 처리되어야 할 데이터의 양, 동시에 처리되는 필터링 조건의 수 등을 고려하여 데이터 처리 방법을 채택하여야만 데이터의 손실없이 실시간 처리가 가능하게 된다.

<22> 근래에 RFID 미드루에어 시스템 또는 솔루션들이 다양한 형태로 제안되고 있으나, 주로 EPC 등과 같은 간단한 형식의 데이터를 처리하는데 국한되어 있어, 대량의 데이터 처리에 대한 고려가 미진한 상태이다. 또한 기존의 RFID 미들웨어 시스템은 단순히 비즈니스 애플리케이션과 핵심 인프라스트럭처 사이의 커뮤니케이션만 지원하는데 국한되는 것이 일반적이다.

발명이 이루고자 하는 기술적 과제

<23> 따라서 본 발명의 목적은 상호 상이한 복수의 프로토콜에 따른 플랫폼의 접근이 가능하며, 실시간으로 대용량의 데이터 처리가 가능한 RFID 미들웨어 시스템을 제공하는데 있다.

<24> 또한, 본 발명의 목적은 인프라스트럭처, 즉 응용 서비스인 클라이언트가 사용하는 프로토콜의 상이함에 따른 다양한 플랫폼에 대한 확장성 및 그 통합을 지원하면서도 대용량의 데이터 처리와 가능한 RFID 미들웨어 시스템을 제공하는데 있다.

발명의 구성 및 작용

<25> 상기 목적은 본 발명에 따라, 상호 상이한 기종의 복수의 RFID 리더에 네트워크를 통해 각각 연결된 RFID 어댑터를 가지며, 상기 복수의 RFID 리더에 의해 읽힌 스트림 태그 데이터를 상기 RFID 어댑터를 통해 수집하는 리더 인터페이스부와; 상호 상이한 프로토콜을 사용하는 복수의 클라이언트와 상기 각 프로토콜에 따라 상기 복수의 클라이언트와 데이터를 교환하는 응용 인터페이스부와; 상기 리더 인터페이스부를 통해 수집된 상기 스트림 태그 데이터를 관리하고, 상기 응용 인터페이스부를 통해 상기 클라이언트로부터 수신되는 필터링 조건에 따라 상기 스트림 태그 데이터를 가공 및 필터링하여 응용 데이터를 생성하며, 상기 응용 인터페이스부를 통해 상기 복수의 클라이언트 중 적어도 하나 이상에 상기 응용 데이터를 전송하는 이벤트 매니저를 포함하는 것을 특징으로 하는 RFID 미들웨어 시스템에 의해서 달성된다.

<26> 여기서, 상기 응용 인터페이스부는 JMS(Java Message Service) 프로토콜, SOAP(Simple Object Access Protocol), 및 HTTP(HiperText Transfer Protocol) 중 적어도 어느 하나로 상기 클라이언트와 통신할 수 있다.

<27> 그리고, 상기 이벤트 매니저는 상기 리더 인터페이스부를 통해 수집된 상기 스트림 태그 데이터가 저장되는 태그 저장부와, 상기 필터링 조건에 따라 상기 스트림 태그 데이터를 가공 및 필터링하여 응용 데이 터를 생성하

는 필터 매니저와, 상기 필터 매니저에 의해 생성된 상기 응용 데이터를 상기 응용 인터페이스부를 통해 해당 클라이언트로 전송하는 EM 제어부를 포함할 수 있다.

<28> 여기서, 상기 필터 매니저와 상기 각 클라이언트 간의 필터링 클래스 구조는, 상기 각 프로토콜을 이용하여 필터를 관리하는 복수의 개별 프로토콜 클래스와; 상기 각 필터에 대한 필터정보가 기록된 FilterSet 클래스와, 상기 필터정보에 대한 연산요청을 위한 API(Application Program Interface)를 갖는 FilterMgr 클래스와, 상기 각 개별 프로토콜 클래스가 상기 FilterMgr 클래스의 API(Application Program Interface)를 통해 상기 FilterSet 클래스에 동시 접속할 때 소정의 임계영역을 설정하여 접근을 제어하는 FlterServer 클래스로 구성된 공통 클래스를 포함할 수 있다.

<29> 여기서, 상기 FilterSet 클래스는 매소드가 존재하지 않는 복수의 맴버로 구성되며; 상기 FilterSet 클래스를 구성하는 상기 멤버는 상기 각 필터에 대한 필터규칙을 포함할 수 있다.

<30> 또한, 상기 FlterServer 클래스는 상기 각 개별 프로토콜 클래스가 상기 FilterMgr 클래스의 상기 API를 통해 상기 FilterSet 클래스에 접속하는 경우 쓰레드(Thread)를 생성하며; 상기 쓰레드는 상기 FilterSet 클래스의 상기 필터정보가 기록된 파일에 록(Lock)을 걸고 상기 FilterMgr 클래스를 통해 수신된 상기 연산요청을 수행할 수 있다.

<31> 한편, 상기 개별 프로토콜 클래스는 상기 JMS(Java Message Service) 프로토콜을 이용하여 상기 필터를 관리하는 JMS 클래스와, AXIS 상에서 상기 SOAP(Simple Object Access Protocol)를 이용하여 상기 필터를 관리하는 SOAP 클래스를 포함할 수 있다.

<32> 여기서, 상기 필터링 클래스 구조는 상기 JMS 클래스에 대한 상기 JMS(Java Message Service) 프로토콜을 지원하는 JMS 서비스 제공자를 더 포함하며; 상기 JMS 클래스는 상기 JMS(Java Message Service) 프로토콜을 위한 API(Application Program Interface)를 제공하며, 상기 JMS 서비스 제공자에 접속하여 상기 필터정보에 대한 연산요청을 상기 JMS 서비스 제공자에 저장하는 JMSController 클래스와, 상기 JMS 서비스 제공자에 저장된 상기 필터정보에 대한 연산요청을 독출하여 상기 공통 클래스의 상기 FilterMgr 클래스를 통해 상기 FilterSet 클래스에 전달하는 ListenerBody 클래스와, 상기 JMS 서비스 제공자에 연결되며, 상기 ListenerBody 클래스를 생성하여 비동기적으로 상기 필터정보에 대한 연산요청을 처리하는 Listener 클래스를 포함할 수 있다.

<33> 또한, 상기 SOAP 클래스는 상기 SOAP(Simple Object Access Protocol)를 위한 API(Application Program Interface)를 제공하며, 상기 AXIS에 접속해서 상기 필터정보에 대한 연산요청을 수행하는 SOAPController 클래스와, 상기 SOAPController 클래스로부터의 상기 연산요청을 상기 공통 클래스의 상기 FilterMgr 클래스를 통해 상기 FilterSet 클래스에 전달하는 FilterMgr.jws 클래스를 포함할 수 있다.

<34> 여기서, 상기 RFID 어댑터는 상기 각 RFID 리더와 TCP/IP를 통해 연결되며; 상기 리더 인터페이스부를 통해 수집된 상기 스트림 태그 데이터는 XML 파일 형태로 상기 이벤트 매니저에 전달되어 상기 태그 저장부에 저장될 수 있다.

<35> 이하에서는 도면을 참조하여 본 발명을 보다 상세하게 설명한다.

<36> 도 1은 본 발명에 따른 RFID 미들웨어 시스템(10)이 적용된 RFID 시스템의 구성을 도시한 도면이고, 도 2는 본 발명에 따른 RFID 미들웨어 시스템(10)의 구성을 도시한 도면이다.

<37> 도 1을 참조하여 설명하면, 본 발명에 따른 RFID 시스템은 복수의 RFID 리더(30a,30b,30c), RFID 미드웨어 시스템, 및 응용 서비스인 복수의 클라이언트(50a,50b,50c)를 포함한다.

<38> 본 발명에 따른 RFID 미들웨어 시스템(10)은 RFID 리더(30a,30b,30c)는 상호 상이한 기종의 RFID 리더(30a,30b,30c)로부터 스트림 태그 데이터를 수신할 수 있도록 마련된다. 본 발명에서는 Matrics 사, Alian 사, Intermec 사의 RFID 리더(30a,30b,30c)를 RFID 미들웨어 시스템(10)이 인식 가능한 것을 일 예로 하여 설명한다.

<39> 한편, RFID 미들웨어 시스템(10)은, 도 2에 도시된 바와 같이, 리더 인터페이스부(16), 응용 인터페이스부(17), 및 이벤트 매니저(11)를 포함한다.

<40> 리더 인터페이스부(16)는 네트워크(90)를 통해 RFID 리더(30a,30b,30c)와 연결되어 RFID 리더(30a,30b,30c)에 의해 읽힌 스트림 태그 데이터를 수집한다. 여기서, 리더 인터페이스부(16)는 상술한 바와 같은 상이한 기종의 RFID 리더(30a,30b,30c)와 네트워크(90)를 통해 연결되는 각 RFID 리더(30a,30b,30c)의 기종에 대응하는 RFID

어댑터(미도시)를 포함하며, 각 RFID 어댑터를 통해 수신되는 스트림 태그 데이터를 수집한다.

<41> 여기서, RFID 어댑터는 각 RFID 리더(30a,30b,30c)와 TCP/IP를 통해 연결될 수 있고, RFID 어댑터를 통해 수집되는 스트림 태그 데이터는 XML 파일 형태로 관리될 수 있다.

<42> 응용 인터페이스부(17)는 상호 상이한 프로토콜을 사용하는 복수의 클라이언트(50a,50b,50c)와 각 프로토콜에 따라 복수의 클라이언트(50a,50b,50c)와 데이터를 교환한다. 여기서, 각 클라이언트(50a,50b,50c)는 해당 프로토콜에 따른 플랫폼을 가지게 되며, 본 발명에 따른 RFID 미들웨어 시스템(10)의 응용 인터페이스부(17)는 상이한 플랫폼의 클라이언트(50a,50b,50c)의 접속을 가능하게 한다.

<43> 본 발명에서는 클라이언트(50a,50b,50c)가 JMS(Java Message Service) 프로토콜, SOAP(Simple Object Access Protocol), 및 HTTP(HiperText Transfer Protocol)에 따른 플랫폼을 갖는 것을 일 예로 하며, 이에 따라 응용 인터페이스부(17)는 JMS(Java Message Service) 프로토콜, SOAP(Simple Object Access Protocol), 및 HTTP(HiperText Transfer Protocol)에 따라 해당 클라이언트(50a,50b,50c)와 데이터를 교환한다.

<44> 한편, 이벤트 매니저(11)는 리더 인터페이스부(16)를 통해 수집된 스트림 태그 데이터를 관리한다. 그리고 이벤트 매니저(11)는 응용 인터페이스부(17)를 통해 클라이언트(50a,50b,50c)로부터 수신되는 필터링 조건에 따라 스트림 태그 데이터를 가공 및 필터링하여 응용 데이터를 생성한다. 그리고 이벤트 매니저(11)는 필터링 조건에 따라 생성된 응용 데이터를 해당 클라이언트(50a,50b,50c)로 응용 인터페이스부(17)를 통해 전송한다.

<45> 도 2를 참조하여 설명하면, 본 발명에 따른 이벤트 매니저(11)는 리더 인터페이스부(16)를 통해 수집된 스트림 태드 데이터가 저장되는 태그 저장부(12)와, 응용 인터페이스부(17)를 통해 클라이언트(50a,50b,50c)로부터 수신되는 필터링 조건에 따라 스트림 태그 데이터를 가공 및 필터링하여 응용 데이터를 생성하는 필터 매니저(14)와, 필터 매니저(14)에 의해 생성된 응용 데이터를 응용 인터페이스부(17)를 통해 해당 클라이언트(50a,50b,50c)로 전송하는 EM 제어부(13)를 포함할 수 있다.

<46> 이하에서는 도 3 내지 도 6을 참조하여 본 발명에 따른 이벤트 매니저(11)의 필터 매니저(14)가 필터에 대한 연산요청을 수행하는 방법에 대해 상세히 설명한다.

<47> 본 발명에 따른 필터 매니저(14)는 RFID 미들웨어 시스템(10)에서 사용되는 필터를 관리하는 프로그램 형태로 마련된다. 그리고, 이벤트 매니저(11)는 각 클라이언트(50a,50b,50c)의 다양한 프로토콜을 지원하여 다양한 플랫폼의 클라이언트(50a,50b,50c)가 접근 가능하도록 마련된다.

<48> 여기서, 필터 매니저(14)는 다수의 클라이언트(50a,50b,50c)가 하나의 필터 파일, 즉 태그 저장부(12)에 저장된 스트림 태그 데이터가 기록된 필터파일, 예컨대 XMLFilter.xml에 접근할 때 생기는 동기화 문제를 해결하기 위해, 도 3에 도시된 바와 같이, 각 프로토콜에 따른 클라이언트(50a,50b,50c)로부터 전송되는 필터링 조건을 포함하는 필터 제어 명령을 처리하는 개별 프로세서(14a,14b,14c)와, 필터파일을 통합 관리하며 개별 프로세서(14a,14b,14c)에 의해 처리된 필터 제어 명령에 따라 필터파일을 관리하는 통합 프로세서(14d)를 포함할 수 있다.

<49> 한편, 본 발명에 따른 RFID 미들웨어 시스템(10)에서 각 클라이언트(50a,50b,50c)의 플랫폼과 필터 매니저(14) 간의 필터링 클래스 구조를 상세히 설명하면, 필터링 클래스 구조는 각 프로토콜을 이용하여 필터를 관리하는 복수의 개별 프로토콜 클래스와, 이를 모든 프로토콜이 사용하여 이를 통합 관리하는 공통 클래스를 포함할 수 있다.

<50> 도 4는 본 발명에 따른 필터링 클래스 구조에서의 공통 클래스의 구조를 설명하기 위한 도면이다. 도 4를 참조하여 설명하면 공통 클래스는 FilterSet 클래스와, FilterMgr 클래스와, FlterServer 클래스를 포함할 수 있다.

<51> FilterSet 클래스는 각 필터에 대한 필터정보가 기록된다. FilterSet 클래스는 각 필터들을 하나의 객체로 저장하여 네트워크(90)를 통해 클라이언트(50a,50b,50c)로 전송하기 위해 사용되는 바, 등록하고자 하는 필터의 묶음으로 정의될 수 있다.

<52> 여기서, FilterSet 클래스는 매소드(Method)가 존재하지 않는 복수의 맴버로 구성되는데, 각 맴버는 TagID, reader1, readerNum, maxDiscoveryDay 등과 같이 등록하고자 하는 필터에 대한 필터규칙을 포함할 수 있다.

<53> FilterMgr 클래스는 필터정보에 대한 연산요청을 위한 API(Application Program Interface)를 포함한다. 도 4에 도시된 바와 같이, FilterMgr 클래스는 API(Application Program Interface)를 가지고 실제 필터정보가 기록되어 있는 필터파일, 즉, 전술한 바와 같이 XMLFilter.xml에 접근하는 FilterServer 클래스와 통신한다.

<54> 여기서, XMLFilter.xml은 여러 개의 파일이 동시에 접근할 수 있으므로 FilterServer 클래스가 임계영역을 설정하고 접근을 제어한다. 이에 따라, FilterServer 클래스는 FilterMgr 클래스에서 제공하는 함수를 가지고, XMLFilter.xml로의 동시 접근에 따른 파일 손상을 제거하고, 간단히 필터파일에 대한 필터정보의 기록,삭제, 읽기 등의 연산요청을 수행할 수 있게 된다.

<55> 아래의 [표 1]은 FilterMgr 클래스의 매소드들의 예를 나타낸 것이다. [표 1]에 나타낸 함수들을 이용하여 FilterMgr 클래스에 필터정보를 설정하고, FM_XMLInsert, FM_XMLDelete, FM_GetXMLFile 함수를 이용하여 필터정보를 제어할 수 있다.

<56> [표 1]

<57>

매소드	역 할
setTagID	(필더설정) TagID를 설정한다.
setReader1	(필더설정) Reader1을 설정한다.
setReaderNum	(필더설정) ReaderNum을 설정한다.
setMinDiscoveryDay	(필더설정) MinDiscoveryDay를 설정한다.
setMinDiscoveryTime	(필더설정) MinDiscoveryTime을 설정한다.
setMaxDiscoveryDay	(필더설정) MaxDiscoveryDay를 설정한다.
setMaxDiscoveryTime	(필더설정) MaxDiscoveryTime을 설정한다.
setAntenna	(필더설정) Antenna를 설정한다.
setRemoteAddr	(필더설정) RemoteAddr를 설정한다.
FM_XMLInsert	(API) 설정된 필터정보를 기록한다.
FM_XMLDelete	(API) 설정된 필터정보를 삭제한다.
FM_GetXMLFile	(API) 설정된 필터정보를 String으로 얻는다.

<58> FilterServer 클래스는 필터정보를 저장하는 FMLFilter.xml을 여러 개의 개별 프로토콜 클래스가 동시에 접근할 때, 필터파일의 손상을 막기 위해 소켓을 사용하여 해당 연산요청을 받아들이고, 필터파일로의 접근시 임계영역을 설정하여 동기화를 지원한다. 도 4에 도시된 바와 같이, FilterServer 클래스는 FilterMgr 클래스를 이용해서 접속이 오면 쓰레드를 하나 생성하고, 개별 쓰레드는 연산요청에 따라 필터파일에 록(Lock)을 걸고 연산요청에 따른 작업을 수행한다.

<59> 한편, 개별 프로토콜 클래스는 JMS(Java Message Service) 프로토콜을 이용하여 필터를 관리하는 JMS 클래스와, AXIS 상에서 SOAP(Simple Object Access Protocol)를 이용하여 필터를 관리하는 SOAP 클래스를 포함할 수 있다.

<60> 도 5는 본 발명에 따른 JMS 클래스의 구성을 설명하기 위한 도면이다. 도면에 도시된 JMS 클래스는 JMSController 클래스와, ListenerBody 클래스와, Listener 클래스를 포함할 수 있다.

<61> 여기서, 본 발명에 따른 필터 매니저(14)는 JMS 클래스에 대한 JMS(Java Message Service) 프로토콜을 지원하는 JMS 서비스 제공자를 포함할 수 있는데, 본 발명에서는 JMS 서비스 제공자로, 도 5에 도시된 바와 같이, OPENJMS를 사용하는 것을 일 예로 한다.

<62> 도 5를 참조하여 JMS 클래스의 처리 과정을 전체적으로 요약하면, JMSController 클래스에서 API를 이용하여 필터에 대한 연산요청을 수행한다. 이 연산요청은 OPENJMS의 testQueue에 쌓이게 되고, Listener 클래스에서 읽어와 공통 클래스의 FilterMgr 클래스를 이용하여 실제 필터파일인 XMLFilter.xml에 접속한다. FilterMgr 클래스에 의해 접속된 필터파일은 OPENJMS의 InterMiddlewareQueue에 쌓이게 되고, JMSController 클래스에서 최종적으로 스트링으로 XMLFilter.xml의 내용을 받을 수 있게 된다.

<63> JMSController 클래스는 JMS(Java Message Service) 프로토콜을 위한 API(Application Program Interface)를 제공하며, JMS 서비스 제공자인 OPENJMS에 접속하여 상기 필터정보에 대한 연산요청을 전달하고, 이는 OPENJMS의 testQueue에 저장된다. 여기서, JMSController 클래스는 클라이언트(50a,50b,50c)의 플랫폼 상에서 동작한다.

<64> 아래의 [표 2]는 JMSController 클래스의 메소드 및 그 역할의 일 예를 나타낸 것이다.

<65> [표 2]

<66>

매소드	역 할
JMS_Connect	JMS에 접속한다.
JMS_XMLInsert	필터정보를 저장한다.(FilterSet 이용)
JMS_XMLDelete	필터정보를 삭제한다.(FilterSet 이용)
JMS_GetXML	필터정보를 String으로 얻는다.
JMS_Exit	JMS에 접속 종료 요청

<67> Listener 클래스는 OPENJMS에 연결되며, ListenerBody 클래스를 생성하여 비동기적으로 연산요청을 처리한다. 그리고, ListenerBody 클래스는 실질적으로 연산요청을 처리하며, Listener 클래스에서 비동기적인 연산요청의 처리를 위해 해당 클래스를 사용한다. ListenerBody 클래스는 OPENJMS의 testQueue에서 JMSController 클래스가 요청한 연산요청을 받아서 연산요청에 따른 작업을 수행한다. 여기서, ListenerBody 클래스는 내부적으로 FilterMgr 클래스의 객체를 소유하고, 이를 이용하여 연산요청을 수행하게 된다.

<68> 한편, 도 6은 본 발명에 따른 SOAP 클래스의 구성을 설명하기 위한 도면이다. 본 발명에 따른 SOAP 클래스는 AXIS에서 SOAP(Simple Object Access Protocol)을 이용하여 필터를 관리한다. 도 6에 도시된 바와 같이, SOAP 클래스는 SOAP(Simple Object Access Protocol)의 통신 방식인 XML 방식을 이용하여 아파치(도 6의 'Apache tomcat') 상에서 구동되는 AXIS에 데이터를 전송한다. 이는 XML로 직접 통신이 가능하고, 제공하는 API(Application Program Interface)로도 통신이 가능하다.

<69> 또한, SOAP 클래스는, 도 6에 도시된 바와 같이, SOAPController 클래스와, FilterMgr.jws 클래스를 포함할 수 있다.

<70> SOAPController 클래스는 SOAP(Simple Object Access Protocol)를 위한 API(Application Program Interface)를 제공하며, AXIS에 접속해서 필터정보에 대한 연산요청을 수행한다. 아래의 [표 3]은 SOAPController 클래스의 메소드와 그 역할의 일 예를 나타낸 것이다. SOAPController 클래스는 클라이언트(50a,50b,50c)의 플랫폼 상에서 동작한다.

<71> [표 3]

<72>

매소드	역 할
SOAP_XMLInsert	필터정보를 저장한다.(FilterSet 이용)
SOAP_XMLDelete	필터정보를 삭제한다.(FilterSet 이용)
SOAP_GetXML	필터정보를 String으로 얻는다.

<73> FilterMgr.jws 클래스는 필터정보에 대한 연산요청을 수행하는 API(Application Program Interface)를 포함하고, 이는 전술한 공통 클래스의 FilterMgr와 유사한 역할을 하지만 AXIS 환경을 위하여 변경된 것이다. 이에 따라, 공통 클래스의 FilterMgr 클래스와 FilterMgr.jws 클래스에서 처리되는 파일은 FilterMgr.java와 FilterMgr.jws로 구분될 수 있다. 정리하면, 공통 클래스의 FilterMgr 클래스는 HTTP(HiperText Transfer Protocol)와 JMS(Java Message Service) 프로토콜을 위해 사용되고, FilterMgr.jws 클래스는 SOAP(Simple Object Access Protocol)를 위해 사용된다.

<74> 상기와 같은 필터링 클래스 구조를 통해 필터 매니저(14)는 각 클라이언트(50a,50b,50c)로부터의 필터정보에 대한 필터링 조건에 따라 스트림 태그 데이터로부터 응용 데이터를 생성하고, EM 제어부(13)는 필터 매니저(14)에 의해 생성된 응용 데이터를 해당 클라이언트(50a,50b,50c)로 전송하게 된다.

<75> 다시, 도 2를 참조하여 설명하면, 본 발명에 따른 이벤트 매니저(11)는 EM 제어부(13)의 제어에 따라 모니터(70) 상에 그래픽 유저 인터페이스(Graphic User Interface) 화면을 표시하는 GUI 생성부(15)를 더 포함할 수 있다.

<76> 도 7 내지 도 11은 본 발명에 따른 GUI 생성부(15)가 제공하는 그래픽 유저 인터페이스 화면의 예들을 도시한 도면이다.

<77> 도 7은 리더 인터페이스부(16)의 어댑터에 접속된 RFID 리더(30a,30b,30c)의 접속 상황을 나타낸 화면의 예를 도시한 도면이다. 도면에 도시된 바와 같이, 화면에는 RFID 리더(30a,30b,30c)의 종류, IP 주소, 해당 어댑터

의 포트 번호, On/Off 상태 등이 표시된다.

<78> 또한, 화면의 하단에 마련된 선택 버튼 중 '정보추가', '정보수정', '모두삭제' 버튼을 이용하여 RFID 리더(30a,30b,30c)의 정보를 파일로 관리할 수 있으며, '선택접속/해제'버튼을 통해 RFID 리더(30a,30b,30c)를 개별적으로 접속 및 접속해제시킬 수 있다.

<79> 도 8은 도 7의 화면을 통해 접속 설정이 된 상태에서 접속된 RFID 리더(30a,30b,30c)로부터 수집된 스트림 태그 데이터가 화면상에 표시되는 일 예를 도시한 도면이다.

<80> 도 9는 도 7의 화면 상에서 새로운 RFID 리더(30a,30b,30c)를 추가하는 과정을 설명하기 위한 도면으로, '정보추가'버튼을 선택하는 경우 '리더기 정보 추가'화면이 표시되고 이를 통해 RFID 리더(30a,30b,30c)의 종류, IP 주소 및 포트 등을 선택함으로써 새로운 RFID 리더(30a,30b,30c)의 추가가 가능하게 된다. 또한, 도 10은 도 8에서 RFID 리더(30a,30b,30c)가 안테나가 분리형인 경우 안테나 별도 태그 읽기를 선택할 수 있도록 하는 화면의 예를 도시한 도면이다.

<81> 한편, 도 11은 특정 필터링 조건에 따라 필터링되어 화면 상에 표시되는 스트림 태그 데이터의 예를 도시한 도면이다. 여기서, 도 11은 TagID, RFID 리더(30a,30b,30c)의 종류(도 11의 'Reader'항목의 'Alien'), 기간(도 11의 'Min_Day' 및 'Max_Day' 항목)이 필터링 조건으로 설정되어, 해당 필터링 조건에 해당하는 스트림 태그 데이터가 실시간으로 추출되어 화면 상에 표시되는 예를 도시한 도면이다.

<82> 한편, 도 12는 본 발명에 따른 RFID 시스템의 클라이언트(50a,50b,50c)에 표시되는 그래픽 유저 인터페이스 화면의 예를 도시한 도면이다.

<83> 도면에 도시된 바와 같이, 클라이언트(50a,50b,50c)에 표시되는 화면을 통해서는, API(Application Program Interface)의 설정, 예컨대, 전술한 바와 같이 JMS(Java Message Service) 프로토콜, SOAP(Simple Object Access Protocol), 및 HTTP(HiperText Transfer Protocol) 중 해당 클라이언트(50a,50b,50c)가 지원하는 프로토콜을 선택하고, TagID, RFID 리더(30a,30b,30c) 종류의 선택, 기간의 설정 및 안테나 선택 등을 선택함으로써 필터링 조건을 선택하기 위한 화면이 표시된다.

<84> 그리고, 도 12에 도시된 화면을 통해 설정된 필터링 조건은 RFID 미들웨어 시스템(10)에 전송되고, RFID 미들웨어 시스템(10)의 필터 매니저(14)에 의해 해당 필터링 조건에 맞게 스트림 태그 데이터 중 추출되어 응용 데이터로 생성된다. 그리고, 응용 데이터는 다시 클라이언트(50a,50b,50c)로 전송되어 도 13에 도시된 바와 같이 표시된다.

<85> 이상에서 본 발명의 바람직한 실시예에 대하여 상세하게 설명하였지만 본 발명의 권리범위는 이에 한정되는 것은 아니고 다음의 청구범위에서 정의하고 있는 본 발명의 기본 개념을 이용한 당업자의 여러 변형 및 개량 형태 또한 본 발명의 권리범위에 속하는 것이다.

발명의 효과

<86> 이상 설명한 바와 같이, 본 발명에 따르면, 본 발명에 따르면, 상호 상이한 복수의 프로토콜에 따른 플랫폼의 접근이 가능하며, 실시간으로 대용량의 데이터 처리가 가능하게 된다.

<87> 또한, 본 발명에 따르면, 인프라스트럭처, 즉 응용 서비스인 클라이언트가 사용하는 프로토콜의 상이함에 따른 다양한 플랫폼에 대한 확장성 및 그 통합을 지원하면서도 대용량의 데이터 처리가 가능하게 된다.

도면의 간단한 설명

<1> 도 1은 본 발명에 따른 RFID 미들웨어 시스템이 적용된 RFID 시스템의 구성을 도시한 도면이고,

<2> 도 2는 본 발명에 따른 RFID 미들웨어 시스템의 구성을 도시한 도면이고,

<3> 도 3은 본 발명에 따른 RFID 미들웨어 시스템의 필터 매니저와 클라이언트 간의 데이터 교환을 설명하기 위한 도면이고,

<4> 도 4는 본 발명에 따른 필터링 클래스 구조에서의 공통 클래스의 구조를 설명하기 위한 도면이고,

<5> 도 5는 본 발명에 따른 JMS 클래스의 구성을 설명하기 위한 도면이고,

<6> 도 6은 본 발명에 따른 SOAP 클래스의 구성을 설명하기 위한 도면이고,

<7> 도 7 내지 도 11은 본 발명에 따른 RFID 미들웨어 시스템의 GUI 생성부가 제공하는 그래픽 유저 인터페이스 화면의 예들을 도시한 도면이고,

<8> 도 12 및 도 13은 본 발명에 따른 RFID 시스템의 클라이언트에 표시되는 그래픽 유저 인터페이스 화면의 예들을 도시한 도면한 도면이다.

<9> <도면의 주요 부분에 대한 부호의 설명>

<10> 10 : RFID 미들웨어 시스템 11 : 이벤트 매니저

<11> 12 : 태그 저장부 13 : EM 제어부

<12> 14 : 필터 매니저 15 : GUI 생성부

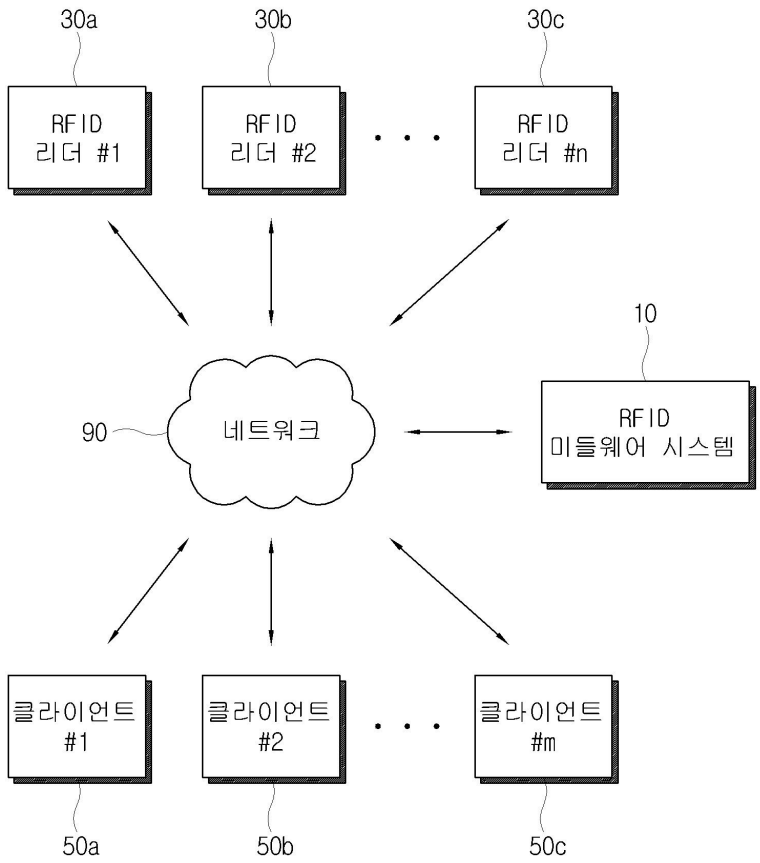
<13> 16 : 리더 인터페이스부 17 : 응용 인터페이스부

<14> 30a,30b,30c : RFID 리더 50a,50b,50c : 클라이언트

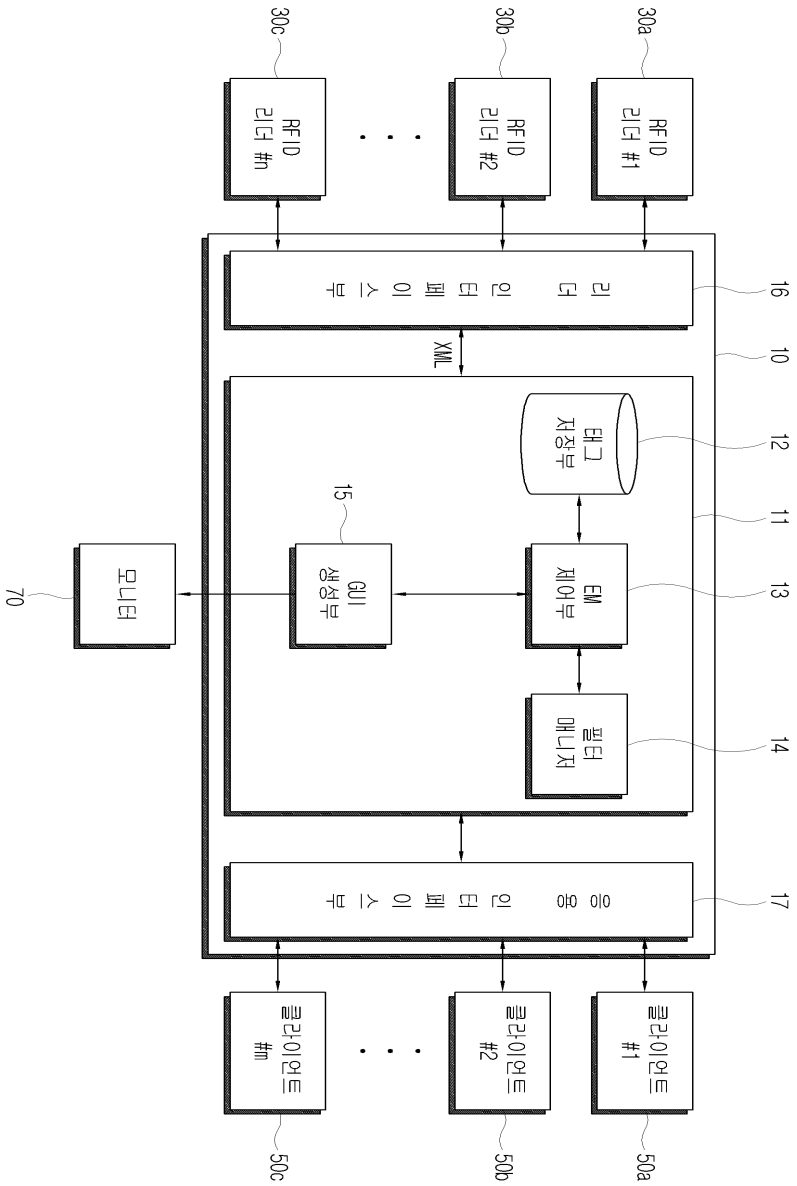
<15> 70 : 모니터

도면

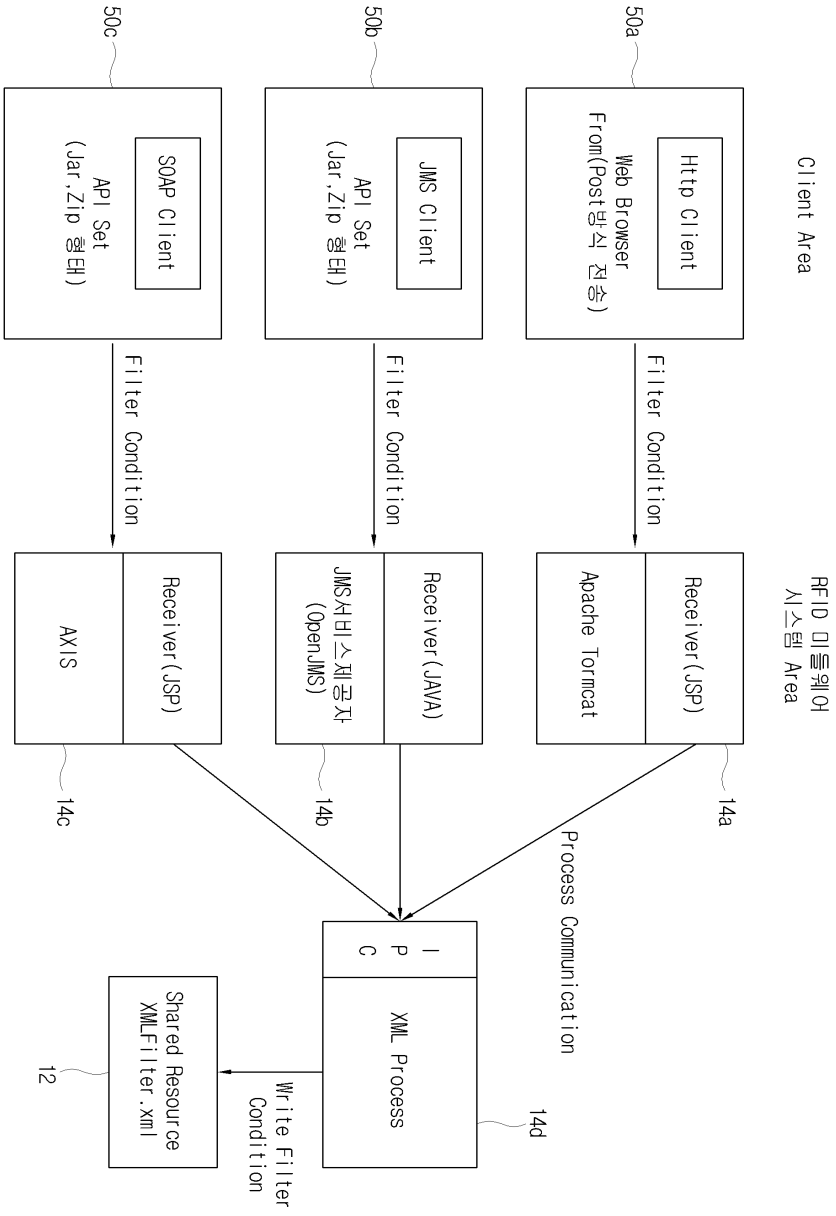
도면1



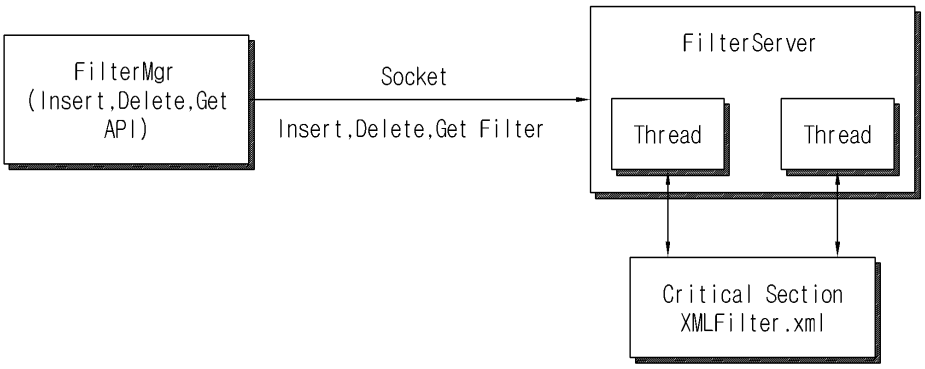
도면2


30c
30b
30a
RFID 리더 #n
RFID 리더 #2
RFID 리더 #1
리더 인터페이스부
16
XML
10
태그 저장부
12
15
11
모니터
70
GUI 생성부
EM 제어부
13
필터 매니저
14
애플리케이션 인터페이스부
17
클라이언트 #m
클라이언트 #2
클라이언트 #1
50c
50b
50a

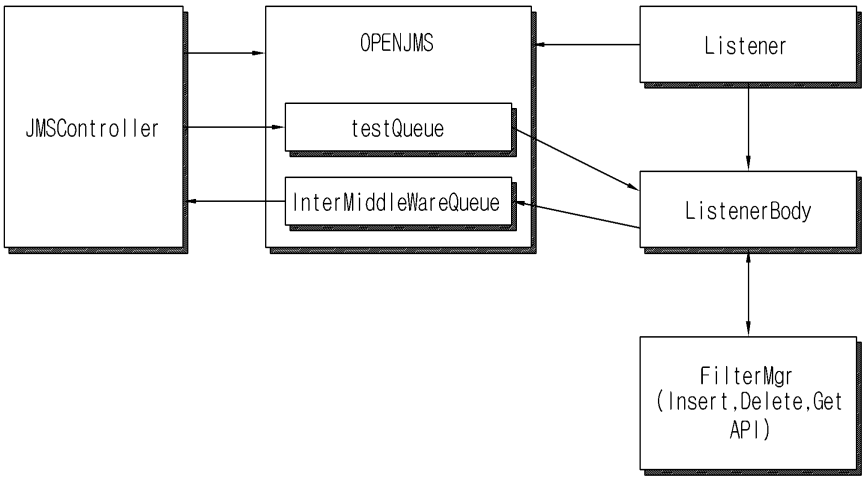
도면3


Client Area
RFID 미들웨어 시스템 Area
50a
Http Client
Web Browser From(Post방식 전송)
50b
JMS Client
API Set (Jar,Zip 형태)
50c
SOAP Client
API Set (Jar,Zip 형태)
Filter Condition
Filter Condition
Filter Condition
Receiver(JSP)
Apache Tormcat
Receiver(JAVA)
JMS서비스제공자 (OpenJMS)
Receiver(JSP)
AXIS
14a
14b
14c
Process Communication
IPC
XML Process
14d
Write Filter Condition
Shared Resource XMLFilter.xml
12

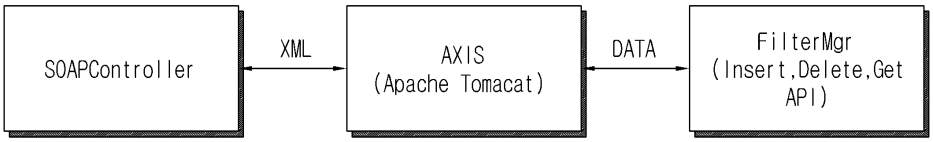
도면4


FilterMgr
(Insert,Delete,Get
API)
Socket
Insert,Delete,Get Filter
FilterServer
Thread
Thread
Critical Section
XMLFilter.xml

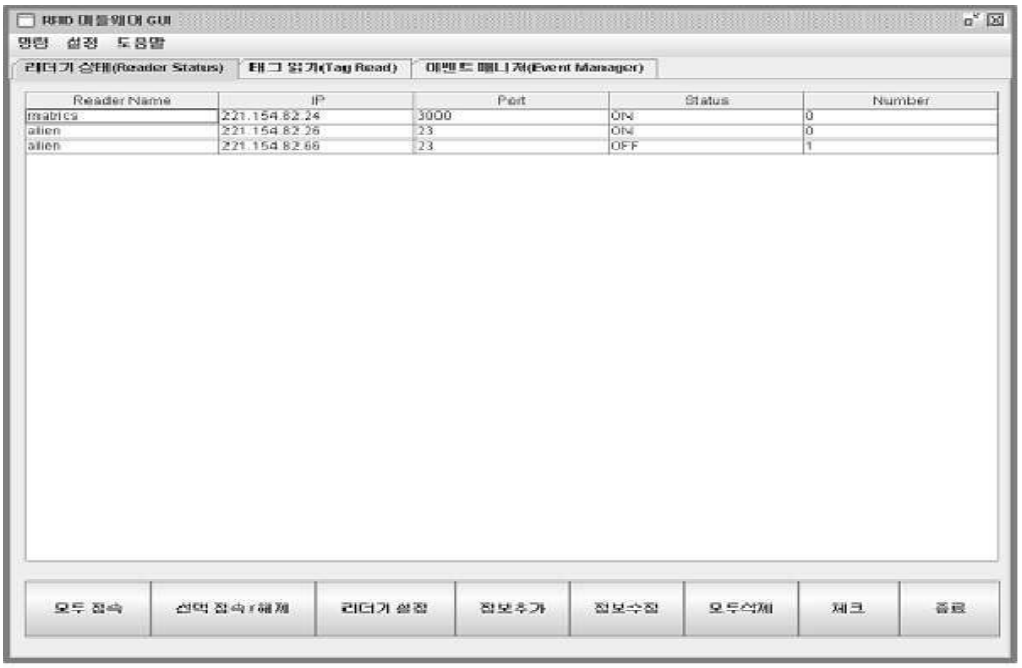
도면5


JMSController
OPENJMS
testQueue
InterMiddleWareQueue
Listener
ListenerBody
FilterMgr
(Insert,Delete,Get
API)

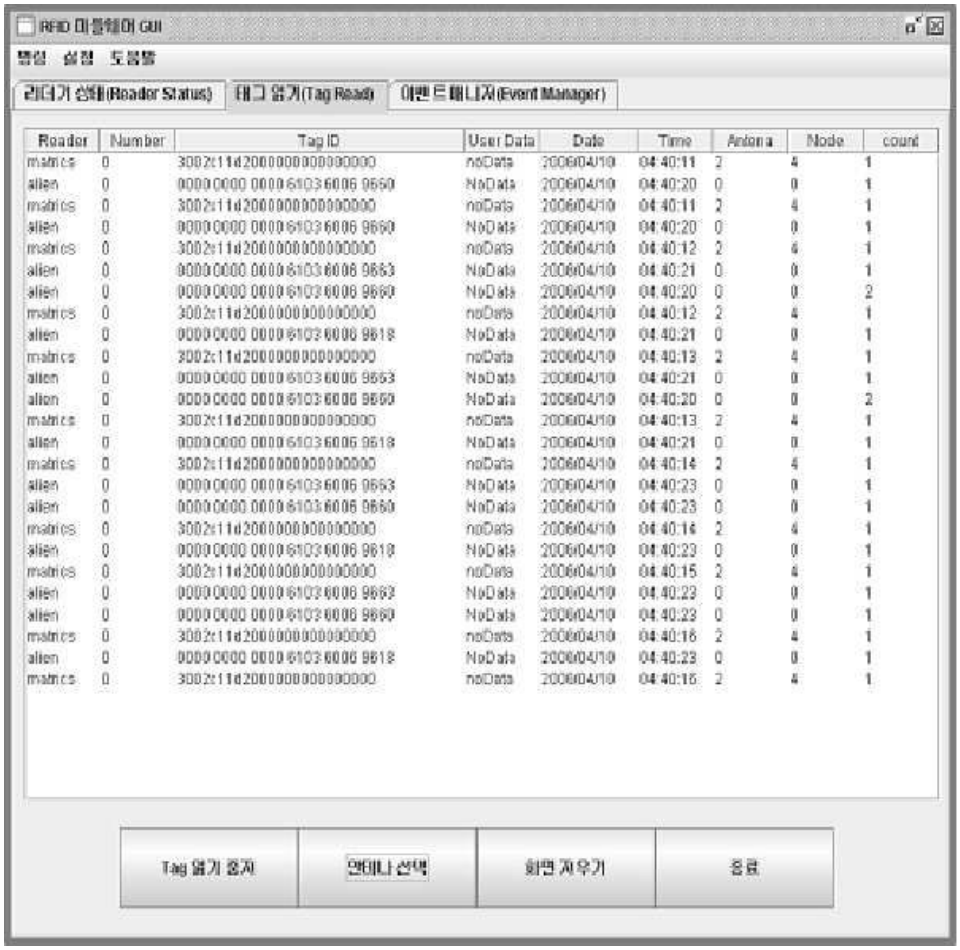
도면6


SOAPController
XML
AXIS
(Apache Tomacat)
DATA
FilterMgr
(Insert,Delete,Get
API)

도면7


RFID 미들웨어 GUI
명령 설정 도움말
리더기 상태(Reader Status)
태그 읽기(Tag Read)
이벤트 매니저(Event Manager)
Reader Name
IP
Port
Status
Number
matrics
221.154.82.24
3000
ON
0
alien
221.154.82.26
23
ON
0
alien
221.154.82.66
23
OFF
1
모두 접속
선택 접속/해제
리더기 설정
정보추가
정보수정
모두삭제
체크
종료

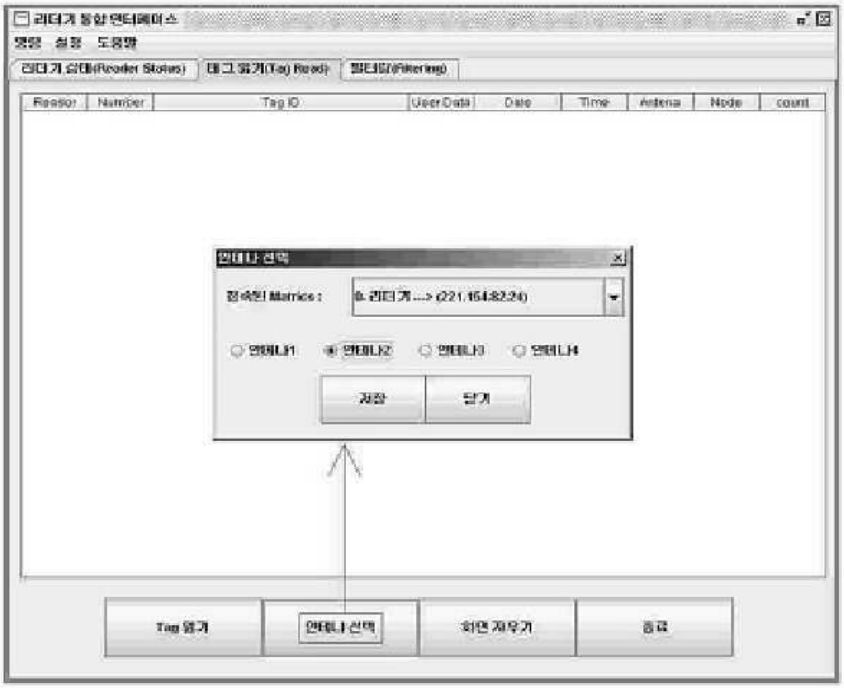
도면8


RFID 미들웨어 GUI
명령 설정 도움말
리더기 상태(Reader Status)
태그 읽기(Tag Read)
이벤트 매니저(Event Manager)
Reader
Number
Tag ID
User Data
Date
Time
Antenna
Node
count
Tag 읽기 중지
안테나 선택
화면 지우기
종료

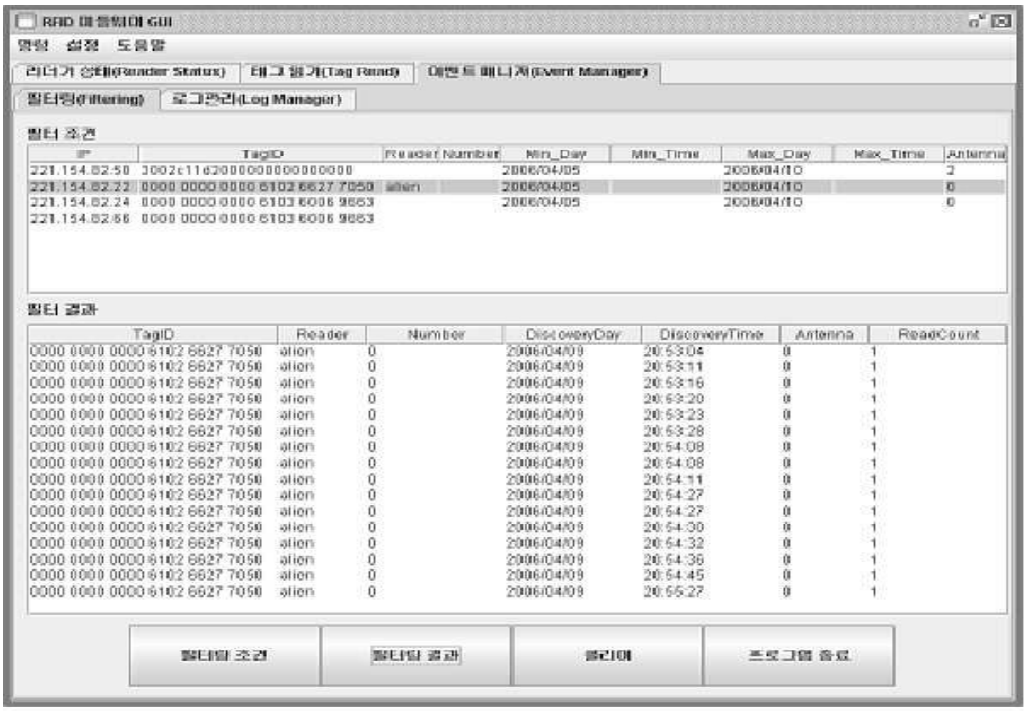
도면9


Reader Name
IP
Port
Status
Number
ALIEN
MATRICS
221.154.82.24
3009
확인
취소
추가되었습니다.
확인

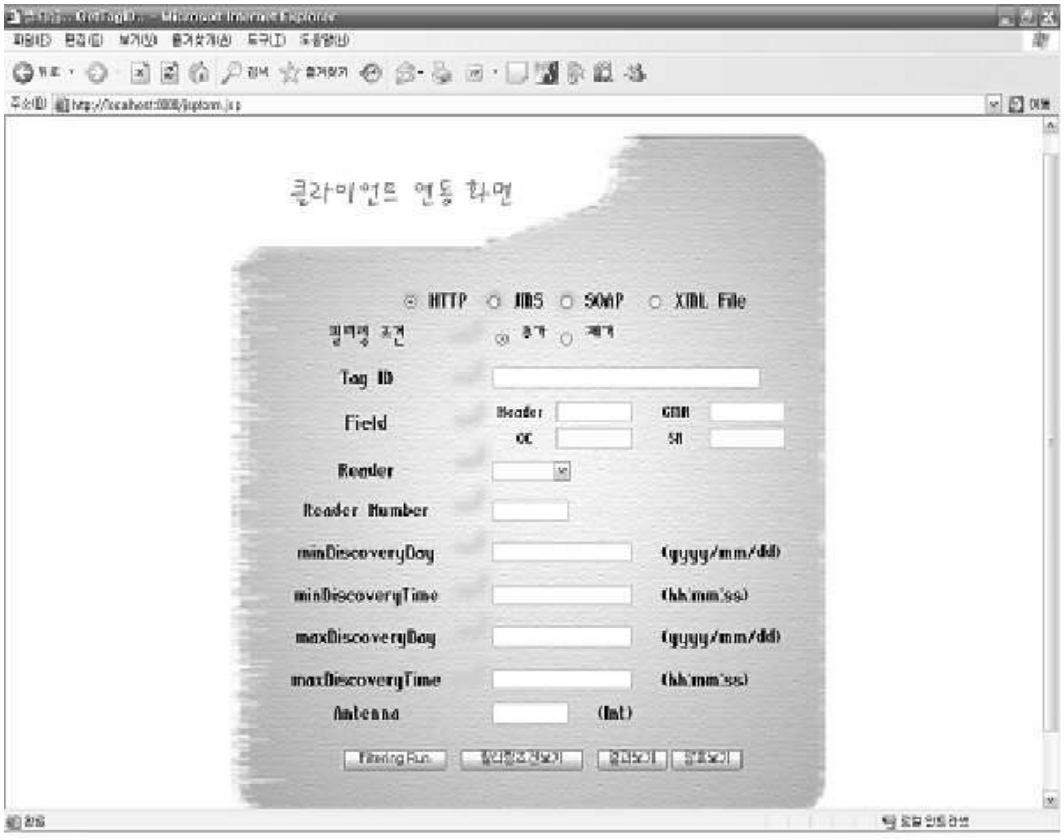
도면10


Reader
Number
Tag ID
User Data
Date
Time
Antena
Node
count
확인
취소

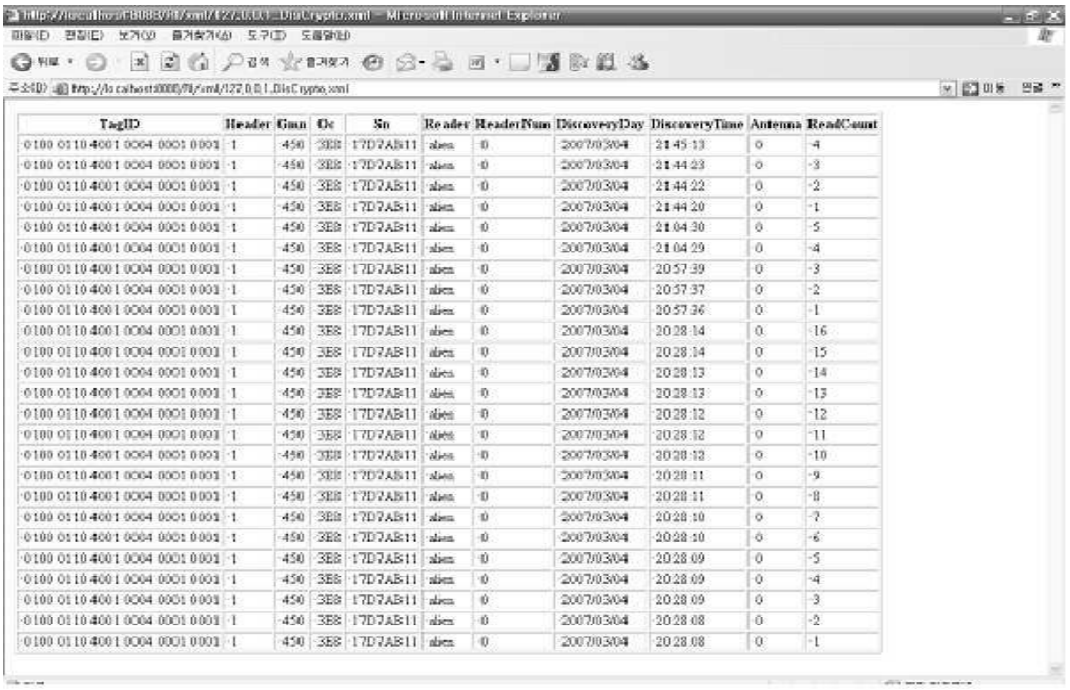
도면11


RFID 미들웨어 GUI
명령 설정 도움말
리더기 상태(Reader Status)
태그 읽기(Tag Ready)
이벤트 매니저(Event Manager)
필터링(Filtering)
로그관리(Log Manager)
필터 조건
TagID
Reader Number
Min_Day
Min_Time
Max_Day
Max_Time
Antenna
필터 결과
TagID
Reader
Number
DiscoveryDay
DiscoveryTime
Antenna
ReadCount
필터링 조건
필터링 결과
클리어
프로그램 종료

도면12


클라이언트 연동 화면
HTTP
JMS
SOAP
XML File
필터링 조건
Tag ID
Field
Reader
Reader Number
minDiscoveryDay
minDiscoveryTime
maxDiscoveryDay
maxDiscoveryTime
Antenna
(yyyy/mm/dd)
(hh:mm:ss)
(int)
Filtering Run

도면13



TagID	Header	Gmn	Oc	Sn	Reader	ReaderNum	DiscoveryDay	DiscoveryTime	Antenna	ReadCount
0100 0110 4001 0004 0001 0001	1	450	3E8	17D7AB11	alien	0	2007/03/04	21:45:13	0	-4
0100 0110 4001 0004 0001 0001	1	450	3E8	17D7AB11	alien	0	2007/03/04	21:44:23	0	-3
0100 0110 4001 0004 0001 0001	1	450	3E8	17D7AB11	alien	0	2007/03/04	21:44:22	0	-2
0100 0110 4001 0004 0001 0001	1	450	3E8	17D7AB11	alien	0	2007/03/04	21:44:20	0	-1
0100 0110 4001 0004 0001 0001	1	450	3E8	17D7AB11	alien	0	2007/03/04	21:04:30	0	-5
0100 0110 4001 0004 0001 0001	1	450	3E8	17D7AB11	alien	0	2007/03/04	21:04:29	0	-4
0100 0110 4001 0004 0001 0001	1	450	3E8	17D7AB11	alien	0	2007/03/04	20:57:39	0	-3
0100 0110 4001 0004 0001 0001	1	450	3E8	17D7AB11	alien	0	2007/03/04	20:57:37	0	-2
0100 0110 4001 0004 0001 0001	1	450	3E8	17D7AB11	alien	0	2007/03/04	20:57:36	0	-1
0100 0110 4001 0004 0001 0001	1	450	3E8	17D7AB11	alien	0	2007/03/04	20:28:14	0	-16
0100 0110 4001 0004 0001 0001	1	450	3E8	17D7AB11	alien	0	2007/03/04	20:28:14	0	-15
0100 0110 4001 0004 0001 0001	1	450	3E8	17D7AB11	alien	0	2007/03/04	20:28:13	0	-14
0100 0110 4001 0004 0001 0001	1	450	3E8	17D7AB11	alien	0	2007/03/04	20:28:13	0	-13
0100 0110 4001 0004 0001 0001	1	450	3E8	17D7AB11	alien	0	2007/03/04	20:28:12	0	-12
0100 0110 4001 0004 0001 0001	1	450	3E8	17D7AB11	alien	0	2007/03/04	20:28:12	0	-11
0100 0110 4001 0004 0001 0001	1	450	3E8	17D7AB11	alien	0	2007/03/04	20:28:12	0	-10
0100 0110 4001 0004 0001 0001	1	450	3E8	17D7AB11	alien	0	2007/03/04	20:28:11	0	-9
0100 0110 4001 0004 0001 0001	1	450	3E8	17D7AB11	alien	0	2007/03/04	20:28:11	0	-8
0100 0110 4001 0004 0001 0001	1	450	3E8	17D7AB11	alien	0	2007/03/04	20:28:10	0	-7
0100 0110 4001 0004 0001 0001	1	450	3E8	17D7AB11	alien	0	2007/03/04	20:28:10	0	-6
0100 0110 4001 0004 0001 0001	1	450	3E8	17D7AB11	alien	0	2007/03/04	20:28:09	0	-5
0100 0110 4001 0004 0001 0001	1	450	3E8	17D7AB11	alien	0	2007/03/04	20:28:09	0	-4
0100 0110 4001 0004 0001 0001	1	450	3E8	17D7AB11	alien	0	2007/03/04	20:28:09	0	-3
0100 0110 4001 0004 0001 0001	1	450	3E8	17D7AB11	alien	0	2007/03/04	20:28:08	0	-2
0100 0110 4001 0004 0001 0001	1	450	3E8	17D7AB11	alien	0	2007/03/04	20:28:08	0	-1