

# RFID 미들웨어 시스템을 위한 가용성 관리 설계 방안

박재걸<sup>o</sup> 하성호 재흥석

부산대학교 컴퓨터공학과

{jgpark<sup>o</sup>, shha, hschae}@pusan.ac.kr

## A Design Approach to Managing Availability of RFID Middleware System

Jaegeol Park<sup>o</sup>, Sungho Ha, Heung Seok Chae

Department of Computer Science and Engineering, Pusan National University

### 요 약

RFID기술의 발전과 더불어 RFID 미들웨어 시스템의 중요성은 커지고 있으나 가용성 제공 수준은 아직 미흡하다. 현재 출시된 RFID 미들웨어 시스템은 가용성을 제공한다 하더라도 범위가 한정되거나 범용성이 없는 자사 기술 집약적인 형태를 보인다. 현 기술 현황으로는 다양한 도메인과 다양한 수준에서의 가용성을 보장하기란 한계가 있다. 본 논문에서는 가용성 관리를 위한 프레임워크를 제공함으로써 프레임워크 기반으로 RFID 미들웨어 시스템의 가용성을 보장하는 방안을 제시한다. 가용성 관리 프레임워크는 가용성 보장을 위한 기본 로직을 프레임워크를 통해 제공하므로 RFID 미들웨어 시스템은 원하는 수준의 가용성을 보장 받을 수 있다. 가용성 관리 프레임워크를 통한 가용성 제공은 재사용성, 확장성, 유연성, 이식성의 장점을 지니며 가용성을 보장하는 RFID 미들웨어 시스템 구현에 대한 용이성을 제공할 것이다.

### 1. 서 론

RFID 미들웨어는 RFID태그 리더로부터 수집된 데이터를 통합, 여과, 기록하는 핵심요소이다. RFID 리더 또는 리더 네트워크와 비즈니스 정보 시스템 사이의 데이터 교환이 가능하도록 해준다. 리더가 읽은 실시간 이벤트 데이터와 정보를 관리하고 문제가 발생하면 경고를 제공하는 역할도 하며 EPC Information Services(이하 EPCIS) 또는 다른 정보 시스템과의 의사소통이 가능하도록 하는 중추 역할을 한다[1]. RFID 어플리케이션은 RFID 미들웨어를 통해 비즈니스를 위한 핵심적이고 중요한 정보를 수집하여 비즈니스 프로세스를 자동화 하게 된다.

RFID시스템에서 RFID 미들웨어 시스템의 가용성[2]을 보장하지 못할 경우 해당 시스템은 물론 모든 비즈니스 플로우에 문제가 발생할 수 있다. RFID미들웨어의 수행이 중단 된다면 RFID 리더를 통해 수집된 데이터가 RFID 미들웨어로 전달되지 않음은 물론 RFID 어플리케이션은 불완전한 데이터를 근거로 비즈니스 프로세스를 수행하게 된다. 이러한 처리과정을 통해 도출된 비즈니스 프로세스의 산출물은 그 결과를 보장할 수 없음은 물론 더 큰 문제를 야기 시킨다.

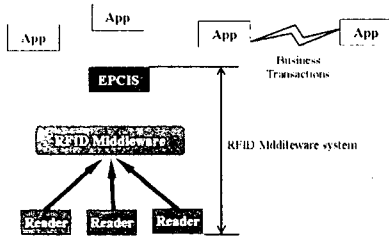


그림 1. RFID 미들웨어 시스템

본 논문은 다음과 같이 구성된다. 2절에서는 현재 출시된 RFID 미들웨어 제품의 가용성 제공 수준에 대한 현황을 소개한다. 3절에서는 RFID 미들웨어 시스템의 가용성을 보장하기 위한 가용성 관리 설계 방안을 제시한다. 4절에서는 제시된 가용성 관리 설계 방안의 장점과 전망에 대해 설명한다.

### 2. RFID 미들웨어 시스템의 가용성 현황

본 절에서는 RFID 미들웨어를 제공하고 있는 벤더들의 출시 제품을 가용성 측면에서 소개한다. 각 벤더들의 웹 사이트로부터 수집된 RFID 미들웨어 제품에 대한 소개 자료를 바탕으로 분석을 수행하였다.

이 논문은 교육인적자원부 지방연구중심대학육성사업 (차세대물류IT기술연구사업단)의 지원에 의하여 연구되었음.

- Sun Java System RFID Software[3] : Jini 기술을 이용해서 Event Manager의 장애를 감시한다. 장애가 탐지된 경우 장애가 발생한 Event Manager를 대신할 Event Manager를 선정하여 사용하는 방식으로 가용성을 제공한다. 자바 플랫폼을 기반으로 동작하는 미들웨어 시스템만이 가용성을 제공 받을 수 있다.
- TagsWare[4] : CapTech사의 제품으로 여러 제조사의 RFID 장치를 쉽게 통합 할 수 있도록 인터페이스를 추상화 하여 제공한다. 다양한 장치의 동작 상태에 대한 감시가 아닌 RFID 리더에 대한 모니터링만 제공하는 등 제한된 수준의 가용성을 제공한다.
- RFTagAware[5] : ConnecTerra사의 제품으로 안정적인 수행을 위해 RFID리더와 EdgeServer를 관리하기위한 중앙집중식 모니터링 시스템을 제공한다. 중앙 집중식 감시와 관리를 통해 가용성을 제공하고 있다.
- Singularity[6] : SCM(Supply Chain Management)과 EPCglobal Network에 사용하기 위한 RFID 미들웨어 기술 개발을 위한 오픈소스 프로젝트이다. Jini 기술을 이용하여 네트워크 또는 컴포넌트에 문제가 생기더라도 이벤트 정보를 잃지 않도록 하고, Fault tolerant를 제공하여 가용성을 제공할 예정이다. 가용성 제공이 자바 플랫폼에 종속된다.
- Manhattan Associates[7] : 유연성, 신뢰성, 확장성에 중점을 두고 있다. 보충 하드웨어를 두고 핸드온 트레이닝 코스를 통해 지속적인 서비스가 이루어질 수 있도록 하겠다는 의지를 나타낼 뿐 가용성에 대한 구체적인 방안을 제시하지 못하고 있다.
- OAT Foundation Suite[8] : 중앙관리를 위한 OATaxiom, RFID 배치에 대한 Best practice를 관장하는 OATxpress, Drag-and-drop 유저 인터페이스로 표준 빌딩 블록을 제공하는 OATlogic, EPC 넘버 관리를 위한 OATepc, 확실한 데이터 캡처를 위한 OATwm으로 구성된다. OATxpress를 통해 제공하는 Device monitoring dashboard를 통해 RFID 시스템 하부구조의 상태를 감시하고 장애 발생시 경고 기능을 제공한다. 장애 탐지 및 장애 보고에 중점을 두는 반면 장애 복구에 대한 정책이 결여되어있다.
- URIS[9] : 국내의 RFID 시스템에 관련된 다수의 프로젝트를 수행한 국내 기업인 (주)윙릭스사의 제품이다. ERP, SCM, WMS, CRM 등 기존 엔터프라이즈 어플리케이션과의 연동 측면에서 우수한 평가를 받고 있으나 가용성을 보장하기 위한 기술에 대해서는 특별한 진척 사항이 없다.

대부분의 RFID 미들웨어 제품은 자사 제품이 확장성과 성능이 우수하다는 측면을 강조한 반면 가용성 보장 수준은 미흡하거나 제한된 수준에서 행해진다. 이는 가용성이 RFID 미들웨어 시스템에서 중요한 특징이 아니라고 보다는 해당 미들웨어 벤더가 아직 가용성 부문에 많은 투자를 하지 못하고 있는 것이다. 원하는 시점에 항상 서비스를 제공할 수 있는 높은 가용성을 제공해야 하는 것은 미들웨어에 있어서 기본적으로 요구되는 특징이다.

### 3. 가용성 관리 프레임워크

본 절에서는 RFID 미들웨어 시스템의 가용성을 보장하기 위한 방법으로서 가용성 관리 프레임워크(Availability Management Framework, 이하 AMF)를 소개한다.

3.1 AMF의 개요

AMF는 가용성의 표준이 되는 프레임워크를 제공하여 노드 또는 컴포넌트 레벨에서 RFID 미들웨어 시스템의 모든 구성요소에 대해 동등한 수준 또는 필요한 수준에 맞는 가용성을 제공한다.

RFID 미들웨어 시스템의 가용성을 지원하기 위해서는 기본적으로 2 단계 방식을 취한다. 우선 RFID 미들웨어 시스템의 가용성 지원을 위한 일반적인 프레임워크로서 AMF를 정의한다. 그리고 AMF를 바탕으로 특정 RFID 미들웨어 시스템의 가용성 지원을 위해 AMF가 요구하는 부분만 해당 노드 또는 컴포넌트에 맞게 직접 구현한다. (그림2)는 RFID 미들웨어 시스템과 AMF의 관계를 보여준다. AMF는 RFID 리더, RFID 미들웨어, EPC Information Server(이하 EPCIS) 등으로 구성되는 RFID 미들웨어 시스템의 하단에 위치하여 프레임워크를 제공함으로써 가용성을 보장한다.

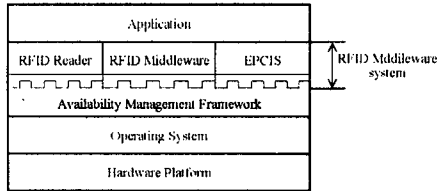


그림 2. AMF의 역할

AMF가 제공하는 부분과 개별 RFID 미들웨어 시스템이 구현하는 부분이 합쳐지면 실질적인 특정 RFID 미들웨어 시스템의 가용성을 보장하게 된다.

3.2 AMF 설계 방안

본 절에서 제시하는 AMF는 가용성을 제공해야 할 RFID 미들웨어 시스템에 속하는 클라이언트들 중 RFID 리더와 RFID 미들웨어에 중점을 두고 설명한다.

AMF의 구성을 최상위 수준에서 보면 AMFSrv와 AMFCli로 구분할 수 있다. AMFSrv는 AMF 서버로서의 역할을 할 AM Server(가용성 관리 서버) 구현을 위해 제공되는 프레임워크이다. AMFCli는 AMF 클라이언트로서 해당 클라이언트가 AMF를 통해 가용성을 보장받기 위해 필요한 가용성 관리자(Availability Manager, 이하 AM) 구현과 클라이언트 측이 AMFSrv와 상호작용하기 위해 제공되는 프레임워크이다. (그림3)과 같이 RFID 미들웨어 또는 RFID 리더와 같은 클라이언트는 AM 생성을 위해 클라이언트 측 프레임워크인 AMFCli를 제공받으므로 가용성 기술 구현이 용이하다.

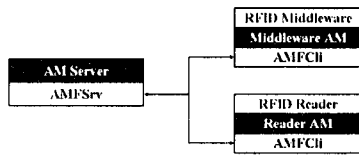


그림 3. AMF와 가용성 관리자

● AMFSrv 패키지

AMFSrv는 AMF의 서버 측 프레임워크를 제공한다. (그림4)는 AMFSrv를 서브시스템 수준에서 패키지 다이어그램으로 나타낸 것이다.

- AMFNodeConnectionMgr : 가용성을 보장받기 원하는 클라이언트로부터 오는 메시지를 AMFSrvMgr로 전송하며 AMFSrvNodeMgr로부터 오는 메시지는 AMFCli로 보낸다.
- AMFSrvMgr : AMFNodeConnectionMgr를 통해 AMFCli로부터 들어오는 메시지를 처리하여 해당 메시지를 필요로 하는 서브시스템으로 보낸다. 장애 탐지 전략과 복구 전략을 지시한다. 즉, 등록된 컴포넌트에 대한 장애 탐지를 시작하도록 명령하고 컴포넌트의 장애가 탐지되면 장애 복구를 위한 정보를 AMFSrvNodeMgr로 보낸다.
- AMFSrvNodeMgr : 각 노드의 컴포넌트들을 그룹으로 관리한다. 가용성을 보장할 위해 서비스 그룹, 노드, 컴포넌트를 등록한다. AMFSrvMgr로부터 받은 장애 복구 정보를 해당 컴포넌트에게 전달한다. 같은 기능을 제공하는 클라이언트 또는 컴포넌트들을 그룹으로 묶고 그룹간의

의존성을 설정하는 것은 가용성 향상을 위한 중요한 기술이다. 이러한 그룹간의 의존관계 관리 통해 의존관계에 있는 클라이언트 또는 컴포넌트의 장애가 탐지되면 해당 그룹의 노드 또는 컴포넌트들 중에서 대체할 대상을 찾아 지속적인 서비스가 이루어지도록 한다.

- AMFDependencyMgr : 그룹간의 의존성과 컴포넌트간의 의존성을 등록하고 검색요청이 있으면 의존관계를 제시하는 등의 의존성 관리 역할을 한다.
- AMFRecoveryMgr : 관리중인 클라이언트의 AMFCli로부터 장애가 보고 되면 AMFDependencyMgr를 통해 의존성을 조사하고 해당 컴포넌트의 장애에 대한 복구 전략을 생성한다.

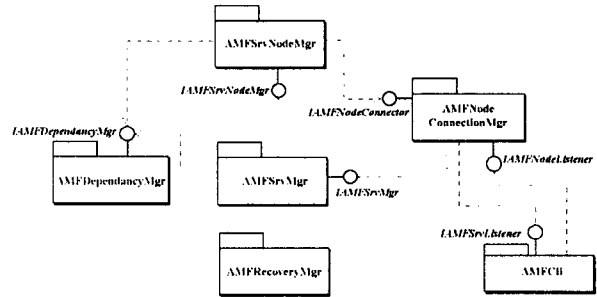


그림 4. AMFSrv Package Diagram

● AMFCli 패키지

AMFCli는 AMF의 클라이언트 측 프레임워크를 제공한다. (그림5)는 AMFCli를 서브시스템 수준에서 패키지 다이어그램으로 나타낸 것이다.

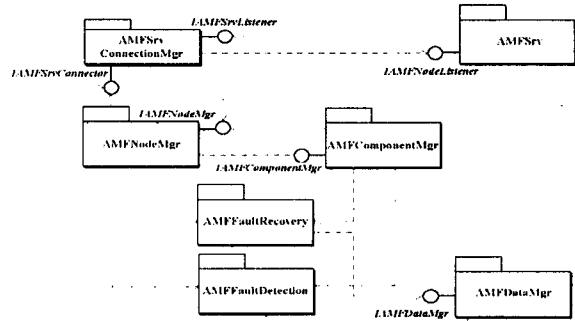


그림 5. AMFCli Package Diagram

- AMFSrvConnectionMgr : AMFSrv로부터 오는 메시지를 AMFNodeMgr로 전달하거나 AMFNodeMgr로부터 오는 메시지를 AMFSrv로 전달한다.
- AMFNodeMgr : 관리 대상 컴포넌트를 가지고 있는 노드를 관리한다.
- AMFCComponentMgr : 관리 대상 컴포넌트들을 관리한다. 컴포넌트 등록을 요청하는 기능을 하며, 컴포넌트에 대한 장애 탐지 및 장애 복구 전략을 수행하는 역할도 한다.
- AMFFaultDetection : AMFSrv에 등록되어 관리되고 있는 컴포넌트의 상태를 감시하기 위한 기능을 제공한다.
- AMFFaultRecovery : 감시하고 있는 컴포넌트에 대한 장애 발생시 장애 복구를 위한 기능을 제공한다.
- AMFDataMgr : 의존관계에 있는 컴포넌트에 장애가 발생하였을 때 장애 복구를 위해 필요한 상태정보 또는 관련 정보를 저장하고 관리한다.

AMFCli의 AMFCComponentMgr, AMFFaultDetection, AMFFaultRecovery, AMFDataMgr은 특정 클라이언트에 맞도록 직접 구현해야 하는 부분인 AM을 구현할 때 이용하게 되는 서브시스템들이다.

3.4 장애 탐지와 장애 복구

그림6, 그림7, 그림8은 RFID 미들웨어 시스템에 속하는 한 클라이언트의 AComponent에 대한 장애 탐지, 장애 보고, 장애 복구 시나리오를 보여주는 순차 다이어그램이다. 제시한 복구 시나리오는 특정 클라이언트의 컴포넌트인 AComponent를 가용성 관리 대상으로 한다. 장애 탐지 전략은 Ping/Echo를 가정하고 장애 복구 전략은 컴포넌트 재시작을 기준으로 작성되었다.

장애 탐지 전략으로 Heartbeat를 사용하거나 장애 복구 전략으로 다양한 Redundancy 전략 또는 Repair 전략을 사용할 수 있다. 전략에 따라 순차 다이어그램의 일부(AM 관련 부분)가 달라질 수 있다.

AMFCII를 이용하여 구현된 AM의 일부인 AComponentAgent에 의해 가용성 관리 대상이 될 AComponent, 장애 탐지를 지원하기 위한 AComponentPingEchoProvider, 장애 복구를 위한 AComponentRecoveryProvider가 생성된다. AComponent를 AMFSrv에 등록하게 되면 AMFSrv는 AMFCII와 해당 노드 또는 컴포넌트에 맞게 구현된 AM을 통해 장애 탐지와 장애 복구 전략을 수행하여 AComponent의 가용성을 관리하게 된다.

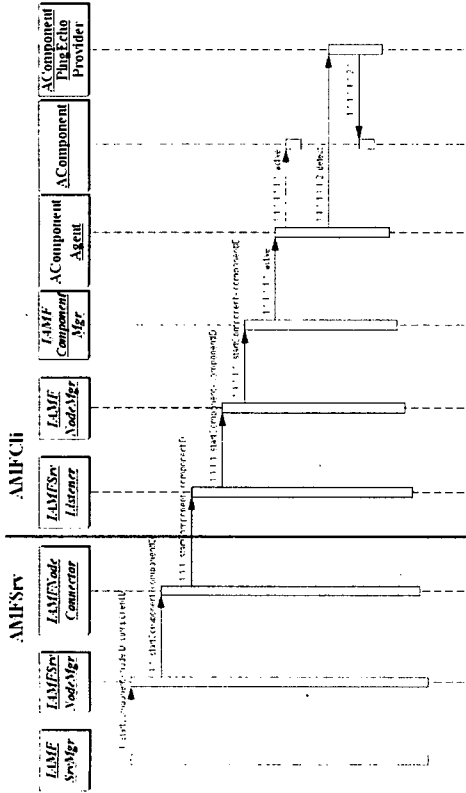


그림 6. 장애 탐지 시나리오

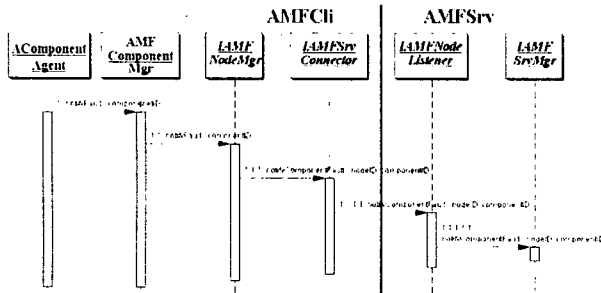


그림 7. 장애 발생 보고 시나리오

RFID 미들웨어 시스템의 구성요소가 노드 또는 컴포넌트 레벨에서의 가용성을 보장하기 위해 AMFCII를 통해 AMFSrv에게 자신을 등록해 줄 것을 요청하며, AMFSrv는 등록을 요청한 노드 또는 컴포넌트에게 가용성을 제공하기 위해 필요한 정보를 제공하고 가용성을 관리한다. AMFSrv는 등록된 노드 또는 컴포넌트의 장애 발생 여부를 확인하기 위해 장애 탐지 명령을 내리고 AMFCII가 그 지시를 받아 장애 탐지를 실시(그림6)하게 된다. 장애가 탐지되면 AMFCII는 AMFSrv로 보고(그림7)하게 되고 AMFSrv

는 AMFCII와 해당 AM을 통해 노드 또는 컴포넌트에 대한 장애 복구 전략을 수행(그림8)한다.

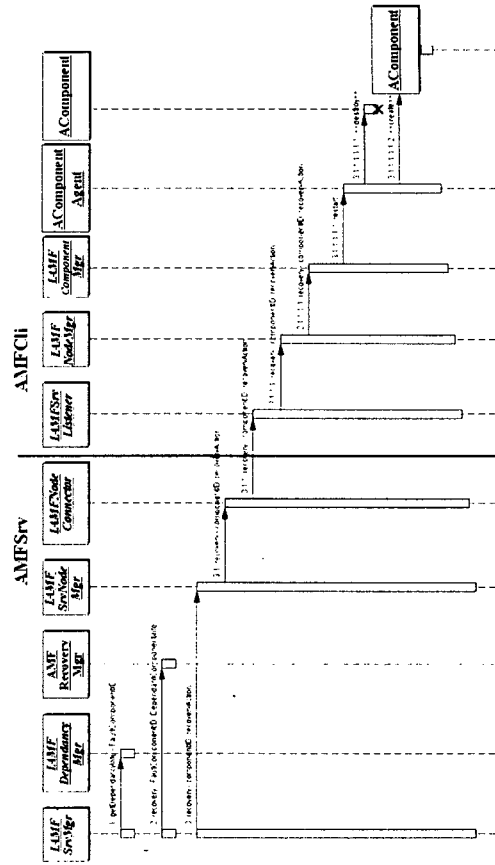


그림 8. 장애 복구 시나리오

#### 4. 결론

가용성 관리 프레임워크 객체지향 프레임워크 기반으로서 재사용성 및 확장성이 뛰어나다. 장애 탐지 및 장애 복구 기법을 한정시키지 않으므로써 상황에 따라 그에 적합한 수준의 가용성 관리가 가능하다. RFID 미들웨어 시스템을 다른 플랫폼으로 이식하여야 할 때에도 해당 플랫폼에 맞는 프레임워크를 사용하여 자신의 고유 기능과 최소한의 가용성 코드만을 이식하여 안정적인 가용성을 보장받을 수 있다. 가용성 관리 프레임워크 자체가 하나의 유연적인 상태를 취하고 있으므로 RFID 미들웨어 시스템 벤더는 가용성 기술을 어렵게 구현할 필요 없이 가용성 관리 프레임워크를 도입하여 일부 기만 구현함으로써 해당 RFID 미들웨어 시스템의 가용성을 보장 받을 수 있게 된다.

#### 5. 참고문헌

- [1] EPCglobal, The Network Components, [http://www.epcglobalus.org/Network/network\\_components.html](http://www.epcglobalus.org/Network/network_components.html)
- [2] Evan Marcus & Hal Stern, Blueprints for High Availability, Wiley, 2003.
- [3] The Sun Java System RFID Software, <http://www.sun.com/software/products/rfid/datasheet.xml>
- [4] TaggWare, Product Specification <http://www.taggware.com/architecture.jsp>
- [5] RFTagAware, <http://www.connecterra.com/products/rftagaware.php>
- [6] Singularity <http://www.i-konect.com/singularity/architecture.htm>
- [7] Manhattan Associates, <http://www.manh.com/rfid/index.html>
- [8] OAT Foundation Suite, <http://www.oatsystems.com/products/index.html>
- [9] URIS, <http://www.allixon.com>