

IoT 및 네트워크 관리 지원을 위한 컴포넌트 아키텍처 개발

(Development of component architecture to support IoT management)

서희경*

(Hee Kyoung Seo)

요약

IoT에서는 인간과 사물, 사물과 사물들이 통신하며 공통적인 네트워크를 형성하여 자동화 서비스를 실현하는 것이 무엇보다 중요하다. 인간과 사물, 사물과 사물을 공통적인 네트워크로 공유하는 방식으로 웹은 가장 강력한 수단 중의 하나이다. 따라서 IoT에서 사용되는 각 디바이스와 웹과의 통신방식의 효율성은 IoT의 성패를 가름할 수 있다. 웹 응용에서 소프트웨어의 재사용은 소프트웨어의 고품질과 고생산성을 위한 최상의 방법으로 여겨진다. 모듈, 클래스, 패턴, 프레임워크, 비즈니스 컴포넌트들은 다양한 관점의 재사용 요소들이다. 컴포넌트는 잘 정의된 인터페이스를 통해서 다른 것들과 쉽게 플러그인함으로써 응용 개발의 근본 문제인 복잡성을 극복하고 개발과 운용의 다양성을 제공할 수 있다. IoT 및 Network Management를 위한 웹 기반의 분산 환경은 각 종 디바이스에서 수집되는 정보를 이용하는 응용의 개발과 활용을 위한 표준 아키텍처이다. 따라서 IoT 및 NM 응용에서 최상의 서비스 제어를 위해 구성 자원들을 관리, 체계화하는 네트워크 관리는 개별 응용 뿐 아니라 응용의 대부분에서 하부 계층 지원 서비스로 요구된다. 본 논문에서는 이질적인 인터넷 네트워크상에서 컴포넌트 기반 IOT 및 관련 네트워크 관리 시스템 개발을 목적으로 한다. 이를 위해 필요한 컴포넌트들의 분류 계층화를 위한 컴포넌트 아키텍처를 정의하고 네트워크 도메인에서 필요로 하는 컴포넌트를 식별, 분류하며 실제 네트워크 관리 시스템의 유형을 제시한다.

■ 중심어 : 사물인터넷 ; 컴포넌트 ; 네트워크 관리 시스템 ; 소프트웨어의 재사용

Abstract

It is important to realize automation services by communicating in IoT with humans, objects & objects, and forming a common network. People used web like the most powerful network way to sharing things and communication. Therefore the efficiency method communication between each device and the web in IoT could be different from ones. The best method for high quality software product in web applications is software reuse ; Modules, classes, patterns, frameworks, and business components are reusable elements of various perspectives. Components is plugged with others through well-defined interfaces, which can overcome the operation and complexity of application development . A web-based distributed environment for IoT applications is a standard architecture use information collected from various devices for developing and using applications . For that reason, the network management which manages the constituent resources for the best service control in IoT application is required as a sub-layer support service in most applications as well as individual applications. In this paper, we measure to develop a network management system based not only by components but on heterogeneous internetworks. For procedure this, we clarify a component architecture for classifying and classify also the component needed in the IOT and network domain or order the type of real network management system.

■ keywords : IoT ; component ; network management system ; software reuse

I. 서론

IoT란 인간과 사물, 서비스 분산된 세 가지 환경 요소에 대

해 인간의 명시적 개입 없이 상호 협력적으로 네트워킹, 센싱, 정보 처리 등 지능적인 관계를 형성하는 사물 공간 연결망을 뜻한다.[1] 1989년 처음 등장한 웹은 URI, HTML, HTTP 라는 간단한 기술을 통하여 자유롭게 문서공유를 가능케 했으며, 친

*정회원, 영진전문대학 IT정보계열 서희경

접수일자 : 2017년 05월 16일

수정일자 : 2017년 06월 26일

게재확정일 : 2017년 06월 26일

교신저자 : 서희경 e-mail : seohk17@yjc.ac.kr

속한 이용환경 덕에 확산이 급격하게 일어나 인터넷 발전에 혁명적으로 기여를 해왔다.[2] IoT 시대에 웹은 인간과 사물들을 통합하여 연결하기 위한 중요한 수단으로 떠오르고 있다. WoT란 주변 사물을 웹에 바로 연결해주고, 사물인터넷 기반의 기기를 통해 수집, 분석한 사용자 맞춤형 정보를 언제 어디서나 제약 없이 활용할 수 있게 해주는 기술이다.[3] 가장 보편화된 컴퍼넌트인 COM, CORBA, EJB 등은 개발자에게 인터페이스만 제공하여 내부의 상세한 부분을 숨김으로서 쉽고 빠르게 응용을 개발할 수 있는 실행 블록이다. 재사용 컴퍼넌트의 사용은 소프트웨어 개발에서 모든 단위 모듈을 부품화하여 소프트웨어의 산업화와 대량화를 위해 재사용의 정수를 실현할 수 있다. 분산 컴퓨팅을 위한 기본 서비스 구현과 멀티 벤더/솔루션의 표준 모델로서 웹과 Java, CORBA는 응용의 표준 아키텍처로 자리잡고 있다. 이들 플랫폼들은 이질적인 네트워크 환경에서 통신하는 분산 사용자의 요구를 지원한다. 네트워크 관리 시스템은 네트워크상에서 운영되는 모든 자원들의 관리와 제어를 위한 도구들의 집합이다. 따라서 네트워크 서비스를 필요로 하는 모든 응용에서 웹과 Java, CORBA 기술을 포함하는 네트워크 관리는 핵심적인 응용 도메인이 된다. 그러므로 하부 계층에서 분산 컴퓨팅을 위한 통합 API 및 객체 서비스 뿐 아니라 관리 도메인에 한정적인 수직적 서비스들을 패턴과 컴퍼넌트 등의 재사용 요소로 정의하고, 활용함으로써 응용의 개발과 운영을 위한 생산성을 확보할 수 있다. 현재 관리 시스템 대부분은 특정한 관리 목적을 위해 특정 벤더나 프로토콜, 플랫폼에 의존적이기 때문에 관리자는 비 일치적인 하부 구조에 대해 충분한 이해와 네트워크 운영의 상해함을 파악해야 한다. 그러므로 CORBA를 이종 네트워크 자원의 추상적인 결합 매개체로, 웹 브라우저를 불일치 요소들의 통합된 액세스 인터페이스로, JMAPI를 서비스 관리 솔루션 개발 도구로써 이용해 이 문제들을 해결하고자 한다. 이들 플랫폼들은 전형적인 컴퍼넌트, 객체 서비스이다[4]. 따라서 이들 기술을 이용함은 하부 계층의 아키텍처를 컴퍼넌트나 패턴으로 패키지화한 독립적인 요소들 간의 상호작용으로 구성할 수 있으므로 CBD를 위한 좋은 도메인을 제시한다. 본 논문에서는 컴퍼넌트 기반 소프트웨어 개발 기술의 관점에서 웹 융합 IoT환경 지원을 위한 관리 시스템을 구축하고자 한다. 따라서 컴퍼넌트 아키텍처 모델의 정의를 통해 관리 도메인에 적용할 수 있는 패턴들과 컴퍼넌트들을 식별하고 계층화함으로써 공통적인 재사용 자원을 확보하고 실제 웹 융합 IoT 응용을 위한 프로토타이핑에 적용한다.

II. 관련 연구

2.1 디자인 패턴 및 프레임워크

객체지향 설계에서 어느 도메인에서 빈번히 반복되는 구조를 일반화 한 디자인 패턴은 특정 도메인에 대한 지식과 경험을 재사용 할 수 있다. 이는 자주 발생하는 설계 문제와 그 해답을 기록한 것으로 재사용 가능한 설계 생성을 위해 적용되는 추상화된 구조를 가지고 있으며 유용한 클래스 구조와 관련성 및 그 적용 시점을 표현해서 시스템 복잡도를 감소시키고, 소프트웨어 재사용의 바탕을 제공한다. 또, 검증된 기술을 패턴으로 표현하여 새로운 시스템 개발자들에게 패턴을 더 쉽게 활용하도록 도와주며, 클래스와 객체 상호작용 및 그것들이 품고 있는 의미의 명세를 공통적인 어휘로 제공하여 기존 시스템의 문서화와 유지보수를 편리하게 한다[5]. 크기와 추상화 수준에 따라 매우 다양한 패턴을 Gamma는 크게 두 가지 기준으로 분류하여 관련된 패턴들의 동일한 부분을 참조할 수 있게 카탈로그를 만들었다. 첫 번째 분류 기준은 목적(purpose)으로써, 패턴이 무엇을 하는지 반영하고, 패턴들은 구조적, 생성적, 혹은 행위적 목적 중에서 하나를 가질 수 있다. 두 번째는 적용 될 수 있는 범위(scope)로써, 패턴이 클래스나 객체 중에 어디에 사용되는지를 제시한다.

		Purpose		
		Crational	Structural	Behavioral
Scope	Class	Factory Method	Adapter	Interpreter Template Method
	Object	Abstract Factory Builder Prototype Singleton	Adapter Bridge Composite Decorator Facade Flyweight Proxy	Chain of Responsibility Command Iterator Mediator Observer State Strategy Visitor

그림 1. Gamma의 디자인 패턴 분류

프레임워크란 재사용 가능한 설계구조와 추상화된 클래스 집합, 또는 특정영역에서 재사용 가능한 설계를 구성하는 상호 관련된 클래스 집합이다. 프레임워크를 이용한 시스템 구축은 프레임워크에서 제공하는 추상적인 클래스를 상속받아 원하는 클래스를 구현하는 방식으로 이루어진다. 비유적으로 표현하면 프레임워크는 구축하려는 시스템의 전체적인 틀이라고 할 수 있으며, 실제 개발과정에서는 이와 같은 틀을 바탕으로 원하는 기능을 구현하는 코드를 추가하여 작업하게 된다. 애플리케이션을 개발할 수 있는 아키텍처를 제공하는 면에서 프레임워크와 라이브러리는 차이가 있다. 프레임워크 설계와 코드부분을 재사용하여 개발비용과 시간이 감소되고 사용이 용이하지만 구현이 힘들다[6]. 보통 프레임워크 도메인은 내부구조 혹은 문제영역에 의한 분류로 나뉜다.

우선, 내부구조에 의한 분류는 클래스를 인스턴스화 시켜 이들의 멤버함수를 호출해 프레임워크의 기본 기능을 사용하는 블랙박스, 그리고 새로운 클래스를 만들거나 멤버함수 재정의를 통해 기능을 확장, 수정하는 화이트 박스로 크게 나뉜다.

두 번째, 문제영역에 의한 분류는 도메인 프레임워크, 애플리케이션 프레임워크, 지원 프레임워크로 나뉜다.

- 도메인 프레임워크(Domain Framework) : 특정 클라이언트 내의 문제에 대한 해결을 위해 특정한 문제영역내의 전문 지식을 캡슐화 시킨다.
- 애플리케이션 프레임워크(Application Framework) : 매우 넓은 영역의 프로그램에 적용될 수 있는 전문지식을 캡슐화 하는 것으로, 모든 GUI(Graphical User Interface) 애플리케이션이 요구하는 표준기능을 지원하는 현재의 상업적인 GUI 애플리케이션 프레임워크를 예로 들 수 있다.
- 지원 프레임워크(Support Framework) : 파일접근, 분산 컴퓨팅 지원, 혹은 장치 드라이버같이 시스템 레벨의 서비스를 제공하며, 애플리케이션 개발자는 전형적으로 직접 지원 프레임워크를 사용하거나 시스템 제공자가 제공하는 수정본을 사용한다.

2.2 네트워크 관리 도메인(NM Domain)

네트워크 관리는 상호 연결된 네트워크 서비스 사용을 위해서 계획하고 조직화하며 감시하고 제어한다. 따라서 실시간인 네트워크의 상황 변화를 반영하고 예상 가능한 통신 행위들을 보증한다. NM을 위한 OSI의 관리 아키텍처 모델은 Fault management, Configuration management, Accounting management, Performance management, Security management 등 총 5개의 기능적 관리 영역을 설정하고 있다.[4] NM은 근본적으로 웹과 같은 간단한 인터페이스를 통해서 모든 관리 기능을 수행해야 하고 기존 자원과의 호환성을 원칙으로 한다. 표 1은 OSI의 네트워크 관리 정보이다[5].

2.3 컴포넌트 기반 소프트웨어 개발 : CBD

‘객체지향 CBD 개발 방법론’이란, 세분화된 객체를 사용에 따라 그룹화 하여 구성된 컴포넌트 단위로 소프트웨어를 개발하는 방식이다. 시스템을 구성하는 컴포넌트를 조립해 시스템을 구축하는 개발방법에 속한다[6]. CBD에서 인터페이스는 컴포넌트와 분리되어 구현한다. 예를 들어 코바(CORBA)나 엔터프라이즈 자바 빈즈(ENterprise Java Beans)같은 컴포넌트 기반의 프로그래밍 언어에서 컴포넌트와 인터페이스를 구현하기 위한 구성 요소를 각각 제공하고 있다[7]. 컴포넌트 기반의 소프트웨어는 충분한 기능을 제공하는 표준화된 컴포넌트들의 인

터페이스 결합을 통해 아웃소싱 되어 짐으로써 개발된다.

표 1. 네트워크 관리 정보

OSI 영역	기능
Fault Management	오류나 예외적인 상황을 검색, 추출 및 해결하고 보고하는 기능
Configuration Management	네트워크 장치(하드웨어, 소프트웨어 자원)들에 대한 모든 정보를 수집하고 유지관리하며 데이터베이스화 하여 네트워크 운영의 계획, 제어 변경을 하는 기능
Accounting Management	네트워크 자원 사용을 분석해서 이용자에게 비용을 부과하고 효율적인 네트워크 사용 방법을 제시하는 기능
Performance Management	전체 네트워크 구성 장비의 통신량을 수집하고 분석, 판단하여 최적의 성능이 유지되도록 관리하는 기능
Security Management	불법적인 데이터 유출을 예방하고 사적 비밀 정보의 노출을 최소화하며 정보 변경을 예방하는 기능

CBD의 핵심인 컴포넌트 조립에 의한 빠른 응용 개발에 앞서 고품질의 재사용 요소의 구축은 필수적인 것이다.

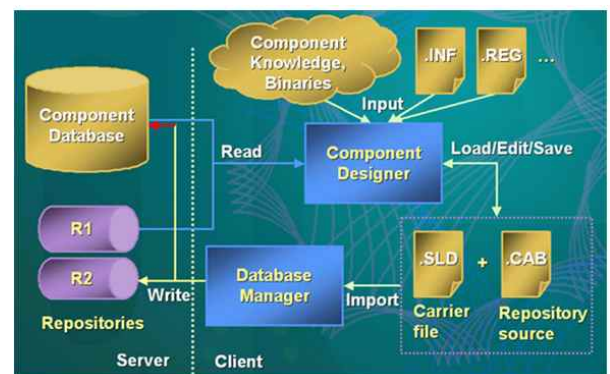


그림 2. 재사용 저장소와 컴포넌트

CBD를 위한 재사용 저장소는 여러 형태의 재사용 가공물 즉, 클래스, 플러깅 가능한 컴포넌트, collaboration 프레임워크, 패턴, 명세 등이 포함되어져 파라미터화 되거나 커스터마이져 되어져 활용된다. 컴포넌트의 개별적인 개발, 구매, 적용을 위해서는 특정 도메인의 수직적인 프로세스에 적용되는 컴포넌트 계층의 정의 뿐 아니라 공통 도메인에 적용가능한 수평적 공용 컴포넌트의 정의가 필요하다. 그림 2는 재사용 저장소에서의 컴포넌트 의미이다. 그림 3은 IOT와 네트워크 관리 서비스의 구조를 보여준다.

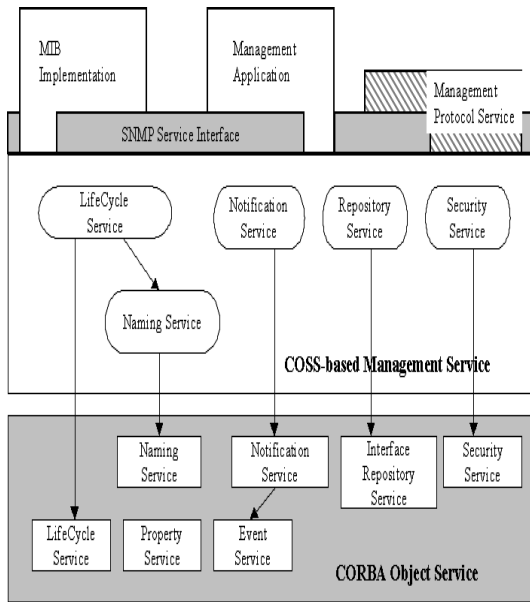


그림 3. IOT와 네트워크 관리 서비스

2.4 컴포넌트 아키텍처와 NM 도메인

컴포넌트 아키텍처는 관련된 여러 종류의 컴포넌트들을 연관시키기 위한 표준 계층으로 컴포넌트의 획득과 이해, 조립을 위한 레이아웃을 제시하여 사용자들이 필요로 하는 컴포넌트들을 식별, 검색하며 커스터마이징할 수 있도록 가이드라인을 제공한다. 따라서 컴포넌트의 생성과 저장을 위한 스키마의 자연스러운 확보가 가능하게 되고 소프트웨어 전개를 위한 응용 모델을 제시한다. 컴포넌트 기술이 보다 더 강력한 기능을 제공하기 위해 해션 아키텍처 기반으로 컴포넌트의 생성과 조립 및 추출이 가능해야 하며 현재 소프트웨어 아키텍처 기술을 컴포넌트 기술 분야에 적용하려는 연구가 활발히 진행 그리고 현장에서 다양하게 적용되고 있다[8]. 또한 최근의 빅데이터 처리 및 IOT 등 4차 산업의 다양하고 복잡한 응용 소프트웨어 개발에서 실질적이고 효율적으로 적용중이다.

재사용을 통한 개발을 하기 위해서는 사용가능한 재사용 자원을 효율적으로 관리하고 신뢰성 있는 자원으로 제공하기 위해서는 도구를 통한 소프트웨어 개발 환경과 사용자 애플리케이션의 수행 환경 등을 고려한 재사용 관리 시스템의 구축은 중요하다.

그 하나의 예로 IBM의 RSL(Reusable Software Library) 재사용 시스템은 장기적이고 단계화 된 방대하고 다양한 기업 환경에서 소프트웨어 부품을 재사용할 수 있게 하는 도구다. 이 시스템에서 사용하는 검색방법은 패킷 분류 기법을 기반으로 14개의 항목으로 패킷을 구성한다. RSL 시스템은 다음(그림 3)과 같이 RSL 데이터베이스, 라이브러리 관리, 사용자 질의, 소프트웨어 부품 추출 및 평가, 소프트웨어 설계와 같은 네 가지

서브시스템으로 이루어져 있다. IBM Sanfrancisco는 객체지향 하부 구조와 응용 모델, 디폴트 업무 로직을 위한 비즈니스 프로세스 컴포넌트를 객체의 관점에서 제공하며 재사용 객체를 위한 3계층을 정의한다[9].

- ① Foundation 계층 : 객체 모델 베이스 클래스, 유틸리티, 객체 서비스 집합 제공
- ② Common Business Object 계층 : 비즈니스 도메인 상에서 공통적으로 요구되는 행위와 비즈니스 객체 및 설계 패턴 포함
- ③ Core Business Process 계층 : 수직 도메인에 대한 비즈니스 객체와 디폴트 로직을 제공

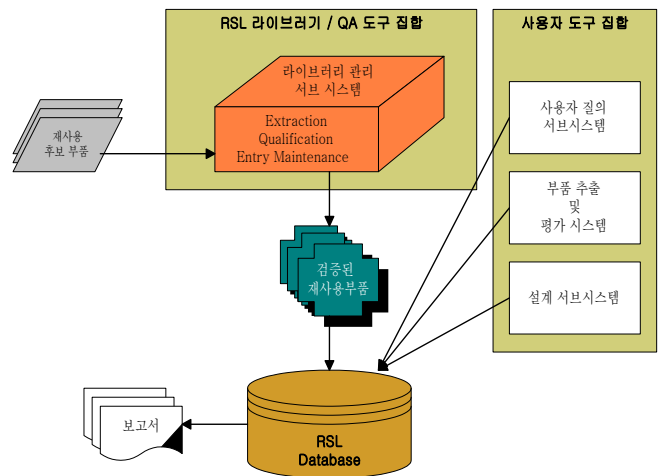


그림 4. RSL 시스템의 구성

- RSL 데이터베이스 : 재사용 소프트웨어의 모든 부품 속성은 RSL 데이터베이스에 저장된다. 모든 데이터베이스의 정보는 RSL의 서브 시스템 모두에서 접속이 가능하지만 내용 수정은 라이브러리 관리 서브 시스템을 통해서만 가능하다. RSL 데이터베이스에는 Ada로 작성된 소프트웨어 부품을 저장하며 함수, 프로시저, 패키지, 프로그램 등의 다양한 타입의 부품을 저장한다.
- 소프트웨어 설계 서브 시스템 : RSL 프로토타입은 소프트웨어 생명주기에 통합되어 일부를 이루고 소프트웨어 설계 서브 시스템인 SoftCAD를 이용해 설계자는 객체지향 그래프를 생성한다. 또한, Ada 프로그램 설계 언어를 사용하여 상위 단계의 시스템 설계를 이행할 수 있으며, 상세 설계를 위한 템플릿으로 사용된다.
- 라이브러리 관리 시스템 : 소스코드나 설계 문서로부터 재사용 정보를 추출하여 후보 부품의 품질을 평가한다. 검증된 부품을 RSL 데이터베이스에 입력하고, RSL을 관리하는 기능을 제공한다. 또한, 표준화된 입력자료, 자동화된

자료 수집, 생명주기 상에서 재사용 정보의 일치성과 지속성, 재사용 정보의 합리성과 완전성, 재사용 정보의 검사를 위한 도구를 제공한다.

- 사용자 질의 서브 시스템 : 메뉴 방식의 인터페이스를 제공해 특정 속성 값을 가진 부품을 찾아주며 속성에 대한 보고서를 생성한다.
- 소프트웨어 추출 및 평가 서브 시스템 : 사용자의 질의에 대해 두 방식의 결과를 제공한다. 먼저 사용자의 질의에 대해 가장 적절한 부품을 찾고 그 속성들과 함께 사용자에게 제공한다. 그리고 질의에 적합한 순서대로 추천 부품의 목록을 제공하여 사용자가 선택하여 사용할 수 있도록 하였다.

III. 컴포넌트 아키텍처와 IOT 와 NM (Network Management) 도메인

3.1 NM 시스템의 아키텍처

네트워크 관리자는 인터넷워킹 상에서 웹 브라우저 상의 단일 인터페이스 시스템을 통해 네트워크 상태를 확인, 모니터링하며 비 관리자들도 원하는 네트워크 구성물의 상태 정보를 액세스할 수 있다. 이는 CORBA 기반의 통합 네트워크 환경에서 웹 브라우저상의 클라이언트 Java 응용을 통해 분산된 MIB로부터 관리 정보를 액세스함을 의미한다. 이를 위해 표준 인터페이스를 활용하여 멀티 플랫폼과 장비, 응용 자원을 관리한다. 본 논문에서는 IOT 및 네트워크 관리를 위한 시스템 아키텍처를 그림 5와 같이 제시한다.

IOT 및 네트워크관리 시스템은 구조가 구조와 일관성이 있으며 프레임워크가 잘 정의되어있으며 필요한 client 부분에서 어답팅하면 되기 때문에 본 논문에서 제시하는 아키텍처의 활용이 매우 적절하다. 즉, 개발자가 추가적인 부분만 파라미터로 패싱하여 관련된 IOT 및 NM 응용 시스템의 개발이 용이하며 또한 생산성 및 품질을 보증할 수 있는 장점이 있다.

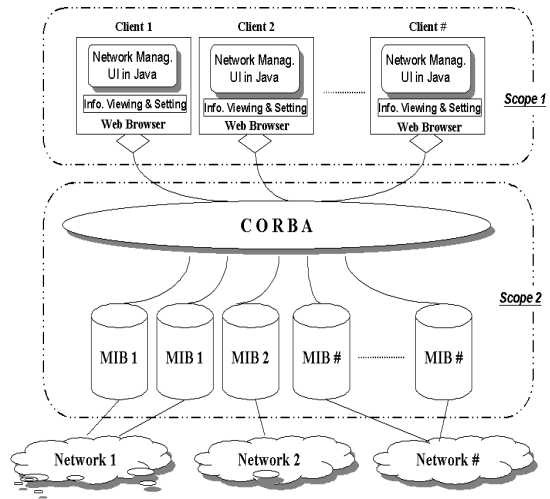


그림 5. IOT 및 네트워크 관리 시스템의 구조

3.2 IOT 및 네트워크 관리 컴포넌트

본 논문의 목적은 웹과 CORBA, Java 기술 등을 이용하여 네트워크 관리 시스템 개발에 필요한 시스템 자원들의 패턴화 및 컴포넌트화이다. 즉, 관리 시스템 컴포넌트들 간의 관련성을 정의, 일반화하며 실행성의 컴포넌트를 식별하여 관리를 위한 표준 자원을 제공하는 것이다. 본 논문에서는 그림 7과 같은 컴포넌트 아키텍처 모델(ABCD)을 정의하였다. IBM의 Sanfrancisco 모델을 기반으로 전체 4개의 계층으로 구성된다. 특히, 공통 컴포넌트 영역을 두 계층으로 구분하여 CBD를 위한 풍부한 조립 자원을 제공한다. 또한 확장성 있는 컴포넌트 제공을 위해 비즈니스 프레임워크 구축의 최상의 재사용 요소로서의 설계 패턴을 포함한다.

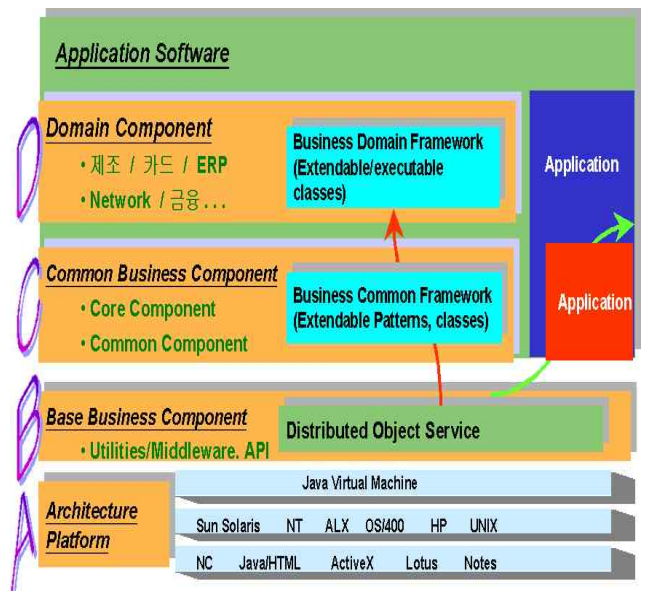


그림 6. 제안하는 IOT 및 NM 컴포넌트 아키텍처

표 2. 컴포넌트 아키텍처의 각 계층들

계층	이름	특징
A	Architecture Platform	분산 컴퓨팅 환경에서 멀티 벤더/응용 구축을 위한 하부적인 물리적 플랫폼
B	Base Business Component	· 분산 컴퓨팅을 위한 미들웨어적인 통합 API 및 기존의 분산 객체 서비스 · 모든 응용에 하부적인 기능 수행에 필요한 공통적인 컴퍼넌트 · 공통 기능의 멀티 응용 컴퍼넌트
C	Common Business Component	· 비즈니스 응용을 위한 실행성의 기능성 컴퍼넌트
D	Domain Component	· 각 비즈니스 영역별 필요 컴퍼넌트

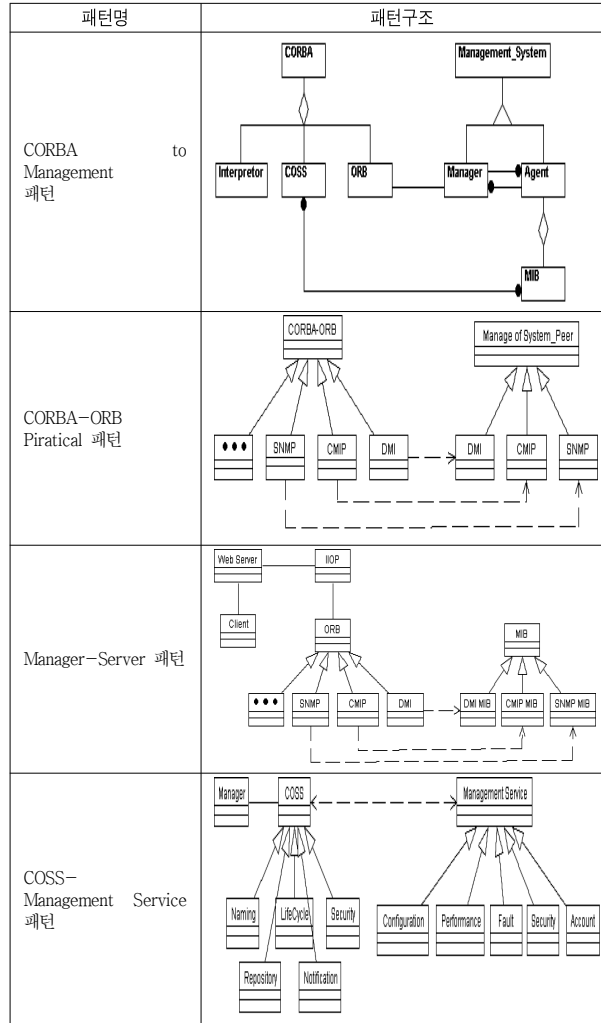
표 2는 컴포넌트 아키텍처의 각 계층을 나타낸다. 컴포넌트의 적용 범위에 따라 계층 B는 미들웨어적인 API와 분산 객체 서비스의 기본 포맷 제공 컴포넌트이다. 따라서 CORBA나 EJB와 같은 분산 객체 컴포넌트들이 여기에 속한다. 공용 계층의 C는 다수의 응용에서 요구되는 일반적 메카니즘의 컴포넌트이다. 그러므로 비즈니스 응용의 핵심 컴포넌트가 된다. 또 커스터마이징 기능만 하는 공통적인 저수준 서비스이다. 따라서 모든 NM 응용에서 요구되는 컴포넌트가 된다. 또 커스터마이징 가능한 설계 패턴을 포함시킴으로써 컴포넌트의 재사용 스킵를 확장한다. 계층 D는 수직적 도메인을 위한 특정 응용 컴포넌트이다. 네트워크 관리 도메인은 본 논문에서 제시한 아키텍처에 비추어 볼 때 두 계층의 컴포넌트로 정리할 수 있다. 만약 네트워크 관리를 모든 응용에서 공통적으로 요구되는 기반 기능으로 정의하면 B계층과 공용 컴포넌트인 C계층 컴포넌트가 되지만 NM을 독립적인 시스템으로 고려하면 D 계층인 도메인 영역이 된다.

3.3 관리 도메인에서 식별 되는 컴포넌트

논문에서 제시한 컴포넌트 아키텍처에서는 확장성 있는 사용자 커스터마이징으로 구성된다. 그림 8은 네트워크 관리 시스템 개발을 위해 식별되어진 컴포넌트들을 본 논문에서 제시한 아키텍처에 매핑시켜 제시한 것이다. B계층 컴포넌트는 네트워킹을 위한 하부 컴포넌트로 통신과 데이터베이스를 위한 기본으로 서비스 제공을 위한 것이다. C 계층의 컴포넌트는 네트워크 관리 도메인의 업무 수행을 위한 로직 프로세스로 B 계층의 컴포넌트의 조합에 의해 형성 사용될 수 있다.

네트워크 관리를 위한 컴포넌트들은 하나의 도메인으로 뿐 아니라 분산 컴퓨팅을 위한 지원 컴포넌트까지 모두의 관점에서 이용될 수 있다. 도메인 분석을 통해 모델링을 작성한 후 이 정보를 바탕으로 네트워크 패턴을 생성 할 수 있다. 설계 패턴은 구현의 한정성과 투명한 설계 단계의 재사용 요소이다.

표 3. 네트워크 관리를 위한 설계 패턴



이질적인 네트워크 환경에서도 단일화 가능한 미들웨어 메카니즘을 제공한다. 그림 7은 본 논문에서 제안하는 IOT 및 네트워크 관리 도메인에서 식별된 컴포넌트들을 보여준다.

3.4 제안 아키텍처에 컴포넌트 적용

가. 개요

IOT 및 NMS를 구축하기 위해 도메인 분석을 통해 정의된 패턴과 컴포넌트를 CBD 프로세스에 따라 이용한다. 관리 시스템의 구성 컴포넌트와 패턴은 일반화되고 독립적인 선택과 다른 서비스 추가를 위한 표본 모델을 정의한다. 또한 관련성을 확정지으며 구현의 템플리트를 제공할 수 있어 동적이며 분산되고 변화 가능성이 산재한 네트워크 관리 서비스의 구현에 적합하다. 그림 8은 본 논문에서 제시한 아키텍처에 기반 하여 식별된 컴포넌트와 패턴의 조합을 통해 NM 도메인의 응용을 구축하기 위한 의미적인 컴포넌트 계층이다.

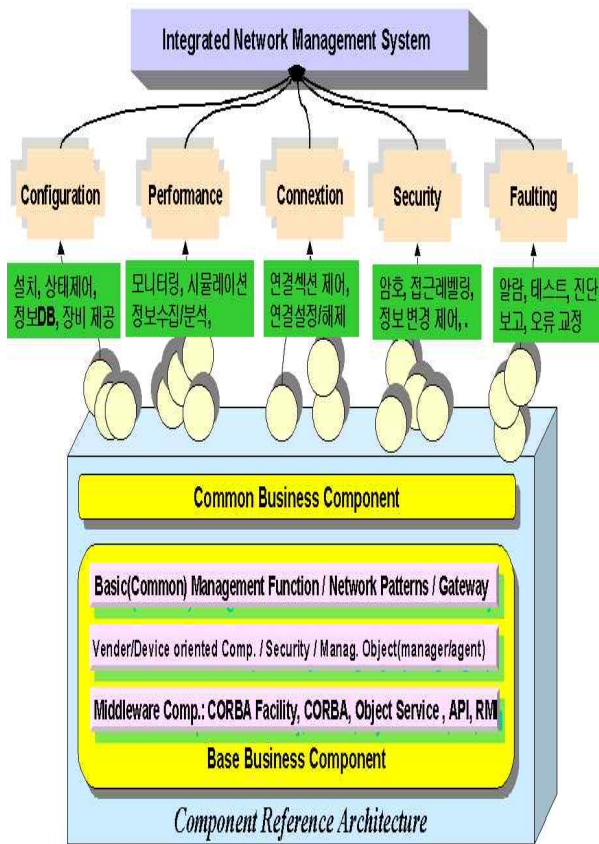


그림 7. 본 논문 제안 IOT 및 네트워크 관리 컴포넌트

나. 패킷 구성

패킷 방법은 컴포넌트의 성질을 잘 보여줄 수 있는 패킷을 정의하고 각 패킷에 대한 제한된 어휘(controlled vocabulary)를 제공하여 그 어휘를 사용해 컴포넌트의 성질을 나타내는 방법이다. 컴포넌트의 성질을 나타내는 패킷은 크게 컴포넌트의 기능적인 면을 설명하여 주는, 즉 컴포넌트가 어떠한 일을 하는가 하는 것을 설명하는 부분과 컴포넌트가 실행되는 환경에 대해 설명하여 주는, 즉 컴포넌트가 어디에서 실행되는가 하는 것을 설명하는 부분으로 나눌 수 있다. 또한, 단지 명세를 위해 존재해야 될 항목과 패킷으로 구성해야될 항목을 분리하여 구성하였다. 구성된 패킷은 식별된 비기능적인 요소를 일련의 코드 부여가 가능한 항목과 기능적인 부분에서 고려되지 못한 부분을 고려하여 그림 8과 같이 IOT 및 NM 그리고 구현관련 현상 정보를 위한 11개의 제한된 어휘로 작성되었으며, 질의 시스템에서 원하는 컴포넌트를 찾기 위한 패킷 검색에 사용 가능하다.

IOT 식별 ID	운영체제	아키텍처	컴포넌트 타입	플 타입	구현언어
NM 식별 ID	운영체제	아키텍처	컴포넌트 타입	플 타입	구현언어
구현 도구	컨테이너	지원 언어	데이터베이스	개발 회사	

그림 8. 구성된 패킷

패킷으로 구성된 각각의 어휘는 컴포넌트의 성질에 따라 하나 이상 대응되는 세부항목을 가질 수 있다. 패킷에서 제외된 나머지 정보는 컴포넌트의 이해를 위한 비기능적인 요소에 포함하여 명세하도록 한다.

개발될 컴포넌트 관리 시스템에서 검색을 통한 결과로 제시되거나, 카탈로그에 의해서 획득된 컴포넌트의 정보를 제시할 경우, 사용자가 원하는 컴포넌트를 선택하기 위해서 비기능적인 요소의 명세는 중요한 부분으로 사용된다.

그림 9는 본 논문에서 실제 NM에 적용한 최종 시스템 구조이다.

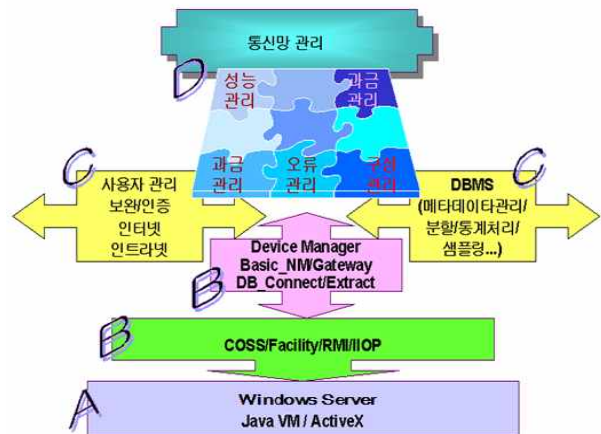


그림 9. IOT 및 Network Management 컴포넌트 아키텍처

IV. 결론

IoT 및 Network Management를 위한 웹 기반의 분산 환경은 각 종 디바이스에서 수집되는 정보를 이용하는 응용의 개발과 활용을 위한 표준 아키텍처이다. 따라서 IoT 및 NM 응용에서 최상의 서비스 제어를 위해 구성 자원들을 관리, 체계화하는 네트워크 관리는 개별 응용으로서 뿐 아니라 대부분의 응용에서 하부 계층 지원 서비스로 요구된다. 본 논문에서는 이질적인 인터 네트워크상에서 컴포

넌트 기반 IOT 및 관련 네트워크 관리 시스템 개발을 목적으로 한다. 이를 위해 필요한 컴포넌트들의 분류 계층화를 위한 컴포넌트 아키텍처를 정의하고 네트워크 도메인에서 필요로 하는 컴포넌트를 식별, 분류하며 실제 네트워크 관리 시스템의 유형을 제시한다.

네트워크 관리 도메인은 분산 컴퓨팅 응용의 대부분이 하부 기능뿐 아니라 독립적인 응용으로서 중요성이 증가하고 있다. 그러나 현재의 관리 서비스는 특정 프로토콜, 벤더 그리고 응용에 대한 한정성 때문에 큰 복잡성과 유지 보수 비용을 가지고 있다. 따라서 CORBA를 통해 투명한 네트워킹 환경을 설정하고 단일 UI로 NM 정보를 획득, 관리함으로써 NMS에 대한 요구를 해결할 수 있다. NM에서 사용되는 기존의 분산 객체 서비스들은 컴퍼넌트와 패턴으로 형성시킴으로서 실시간 변화에 민감하고 사용자의 요구를 반영시킬 수 있다. 본 논문에서는 인터넷 네트워크 상에서 컴포넌트와 설계 패턴에 기반 한 네트워크 관리 시스템 개발을 목적으로 한다. 이를 위해 컴포넌트 아키텍처를 정의하고 NMS 도메인에서 필요한 설계 패턴과 컴포넌트들 식별하여 아키텍처에 매핑시킨다. 그러므로 NM 도메인에서의 컴포넌트들의 분류 계층을 형성하여 실제 네트워크 관리 시스템의 프로토타이핑에 적용하고 실행 예를 제시하였다. 본 논문에서 식별되어진 패턴과 컴포넌트들은 네트워크 관리를 위한 표준화된 컴포넌트 계층과 패턴의 연관성 확립으로 효율적인 관리 시스템 개발 모델을 제시할 수 있다. 향후 식별된 패턴과 컴포넌트들의 재사용 저장소를 구축하고 이를 체계적으로 이용할 수 있는 재사용 환경으로 전개할 것이다.

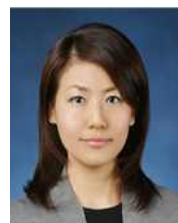
References

- [1] 민경식, "사물 인터넷(Internet of Things)," *한국인터넷진흥원*, pp. 1, 2013년
- [2] 박진태, 표경수, 문일영, "IoT와 웹 기술의 표준화 현황 및 동향 분석," *한국정보통신학회지*, 제16권, 제2호, pp. 1, 2015년 12월
- [3] 이승윤, "[ETRI의 사물인터넷]IoT, 웹으로 실현된," <http://www.hellodd.com/?md=news&mt=view&pid=53640>, 2015년
- [4] Jeff Gehlbach, "Managing Networks in a Software-Defined Future," *Southern California Linux Expo 2015 (SCaLE13x)*, pp.9, 2015.
- [5] Justin Koh, "The 6 Pillars Of Network Management," <http://www.jisikbox.tk/163>, pp. 1, March 2016.
- [6] 장운정 채태병, "CBD 개발 방법론 기반의 촬영 계획 시스템 최적화 방안 연구," *한국정보과학회 학술발표논문집*, pp. 410, 2016년 12월
- [7] 천두완, 조성현, 김수동, "컴포넌트 기반 개발

(CBD) 설계의 서비스 지향 아키텍처(SOA)설계로의 실용적인 변환 기법," <http://dx.doi.org/10.463>, 2011년

- [8] 박상현, 황보선애, 박만곤, "UML 을 사용한 재사용가능 소프트웨어 컴포넌트들의 아키텍처 모델링에 관한 연구," *한국멀티미디어학회 춘계학술발표대회 논문집*, pp.409, 2010년 5월
- [9] Sanfrancisco Technical Report, <http://www-4.ibm.com/software/ad/sanfrancisco>
- [10] Rudolf K. Keller, Jean Tessier, and Gregor von Bochmann, "A Pattern System for Network Management Interfaces," *Communication of the ACM*, vol. 41, no. 9, 1998.
- [11] <http://en.wikipedia.org/wiki/Component>
- [12] <http://www.eclipse.org/modeling/gmp>
- [13] D. Romero, R. Rouvoy, L. Seinturier and P. Carton, "Service Discovery in Ubiquitous Feedback Control Loops," *Proceedings of the 10th IFIP International Conference on Distributed Applications and Interoperable System*, pp. 113-126, 2010.
- [14] L. Jedrzejczyk, A. P.Blaine, K. B. Arosha and B. Nuseibeh, "On the Impact of Real-Time Feedback on User's Behaviour in Mobile Location-Sharing Applications," *Proceedings of the Sixth Symposium on Usable Privacy and Security*, 2010.
- [15] 신수혜, 박준석, 염근혁, "모바일 도메인의 피드백 환경 구축을 위한 상호작용 모델링 방법," *한국 소프트웨어공학 학술대회*, 2012

저자 소개



서희경 (정희원)

2013년 계명대학교 통번역학 학과 박사 수료.
2016년 대구가톨릭대학교 컴퓨터공학 학과 석사 수료.
2017년 영진전문대학 컴퓨터정보계열 조교수

<주관심분야 : 소프트웨어공학, 소프트웨어모델링, 융합 모델링, IOT개발 방법론, 빅데이터응용 모델, 클라우드 컴퓨팅 개발 모델>