

신뢰성 향상을 위한 IoT 기반 상황 인식 시스템 서버 설계

송준석*, 김경태*, 정성민⁰, 윤희용*

⁰성균관대학교 전자전기컴퓨터공학과

e-mail:alskpo@skku.edu*, kyungtaekim76@gmail.com*,

{gearfried, youn7147}@skku.edu⁰*

Design of context awareness system server for improving reliability based on IoT

Jun-Seok Song*, Kyung-Tae Kim*, Sung-Min Chung⁰, Hee-Yong Youn*

⁰Dept. of Electrical and Computer Engineering, Sungkyunkwan University

● 요약 ●

최근 무선통신기술과 소형 하드웨어의 발전으로 모든 환경에서 네트워크 및 컴퓨팅이 가능한 IoT(Internet of Things) 기기들이 지속적으로 증가하는 추세이며, 이러한 IoT 트렌드는 지속될 전망이다. 기존 IoT 플랫폼은 일반적으로 전력 사용적인 측면에서 In-memory 컴퓨팅을 지향하지만 이러한 방식은 시스템 에러 발생 시 메모리의 휘발성으로 인해 신뢰성이 저하되는 단점이 있다. 본 논문에서는 전체 시스템의 신뢰성 향상을 위해 XML 형식의 파일과 관계형 데이터베이스를 이용한 IoT 기반 상황 인식 시스템 서버를 설계한다. 설계된 상황 인식 시스템 서버는 추론 엔진을 포함하고 있으며, 사용자가 등록한 룰과 수집된 상황 데이터를 이용하여 전체 시스템을 동작시킨다. 또한, 전체 시스템의 신뢰성을 검증하기 위해 3가지 동작 시나리오를 제안한다.

키워드: 사물인터넷(Internet of Things), 상황 인식(context awareness), 추론 엔진(Inference engine)

I. Introduction

다바이스 크기의 소형화, 저전력·고성능 프로세서의 증가는 높은 인터넷 보급률과 함께 사물인터넷(Internet of Things)을 탄생시켰다 [1]. IoT는 모든 사물들이 인터넷에 연결되어 사람의 개입 없이 스스로 다른 사물, 사람, 시스템과 연동하여 정보를 생성하고 동작하는 환경을 의미한다[2][3]. IoT가 주목받기 시작하며 다양한 사양의 IoT 플랫폼들이 개발되고 있으며, IoT 기기 및 플랫폼을 효율적으로 사용하기 위하여 스마트 홈, 스마트 팩토리 등 IoT 환경 구축에 대한 연구가 활발히 이루어지고 있다.

기존 IoT 기반 시스템 환경은 IoT 기기의 제한된 파워로 인해 저전력 소비를 지향하여 대부분의 데이터를 메모리에서만 유지한다. 메모리 기반의 데이터 관리는 디스크 기반 데이터 관리보다 처리 속도가 빠르지만 시스템 에러 발생 시 데이터 손실이 발생하여 시스템의 신뢰성이 낮아질 수 있다. 따라서 대규모 센서 데이터가 발생하는 IoT 환경에서 메모리 기반 데이터 관리는 저장된 데이터의 신뢰성을 보장할 수 없다[4].

본 논문에서는 IoT 기반 상황 인식 시스템에서 데이터의 신뢰성을 보장하기 위해 XML 형식의 파일과 관계형 데이터베이스를 이용한 IoT 기반 상황 인식 시스템 서버를 설계한다. 설계된 상황 인식 시스템 서버는 사용자가 등록한 룰, 수집된 상황 데이터와 이를 바탕으로

로 현재 시스템의 주변 상황을 판단하는 추론 엔진을 통해 전체 시스템을 동작시킨다. 또한, 전체 시스템의 신뢰성을 검증하기 위해 다바이스 등록, 센서 데이터 수집, 추론 및 실행 시나리오를 제안한다.

II. Preliminaries

1. Related works

1.1 상황 인식 시스템

상황 인식(context awareness)에 대한 최초의 정의는 Schilit 과 Theimer 가 1994년에 소개했으며, 상황정보를 각각 컴퓨팅 상황정보 (computing context information), 사용자 상황정보(user context information), 물리적 상황정보(physical context information) 나누어 정의하였다. 근래에 널리 사용되는 상황과 상황 인식에 대한 정의는 Abowd 등에 의해 제안되었다. 상황이란 사람, 장소, 물체 등 개체의 상태를 특성화하기 위해 사용될 수 있는 임의의 정보이다[5]. 사용자에게 서비스나 정보를 제공할 때에 상황 정보를 이용하는 시스템을 상황 인식 시스템이라 한다. 대표적인 상황 인식 시스템으로는 스마트 홈, 모니터링 시스템, 방법 시스템 등이 있다.

1.2 상황 추론

추론(inference)이란 기존에 주어진 룰과 지식을 이용하여 새로운 정보를 생성하는 행위이다. 추론에 사용되는 기법에는 정방향 추론과 역방향 추론 등 다양한 기법이 있으며, 이러한 추론 기법을 이용하여 추론하는 소프트웨어 도구를 추론 엔진이라 한다. 상황 추론(context inference)은 추론을 통해 low-level 상황 정보에서 high-level 상황 정보를 생성하는 것이다. 상황 추론은 단일 또는 복수의 상호작용을 통해 high-level 상황 정보를 생성할 수 있으며, 반복될수록 기존보다 정확하고 의미 있는 정보를 제공한다. 상황 인식 연구에서 많이 사용되는 규칙기반 추론 기법은 그림 1. 과 같다[6]. 추론 엔진에 정의된 룰과 상황 정보가 매치(Match)되면 매치된 룰 중 최선의 룰을 선택(Select)하여 실행(Execute)된다. 실행 후 생성된 상황 정보는 다시 룰과 매치될 수 있다.

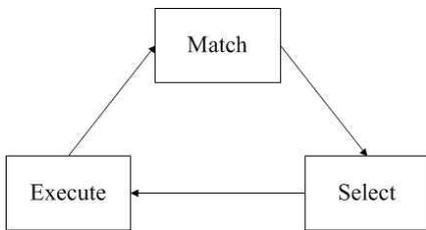


Fig. 1. Rule-based inference cycle

III. The Proposed Scheme

본 논문에서는 XML 형식의 파일과 관계형 데이터베이스인 MySQL을 이용한 IoT 기반 상황 인식 시스템 서버를 설계한다. 상황 인식 시스템 서버의 전체 시스템 구조도는 그림 2. 와 같다.

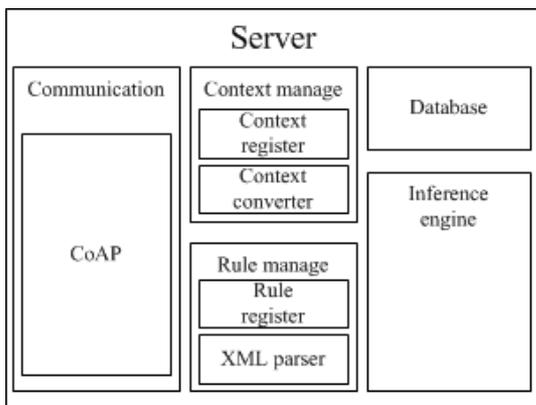


Fig. 2. Structure of context-awareness system server based on IoT

상황 인식 시스템 서버는 CoAP 통신으로 클라이언트와 통신하며, Context manage와 Rule manage 모듈을 통해 상황 데이터와 룰을 관리한다. Database에는 추론에 필요한 모든 상황 데이터 및 룰이 저장되어 있으며, 추론 엔진은 Database에 저장된 데이터를 이용하여

새로운 상황 데이터를 생성한다. 상황 인식 시스템 서버는 추론 엔진에서 생성된 상황 데이터를 바탕으로 상황 인식 서비스를 제공한다.

상황 인식 시스템 서버의 추론 엔진은 사용자가 작성한 XML 문서 형태의 룰과 상황 인식 시스템 클라이언트에서 수집된 상황 데이터를 이용하여 작동한다. 이를 위해, 사용자는 상황 인식 시스템의 목적에 맞는 룰을 생성하여 그림 3. 과 같이 XML 문서 형태로 서버에 저장한다. 사용자는 필요에 따라 서버에 등록된 룰을 변경함으로써 다양한 IoT 환경을 구축할 수 있다.

```
<Activity Id="rule_1" Name="check window">
  <Rule>Window movement null</Rule>
  <Description>Detect movement in the window</Description>
  <BlockActivity BlockId="" />
  <ExtendedAttributes>
    <ExtendedAttribute Name="ID" Value="Analysis"/>
    <ExtendedAttribute Name="OFFSET" Value="76,149"/>
  </ExtendedAttributes>
</Activity>
```

Fig. 3. Rule for Inference engine

상황 인식 서비스 제공을 위해 상황 인식 서버는 서비스 제공에 필요한 모든 데이터를 관계형 데이터베이스인 MySQL에 저장한다. MySQL에 저장되는 데이터는 IoT 기반 상황 인식 클라이언트 정보, 클라이언트에서 수집된 상황 데이터, 추론 엔진에 사용되는 룰 정보로 쪼개 전체 데이터 테이블 구조는 그림 4. 와 같다.

client_data		context_data		Rule		Execute	
PK	ip	PK	subject	PK	R_ActivityId	PK	E_ActivityId
PK	subject	PK	predicate		R_ActivityName		E_ActivityName
PK	predicate	PK	object		R_Rule		E_Rule
		PK	sensing_time		R_Subject		E_Subject
		PK	millisecond		R_Predicate		E_Predicate
					R_ContextD		E_ContextD
			db_time				

Fig. 4. Data of context-awareness system

데이터 테이블은 상황 인식 시스템 클라이언트 정보를 저장하는 client_data, 상황 데이터 정보를 저장하는 context_data, 사용자가 등록한 룰을 저장하는 Rule, 사용자 룰과 상황 데이터가 매치 시 생성되는 상황 데이터 정보를 저장하는 Execute로 이루어져 있다.

이와 같이 설계된 IoT 기반 상황 인식 시스템은 서비스 제공에 필요한 사용자 룰과 상황 데이터가 XML 문서와 관계형 데이터베이스에 저장되어 있어 시스템 에러 발생 시에도 저장된 데이터의 신뢰성을 보장한다.

IV. System scenario

본 논문에서 설계된 IoT 기반 상황 인식 시스템의 구현 후 신뢰성을 검증하기 위해 시스템의 3가지 동작 시나리오를 제안한다. 전체 시스템 시나리오는 디바이스 등록(device register), 데이터 수집(data collect), 추론 및 실행(inference & activate device)으로 이루어져 있으며, 상황 인식 시스템 서버와 상황 인식 시스템 클라이언트를 모두 포함한다. 서버에 클라이언트의 정보를 저장하여 클라이언트를 등록하는 디바이스 등록 시나리오는 그림 5. 와 같다.

상황 인식 시스템 서버는 상황 데이터 수집과 상황 인식 서비스 제공을 위해 다수의 상황 인식 시스템 클라이언트와 연동되는데, 각각의 클라이언트는 실행과 동시에 해당 클라이언트에 내장되어 있는 클라이언트 정보를 CoAP 통신의 PUT message을 이용하여 서버에 전송한다. 서버는 전송받은 클라이언트 정보를 서버의 데이터베이스에 저장하여 클라이언트를 등록한다.

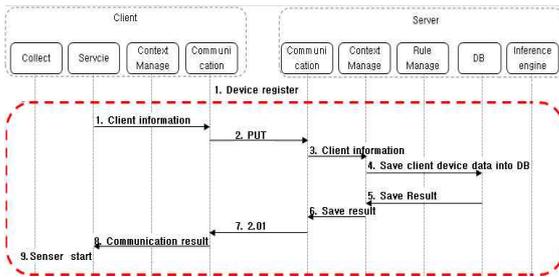


Fig. 5. Flowchart for Device register

클라이언트 등록 후 서버는 등록 결과를 다시 클라이언트에게 전송하며, 클라이언트는 등록 결과를 확인 후 연결된 센서를 동작시킨다. 서버에서 저장하는 클라이언트 정보는 클라이언트 IP, 클라이언트에서 동작 가능한 센서 정보, 센서 동작 여부로 이루어져 있다. 클라이언트 등록 후 클라이언트의 센서 데이터를 이용하여 생성한 상황 데이터를 서버에 저장하는 시나리오는 그림 6. 과 같다.

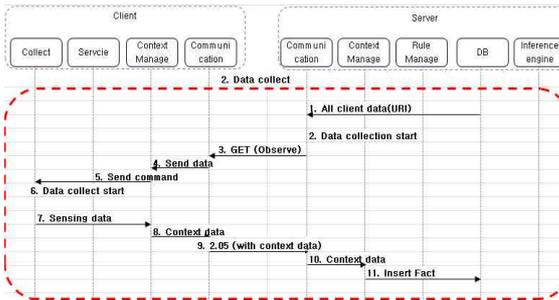


Fig. 6. Flowchart for Data collect

상황 인식 시스템 서버는 상황 데이터 수집을 위해 서버에 등록된 모든 상황 인식 시스템 클라이언트에게 CoAP 통신의 GET message를 보낸다. GET message를 수신한 모든 클라이언트들은 저장된 센싱 데이터를 클라이언트에 내장되어 있는 메타데이터를 이용하여 상황 데이터로 전환하여 서버에게 전송한다. 서버는 클라이언트로부터 전송받은 상황 데이터를 데이터베이스에 저장한다. 서버에서 저장하는 상황 데이터는 주어(subject), 술어(predicate), 목적어(object)로 이루어져 있으며 센싱 데이터의 생성 시간과 서버의 데이터베이스에 상황 데이터가 저장된 시간을 포함한다.

서버에 상황 데이터를 저장 후 추론 엔진을 통해 새로운 상황 정보를 추론하여 상황 인식 서비스를 제공하는 시나리오는 그림 7. 과 같다.

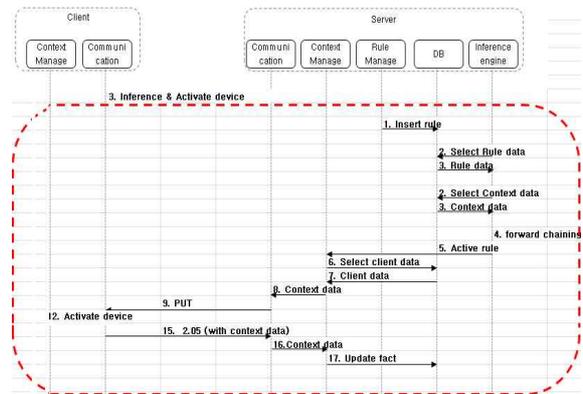


Fig. 7. Flowchart for Inference & Activate device

상황 인식 시스템 서버는 추론 엔진을 동작시키기 위해 추론 엔진에 필요한 사용자 룰을 데이터베이스에 저장한다. 사용자 룰을 데이터베이스에 저장한 후, 추론 엔진은 데이터베이스에 저장되어 있는 모든 사용자 룰 데이터와 상황 데이터를 이용하여 추론을 시작한다. 추론 중 새로운 상황 데이터가 발생하면 추론을 중단하고 발생한 상황 데이터를 데이터베이스에 저장한다. 새로 데이터베이스에 저장된 상황 데이터를 사용자에게 제공해야 하는 경우 클라이언트 정보에서 상황 데이터의 주어, 술어와 매치되는 클라이언트를 찾아 해당 클라이언트로 CoAP 통신의 PUT message를 보낸다. Put message를 받은 클라이언트는 연결된 센서를 동작하여 사용자에게 상황 인식 서비스를 제공한 후 결과를 다시 서버로 전송한다. 서버는 클라이언트의 센서 실행 결과를 데이터베이스에 저장하고 다시 추론 엔진을 실행한다. 추론 엔진은 데이터베이스에 저장된 모든 상황 데이터를 이용하여 추론하고 종료한다.

V. Conclusions

본 논문에서는 대규모 센서 데이터를 사용하는 IoT 기반 상황 인식 시스템의 데이터 신뢰성을 보장하기 위해 XML과 데이터베이스를 이용하여 데이터를 저장 및 관리하는 상황 인식 시스템 서버를 설계했다. 설계된 시스템 서버는 클라이언트에서 생성된 상황 데이터와 사용자가 등록한 룰을 이용하여 사용자에게 상황 인식 서비스를 제공한다. 또한, 설계된 시스템의 구현 후 신뢰성 검증을 위해 시스템의 3가지 동작 시나리오를 제안했다.

향후 연구방향으로는 설계된 IoT 기반 상황 인식 시스템 서버를 구현하고 제안한 시나리오를 바탕으로 전체 시스템의 신뢰성을 검증할 예정이다.

Acknowledgment

본 연구는 BK21Plus 사업, 한국연구재단 기초연구사업 (2013R1A1A2060398), 삼성전자, 미래창조과학부 및 정보통신기술 연구진흥센터의 정보통신·방송 연구개발사업 (1391105003), 미래부/정보통신방송연구개발사업의 일환으로 수행하였음.

References

- [1] G. Chmaj and H. Selvaraj, "Energy-Efficient Distributed Computing Solutions for Internet of Things with Zigbee Devices," IEEE Conference Computer and Information Science, pp. 437-442, June 2015.
- [2] D. A. Reed, D. B. Gannon, J. R. Larus, "Imagining the Future: Thoughts on Computing," IEEE Computer, Vol. 41, No. 1, pp. 25-30, 2012.
- [3] D. Uckelmann, M. Harrison, and F. Michachelles, "An Architectural Approach Towards the Future Internet of Things," Architecting the Internet of Things, Springer, pp. 1-24, Mar. 2011.
- [4] J. W. Kim, "An Effective Data Distribution Scheme in Sensor Network for Internet of Things," The Korea Institute of Electronic Communication Sciences, Vol. 10, No. 7, pp. 769-774, July 2015.
- [5] C. Perera, A. Zaslavsky, P. Christen, and D. Georgakopoulos, "Context Aware Computing for The Internet of Things : A Survey," IEEE Communications Surveys & Tutorials, Vol. 16, No. 1, pp. 414-454, 2014.
- [6] T. H. Kim, J. H. Kim, K. H. Won, K. H. Lee, K. R. Sohn, "A Rule-based Reasoning Engine supporting Hierarchical Taxonomy," The Institute of Electronics Engineers of Korea - Computer and Information, Vol. 45, No. 5, pp. 148-154, Sept. 2008.