KIPS Trans. Softw. and Data Eng.
Vol.8, No.4 pp.145~152 pISSN: 2287-5905

https://doi.org/10.3745/KTSDE.2019.8.4.145

A Method to Provide Context from Massive Data Processing in Context-Aware System

Yoo Sang Park† · Jong Sun Choi†† · Jae Young Choi†††

ABSTRACT

Unlike a single value from a sensor device, a massive data set has characteristics for various processing aspects; input data may be formed in a different format, the size of input data varies, and the processing time of analyzing input data is not predictable. Therefore, context aware systems may contain complex modules, and these modules can be implemented and used in different ways. In order to solve these problems, we propose a method to handle context information from the result of analyzing massive data. The proposed method considers analysis work as a different type of abstracting context and suggests the way of representing context information. In experiment, we demonstrate how the context processing engine works properly in a couple of steps with healthcare services.

Keywords : Context Aware System, Context Abstraction, Ontology, Massive Data Processing

상황인지 시스템에서 대용량의 데이터 처리결과를 컨텍스트 정보로 제공하기 위한 방법

박 유 상† · 최 종 선†† · 최 재 영†††

요 약

단일 센서기기로부터 수집된 데이터와는 다르게 대용량의 데이터는 입력데이터의 구성 및 크기가 가변적이고, 처리 완료시점을 예측할 수 없는 특징을 갖고 있다. 상황인지 시스템이 이러한 환경의 요구사항을 적용하게 되면 컨텍스트 표현방법과 처리모듈들이 개별로 구성되어 해당 입력자료에 대한 호출 및 처리루틴이 복잡하게 구현될 수 있는 문제점이 있다. 이러한 문제점을 해결하기 위해서 본 논문에서 제안하는 처리방법은 온톨로지 기반의 지식표현을 통해 컨텍스트를 표현하고, 대용량의 데이터 처리결과를 반환하는 모듈의 중복 실행을 방지하여 컨텍스트 정보를 제공하기 위한 동작순서를 함께 기술한다. 실험에서는 헬스케어 환경에서 발생하는 센싱데이터 중 대용량의 데이터 처리결과를 필요로 하는 서비스에 대해 기술하고, 기존의 센싱데이터를 바탕으로 서비스를 제공하는 처리과정과 함께 대용량의 데이터 처리결과를 컨텍스트 정보로 제공하는 과정을 보인다.

키워드 : 상황인지 시스템, 컨텍스트 추상화, 온톨로지, 대용량 데이터 처리

1. 서 론

주변 환경의 변화에 따라 사용자가 필요로하는 서비스를 제공해주는 상황인지 시스템은 사용자와 상호작용 할 수 있는 센서기기 및 센싱데이터 수집기술이 발전함에 따라 다양한 입력장치와 수집경로를 통해 주변 환경을 인식하고, 서비스 도메인에 따라 서비스를 제공할 수 있게 되었다[1-3].

상황인지 시스템에서 언급되는 컨텍스트의 개념은 과거 Schilit의 연구[4] 및 Dey의 연구[5]에서 언급되는 '사용자와 그 주변에 배치된 사물(Object)의 위치정보, 식별정보 및 상태정보' 의 정의를 따르며, 유비쿼터스 컴퓨팅(Ubiquitous Computing)과 Internet of Things(IoT) 환경의 컴퓨팅 환경에서도 컨텍스트의 개념은 동일한 의미로 폭넓게 적용된다[6-8]. 상황인지 시스템은 일반적으로 원시정보 계층, 상황인지 계층, 어플리케이션 계층 등의 세 계층의 구성을 바탕으로 사용자에게 서비스를 제공하며, 각 계층의 역할과 구성되는 모듈들은 다음과 같다. 첫 번째, 원시정보 계층은 데이터의

※ 본 논문은 2016년도 정부(미래창조과학부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업(No.2016R1C1B2009744)과 한국 산업통상자원부의 산업핵심기술개발사업 프로그램(No.10062501)의 지원으로 수행되었음.

† 준 회 원 : 숭실대학교 컴퓨터학부 박사과정
†† 정 회 원 : 숭실대학교 컴퓨터학부 조교수
††† 종신회원 : 숭실대학교 컴퓨터학부 교수

Manuscript Received : November 29, 2018
Accepted : February 20, 2019

* Corresponding Author : Jong Sun Choi(jongsun.choi@ssu.ac.kr)

생성 및 저장과 관련된 모듈이 구성된다. 두 번째, 상황인지 계층은 수집정보의 처리를 통해 어플리케이션 계층의 서비스들을 실행하기 위한 컨텍스트 정보를 생성하는 모듈이 구성된다. 이 때 생성되는 컨텍스트 정보는 서비스 실행을 담당하는 모듈, 관련 시스템 및 서비스 개발자들이 적용하기 쉽도록 컨텍스트 정보의 표현방법을 함께 고려해야 한다[8-9]. 세 번째, 어플리케이션 계층은 사용자에게 서비스를 제공하기 위한 모듈이 구성된다[10, 11].

어플리케이션 계층에서 제공되는 서비스들은 컨텍스트의 수준인 사용자 프로파일 정보, 위치정보, 센싱데이터 등을 일반적인 컨텍스트 정보로 사용하는 반면에, 분석정보를 필요로 하는 서비스들은 센싱데이터들이 컨텍스트 정보로 직접 사용되지 않는다. 예를 들면, 헬스케어 분야에서 제공되는 건강정보 분석결과 서비스는 수집된 센싱데이터를 분석하고, 그 결과를 컨텍스트 정보로 사용한다. 이 때 수행되는 분석작업을 수행하는 모듈들의 특징은 일반적으로 두가지가 있다. 첫 번째, 입력 센싱데이터들의 크기가 가변적이다. 두 번째, 입력자료들에 대한 처리 완료시점을 예측할 수 없다[12]. 이러한 요인들을 어플리케이션 계층에서 반영하여 요구사항을 구성하게 되면 서비스를 제공하는 처리루틴이 복잡해질 수 있는 문제가 발생한다. 따라서 어플리케이션 계층에서 요구사항을 최소로 고려할 수 있도록 상황인지 계층에서 분석과정에 필요한 컨텍스트 정보 및 분석결과에 대한 컨텍스트 정보를 관리할 수 있어야 한다.

본 논문에서는 상황인지 시스템이 대용량의 데이터 처리결과를 컨텍스트 정보로 사용하기 위한 컨텍스트 처리 방법을 제안한다. 제안하는 처리방법은 상황인지 계층에서 동작하고, 온톨로지 기반의 지식 표현방법을 통해 컨텍스트 정보를 기술하는 과정과, 대용량의 데이터를 처리하는 모듈의 구성에 따라 발생할 수 있는 분석모듈의 중복 실행을 방지하기 위한 처리순서를 함께 기술한다.

2. 관련 연구

상황인지 시스템에서 언급되는 컨텍스트의 개념을 이해하기 위해서는 Schilit 연구[4]에서 언급된 “상황인지 컴퓨팅에서 사용되는 소프트웨어는 인간과 사물로부터 얻은 수집정보, 위치정보 등에 적응할 수 있는 소프트웨어를 말한다”와 Dey의 연구[5]에서 언급된 “컨텍스트는 사용자 또는 사물의 위치, 식별정보, 개체의 상태 등을 나타내기 위한 정보”를 우선적으로 함께 접목시켜 개념을 정리할 필요가 있다. 기술 집적도가 개선되어 센서기기들이 소형화되고, 사용자 또는 주변환경을 측정할 수 있는 정보의 종류가 다양해짐에 따라, 컨텍스트의 정의는 서비스 도메인별로 세분화되고, 컨텍스트로 언급되는 데이터의 종류 및 이들을 표현하기 위한 수단도 다양하게 구성된다[13-15]. 이 때, 컨텍스트의 표현은 시스템과 사용자를 서로 연결해주는 가교역할을 수행한다. 상황인지 시스템들을 자세히 들여다보면, 사용자가 센서데이터에 대한 세부내용을 인지하지 않더라도, 상황인지 시스템이 센서데이터를 수집, 정제, 분석, 평가 등의 과정을 거쳐 사용자가 이해할 수 있는 정보로 제공된다.

Dmitry의 연구[13]에서는 M2M환경에서 발생하는 수집정보를 바탕으로 어플리케이션 서비스를 제공하기 위해 필요한 컨텍스트 정보를 XML 포맷으로 정의하여 표현하였다. 구조화된 명세정보를 표현할 수 있는 XML 포맷은 어플리케이션 계층에서 처리하는 입력자료의 구성요소를 검사하고, 명시되지 않은 자료에 대한 피드백을 시스템에 알려줄 수 있는 장점을 가진다. Vivian의 연구[14]에서는 컨텍스트 정보의 표현을 위해 CARF(Context-aware Reference Framework)를 제안하였으며, 제안하는 프레임워크에서는 사용자가 이해할 수 있는 식별정보(Where, When, Who)를 특정 상태(What, Why, How, to What)와 함께 맵핑할 수 있는 관계를 Adaptation으로 정의한 구조를 보여준다. 선행연구[15]에서는 컨텍스트 정보를 Triplet 기반의 표현방법을 이용하였으며, 이는 CARF에서 제안한 식별정보와 상태를 관계정보인 Adaptation으로 정의하여 표현한 예를 보여준다.

XML 또는 Triplet 포맷으로 정의되는 컨텍스트 정보는 사용자에게 서비스를 제공하기 위한 근거자료로 사용되며, 어플리케이션 계층에 정의된 서비스 명세의 입력정보로 사용된다[16]. 어플리케이션 계층에 정의된 서비스들은 주변환경 정보를 확인하기 위해서 컨텍스트 정보를 이용하는데, 일반적인 고려대상인 센싱데이터를 컨텍스트 정보로 표현하는데는 어려움이 없어 보인다. 하지만, 수집정보들을 분석한 결과를 사용자에게 제공하는 서비스의 경우 센싱데이터의 수집과는 별개로 분석을 위한 작업이 추가로 필요하다[12]. 이러한 분석작업에서의 입력자료는 빅데이터 또는 대용량의 데이터라고 불리는 것들이 대표적인 예이며, Hadoop 기반의 시스템 또는 원격지에 위치한 별도의 분석 플랫폼을 통해 분석작업이 수행된다. 일반적인 분석 시스템들은 데이터의 크기가 커지면 커질수록 분석작업의 처리완료 시간을 예측할 수 없고, 저장공간 또는 컴퓨팅 성능과 연관된 자원할당 작업에 민감하며, 어플리케이션 계층의 서비스들과 연계하기 위한 인터페이스가 별도로 개발되어야 하는 이유로 모든 처리과정이 자동화되어있지 않고, 데이터 전처리 또는 배치작업 등과 같이 일부분의 기능만 자동화되어 있다.

상황인지 시스템에서 분석작업을 필요로하는 서비스가 제공되려면 크게 두가지의 문제를 고려할 수 있다. 첫 번째는 분석 시스템 또는 플랫폼들에서 제공되는 기능들을 연동하는 것인데, 이는 시스템 또는 플랫폼 별로 제공되는 기능이 다를 수 있기 때문에 공통되는 작업에 대한 기능부터 연동하는 것이 선행되어야 한다. 두 번째는 분석작업을 요청하였을 때 이들을 반복적으로 요청하였을 때 발생할 수 있는 잉여 자원할당을 야기하지 않도록 요청작업들을 사전에 제어하는 매개체가 구성되어 있는지의 여부이다. 컨텍스트 표현방법이 상기 언급한 문제에 대한 매개체역할을 수행할 수 있으며, 그 결과 잉여 자원이 할당되지 않고 분석작업을 수행할 수 있다.

Alastair[17], 박지형[18], 그리고 Irena[19]의 연구들은 데이터를 획득하는 과정 또는 결과정보를 획득하는 과정의 특징

을 반응형/비반응형 응답(Responsive/Unresponsive feedbacks)와 같이 두가지의 범주로 나누어 기술한 것을 보여준다. 이러한 접근방법은 대용량의 데이터 처리과정이 본질적으로 갖는 입력자료 크기의 가변성과 처리시간을 예측할 수 없는 문제를 별개로 고려할 수 있도록 한다. 따라서 컨텍스트 표현방법은 센서데이터 또는 대용량의 수집정보 등과 같은 입력자료들에 대한 반응형/비반응형별로 처리될 수 있도록 기술되어야 한다.

3. 대용량의 데이터 처리결과를 컨텍스트 정보로 제공하기 위한 방법

본 장에서는 상황인지 시스템의 개요와 제안하는 대용량의 데이터 처리결과를 컨텍스트 정보로 제공하기 위한 방법을 기술한다. Fig. 1은 어플리케이션 계층에서 적용할 수 있는 분야 중 하나인 헬스케어 분야를 상황인지 시스템과 함께 구성할 때, 각 계층별 수행하는 기능들의 구성을 나타낸 것이다.

상황인지 시스템을 통해 제공되는 헬스케어 서비스는 3종의 시나리오를 예로 들어 기술한다. Table 1은 헬스케어 서비스 시나리오들의 요약정보를 기술한 것이다.

Table 1에 기술된 3종의 시나리오는 헬스케어 분야에서 로봇을 통해 사용자에게 서비스를 제공하는 서비스들을 간추린 것이다. 앞서 기술한 상황인지 시스템이 갖는 어플리케이션 계층의 서비스 도메인이 헬스케어 분야인 경우 구성된 서비스들이 제공되는 전체 과정은 다음과 같이 요약할 수 있다. 수집기기 정보 및 세부내용은 실험에서 기술한다.

1) 서비스 실행조건 검사
   상황인지 계층에서 제공되는 컨텍스트 정보를 검사
2) 컨텍스트 추상화
   수집정보를 바탕으로 컨텍스트 정보를 생성
3) 수집정보
   센싱데이터, 사용자 프로파일 등의 정보가 구성됨

본 논문에서 제안하는 컨텍스트 처리 엔진의 목적은 컨텍스트 추상화 과정을 통해 수집정보와 서비스 실행조건을 검사하는 과정을 서로 연결하는 것으로, 센싱데이터들을 컨텍스트 정보로 변환하기 위한 처리모듈, 센싱데이터를 적재하는 저장소의 관리 모듈, 서비스 실행 엔진의 요청을 처리하기 위한 인터페이스로 구성된다. 컨텍스트 처리 엔진의 세부적인 처리과정은 세 항목으로 나누어 각각 트리플릿 기반의 컨텍스트 표현방법, 반응형 수집정보 처리, 비반응형 수집정보 처리 과정에 대해 기술한다.

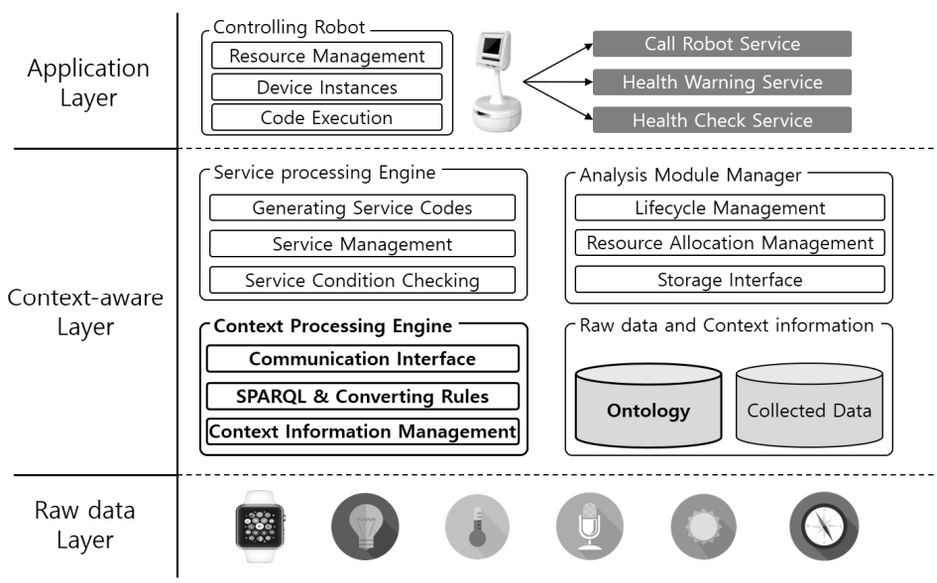

Fig. 1. Layered Architecture of Context-aware System for Healthcare Services

Table 1. Summary for healthcare service scenario examples

Service Name	Input device	Input data	Service
Call Robot Service	• Sensor devices	• Button action	• Robot moves to the sensor location
Health Check Service	• Smart scale • Wearable watch	• Location • Motion recognition	• Robot moves to the health checking location and informs to user
Health Warning Service	• Computer system for analysis	• Scale information • Activity information	• Robot moves to the user and reports

3.1 트리플릿 기반의 컨텍스트 표현방법

본 논문에서 사용되는 컨텍스트 표현방법은 온톨로지에 적용할 수 있는 지식표현 방법 중 하나인 트리플릿(Triplet)을 이용한다. 트리플릿은 <Subject, Predicate, Object>와 같이 표현되며, 정보를 표현하는 개체(Entity)를 트리플릿의 구성순서에 따라 각 Subject, Predicate, 그리고 Object 등의 위치에 배치하여 표현한다. 이렇게 구성된 트리플릿은 Subject와 Object는 Predicate의 관계를 통해 정의된다는 의미를 가지며, 이러한 트리플릿의 개념은 시멘틱웹 분야에서 널리 쓰이고 있다. 상황인지 시스템에서 트리플릿을 통해 컨텍스트 정보를 표현할 수 있는데, 예를 들면, 화장실에 부착된 움직임 감지 센서장치가 움직임 여부를 판단한 결과는 <MotionSensor_Bathroom, isNow, checkedIN> 또는 <MotionSensor_Bathroom, isNow, checkedOUT>의 형태와 같이 정의하고, 표기할 수 있다. 컨텍스트 정보를 트리플릿으로 표현되었을 때 상황인지 시스템에서 갖는 특징은 세 가지가 있다. 첫 번째, 개체의 정보를 단일 센서기기 또는 다수개의 센서기기로 표현할 수 있고, 나아가 시스템에서 제공되는 서비스의 명세로 표현할 수 있다. 두 번째, 상황인지 시스템은 각 센서기기로부터 제공되는 정보를 상태값으로 확인할 수 있다. 세 번째, 대용량의 센서데이터를 분석하는 과정이 서비스 제공과정에 포함되어 있을 경우, 분석작업들을 제어하고, 결과정보를 컨텍스트 정보로 표현할 수 있다. Fig. 2는 본 절에서 언급한 예시를 트리플릿으로 변환되고 이 컨텍스트 정보가 온톨로지에 저장되는 과정을 도식화한 것으로, 이 때, “isNow” Predicate은 최신의 센싱데이터를 입력정보로 이용한다.

3.2 반응형 수집정보 처리

반응형 수집정보 처리방법은 센서기기가 단수개 또는 센서데이터 하나만을 이용하여 서비스를 실행할 때 적용하는 처리방법으로, 분석과정이 포함되지 않는 대부분의 경우에 적용할 수 있는 처리방법이다. 3장에서 언급한 서비스들 중 반응형 수집정보 처리를 거치는 서비스는 로봇 호출 서비스와 건강정보 측정 서비스가 있다. 이들은 입력정보들을 각각의 트리플릿으로 표현하고, 상황인지 시스템이 각각의 트리플릿 정보를 확인하여 서비스를 실행할 수 있다. Fig. 3은 로봇 호출 서비스가 실행되는 과정을 상황인지 시스템, 컨텍스트 처리 엔진, 그리고 어플리케이션 서비스의 관점에서 나타낸 것이다.

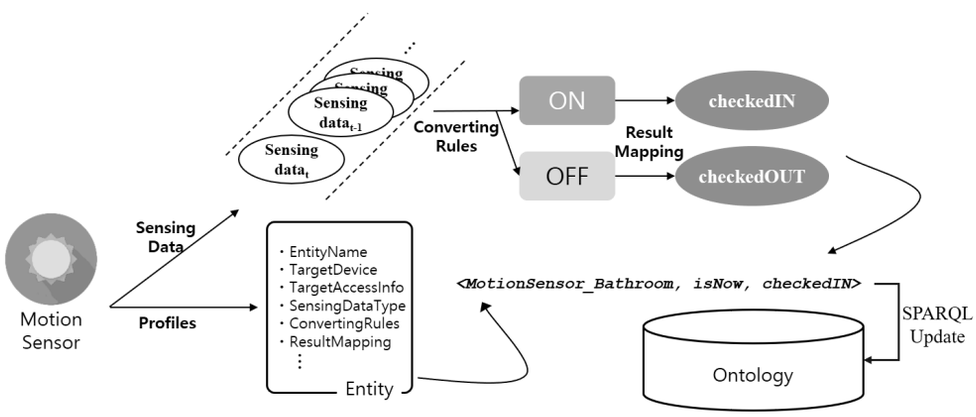

Fig. 2. Conceptual Procedure for Context Abstraction from Sensor Devices

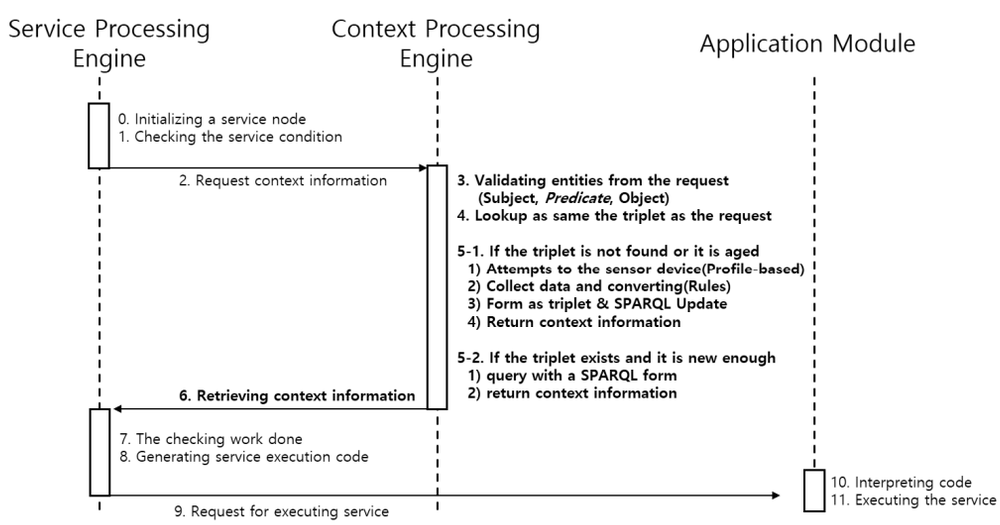

Fig. 3. Sequence Diagram for Processing Responsive Collected Data

3.3 비반응형 수집정보 처리

비반응형 수집정보의 처리는 다수개의 센서기기 또는 대용량의 데이터들을 분석하는 작업이 필요한 서비스들을 실행할 때 적용하는 처리방법으로, 서비스 실행엔진으로부터 요청받는 트리플릿 중 “ResultedAs” Predicate들을 처리하며, 아래 세 항목에 대한 요구사항을 만족할 수 있도록 처리과정을 가진다.

1) 분석결과는 최신의 수집정보를 입력정보로 한다.

2) 분석에 필요한 입력정보의 크기는 가변적이다.

3) 분석과정에 소요되는 시간은 가변적이다.

Fig. 4는 비반응형 수집정보 처리 시퀀스 다이어그램이다. 비반응형 수집정보 처리는 대용량의 데이터를 분석하고, 결과를 반환하는데 소요되는 시간과 컴퓨팅 자원을 컨텍스트 처리 엔진에서 최소한으로 부담할 수 있도록 분석작업에 대한 현황만 게시한다. 분석과정이 처리되는 과정에는 트리플릿의 Object 개체가 “Running” 상태로 갱신되며, 서비스 처리 엔진이 요청하는 동일한 입력자료에 대해서는 최신정보가 없다는 결과값을 리턴한다. 그 결과 서비스 처리 엔진이 요청하는 분석작업의 결과 상태값 대해서 분석과정이 끝나지 않은 작업들은 실행시키지 않고 서비스 기술 문서에 정의된 처리 루틴에 따라 서비스를 제공할 수 있으며, 서비스 처리 엔진 내 구성된 모듈을 번거롭게 수정하지 않고, 3.2절의 반응형 수집정보 및 본절의 비반응형 수집정보를 함께 처리할 수 있다.

4. 실 험

실험에서는 제안한 트리플릿 기반의 컨텍스트 정보 표현 방법에 따라 헬스케어 서비스를 제공하는데 필요한 컨텍스트 정보의 처리 과정을 보인다. Table 2는 세 종류의 헬스케어 서비스에 대해서 적용되는 컨텍스트 정보를 나타낸 것이다.

서비스 시나리오 별 사용되는 입력기기 및 정보에 대한 내용은 Table 3과 같다. 특히 건강적신호 리포트 서비스의 경우, 사용자는 웨어러블 기기를 착용한 상태로 하루 일과를 보내며 활동량 수치를 축적하는데 활동량 측정 모듈을 이용한다. 활동량 측정 모듈은 비반응형 수집정보 처리를 위한 예시를 보이기 위해 적용한 외부 모듈로서, 3축 가속도 센서의 값과 변위를 바탕으로 활동량을 측정하며 10 fps의 빈도로 원시정보와 활동량 데이터를 생성한다. 비반응형 수집정보의 처리과정에서 사용되는 분석모듈은 위 활동량 측정 모듈과 연계하여 활동량 레벨을 산출하며, 분석과정은 3시간, 12시간,

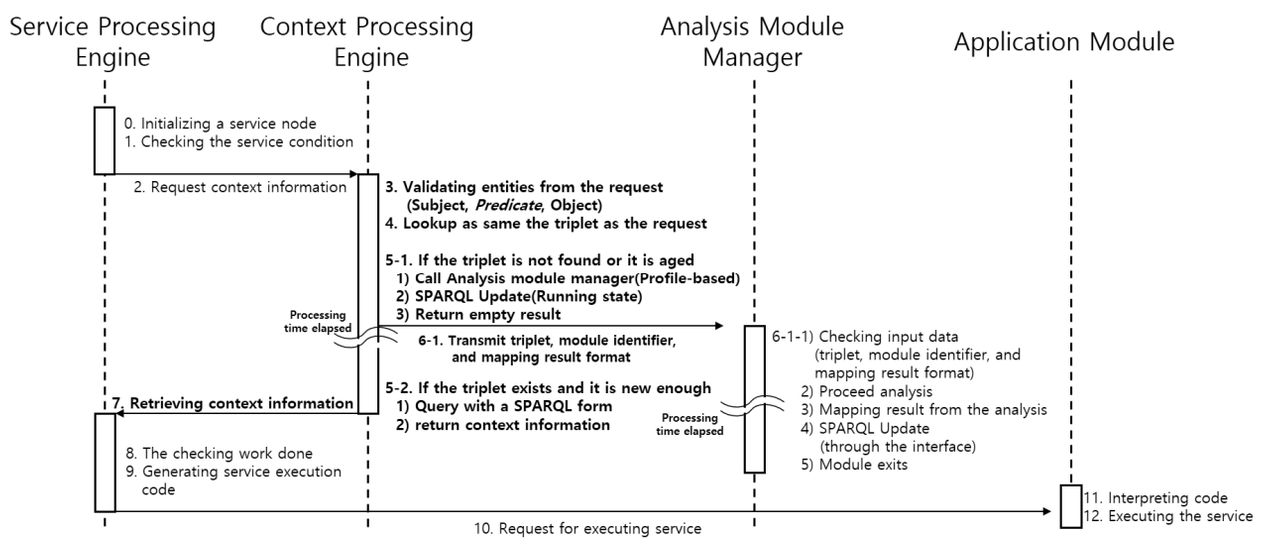

Fig. 4. Sequence Diagram for Processing Unresponsive Collected Data

Table 2. Service Summary with Context Representation for the Experiments

Service Name	Context representation in triplet form	Input device	User action
Call Robot Service	• <CallRobotService, isNow, True> • <CallRobotService, isNow, False>	• SmartThings multipurpose sensor	• Manipulate Input device
Health Check Service	• <HealthCheckService, isNow, True> • <HealthCheckService, isNow, False>	• SmartThings motion sensor • System time	• Moves around the motion sensor to be recognized in a specific time range
Health Warning Service	• <HealthWarningState, ResultedAs, Normal> • <HealthWarningState, ResultedAs, Abnormal>	• Samsung Galaxy Gear S2 • System time	• Activity information for a day life

Table 3. Summary of Input Devices for Experiments

Input device	Specification	Usage example or deployed
SmartThings Multipurpose Sensor	• Consists of two parts and provide 2 kinds of information whether these two parts are put together or not	• Door opened/closed • Used as a button
SmartThings Motion Sensor	• Similarly works as a general IR(Infrared) sensor, this recognizes the movement of objects	• Recognizes enter or out for the specific location
SmartThings Hub	• Hub interconnects the SmartThings devices and transmit data to the SmartThings server	• Gateway for SmartThings sensor devices. one used
Samsung Galaxy Gear S2	• Wearable watch • Consists of 3 axis accelerator sensor	• User puts on the devices • Activity information checking module included

Table 4. Features and Input and Output Data for Context Processing Engine

Submodules	Feature	Input/Output Data	
		Input	Output
Communication Interface	• Interfaces for service processing engine	• Three string set as a triplet	• Boolean Value
	• Interfaces for analysis module manager	• Module Identifier • Data location index for analysis	• None (SPARQL update)
	• Interfaces for users on the webpage	• Vue.js based webpage	
SPARQL & Converting Rules	• Query Generator for SPARQL	• Triplet	• Generated query result for SPARQL
	• Rules for converting sensor data	• Identifier of sensor device • Sensing data	• Mapped result status
Context Information Manager	• Triplet management as instances	• The list of context information	
Development tools & Protocol	• Context processing engine: Springboot based Java webserver application • Ontology for context information: Marklogic 9 • Protocols: RESTful based webservices		

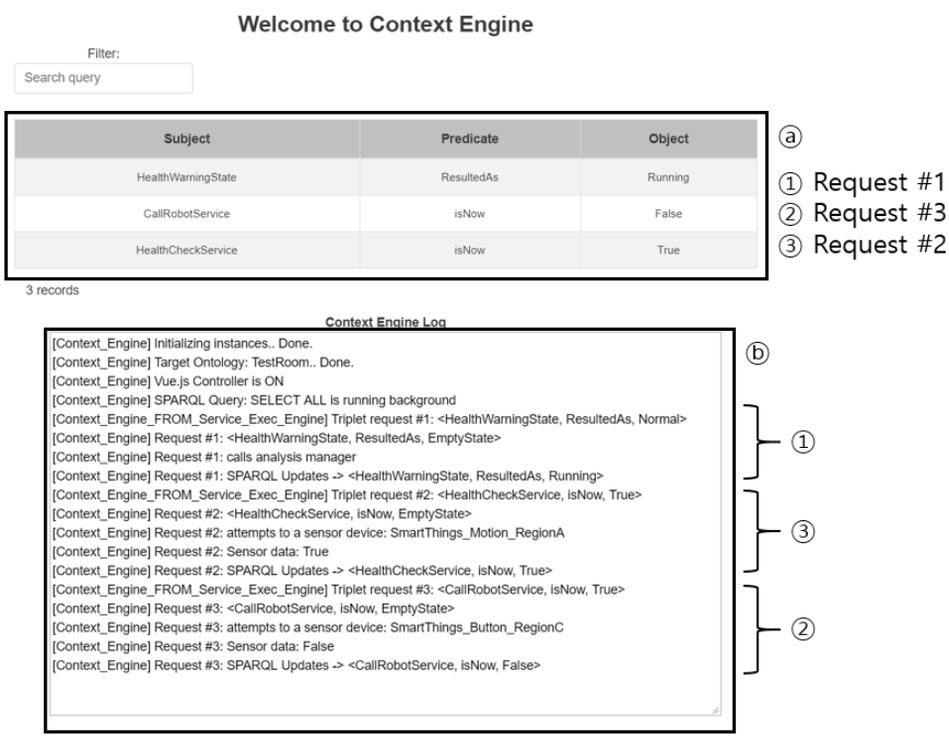

Fig. 5. Webpage for Users to Explore the List and Histories of Context Information

1일(24시간 내), 7일, 1개월 등 다양한 범위의 시간에 따른 분석을 수행하기 위해 가변 크기의 입력자료를 이용한다. 그 결과, 컨텍스트 처리 엔진은 수집 및 분석결과를 확인하는 과정에서 비반응형 수집정보를 제공하는 기기 또는 데이터 셋에서 컨텍스트 정보를 처리하는 작업에 대한 응답시간이 다르게 소요된다.

컨텍스트 처리 엔진은 3장의 Fig. 1과 상기 기술한 시스템 내에서 반응형/비반응형 수집정보를 처리하기 위해서 통신 인터페이스, SPARQL 및 규칙 관리 모듈, 컨텍스트 정보 관리 모듈 등의 3종류의 모듈로 구성된다. 각 모듈의 기능, 입력자료와 출력자료는 Table 4와 같다.

Fig. 5는 컨텍스트 정보의 현황을 보여주는 Vue.js로 구현된 웹페이지 화면이다. ⓐ는 트리플릿 목록 및 각 트리플릿내 구성된 개체가 변동될 경우, 그 현황을 실시간으로 반영하여 사용자에게 보여주며, ⓑ는 컨텍스트 처리 엔진의 동작 내역을 나타낸 것이다. 각 처리과정에 대해 확인할 수 있는 로그는 ①, ②, 그리고 ③번으로 표시하였다. 실험에서는 세 번의 요청을 통해 각 컨텍스트 정보를 요청하고, 1번 요청과 같이 비반응형 수집정보 처리가 필요한 컨텍스트 정보인 경우 인 경우 본 논문에서 제안한 비반응형 수집정보 처리과정을 수행하고 분석작업 중인 경우 Running으로 표기가 된다. 2번 또는 3번 요청들과 같이 반응형 수집정보 처리가 필요한 경우 컨텍스트 처리 엔진이 각 센서기기에 접근하여 센싱데이터를 수집 후 변환규칙에 따라 상태값이 갱신된다.

5. 결 론

본 논문에서는 기존의 상황인지 시스템에서 처리하는 단일 센싱데이터에 대한 컨텍스트 정보 처리와 함께 대용량의 데이터 처리결과를 컨텍스트 정보로 사용하기 위해 필요한 컨텍스트 정보의 정의 및 컨텍스트 처리 엔진의 동작과정을 보였다. 제안하는 방법을 통해서 본 논문에서 보인 상황인지 시스템은 반응형/비반응형 수집정보에 대해 컨텍스트 정보를 검사할 수 있고, 비반응형 수집정보 처리 과정에서 필요한 분석작업은 분석 모듈 관리자가 처리하여 분석결과를 제공하여 이를 컨텍스트 정보로 함께 적용할 수 있다.

특히 대용량의 데이터를 처리하는 경우 발생하는 문제 중 하나인 데이터 분석과정에 대한 응답시간이 상이한 데이터 셋을 처리하는 방법에 대한 문제를 비반응형 수집정보 처리 과정을 통해 해결할 수 있다. 향후에는 컨텍스트 정보를 처리하는 도메인별 서비스들의 요구사항을 반영하여 제안한 처리 방법을 통해 사용자에게 맞춤형 서비스를 제공할 수 있는 후속연구를 진행할 것이다.

References

[1] G. M. Kapitsaki, G. N. Prezerakos, N. D. Tselikas, and I. S. Venierisa, “Context-aware service engineering: A Survey,” *Journal of Systems and Software*, Vol.82, Issue 8, pp.1285-1297, 2009.

[2] C. Emmanouilidis, R. A. Koutsiamanis, and A. Tasidou, “Mobile guides: Taxonomy of architectures, context awareness, technologies and applications,” *Journal of Network and Computer Applications*, Vol.36, Issue 1, pp.103-125, 2013.

[3] U. Alegre, J. C. Augusto, and T. Clark, “Engineering context-aware systems and applications: A survey,” *Journal of Systems and Software*, Vol.117, pp.55-83, 2016.

[4] B. Schilit, N. Adams, and R. Want, “Context-aware computing applications,” *WMCSA, First Workshop on Mobile Computing Systems and Applications*, pp.85-90, 1994.

[5] A. K. Dey, “Understanding and Using Context,” *Personal and Ubiquitous Computing*, Vol.5, Issue 1, pp.4-7, 2001.

[6] C. Bettini, O. Brdiczka, K. Henricksen, J. Indulska, D. Nicklas, A. Ranganathan, and D. Riboni, “A survey of context modelling and reasoning techniques,” *Personal and Ubiquitous Computing*, Vol.5, Issue 1, pp.4-7, 2001.

[7] J. Ye, S. Dobson, and S. McKeever, “Situation identification techniques in pervasive computing: A review,” *Pervasive and Mobile Computing*, Vol.8, Issue 1, pp.36-66, 2012.

[8] J. Aguilar, M. Jerez, and T. Rodríguez, “CAMeOnto: Context awareness meta ontology modeling,” *Applied Computing and Informatics*, Vol.14, Issue 2. pp.202-213, 2018.

[9] H. S. Choi, J. Y. Lee, N. R. Yang, and W. S. Rhee, “Ontology Based User-centric Service Environment for Context Aware IoT Services,” *The Journal of Korea Contents Association*, Vol.14, Issue 7, pp.29-44, 2014.

[10] T. Fissaa, H. Guermah, M. E. Hamlaoui, H. Hafiddi, and M. Nassar, “An Intelligent Approach for Context-Aware Service Selection using Machine Learning,” *LOPAL '18, Proceedings of the International Conference on Learning and Optimization Algorithms: Theory and Applications*, No. 46, Rabat, 2018.

[11] G. Riva, “Ambient Intelligence in Health Care,” *Cyber Psychology & Behavior*, Vol.6, No.3, pp.295-300, 2003.

[12] S. Landset, T. M. Khoshgoftaar, A. N. Richter, and T. Hasanin, “A survey of open source tools for machine learning with big data in the Hadoop ecosystem,” *Journal of Big Data*, Vol.2, Issue 24, pp.1-36, 2015.

[13] D. Namiot, and M. Sneps-Sneppe, “On M2M Software,” *International Journal of Open Information Technologies*, Vol.2, No.6, pp.29-36, 2014.

[14] V. G. Motti and J. Vanderdonckt, “A Computational Framework for Context-aware Adaptation of User Interfaces,” *RCIS, Research Challenges in Information Science, 7th International Conference*, pp.1-12, IEEE, 2013.

[15] Y. S. Park, J. S. Choi, and J. Y. Choi, “Heterogeneous Sensor Data Acquisition Model for Providing Healthcare Services in IoT Environments,” *Korea Information Processing Society Transactions on Software and Data Engineering*, Vol.6, Issue 2, pp.77-84, 2017.

[16] D. G. Kwak, J. Y. Choi, and C. W. Yoo, “Rule-based BPEL System using Aspect Oriented Programming,” *Journal of Korea Information Science Society*: Software and Applications, Vol.39, No.2, pp.153-161, 2012.
[17] A. R. Beresford and F. Stajano, “Location Privacy in Pervasive Computing,” *Pervasive Computing*, Vol.2, No.1, pp.46-55, IEEE, 2003.
[18] J. H. Park and M. W. Park, “A Context-aware System in Ubiquitous Environment - Research Activity Guide Assistant,” *Journal of Korean Society for Precision Engineering*, Vol. 21, No.11, pp.31-37, 2004.
[19] I. Bojanova, G. Hurlburt, and J. Voas, “Imagineering an Internet of Anything,” *Computer*, Vol.47, Issue 6, pp.72-77, IEEE, 2014.

박 유 상
https://orcid.org/0000-0002-7606-820X
e-mail : yspark@soongsil.ac.kr
2014년 동아대학교 컴퓨터공학과(학사)
2016년 숭실대학교 컴퓨터학부(석사)
2016년~현 재 숭실대학교 컴퓨터학부 박사과정
관심분야 : 시스템소프트웨어, 분산처리, 지능형 로봇

최 종 선
https://orcid.org/0000-0001-9648-0667
e-mail : jongsun.choi@ssu.ac.kr
2000년 숭실대학교 컴퓨터학부(학사)
2002년 숭실대학교 컴퓨터학부(석사)
2008년~2010년 유한대학교 e-비즈니스과 전임교원
2010년 숭실대학교 컴퓨터학부(박사)
2011년~2012년 숭실대학교 지능형로봇연구소 연구원
2012년~2013년 서일대학교 인터넷정보과 전임교원
2013년~현 재 숭실대학교 컴퓨터학부 조교수
관심분야 : 시스템소프트웨어, 병렬/분산처리, 지능형 로봇

최 재 영
https://orcid.org/0000-0002-7321-9682
e-mail : choi@ssu.ac.kr
1984년 서울대학교 제어계측공학과(학사)
1986년 미국 남가주대학교 전기공학과 (석사)
1991년 미국 코넬대학교 전기공학부(박사)
1992년~1994년 미국 국립오크리지연구소 연구원
1994년~1995년 미국 테네시 주립대학교 연구교수
1995년~현 재 숭실대학교 컴퓨터학부 교수
관심분야 : 시스템소프트웨어, 병렬/분산처리, 고성능컴퓨팅