Graphgen: REST API를 이용한 시계열 데이터의 실시간 시각화 마이크로서비스

권동우*, 옥기수*, 지영민*
*전자부품연구원
e-mail:{dwkwon, ksok, ym.ji}@keti.re.kr

Graphgen: Real-time Visualization Microservice for Time Series Data Using REST API

Dongwoo Kwon*, Kisu Ok*, Youngmin Ji*
*Korea Electronics Technology Institute

요 약

최근 다양한 분야에서 대량의 데이터를 수집하여 처리하고 분석하는 빅데이터 기술이 활용되고 있다. 빅데이터 분석을 위해서는 데이터 시각화 기술이 필수적이다. 본 논문에서는 REST API를 사용하여 시계열 데이터베이스에 데이터를 질의하고, 응답받은 시계열 데이터를 다양한 형태의 차트로 시각화하는 마이크로서비스(Graphgen)를 설계하고 구현한다. 이 서비스는 데이터의 변동에 따라 실시간으로 시각화 객체를 갱신하며, 대용량 데이터 처리의 성능저하를 최소화하도록 개발된다. Graphgen은 InfluxDB와 OpenTSDB 시계열 데이터베이스와 Bokeh 시각화 라이브러리를 지원하며, 추후 서비스 확장이 용이하도록 개발된다. 또한 부하 분산과 통합 배포 관리를 위하여 컨테이너를 기반으로 개발된다.

1. 서론

최근 다양한 분야에서 대용량의 정형/비정형 데이터를 수집하고 분석하여 기존 분석 방법으로는 알아내기 힘들었던 새로운 현상과 통찰들을 찾아내는 빅데이터 기술이 각광을 받고 있다. 빅데이터 분석에는 다양한 접근 방법들이 존재하지만 데이터 시각화는 분야를 막론하고 필수적이다. 특히 시간 축에 대한 측정값을 가지고 일정한 간격으로 데이터가 수집되는 시계열 데이터의 시각화는 데이터의 경향성과 특성을 파악하기 위해 중요한 분석 방법[1]으로 활용되고 있다.

시계열 데이터의 시각화는 일반적으로 다음의 단계를 따른다. 먼저 시계열 데이터베이스에 시각화하고자 하는 데이터를 질의한다. 응답 받은 데이터를 시각화 라이브러리의 데이터 형식에 부합되도록 전 처리한다. 처리가 완료된 데이터는 시각화 라이브러리를 사용하여 다양한 그래프/차트 형태로 시각화하고 화면에 표시한다. 그리고 시간의 경과에 따라 데이터의 변동을 실시간으로 갱신한다.

그러나 이 과정에서 시계열 데이터의 질의 형식과 질의 방법은 시계열 데이터베이스 관리 시스템의 종류에 따라 달라진다. 시각화를 위한 데이터 변환도 시각화 라이브러리의 종류에 따라 달라지며, 시각화 방법 또한 라이브러리에 따라 상이하다. 따라서 시계열 데이터의 시각화를 위해서는 사용 중인 시계열 데이터베이스와 시각화 라이브러리의 종류에 따른 전문 지식과 데이터 처리 작업이 요구된다. 뿐만 아니라, 시간 경과에 따라 변화하는 대량의 데이터에 대한 실시간 요청 처리와 시각화 객체의 효율적인 업데이트 처리 작업이 요구된다.

본 논문에서는 빅데이터 분석에 있어서 시계열 데이터의 효율적인 시각화를 위해 데이터베이스와 시각화 라이브러리의 종류에 관계없이 시각화 과정을 자동으로 처리하는 서비스(Graphgen)를 설계하고 구현한다. 이를 위해, 시각화 서비스의 기능, 성능, 관리적 측면에서의 요구사항을 분석하고 서비스 구조를 설계한다. 서비스는 REST API를 통한 마이크로서비스 형태로 개발되며, 시계열 데이터베이스와 시각화 라이브러리에 독립적인 API를 제공한다. 개발된 서비스는 InfluxDB [2]와 OpenTSDB [3] 시계열 데이터베이스 관리 시스템을 지원하며, 시각화 라이브러리로는 Bokeh [4]를 지원한다. 그리고 이 외의 다른 데이터베이스와 시각화 라이브러리를 지원하기 위해 확장이 용이하도록 설계되고 구현된다.

본 논문의 구성은 다음과 같다. 2장에서는 제안하는 데이터 시각화 서비스의 요구사항을 분석한다. 3장에서는 서비스의 구조와 REST API 데이터 형식을 설계하고 구현한다. 4장에서는 서비스 개발 결과에 대해서 기술한다. 5장에서는 시각화 관련 연구에 대해서 논의한다. 마지막으로 6장에서는 결론과 향후 연구에 대해 논의한다.

2. 서비스 요구사항 분석

일반적으로 시계열 데이터를 수집하고 시각화 하는 과정은 다음과 같다. (1) 시계열 원천 데이터의 수집, (2) 수집된 데이터를 시계열 데이터베이스에 저장, (3) 시각화하

고자 하는 시계열 데이터의 질의, (4) 시각화 라이브러리의 형식에 따른 시계열 데이터의 전 처리, (5) 처리된 시계열 데이터를 시각화 라이브러리의 처리 형식에 맞게 요청, (6) 시각화 결과를 화면에 표시, (7) 데이터 변동에 따른 시각화 객체의 실시간 갱신과 같은 과정을 거친다.

이 과정 중, (3) 과정에서 시계열 데이터베이스 관리 시스템의 종류에 따라 데이터 질의의 형식과 방법이 상이하다. 또한 (4)와 (5) 과정에서 사용되는 시각화 라이브러리의 종류에 따라 데이터 변환 방법과 시각화 요청 방법이 달라진다. 따라서 서비스 요구사항으로, 다양한 시계열 데이터베이스 및 시각화 라이브러리의 지원과 추후 지원되는 서비스의 확장이 유연하도록 설계되어야 한다.

다음으로는 성능 요건이 만족되어야 한다. 시각화 서비스는 단일 과정이 아니라, 완전히 분리되어 있는 시계열 데이터베이스로의 질의 과정, 데이터 처리 과정, 그리고 시각화 라이브러리의 시각화 과정이 연계되어 있다. 따라서 각 과정 간의 데이터 연동과 통신 방법에 따른 성능 저하를 최소화하기 위한 방법이 강구되어야 한다.

그리고 데이터베이스 질의 처리 모듈과 데이터 전 처리 모듈, 그리고 시각화 처리 모듈을 통합한 배포 관리가 필요하다. 위에서 언급한 (7) 과정과 같이 시각화 라이브러리는 데이터를 시각화한 이후에도 데이터의 변동에 따라 실시간 갱신 기능을 제공한다. 이러한 실시간 업데이트 기능을 위해서 시각화 라이브러리는 서버 형태의 별도 프로세스로 동작한다. 따라서 각 시각화 과정별 필요한 모듈들과 별도의 프로세스로 동작하는 시각화 서버를 포함하는 통합된 배포 관리 기능이 요구된다.

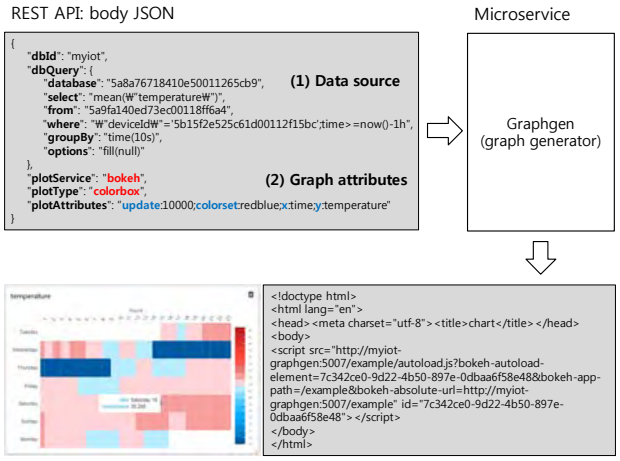
본 논문에서 제안하는 시계열 데이터 시각화 서비스는 빅데이터 분석 뿐 아니라 다양한 응용 서비스 및 애플리케이션 프로그램과 연동되어 동작하기 위해 마이크로서비스 형태로 개발되어야 한다. 또한 다양한 외부 서비스, 애플리케이션 프로그램과 연동하기 위해서 유연한 인터페이스 설계가 필요하다. 마지막으로, 제안하는 서비스가 대량의 데이터를 실시간으로 처리해야하기 때문에 부하 분산 기법 등을 이용한 규모 확장성이 고려되어야 한다.

3. 서비스 설계 및 구현

본 논문에서 제안하는 Graphgen 서비스는 데이터베이스에 저장된 시계열 데이터를 다양한 형태로 시각화하고, 데이터의 변동에 따라 시각화 객체를 실시간으로 업데이트하는 마이크로서비스이다. 그림 1은 제안 서비스의 개념과 서비스 요청 형식/응답 메시지를 나타낸다.

서비스 사용자는 시계열 데이터의 시각화를 위해서 Graphgen에 서비스를 요청한다. 서비스 요청 인터페이스는 REST API를 사용하며, 서비스 요청 형식은 사람이 이해하기 쉬우면서 요청 메시지로 인한 네트워크 혼잡을 줄이기 위해 그림 1의 입력 부분과 같이 경량 데이터 형식인 JSON (JavaScript Object Notation)을 사용한다.

서비스 요청 메시지는 크게 다음 두 가지 정보를 포



(그림 1) Graphgen 시각화 서비스 개념과 메시지 형식

함한다. 첫 번째로 시계열 원천 데이터를 가져오는데 필요한 데이터베이스 질의 정보(dbId, dbQuery)와 두 번째로 시각화하려는 차트의 유형(plotService, plotType)과 속성(plotAttributes) 정보이다. 데이터베이스 질의 정보는 데이터베이스 서버 정보에 대한 식별자인 서비스 ID와 데이터베이스명, 필드명, 조건식 등을 포함한다. 시각화 객체 정보는 차트 유형과 객체 색상, 축 정보, 데이터 갱신 간격 등을 포함한다. 표 1은 차트 속성의 일부를 나타낸다.

서비스 응답 메시지는 그림 1의 출력 부분과 같이 시각화 객체를 표시하고 데이터 변동에 따라 실시간 갱신하는 동적 웹 문서(HTML 콘텐츠)로 전달된다. 동적 웹 문서는 Graphgen 서비스와 지속적인 통신을 통해 실시간 데이터 변동을 시각화 객체에 반영하여 화면에 표시한다.

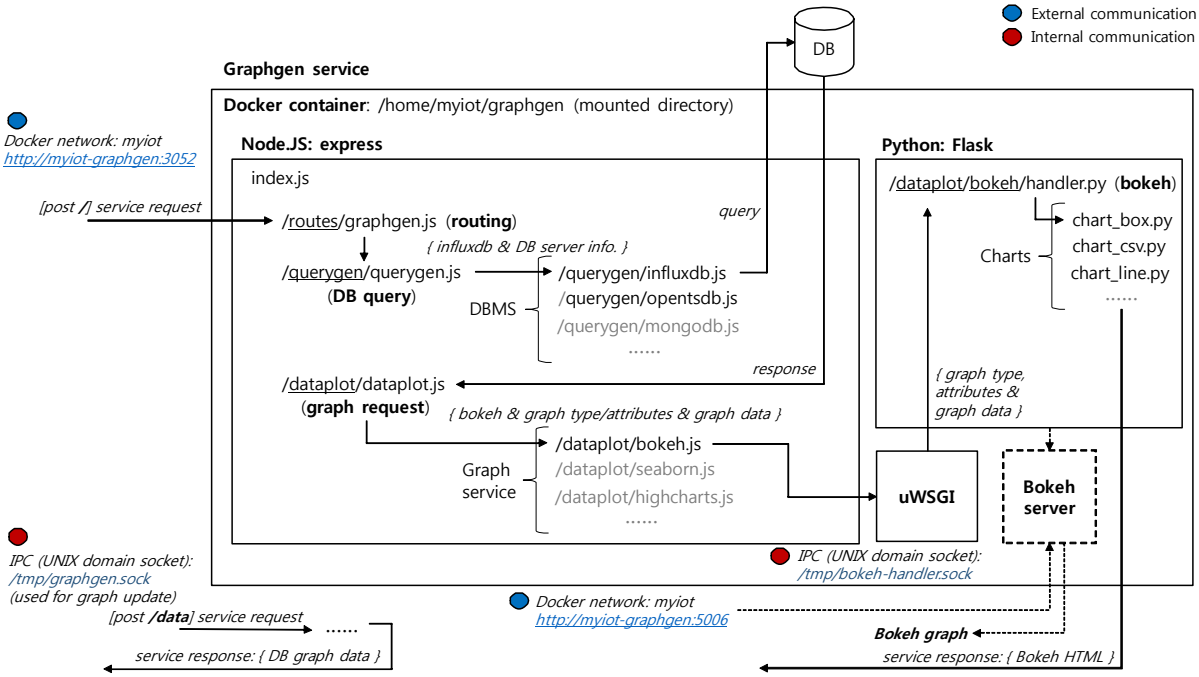
Graphgen 서비스는 Node.js와 Python을 사용하여 구현되며, 그림 2는 Graphgen 서비스 구조와 모듈 간 통신 관계를 나타낸다. Graphgen 서비스 제공에 필요한 처리 모듈들, 관련 라이브러리들, 시각화 서버 등은 배포 관리 요구사항을 만족시키기 위해서 Docker 컨테이너 상에 모두 통합되어 개발되고 동작한다.

Graphgen 서비스 구조는 크게 REST API 처리 모듈, 시계열 데이터베이스 질의/응답 처리 모듈, 시각화 서비스 식별 및 시각화 데이터 전 처리/전달 모듈, 시각화 라이브러리를 사용한 시각화 처리 모듈, 네 가지로 구성된다.

REST API 처리 모듈은 Node.js와 express를 사용하여 개발되었으며, 외부로부터 REST 서비스 요청을 받으

<표 1> 차트 속성 명세

Attribute	Description
update	Chart data update interval
colorset	Predefined color range set for bars
x	X-axis label
y	Y-axis label
colorrange	Selected color range in a color set
linecolorset	Predefined color set for lines
area	Area range for background annotations
areacolorset	Area color set for background annotations
linecolor	Line color
barcolor	Bar color



(그림 2) Graphgen 시각화 마이크로서비스 구조 및 모듈 간 통신 흐름

면 REST 서비스 요청 경로(POST /graphgen)에 따라 routes/graphgen.js 모듈에서 서비스 요청을 처리한다. REST 서비스 인터페이스는 외부 IP 주소와 포트번호를 이용한 외부 통신용 인터페이스와 UNIX domain socket (UDS)을 이용한 내부 통신용 인터페이스가 존재한다.

외부 통신 인터페이스는 외부에서 마이크로서비스 형태로 제공되는 Graphgen 컨테이너에 서비스를 요청할 때 사용된다. 내부 통신 인터페이스는 성능 요구사항을 만족시키면서 시각화 객체를 실시간으로 갱신하기 위해, 프로세스 간 데이터베이스 질의 등의 내부 통신 용도로 사용된다. 내부 통신을 위해서 Loopback 인터페이스를 사용하는 것보다 UDS를 사용하여 프로세스 간 통신을 사용하는 것이 성능 측면에서 큰 향상이 있기 때문에 외부, 내부 통신 인터페이스를 분리한다.

시계열 데이터베이스 질의/응답 처리 모듈(querygen /querygen.js)은 REST API 처리 모듈로부터 전달된 질의 내용을 기반으로 데이터베이스 서버 정보와 데이터베이스 종류에 따른 질의 함수를 가져온다. 그리고 식별된 데이터베이스의 종류에 따라서 실제 질의 요청을 처리한다.

데이터베이스 관리 시스템별 질의 처리는 모듈화 되어 있고, 모듈 간 인터페이스가 규격화되어 있기 때문에 추후 데이터베이스 지원 확장이 용이하도록 설계, 구현되었다. 현재 지원하는 데이터베이스 중 InfluxDB 질의 처리는 querygen/influxdb.js 모듈에서 수행하고, OpenTSDB 질의 처리는 querygen/opentsdb.js 모듈에서 수행한다.

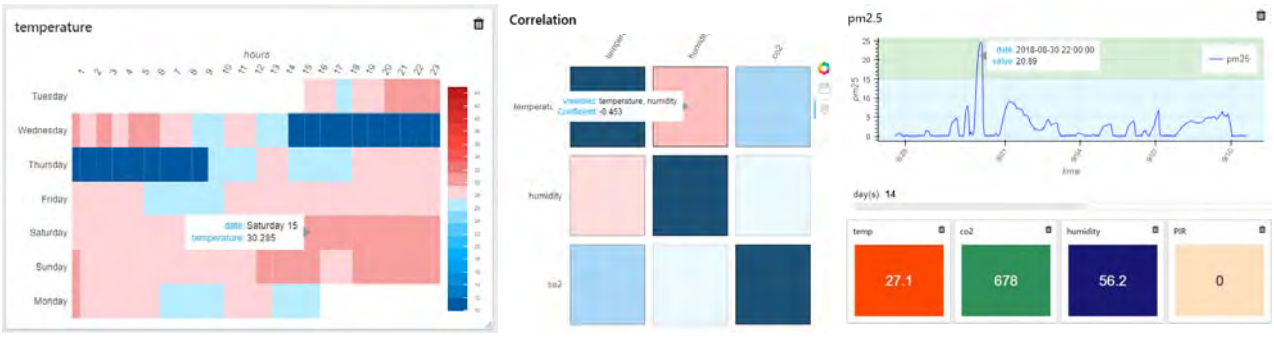
InfluxDB는 SQL-like 질의 구문을 지원하기 때문에 그림 1의 예시와 같은 일반적인 SQL 질의 형식과 유사하게 질의한다. Graphgen은 데이터베이스 서버로 다중 질의를 지원하며 다중 질의를 위해서 JSON 형식의 dbQuery 값을 배열 형태로 정의할 수 있다. 다중 질의는 여러 시계열 데이터 간의 상관관계 분석결과를 시각화하거나 시계열 데이터 셋에 대한 선형 그래프를 나타낼 때 사용된다.

OpenTSDB는 dbQuery 키의 select에서 질의하고자 하는 메트릭(metric)을 ';' 문자를 구분자로 사용하여 나열하여 데이터를 요청한다. 다중 질의를 위해서 InfluxDB와 동일한 방식으로 dbQuery 키 값을 배열 형태로 사용하여 질의하는 방식을 지원한다. 하지만 OpenTSDB에서는 다중 메트릭을 사용하여 다중 질의가 가능하기 때문에 성능 측면에서 다중 질의 각각에 단일 메트릭 데이터를 요청하는 것보다 단일 질의에 다중 메트릭 데이터를 요청하는 것이 불필요한 지연을 줄일 수 있다.

시각화 데이터 전 처리/전달 모듈(dataplot/dataplot.js)은 서비스 요청 메시지 내용을 기반으로 시각화 라이브러리 정보와 시각화 라이브러리 종류에 따른 시각화 요청 함수를 가져온다. 그리고 시계열 데이터베이스로부터의 질의 결과를 변환하고 식별된 시각화 모듈의 POST /charts 경로로 서비스를 요청한다. 이 때, 다중 질의 결과에 대한 시계열 데이터의 형식은 '[[[time, value], [time, value], ...], [[time, value], [time, value], ...] ...]'와 같으며, 단일 질의 결과일 경우 길이가 1인 동일한 배열 형식이다.

Bokeh 시각화 라이브러리를 사용한 시각화 처리 모듈은 Python과 Bokeh 라이브러리를 사용하여 개발되었다. Bokeh 서버는 bokeh.server 라이브러리를 이용하여 모듈에 내장된 형태로 동작한다. Bokeh 시각화 서비스는 Flask와 Flask_restful 모듈(dataplot/bokeh/handler.py)을 사용하여 REST 기반 서비스를 제공한다.

Node.js로 개발한 Graphgen 주 서비스와 Python으로 개발한 Bokeh 시각화 서비스 간의 통신은 REST API로



(그림 3) Graphgen 시각화 마이크로서비스 개발 결과

이루어진다. Python은 웹 서버와 웹 애플리케이션 간의 연동 및 통신을 위해서 Web Server Gateway Interface (WSGI) 규격을 사용한다. 따라서 서비스 간 통신 구조는 Graphgen-(UDS)-uWSGI-Bokeh 모듈과 같다.

그림 2의 서비스 구조와 같이 Graphgen은 컨테이너를 사용하여 REST API를 지원하는 마이크로서비스 형태로 개발되었기 때문에 특정 웹사이트에서의 시각화 서비스 제공이나 응용 애플리케이션 개발에 국한되지 않고 독립적인 시각화 서비스를 제공할 수 있다. 또한 Graphgen 서비스는 컨테이너 기반으로 동작하기 때문에 Docker swarm mode나 Kubernetes 등을 사용하여 서비스 복제를 통한 부하 분산과 규모 확장성을 지원한다.

4. 개발 결과

그림 3은 개발된 Graphgen 시각화 마이크로서비스를 사용하여 사물인터넷 환경 센서 장치로부터 시계열 데이터를 수집하고 시계열 데이터베이스에 저장한 다음, 데이터를 시각화한 것이다. 서비스 요청은 시계열 데이터베이스와 시각화 라이브러리의 종류에 따른 정보나 지식 없이 REST API를 통해 JSON 형식을 구성하여 이루어진다.

그림 3의 왼쪽 상단에서부터 차례로, (1) 주간-시간별 온도 분포 상태를 Colorbar 유형으로 시각화, (2) 온도/습도/이산화탄소 농도 간의 상관분석 결과 시각화, (3) 시간별 초미세먼지(PM-2.5) 측정값과 상태를 상태 영역이 존재하는 Line Annotation 유형으로 시각화, (4) 온도/이산화탄소 농도/습도/PIR 센서의 최신 측정값을 대시보드 형태로 시각화한 결과를 나타낸다.

5. 관련 연구

변정윤과 박용범 [5]은 데이터 시각화 작업 시에 입력된 정량적 데이터의 특성과 사용자의 의도를 고려하여 차트 유형을 추천하는 빅데이터 시각화 가이드라인과 그 도구를 제안하였다. 이러한 도구는 본 논문에서 제안한 서비스에서 시각화 서비스 요청 시의 차트 JSON 속성 정의를 최소화하거나 자동 추천 차트 설정으로 차트 속성 전달을 생략하여 비전문가의 데이터 시각화 편의성을 증진시키는 데 효과적으로 이용될 수 있다.

이중연 등 [6]은 대용량의 과학 데이터를 실시간으로 시각화하기 위한 요구사항을 분석하고, 공개 소프트웨어를 활용한 시각화 도구인 GLOVE를 제안하였다. 이 연구는 컨테이너를 활용한 부하 분산과 규모 확장성을 지원하는 본 논문의 접근 방법과 더불어, 애플리케이션 수준에서 병렬 처리 방법을 도입하여 실시간 차트 생성 및 갱신 성능을 향상시키는 방안으로 사용될 수 있다.

6. 결론

본 논문에서는 시계열 데이터베이스에 수집된 시계열 데이터를 시각화하고 데이터 변동에 따라 실시간으로 업데이트하는 Graphgen 마이크로서비스를 개발하였다. Graphgen은 데이터베이스나 시각화 서비스의 종류에 관계없이 서비스 어댑터 모듈화와 인터페이스 규격화로 서비스 지원 확장이 유연하며 다양한 유형의 차트를 정의하여 사용할 수 있다. 또한 컨테이너를 기반으로 개발되어 통합 배포 관리와 규모 확장성을 지원한다.

향후 연구는 데이터 크기에 따른 시각화 처리 성능을 측정하고, 부하 분산을 통한 서비스 확장 규모를 측정하여 내부 모듈들과 인터페이스를 최적화하는 것이다. 그리고 2D map, 3D floor map 등 다차원 시각화 모듈을 개발하는 것이 향후 연구이다.

감사의 글

본 연구는 산업통상자원부(MOTIE)와 한국에너지기술평가원(KETEP)의 지원을 받아 수행한 연구 과제입니다. (No. 20182010106460)

참고문헌

[1] 엄기순, 이원규, “빅 데이터의 예술적 접근: 데이터 비주얼라이제이션 사례 중심으로”, 한국정보처리학회지, 22권 4호, pp. 28-34, 2015년 7월.

[2] InfluxDB, https://www.influxdata.com/

[3] OpenTSDB, http://opentsdb.net/

[4] Bokeh, https://bokeh.pydata.org/

[5] 변정윤, 박용범, “사용자 의도 기반 정량적 빅데이터 시각화 가이드라인 툴”, 한국정보처리학회논문지: 소프트웨어 및 데이터공학, 5권 6호, pp. 261-266, 2016년 6월.

[6] 이중연, 김민아, 이세훈, 허영주, “GLOVE: 대용량 과학 데이터를 위한 분산공유메모리 기반 병렬 가시화 도구”, 한국정보처리학회논문지: 소프트웨어 및 데이터공학, 5권 6호, pp. 273-282, 2016년 6월.