

SSNO 기반 시공간 시맨틱 센서 웹

Spatio-Temporal Semantic Sensor Web based on SSNO

신인수* · 김수정** · 김정준* · 한기준****

In-Su Shin · Su-Jeong Kim · Jeong-Joon Kim · Ki-Joon Han

요 약 유비쿼터스 컴퓨팅 환경이 발전함에 따라 GeoSensor와 같이 GPS 기능을 보유한 센서로부터 생성된 시공간 센서 데이터 활용이 증가하고 있으며, 시공간 센서 데이터를 사용해 사용자에게 다양한 서비스를 효율적으로 제공해주기 위한 시맨틱 센서 웹이 연구되고 있다. 특히, W3C에서는 OGC의 SWE(Sensor Web Enablement)와 같은 센서 관련 표준들을 활용하고, 센서 데이터를 온톨로지로 표현할 수 있는 SSNO(Semantic Sensor Network Ontology)를 개발하였다. 그러나 이러한 연구들은 비시공간 센서 데이터에 대한 질의 처리는 가능하지만 시간과 공간 정보를 포함하는 시공간 센서 데이터를 효율적으로 처리하기 어렵다는 문제점이 존재한다. 따라서, 본 논문에서는 OGC의 “OpenGIS Simple Feature Specification for SQL”에서 제시한 공간 데이터 타입과 공간 연산자를 확장하여 시공간 데이터 타입과 시공간 연산자를 지원하는 SSNO 기반의 시공간 시맨틱 센서 웹을 개발하였다. SSNO 기반의 시공간 시맨틱 센서 웹은 시공간 센서 데이터인 SensorML(Sensor Model Language)과 O&M(Observations and Measurements) 스키마를 분석하여 SSNO 문서로 변환 및 저장하고, 시공간 연산자와 시공간 추론 규칙을 적용하여 효율적인 질의 처리를 수행한다. 마지막으로, 이러한 SSNO 기반의 시공간 시맨틱 센서 웹을 가상 시나리오에 적용해 봄으로써 본 시스템의 효용성을 검증하였다.

키워드 : 시맨틱 센서 웹, 시맨틱 센서 네트워크 온톨로지, 시공간 질의 처리, 시공간 연산자

Abstract According to the recent development of the ubiquitous computing environment, the use of spatio-temporal data from sensors with GPS is increasing, and studies on the Semantic Sensor Web using spatio-temporal data for providing different kinds of services are being actively conducted. Especially, the W3C developed the SSNO(Semantic Sensor Network Ontology) which uses sensor-related standards such as the SWE(Sensor Web Enablement) of OGC and defines classes and properties for expressing sensor data. Since these studies are available for the query processing about non-spatio-temporal sensor data, it is hard to apply them to spatio-temporal sensor data processing which uses spatio-temporal data types and operators. Therefore, in this paper, we developed the SWE based on SSNO which supports the spatio-temporal sensor data types and operators expanding spatial data types and operators in “OpenGIS Simple Feature Specification for SQL” by OGC. The system receives SensorML(Sensor Model Language) and O&M (Observations and Measurements) Schema and converts the data into SSNO. It also performs the efficient query processing which supports spatio-temporal operators and reasoning rules. In addition, we have proved that this system can be utilized for the web service by applying it to a virtual scenario.

Keywords : Semantic Sensor Web, Semantic Sensor Network Ontology, Spatio-Temporal Query Processing, Spatio-Temporal Operator

1. 서 론

최근 유비쿼터스 환경에서는 u-GIS, u-LBS, u-물류 등과 관련된 다양한 서비스를 제공하기 위해 GeoSensor와 같이 GPS(Global Positioning System) 기능을 보유한 센서로부터 생성된 시공간 센서 데이터를 활용하는

연구가 활발히 진행되고 있다[5,6]. 특히 OGC는 모든 종류의 센서 시스템과 웹에 연결된 센서들을 이용하기 위해 SWE(Sensor Web Enablement)[4,13]를 표준으로 제정하였다.

하지만, SWE는 센서 데이터의 교환, 처리, 모니터링 등을 목적으로 개발되었기 때문에, 컴퓨터가 시공

* In-Su Shin, Doctoral Student, Division of Computer Science and Engineering, Konkuk University. isshin@db.konkuk.ac.kr

** Su-Jeong Kim, Researcher, DCT Team Creative Innovation Center, LG Electronics. wjdd11021@gmail.com

*** Jeong-Joon Kim, Assistant Professor, Division of Computer Science and Engineering, Konkuk University. jjkim9@db.konkuk.ac.kr

**** Ki-Joon Han, Professor, Division of Computer Science and Engineering, Konkuk University. kjhan@db.konkuk.ac.kr (Corresponding Author)

간 센서 데이터를 의미있게 인식하고 이해하여 새로운 정보를 만들어 내기에 어려움이 있다. 이러한 문제를 해결하기 위한 방안으로 SWE와 Semantic Web를 접목한 새로운 웹 기술인 시맨틱 센서 웹(Semantic Sensor Web)이 제안되었다[2,3,8].

시맨틱 센서 웹의 핵심 기술인 W3C의 SSNO(Semantic Sensor Network Ontology)[2]는 센서와 센서 네트워크를 설명할 수 있는 온톨로지를 개발하고, SWE와 같은 표준을 따르는 시맨틱 어플리케이션에서의 온톨로지 활용 방법에 대한 연구로써 진행되었다. 최근 SSNO에 관한 연구 및 사용이 증가됨에 따라 수집된 시공간 센서 데이터를 처리하는 기술이 요구되고 있으나, SSNO는 일반적인 비시공간 온톨로지에 대한 질의 처리를 위해 개발되었기 때문에 시공간 센서 데이터에 대한 효율적인 질의 처리를 지원하지 못하는 단점을 가지고 있다.

따라서, 본 논문에서는 시공간 센서 데이터의 효율적인 질의 처리를 위해 OGC에서 제시한 “OpenGIS Simple Feature Specification for SQL”[11]의 공간 데이터 타입과 공간 연산자를 확장하여 시공간 데이터 타입과 시공간 연산자를 지원하는 SSNO 기반 시공간 시맨틱 센서 웹을 개발하였다. SSNO 기반 시공간 시맨틱 센서 웹은 센서의 메타데이터를 교환하는 SensorML (Sensor Model Language)과 센서에서 수집된 데이터를 표현하는 O&M(Observations and Measurements) 스키마를 분석하여 SSNO 문서로 변환 및 저장하며, 사용자의 요구사항에 따라 SSNO에 추론 규칙(Rule)을 적용하여 시공간 센서 데이터에 대한 온톨로지 질의 처리를 수행한다.

마지막으로 논문에서 개발한 SSNO 기반의 시공간 시맨틱 센서 웹을 기후 변화에 따른 빙판길 예상 시나리오에 적용해 봄으로써 본 시스템의 효용성을 검증하였다.

2. 관련연구

2.1 SWE(Sensor Web Enablement)

국제 표준화 단체인 OGC는 2007년에 지리공간정보를 통합하여 처리하기 위한 플랫폼 기술로서 SWE를 표준으로 제정하였다[12,13]. SWE는 인공물과 자연물에 다양한 센서를 설치하고 이를 웹으로 연결시킨 후 표준 프로토콜과 API를 사용하여 센서 데이터를 처리할 수 있는 다양한 기술들에 대한 표준과 센서들에 의하여 끊임없이 발생하는 실시간 정보를 효율

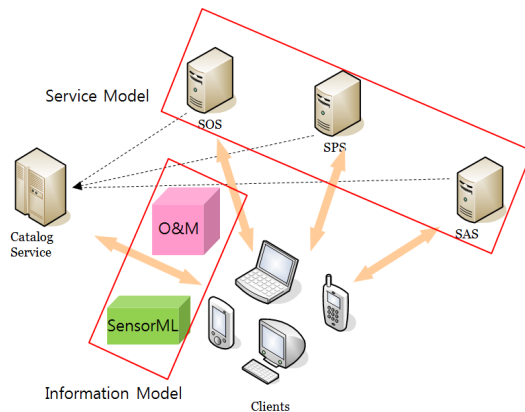


Figure 1. SWE Structure

적으로 얻어내기 위한 기술들에 대한 표준을 제시하고 있다. Figure 1은 SWE에서 제시하고 있는 기술들에 대한 구성도를 보여준다.

Figure 1에서와 같이 SWE는 센서와 센서 데이터를 공간정보 인프라에서 통합하기 위한 표준들의 프레임워크이며, 센서 웹을 통하여 어플리케이션과 서비스들이 웹상에서 다양한 종류의 센서들을 접근할 수 있게 하는 인터페이스와 프로토콜의 집합이다. Information Model은 센서 데이터를 표현하기 위한 표준 모델인 SensorML과 O&M로 구성되며, Service Model은 다양한 센서로부터 관측된 센서 데이터에 대한 접근을 제공하는 서비스인 SOS(Sensor Observations Service), 사용자에게 센서 데이터 수집과 변경 등을 알려주는 서비스인 SAS(Sensor Alert Service), 센서에 데이터 수집을 요청하고 수행을 지원하는 서비스인 SPS(Sensor Planning Service) 등을 포함하고 있다[9,10].

2.2 SSNO(Semantic Sensor Network Ontology)

국제 표준화 단체인 W3C는 2011년에 OGC의 SWE 등과 같은 표준을 따르는 센서 네트워크 및 센서 웹 응용프로그램에서 온톨로지를 사용하기 위한 목적으로 SSNO를 제안하였다[2,14]. SSNO는 iCore, OPENIoT 등 유럽의 FP7 프로젝트들에서 연구에 활용되고 있으며, 사물 인터넷(M2M/IoT) 관련 국제 표준 기구인 oneM2M에서도 SSNO를 확장하는 연구를 진행하고 있다. Figure 2는 SSNO에서 제안하고 있는 주요 클래스와 속성을 보여준다.

Figure 2에서와 같이 SSNO는 온톨로지 구성에 관한 클래스와 속성을 나타내는 Deployment, 플랫폼에 관한 클래스와 속성을 나타내는 PlatformSite, 시스템

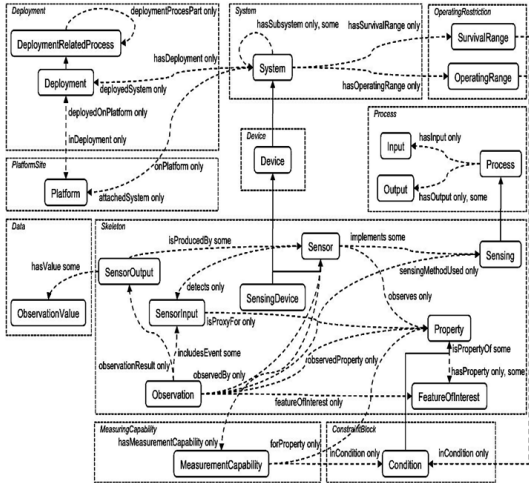


Figure 2. Overview of the SSNO Classes and Properties

에 관한 클래스와 속성을 나타내는 System, 장치에 관한 클래스와 속성을 나타내는 Device, 센서의 생존과 작동에 관한 클래스와 속성을 나타내는 OperatingRestriction, 센서의 입/출력에 관한 클래스와 속성을 나타내는 Process, 센서와 센서 수집 데이터에 관한 클래스와 속성을 나타내는 Skeleton 등으로 구성되어 있다.

2.3 SensorML(Sensor Model Language)

SensorML은 OGC에서 제안한 SWE의 Information Model로서, 다양한 센서 데이터들을 XML 인코딩으로 표현할 수 있는 표준 모델이다[10]. 즉, 센서에 대한 사전 지식이 없이도 센서를 발견하고 센싱 정보를 해석할 수 있도록 하기 위하여, 센서 및 센서 데이터의 다양한 특성과 속성들에 대한 메타데이터를 XML로 인코딩하여 제공한다. 더불어 센서에서 수행 가능한 정보 처리 함수를 제공함으로써, 사용자가 센서에 접근하여 해당 센서에 대한 임무를 부여하거나 원하는 정보 처리를 수행할 수 있다. Figure 3은 SensorML의 스키마를 보여준다.

Figure 3과 같이 SensorML은 접두어 'sml'을 사용한다. 루트 엘리먼트로 SensorML이 위치하며, 애트리뷰트로 version 정보를 기술할 수 있다. 루트 엘리먼트 하위로는 member 엘리먼트를 가지고 있는데, 이는 센서에 대한 정보를 기술하기 위한 엘리먼트로 하나 또는 무한대로 기술이 가능하다. member의 하위 엘리먼트는 선택 구조를 가진다. component 엘리먼트와 componentarray 엘리먼트는 센서를 구성하는 기능에 대해 정의하며, datasource 엘리먼트는 센서 데이터 연

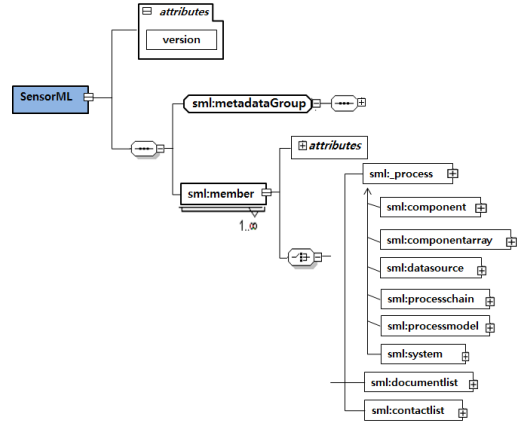


Figure 3. SensorML Schema

결에 대해 정의한다. processchain 엘리먼트는 연결된 process의 집합으로, processmodel과 component를 연결하여 측정 정보를 처리하는 방법을 정의하고, processmodel 엘리먼트는 센서에서 처리되는 정보의 처리 과정 및 물리적인 위치, 인터페이스를 정의한다. 그리고 system 엘리먼트는 복합적인 컴포넌트 그룹을 표현하며, 응용 프로그램과 관련된 물리적 인터페이스의 정보를 정의한다.

2.4 O&M(Observation & Measurements)

O&M은 SensorML과 마찬가지로 OGC에서 제안한 SWE의 Information Model로서, 센서가 관측한 센싱 정보를 교환하고 표현하기 위한 XML 인코딩을 정의한 표준이다[9]. O&M은 센서의 측정값이 특정 센서에 종속되는 데이터 포맷으로만 해석되는 문제를 해결하고, 이기종의 다양한 센서들 사이에서 데이터 교환과 접근이 가능하게 하기 위해 제안되었다. 또한, O&M은 아스키뿐만 아니라 바이너리 블록을 두어서 대량의 데이터 처리가 가능하며, 센서 데이터 자체 외에 센싱 시간, 위치, 특성, 품질, 이벤트 및 데이터 결과를 해석하는데 필요한 함수도 함께 제공한다. Figure 4는 O&M의 스키마를 보여준다.

Figure 4에서 보여주는 바와 같이 O&M은 접두어로 'om'을 사용하며, 루트 엘리먼트로 Observation이 위치한다. 하위 엘리먼트는 순차 구조로 구성되어 있으며, O&M 스키마는 단독으로 사용되기 보다는 다른 인코딩 언어와 함께 사용된다. Observation의 하위 엘리먼트인 samplingtime과 resulttime은 센서가 대상을 관측한 시간을 나타내는 엘리먼트이고, procedure 엘리먼트는 관측 방법을 나타내는 엘리먼트로써, SensorML

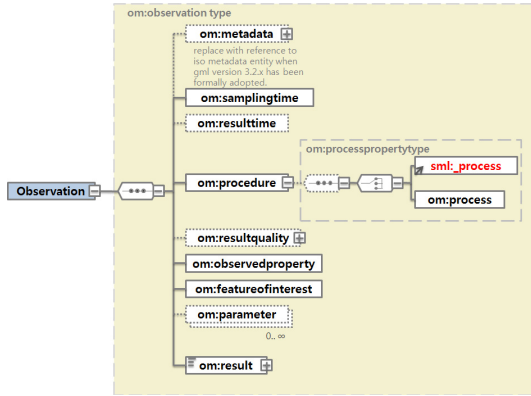


Figure 4. O&M Schema

의 process 엘리먼트와 O&M의 process 엘리먼트 중 선택하여 사용가능하다. 그리고 observedproperty 엘리먼트는 센서가 관측한 데이터의 특성 정보를 나타내며, featureofinterest 엘리먼트는 관측 대상에 대한 정보를 기술한다. 마지막으로 result 엘리먼트는 센서가 관측한 데이터를 나타낸다.

3. 시스템 설계

3.1 시스템 구조

SSNO 기반 시공간 시맨틱 센서 웹은 사용자의 요청에 따라 센서의 메타데이터를 교환하는 SensorML과 센서에서 수집된 데이터를 표현하는 O&M 스키마를 분석하여 SSNO 문서로 변환 및 저장하며, 변환된 SSNO를 대상으로 시공간 연산 및 추론을 수행함으로써 시공간 센서 데이터를 효율적으로 처리할 수 있다. Figure 5는 본 논문에서 개발한 시스템의 전체 구조를 보여준다.

Figure 5에서 보는 바와 같이 SSNO 기반 시공간 시맨틱 센서 웹은 Client와 Semantic Sensor Web Server로 구성된다. Client는 사용자의 질의를 입력받는 Query I/O Module과 질의의 결과를 보여주는 Display Module로 구성되고, Semantic Sensor Web Server는 Ontology Manager와 Query Processing Manager로 구성된다. 그리고 Sensor Observation Service는 다양한 시공간 센서 데이터를 수집하여 SensorML 혹은 O&M 형태로 Semantic Sensor Web Server에 전달해주는 서비스이다.

Ontology Manager는 SensorML의 구문 검사 및 유효성 검사를 실행하고 SensorML 문서를 분석하여 센서의 메타데이터를 생성하는 SensorML Parser Module, O&M 문서의 구문 검사 및 유효성 검사를 실행하고

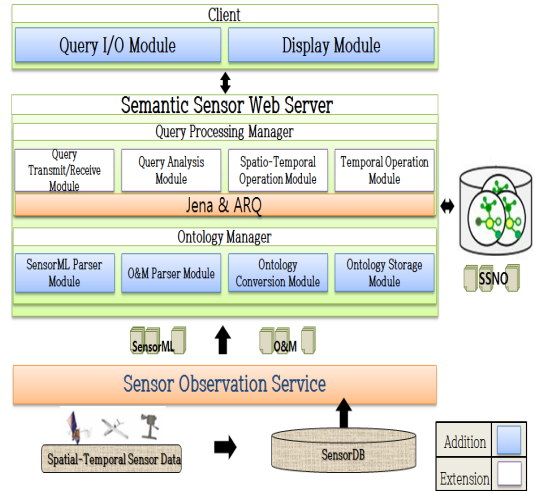


Figure 5. Architecture of Spatio-Temporal Semantic Sensor Web

각 문서를 분석하여 센서의 수집 데이터를 생성하는 O&M Parser Module, 생성된 데이터를 W3C에서 제시한 SSNO 규칙에 맞게 변환하는 Ontology Conversion Module, 정상적으로 변환된 파일을 유효성 검사 후 서버 내의 물리적 저장소에 저장하는 Ontology Storage Module로 구성된다.

Query Processing Manager는 Client로부터 입력된 질의를 Query Processing Manager로 송신하고 검색된 결과를 수신 받는 Query Transmit/Receive Module, 질의에 시공간 연산이 필요하지 판단하고 온톨로지 검색을 수행하는 Query Analysis Module, 시간 데이터 타입과 시간 연산자를 제공하는 Temporal Operation Module, 시공간 데이터 타입과 시공간 연산자를 제공하는 Spatio-Temporal Operation Module로 구성된다.

3.2 Client

Client는 Query I/O Module와 Display Module로 구성되며, Internet Explorer, Firefox와 같은 웹 브라우저를 사용한다.

Query I/O Module은 사용자로부터 입력된 질의를 Semantic Sensor Web Server의 Query Processing Manager로 전달하고, Query Processing Manager로부터 처리된 질의 결과를 받아 Display Module로 전달한다. 이를 위해 비동기적 데이터 교환 방식을 이용하며, 사용자로부터 HTML 형식으로 질의를 입력받는다.

Display Module은 Query I/O Module로부터 전달받은 결과 데이터를 분석하여 웹 브라우저 화면에 텍스트

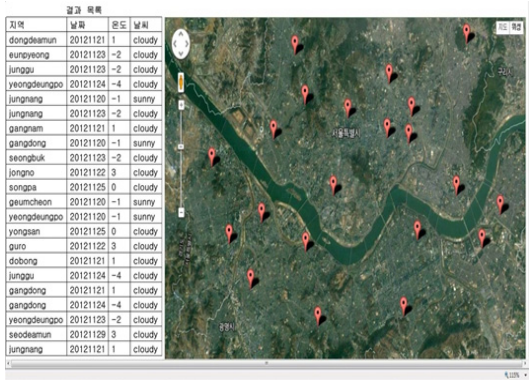


Figure 6. Output Screen for Query Result

트 형태와 지도 형태로 디스플레이 한다. Figure 6은 Display Module의 질의 결과의 디스플레이 화면을 보여준다.

Figure 6과 같이 좌측에는 텍스트 형태로 결과값이 출력되고, 우측에는 지도와 함께 결과값에 해당하는 위치가 표시된다.

3.3 Semantic Sensor Web Server

본 절에서는 Semantic Sensor Web Server를 구성하는 Ontology Manager와 Query Processing Manager에 대해서 상세히 설명한다.

3.3.1 Ontology Manager

Ontology Manager는 SensorML Parser Module, O&M Parser Module, Ontology Conversion Module, Ontology Storage Module로 구성된다.

SensorML Parser Module은 센서의 메타데이터 정보를 가지고 있는 SensorML 문서를 분석하고, 그 결과를 Ontology Conversion Module로 전달한다. Figure 7은 SensorML 문서의 분석 결과를 보여준다.

Figure 7에서 보듯이 SensorML 문서를 읽어와 productName 값으로 WeatherSensor를 추출하고, ModelNumber 값으로는 53759를 추출하여 저장하게 된다.

O&M Parser Module은 센서로부터 수집된 데이터를 저장한 O&M 문서를 읽어와서 구문 검사 및 유효성 검사를 수행한 후, 데이터를 분석한다. 그리고 분석된 결과를 Ontology Conversion Module로 전달한다. Figure 8은 O&M 문서의 분석 결과를 보여준다.

Figure 8에서 보듯이 O&M 문서를 읽어와 시간값인 beginPosition, endPosition과 위치값인 Point를 추출하여 저장하게 된다.

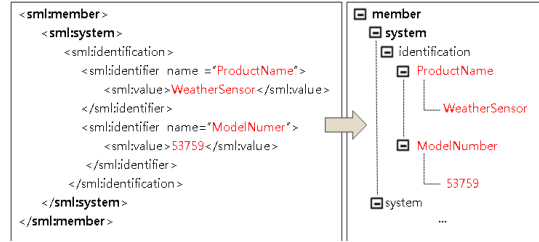


Figure 7. Analysis Result of SensorML Document

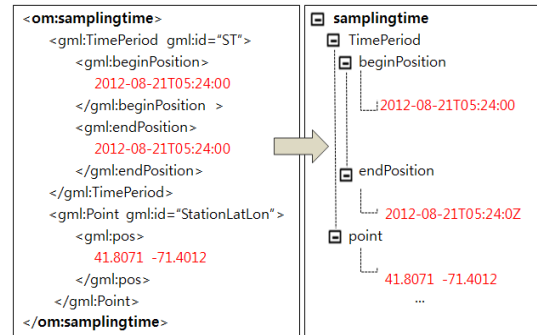


Figure 8. Analysis Result of O&M Document

Ontology Conversion Module은 SensorML Parser Module과 O&M Parser Module로부터 전달받은 센서 메타데이터 및 수집 데이터를 온톨로지 표현을 위해 SSNO로 변환한다. 또한, 변환된 온톨로지의 유효성을 검사하며, 정상적인 온톨로지 변환되었다면 이를 Ontology Storage Module로 전송한다. Figure 9는 SensorML 문서와 O&M 문서를 SSNO로 변환한 결과를 보여준다.

Figure 9에서 보면 O&M의 경우 GML을 내포하고 있으며, om:member, om:observation, om:samplingTime은 각각 ssn:Sensor, ssn:observation, ssn:observation SamplingTime으로 변환된다. SensorML의 경우 sml:System은 ssn:System으로 변환되고, 변환된 결과는 Ontology Storage Module을 통해 물리적으로 저장된다.

Ontology Storage Module은 Ontology Conversion Module에서 주기적으로 변환된 SSNO를 서버의 물리적 저장소에 저장한다. 또한 질의 응답 속도의 향상을 위해 웹에서 읽어온 온톨로지 데이터를 서버의 저장소에 파일 형태로 저장한다. 서버는 주기적으로 이러한 온톨로지 데이터를 자동 갱신함으로써 데이터를 최신으로 유지한다.

3.3.2 Query Processing Manager

Query Processing Manager는 Query Transmit/Receive

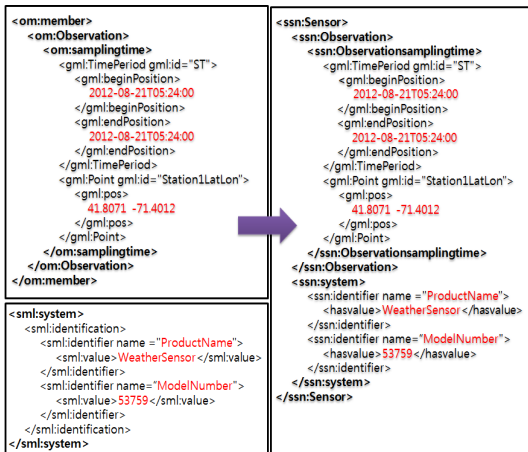


Figure 9. Converted Result of SensorML and O&M Document

Module, Query Analysis Module, Temporal Operation Module, Spatio-Temporal Operation Module로 구성된다.

Query Transmit/Receive Module은 Client의 Query I/O Module로부터 질의를 수신받아 Query Processing Manager의 Query Analysis Module로 전달한다. 그리고 Query Analysis Module에서 반환된 질의 응답 결과를 Client의 Display Module로 전달한다.

Query Analysis Module은 Query Transmit/Receive Module로부터 전달받은 질의에 대한 구문 분석을 수행하고, 온톨로지 검색을 위한 추론 기능을 수행한다. 그리고 시간 또는 시공간 연산 질의의 구문 및 유효성을 검사하고, 오류가 있다면 예외를 발생하여 사용자에게 알려주는 기능을 제공한다. Query Analysis Module에서는 검색 및 추론을 수행하기 위해 Jena[7]의 규칙 기반 추론 엔진을 사용해 추론 규칙을 적용한다. Figure 10은 Jena에서 사용하는 추론 규칙 문법을 보여준다.

Figure 10에서 보는 바와 같이 추론 규칙 문법은 전방향 규칙(Forward Rule)과 역방향 규칙(Backward Rule)을 지원한다. 그리고 용어(Term)들은 트리플 형태의 구조를 가지며, 각 노드(Node)들은 uri 참조, 변수, 리터럴, 숫자 등으로 대체될 수 있다. Figure 11은 Jena의 추론 규칙 문법에 따라 작성한 추론 규칙 예를 보여준다.

Figure 11은 기본적으로 트리플 패턴에 기반을 두어 추론 규칙을 작성한 예를 보여준다. 즉, 온도가 0도 이하이고 눈이 내렸을 때, 해당 지역을 빙판길 위험지역이라고 추론하는 예를 보여준다. 수행이 완료된 결과는 Query Processing Manager의 Query Transmit/Receive Module로 전달된다.

```

Rule := bare-rule.
      or [ bare-rule ]
      or [ ruleName : bare-rule ]

bare-rule := term, ... term -> hterm, ... hterm // forward rule
           or bhterm <- term, ... term // backward rule

hterm := term
       or [ bare-rule ]

term := (node, node, node) // triple pattern
       or (node, node, functor) // extended triple pattern
       or builtin(node, ... node) // invoke procedural primitive

bhterm := (node, node, node) // triple pattern

functor := functorName(node, ... node) // structured literal

node := uri-ref // e.g. http://foo.com/eg
       or prefix:localname // e.g. rdf:type
       or <uri-ref> // e.g. <myscheme:myuri>
       or ?varname // variable
       or 'a literal' // a plain string literal
       or 'lex'^^typeURI // a typed literal, xsd:* type names supported
       or number // e.g. 42 or 25.5
    
```

Figure 10. Jena Reasoning Rule

```

[ICYROADRule :
 (?poly http://db.konkuk.ac.kr/observation.owl# hasTemperature lessThan( ?value 0 ) ),
 (?poly http://db.konkuk.ac.kr/observation.owl# offfeature "snow"),
 -> (?poly http://db.konkuk.ac.kr/observation.owl# statement "icyroad" ) ]
    
```

Figure 11. Example of Reasoning Rule

Temporal Operation Module은 시간 데이터 타입과 시간 연산자를 처리한다. 시간 데이터 타입은 시간의 특성에 맞게 특정 시간(Time)을 의미하는 인스턴트(Instant)와 시작시간(stTime), 종료시간(edTime)을 갖는 피리어드(Period)로 구분된다. 시간 연산자는 타임스탬프(Timestamp) 연산자와 인터벌(Interval) 연산자로 구분되고, 각각은 시간 관계 연산자와 시간 분석 연산자로 구분된다. Table 1은 Temporal Operation Module에서 지원하는 시간 데이터 타입을 보여준다.

Table 1에서와 같이 시간 데이터 타입은 하나의 시간을 가지는 T_Instant, 시작시간과 종료시간을 가지는 T_Period, 하나 이상의 T_Instant를 가지는 T_MultiInstant, 하나 이상의 T_Period를 가지는 T_MultiPeriod, 그리고 위의 4가지 시간 값들을 가지는 T_Collection으로 구분된다. 이러한 시간 데이터 타입은 Temporal 타입으로 정의된다. Table 2는 시간 연산자를 보여준다.

Table 2에서 보는 바와 같이 시간 관계 연산자는 특정 시간 Time 혹은 시간 구간 stTime과 edTime 사이를 기준으로 시간 연산을 수행한다. 시간 관계 연산자에 대한 결과 값으로는 True 혹은 False를 반환하고, 시간 분석 연산자에 대한 결과 값으로는 Temporal 시간 객체를 반환한다. Figure 12는 시간 관계 연산자(T_After)를 활용한 질의 예이다.

Figure 12는 2012년 11월 18일 이후 센서 데이터에

Table 1. Temporal Data Type

Temporal Data Type	Description
T_Instant(Time)	Represent the instant time object
T_Period(stTime, edTime)	Represent the period time object
T_MultiInstant(T_Instant A, T_Instant B, ...)	Represent the multiple instant time objects
T_MultiPeriod(T_Period A, T_Period B, ...)	Represent the multiple period time objects
T_Collection(Temporal A, Temporal B, ...)	Represent the collection of time objects

Table 2. Temporal Operators

Timestamp Temporal Relationship Operators	Description
T_Before(Temporal A, Temporal B)	Whether the A is earlier than the B is returned
T_After(Temporal A, Temporal B)	Whether the A is later than the B is returned
T_Equals(Temporal A, Temporal B)	Whether the A is equal to the B is returned
Timestamp Temporal Analysis Operators	Description
T_Union(Temporal A, Temporal B)	The union of the A and B is returned
T_Difference(Temporal A, Temporal B)	The difference between the A and B is returned
T_Intersection(Temporal A, Temporal B)	The intersection of the A and B is returned
Interval Temporal Relationship Operators	Description
T_Equals(Temporal A, Temporal B)	Whether the A is equal to the B is returned
T_Disjoint(Temporal A, Temporal B)	Whether the A is disjoint from the B is returned
T_Touches(Temporal A, Temporal B)	Whether the A touches the B is returned
T_Overlaps(Temporal A, Temporal B)	Whether the A overlaps the B is returned
T_Contains(Temporal A, Temporal B)	Whether the A contains the B is returned
Interval Temporal Analysis Operators	Description
T_Union(Temporal A, Temporal B)	The union of the A and B is returned
T_Difference(Temporal A, Temporal B)	The difference between the A and B is returned
T_Intersection(Temporal A, Temporal B)	The intersection of the A and B is returned

```

PREFIX ssn: <http://purl.oclc.org/NET/ssn/ssn#>
PREFIX dtp: <http://dtp-126.snacs.abdn.ac.uk#>
PREFIX dbp: <http://dbpedia.org/resource/>
PREFIX spatial: <java.kr.ac.konkuk.db.func.>

SELECT ?sensor ?value ?time
WHERE{
  ?x ssn:sensingdevice ?sensor
  ?sensor ssn:observedProperty ?value
  ?sensor ssn:observationSamplingTime ?time
  ?time spatial:T_After(20121118)
}

```

Figure 12. Example of Temporal Relationship Operation Query

Table 3. Spatio-Temporal Data Type

Spatio-Temporal Data Type	Description
ST_Point(Temporal A, Geometry A)	Represent the spatio-temporal point object
ST_Linestring(Temporal A, Geometry A)	Represent the spatio-temporal line object
ST_Polygon(Temporal A, Geometry A)	Represent the spatio-temporal polygon object
ST_MultiPoint((Temporal A, Geometry A), (Temporal B, Geometry B), ...)	Represent the multiple spatio-temporal point objects
ST_MultiLineString((Temporal A, Geometry A), (Temporal B, Geometry B), ...)	Represent the multiple spatio-temporal line objects
ST_MultiPolygon((Temporal A, Geometry A), (Temporal B, Geometry B), ...)	Represent the multiple spatio-temporal polygon objects
ST_Collection(ST_Geometry A, ST_Geometry B, ...)	Represent the collection of spatio-temporal objects

대한 위치와 온도값을 반환하라는 시간 연산 질의를 보여준다.

Spatio-Temporal Operation Module은 OGC에서 제안한 “OpenGIS Simple Features Specification for SQL”[11]의 공간 연산자를 확장하여 시공간 연산자를 제공한다. 시공간 연산자는 타임스탬프 연산자와 인터벌 연산자로 구분되며, 각각은 시공간 관계 연산자와 시공간 분석 연산자로 구분된다. Table 3은 Spatio-Temporal Operation Module에서 지원하는 시공간 데이터 타입을 보여준다.

Table 3에서와 같이 시공간 데이터 타입은 ST_Point, ST_Linestring, ST_Polygon, ST_MultiPoint, ST_MultiLineString, ST_MultiPolygon, ST_Collection 타입으

Table 4. Spatio-Temporal Operators

Timestamp Spatio-Temporal Relationship Operators	Description
ST_Equals(ST_Geometry A, ST_Geometry B)	Whether the A is equal to the B is returned
ST_Disjoint(ST_Geometry A, ST_Geometry B)	Whether the A is disjoint from the B is returned
ST_Touches(ST_Geometry A, ST_Geometry B)	Whether the A touches the B is returned
ST_Overlaps(ST_Geometry A, ST_Geometry B)	Whether the A overlaps the B is returned
ST_Contains(ST_Geometry A, ST_Geometry B)	Whether the A contains the B is returned
ST_Crosses(ST_Geometry A, ST_Geometry B)	Whether the A crosses the B is returned
Interval Spatio-Temporal Relationship Operators	Description
ST_Equals(ST_Geometry A, ST_Geometry B)	Whether the A is equal to the B is returned
ST_Disjoint(ST_Geometry A, ST_Geometry B)	Whether the A is disjoint from the B is returned
ST_Touches(ST_Geometry A, ST_Geometry B)	Whether the A touches the B is returned
ST_Overlaps(ST_Geometry A, ST_Geometry B)	Whether the A overlaps the B is returned
ST_Contains(ST_Geometry A, ST_Geometry B)	Whether the A contains the B is returned
ST_Crosses(ST_Geometry A, ST_Geometry B)	Whether the A crosses the B is returned
Timestamp Spatio-Temporal Analysis Operators	Description
ST_Union(ST_Geometry A, ST_Geometry B)	The union of the A and B is returned
ST_Difference(ST_Geometry A, ST_Geometry B)	The difference between the A and B is returned
ST_Intersection(ST_Geometry A, ST_Geometry B)	The intersection of the A and B is returned
ST_Distance(ST_Geometry A, ST_Geometry B)	The distance between the A and B is returned
Interval Spatio-Temporal Analysis Operators	Description
ST_Union(ST_Geometry A, ST_Geometry B)	The union of the A and B is returned
ST_Difference(ST_Geometry A, ST_Geometry B)	The difference between the A and B is returned
ST_Intersection(ST_Geometry A, ST_Geometry B)	The intersection of the A and B is returned
ST_Distance(ST_Geometry A, ST_Geometry B)	The distance between the A and B is returned

```

PREFIX ssn: <http://purl.oclc.org/NET/ssnx/ssn#>
PREFIX dtp: <http://dtp.126.snrc.abdn.ac.uk#>
PREFIX dbp: <http://dbpedia.org/resource/>
PREFIX gml: <http://purl.org/fgif/gml/0.2#>
PREFIX spatial: <java:kr.ac.konkuk.db.func.>

SELECT ?sensor ?time
WHERE{
  ?x ssn:sensingdevice ?sensor
  ?sensor ssn:observedProperty ?value
  ?sensor ssn:observationSamplingTime ?time
  ?geo gml:coordinates ?loc.
  ?spatio spatial:makespatio(?time ?loc)
  ?spatio spatial:ST_Contains(ST_Union(20121118 20121120
    POLYGON(37.576024962604066 126.95388793945312,
    37.57738544363852 127.06718444824219, 37.47092082037699
    127.04521179199219, 37.576024962604066 126.95388793945312),
    20121119 20121120) POLYGON(38.5560149628606023
    128.75376773842543, 38.57734544363872 128.06718444824219,
    38.47292385067791 128.02551674129812, 38.5560149628606023
    128.75376773842543))
}
    
```

Figure 13. Example of Spatial-Temporal Relationship & Analysis Operation Query

로 이루어지며, 이러한 시공간 데이터 타입은 ST_Geometry 타입으로 정의된다. Table 4는 시공간 연산자를 보여준다.

Table 4에서 보는 바와 같이 시공간 관계 연산자는 조건 값으로 ST_Geometry를 입력받고, 특정 시간 Time 혹은 시간 구간 stTime과 edTime 사이를 기준으로 시공간 연산을 수행한다. 시공간 관계 연산자는 결과값으로 True 혹은 False를 반환하며, 시공간 분석 연산자는 결과 값으로 ST_Geometry 시공간 객체를 반환한다. Figure 13은 시공간 관계 연산자(ST_Contains)와 시공간 분석 연산자(ST_Union)를 활용한 질의 예이다.

Figure 13은 2012년 11월 18일에서 20일 사이에 해당 Polygon들의 합집합 안에 포함되는 센서 데이터의 위치, 온도값, 시간을 반환하라는 시공간 연산 질의를 보여준다.

4. 시스템 구현

4.1 구현 환경

SSNO 기반 시공간 시맨틱 센서 웹을 구현하기 위하여 구축한 실험 환경은 다음과 같다. 운영체제로는 Windows7 Enterprise K를 사용하였고, 프로그래밍 언어로는 Java 1.7.0_05를 사용하였다. 그리고 웹 서버를 구축하기 위하여 Apache 2.2.11과 Tomcat 7.0을 사용하였으며, 온톨로지에 대한 질의와 추론을 하기 위하여 HP 연구소에서 제시한 JENA 2.6.2[7]와 ARQ 2.8.3[1]을 사용하였다. 또한, 시공간 데이터를 위해 Vivid Solutions에서 제시한 JTS 1.8.0을 사용하였다.

4.2 가상 시나리오

본 논문에서 개발한 SSNO 기반 시공간 시맨틱 센서 웹에 대한 효용성을 검증하기 위한 가상 시나리오는 다음과 같다. 기상청에서 제공하는 날씨 정보 바탕으로 겨울철 기후에 따른 빙판길 예상 지역 선정을 위해 추론 규칙을 적용하였다. 특히, 추론 검색을 통해 선정한 예상 지역 중 사고 발생 빈번 일자 및 지역을 위험 지역으로 선정하기 위해 시공간 연산자를 사용하였다.

Figure 14, Figure 15는 각각 빙판길 예상 지역 선정을 위한 추론 규칙과 이를 적용한 질의를 보여준다.

Figure 14의 추론 규칙에서는 온도가 0도 이하이고 눈이 내렸다면, 해당 지역은 빙판길 예상지역이라는 규칙을 정의하고 있다.

Figure 14의 추론 규칙을 적용한 빙판길 예상 지역을 선정하기 위해 Figure 15의 질의는 (?s <http://db.konkuk.ac.kr/observation.owl#Icyroad> ?poly)를 통해 추론 규

```

|ICYROADRule :
{?poly http://db.konkuk.ac.kr/observation.owl#hasTemperature lessThan(?value 0),
(?poly http://db.konkuk.ac.kr/observation.owl#ofFeature "snow"),
-> (?poly http://db.konkuk.ac.kr/observation.owl#statement "icyroad" )}
    
```

Figure 14. Reasoning Rule about Projected Icy Areas

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX ssn: <http://db.konkuk.ac.kr/observation.owl#>
PREFIX spatial: <java:kr.ac.konkuk.db.func.>
SELECT ?name ?time ?value ?spatio ?weather
WHERE {
?s <http://db.konkuk.ac.kr/observation.owl#Icyroad> poly;
ssn:name ?name;
ssn:hasValue ?value;
ssn:observationSamplingTime ?time;
ssn:ofFeature ?weather;
ssn:spatioTemp ?spatio.
}
    
```

Figure 15. Query about Projected Icy Areas

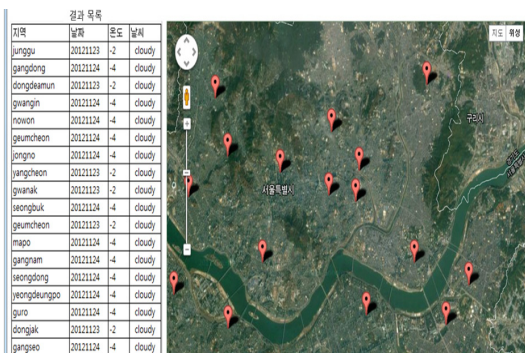


Figure 16. Query Result of Figure 15

칙을 사용하고 있다. Figure 16은 Figure 15의 추론 규칙을 사용한 질의의 수행 결과 화면을 보여준다.

Figure 16에서 보면, 좌측 검색 결과 목록에는 추론 이 적용된 빙판길 예상 지역의 지역명, 날짜, 온도, 날씨 등의 정보를 제공하고, 우측 Map에는 검색 지역의 위치를 지도 형태로 표시한다.

Figure 17은 시공간 관계 연산자 ST_Contains와 시공간 분석 연산자 ST_Union을 사용해 Figure 16에서 검색된 빙판길 예상 지역 중 사고의 발생 빈도가 높은 특정일(2012년 11월 24일)과 사고 다발 발생 지역(즉, 질의에서 사용된 3개의 Polygon)을 추가 검색 조건으로 지정한 질의이고, Figure 18은 그에 대한 수행 결과 화면을 보여준다.

Figure 18에서와 같이 추론 규칙과 시공간 연산을 적용함으로써 추론된 빙판길 예상 지역 중 2012년 11월 24일에 사고 다발 발생 지역 내에 포함된 질의 결과 데이터들이 출력되는 것을 확인할 수 있다.

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX ssn: <http://db.konkuk.ac.kr/observation.owl#>
PREFIX spatial: <java:kr.ac.konkuk.db.func.>
SELECT ?name ?time ?value ?spatio ?weather
WHERE {
?s <http://db.konkuk.ac.kr/observation.owl#Icyroad> poly;
ssn:name ?name;
ssn:hasValue ?value;
ssn:observationSamplingTime ?time;
ssn:ofFeature ?weather;
ssn:spatioTemp ?spatio.
?spatio spatial:ST_Contains(ST_Union(20121124
POLYGON(37.576024962604066 126.95388793945312,
37.57738544363852 127.06718444824219, 37.47092082037699
127.04521179199219, 37.576024962604066 126.95388793945312),
20121124 POLYGON(38.5560149628606023
128.75376773842543, 38.5773454383872 128.06718444824219,
38.472923850677911 128.02551674129812, 38.5560149628606023
128.75376773842543) 20121124 POLYGON(39.5663292102016
129.95788773643312, 39.87832524263832 129.26728242814516,
39.87892585037391 129.05521373129222, 39.5663292102016
129.95788773643312))
ssn:spatioTemp ?spatio.
}
    
```

Figure 17. Spatial-Temporal Query about Projected Icy Areas

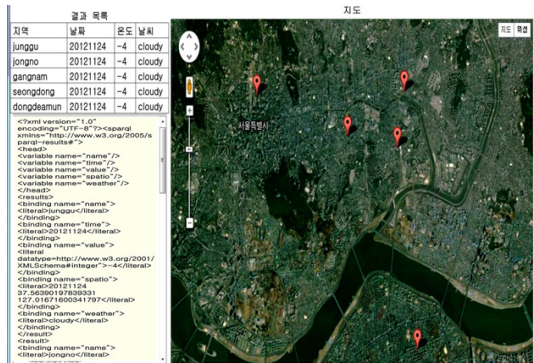


Figure 18. Query Result of Figure 17

5. 결 론

최근 새로운 웹 기술인 시맨틱 센서 웹이 연구되면서 W3C는 시맨틱 센서 웹 환경에서의 온톨로지 표현을 위해 SSNO를 제안하였다. 그러나 SSNO는 일반적인 비시공간 온톨로지에 대한 질의 처리를 위해 제안되었기 때문에 시공간 센서 데이터에 대한 처리를 지원하지 못하는 문제점을 가진다.

따라서 본 논문에서는 시공간 센서 데이터의 효율적인 처리를 지원하는 SSNO 기반 시공간 시맨틱 센서 웹을 개발하였다. SSNO 기반 시공간 시맨틱 센서 웹은 SSNO 사용을 위해 SWE의 교환 언어인 SensorML과 O&M을 분석하는 기능, 분석된 데이터를 SSNO 온톨로지로 변환하는 기능, 변환된 SSNO 온톨로지를 저장하는 기능을 제공한다. 또한, 사용자로부터 전달된 질의에 대해 분석 및 추론을 수행하고, 시공간 데이터를 효율적으로 처리하기 위한 시공간 연산 기능도 제공한다.

마지막으로, 본 논문에서 개발한 SSNO 기반 시공간 시맨틱 센서 웹을 기후 변화에 따른 빙판길 예상 시나리오에 적용해 봄으로써 본 시스템의 효용성을 검증하였다. 그리고 향후 연구로서 SSNO 온톨로지 재사용을 위한 자료 구조, 시공간 센서 데이터의 질의 처리를 위한 인덱스 기법, 다양한 시공간 추론 알고리즘과 같은 연구 등이 필요할 것으로 사료된다.

References

[1] ARQ - A SPARQL Processor for Jena, <http://openjena.org/ARQ/>.

[2] Compton, M; Barnaghi, P; Bermudez, L. 2012, The SSN Ontology of the W3C Semantic Sensor Network Incubator Group, *Journal of Web Semantics*, 17:25-32.

[3] Cory, A; Josh, K. 2009, SemSOS: Semantic Sensor Observation Service, Paper presented at the International Symposium on Collaborative Technologies and Systems, May 18-22.

[4] Daniel, O; Rob, B. 2010, Implementing the Draft

W3C Semantic Sensor Network Ontology, Paper presented at the IEEE International Conference on Pervasive Computing and Communications Workshops, March 29-April 2.

[5] Devaraju, A; Neuhaus, H; Janowicz, K; Compton, M. 2010, Combining Process and Sensor Ontologies to Support Geo-Sensor Data Retrieval, Paper presented at the 6th International Conference on Geographic Information Science, September 14-17.

[6] Janowicz, K. 2012, Observation-Driven Geo-Ontology Engineering, *Transactions in GIS*, 16:351-374.

[7] Jeremy, J. C; Ian, D; Dollin, C. 2003, The Jena Semantic Web Platform: Architecture and Design, HP Laboratories Technical Report HPL.

[8] Kim, J. J; Shin, I. S; Han, K. J. 2010, Development of a Geo Semantic Web System, *Journal of Korea Spatial Information Society*, 18(5):83-92.

[9] Open Geospatial Consortium. 2007, OGC Observation and Measurements(O&M), Version 1.0.

[10] Open Geospatial Consortium. 2007, OGC Sensor Model language(SensorML), Version 1.0.

[11] Open Geospatial Consortium. 2010, OpenGIS Implementation Specification for Geographic Information-Simple Feature Access-Part 1:Common architecture, Version 1.2.1.

[12] Open Geospatial Consortium. 2011, Sensor Web Enablement DWG, <http://www.opengeospatial.org/projects/groups/sensorwebdwg/>.

[13] Open Geospatial Consortium. 2011, SWE Service Model Implementation Standard, Version 2.0.

[14] World Wide Web Consortium. 2011, Semantic Sensor Network XG Final Report, <http://www.w3.org/2005/Incubator/ssn/XGR-ssn/>.

논문접수 : 2014.6.9
수정일 : 2014.10.20
심사완료 : 2014.10.21