

GeoSensor의 공간연산을 확장한 Spatial-Sensor Observation Service

이혁*, 이연*, 정원일**, 배해영*

Spatial-Sensor Observation Service for Spatial Operation of GeoSensor

Hyuk Lee*, Yeon Lee*, weon-IL Chung**, hae-Young Bae*

요약

과학과 기술의 발전은 우리의 삶에서 다양한 변화를 만들고 있다. 특히 센서는 다양한 분야에서 실시간 모니터링과 효율적인 주변 환경 분석을 위해 사용되고 있다. 그러나 기존의 센서 네트워크는 조직별 자체 정의된 프로토콜과 아키텍처를 사용함으로써 네트워크 간의 통신에 추가적인 비용을 발생시켰다. 표준화된 센서 네트워크의 부재로 인한 단점을 해결하고자 OGC에서는 표준화된 센서 및 센서 시스템에 대한 접근, 조작 등을 위해 2007년 SWE v1.0과 2011년 SWE v2.0을 발표했다. SWE는 다양한 하위 컴포넌트들로 구성되어 있는데 그 중 센서 데이터 획득을 위한 SOS에 대한 다양한 시도들이 이루어지고 있다. 그러나 기존의 SOS들은 GeoSensor의 공간연산을 고려하지 않기 때문에 매번 클라이언트에서 공간연산이 구현 및 실행됨으로써 적은 리소스를 갖는 클라이언트에 많은 부하를 초래하고 SOS의 용이성이 떨어지는 문제점이 있다. 본 논문에서는 고정 및 이동형 GeoSensor을 위한 spatialFilter와 spatialFinder 공간 연산을 확장한 S-SOS를 제안한다. 제안된 S-SOS를 사용하여 기존의 기 구축된 SOS를 수정하지 않고 필요에 의해 최소한의 추가로 공간 연산을 지원할 수 있는 아키텍처를 지원하며 이에 필요한 공간 연산의 질의 표현을 설명한다. 따라서 S-SOS는 다양한 위치기반 서비스에 효과적으로 사용될 수 있음을 보인다.

▶ Keyword : SWE, SOS, Spatial, GeoSensor

Abstract

Advances in science and technology have made a lot of changes in our life. Especially, sensors have used in various ways to monitor in real time and analyze the world effectively. Traditional

• 제1저자 : 이혁 • 교신저자 : 이연, 정원일 • 책임저자 : 배해영

• 투고일 : 2011. 07. 07, 심사일 : 2011. 08. 12, 게재확정일 : 2011. 08. 17.

* 인하대학교 컴퓨터정보공학과(Dept. of Computer & Information Engineering, Inha University)

** 호서대학교 정보보호학과(Dept. of Information Security Engineering, Hoseo University)

sensor networks, however, have used their own protocols and architecture so it had to be paid a lot of additional cost. In the past 8 years, OGC and ISO have been formulating standards and protocols for the geospatial Sensor Web. Although the OGC SWE initiatives have deployed some components, attempts have been made to access sensor data. All spatial operations had to calculate on the client side because traditional SOS architecture did not consider spatial operation for GeoSensor. As a result, clients have to implement and run spatial operations, and it caused a lot of overload on them and decreased approachableness. In this paper we propose S-SOS for in-situ and moving GeoSensor that extends 52 North SOS and provides spatialFilter and spatialFinder operations. The proposed S-SOS provides an architecture that does not need to edit already deployed SOSs and can add spatial operations as occasion. Additionally we explain how to express the spatial queries and to be used effectively for various location based services.

▶ Keyword : SWE, SOS, Spatiol, GeoSensor

I. 서 론

센서 기술의 급속한 발전은 CPU, 메모리, 무선 송수신기 등 현재의 센서 노드들을 이전보다 더욱 정교하게 만들었다. 센서 네트워크는 오랜 기간 연구되고 있는 컴퓨팅 시스템들이다. 이들은 센싱 가능한 노드들의 집합으로 구성되어 지는데 예를 들면 빛, 온도, 압력, 위치 등 다양한 종류의 센서들이 이에 포함된다. 무선 센서 네트워크는 전 세계적으로 연구단체 및 기업들의 많은 관심을 끌어들였다. 정확하고 안정적으로 정보를 수집할 수 있는 센서 네트워크의 능력은 쓰나미, 지진, 화재와 같은 재해의 실시간 탐지 및 조기 경보 시스템 등을 가능하게 한다.

1997년 NASA Jet Propulsion Laboratory (JPL)의 Kelvin Delin에 의해 처음 "센서 웹 (Sensor Web)"이라는 용어가 사용되었다[1]. 센서 웹이란 산발적으로 분산되어 있는 이기종 센서들이 마치 하나의 시스템처럼 작동하고 조직화될 수 있는 아키텍처를 의미한다. 그러므로 센서 웹은 수집된 데이터로부터 정보를 추출하고 그 정보를 주변 환경에 지능적으로 반응하고 적용할 수 있도록 한다.

기존의 센서 네트워크는 데이터 전송 시 자체 정의의 프로토콜을 사용하여 왔다. 이것은 TCP/IP 프로토콜이 등장하기 전과 같은 문제점을 갖고 있었다. 비표준화는 특정 위치, 센서 응용 프로그램, 그리고 사용 중인 디바이스들과 강하게 결합된 리소스들의 사용과 실행을 위한 응용들 사이에서 저장된 리소스를 재할당하거나 공유할 수 없음을 의미했다. 이러한 문제를 해결하기 위해 2007년 "Open Geospatial Consortium (OGC)"[2]에서는 개방형 플랫폼으로 표준화된 센서의 표현, 데이터 교환을 위한 형식과 인터페이스들을

정확한 Sensor Web Enablement (SWE)를 발표하였다. 이는 모든 센서와 그에 대한 정보를 XML 표준 프로토콜을 통해 처리하므로 센서 객체와 데이터의 발견 및 접근을 단일화 되어있는 인터페이스로 가능하게 한다. SWE는 센서와 응용 프로그램 간의 미들웨어에 속하며 표준화로 인해 기존 구축된 센서 시스템에도 도입이 가능하며 결론적으로는 비용 절감의 효과를 얻을 수 있다. SWE의 사용으로 실시간으로 새로운 센서들의 추가가 가능하고, 표준화된 웹 서비스들을 통해 실시간으로 측정된 데이터의 요청이 가능한 동적인 시스템이 가능하다.

SWE의 구성 요소 중 특히 센서 또는 센서 시스템으로부터 관측된 데이터에 대한 접근을 제공하는 표준 인터페이스인 SOS에 대한 활발한 연구가 진행되고 있다. OGC 웹사이트에 의하면 현재까지 12개의 기관에서 기관별 목적에 따라 SOS 표준이 구현되었다[3]. 그러나 2007년 OGC에서 발표된 SOS v1.0은 지역적 특성을 갖는 고정 및 이동형 GeoSensor를 위한 공간연산을 제공하지 않는다. 이는 크게 세 가지의 문제점을 갖는다. 첫째, 원하는 지역의 센서 데이터에 접근하기 위해서는 카탈로그 서버에 접속해 해당 서버가 관리하는 모든 센서들의 정보를 획득 후 원하는 각 센서들에게 다시 정보를 요청함으로써 n번의 DescribeSensor와 GetObservation의 호출이 발생하기 때문에 SOS 서버와 클라이언트 사이의 대량의 데이터 통신이 불가피하다. 둘째, 필요시마다 공간연산을 클라이언트 측에서 구현하고 수행함으로써 서버에 비해 적은 리소스를 갖는 클라이언트의 부하가 늘어난다. 이는 또한 요즘의 컴퓨팅 트렌드에 적합하지 않다. 마지막으로 이전의 두 작업은 SOS의 용이성을 급격히 저하시킨다.

2011년 1월 OGC에서는 SWE v2.0을 발표하며 4개의

x, y 좌표를 갖는 사각형 바운더리인 Spatial Filter를 지원하기 시작했다. 그러나 여전히 클라이언트의 측면에서 SOS에 대한 용이성이 낮고 사각형 바운더리만으로 정확한 영역의 이기종 GeoSensor에 대한 공간 연산을 다루기에 적합하지 않다. 본 논문에서는 고정 및 이동형 GeoSensor를 다루는 기 구축된 SOS 서버의 수정을 최소화하며 더욱 정밀한 공간 연산을 지원하고 공간 연산에 대한 클라이언트의 용이성을 고려해 기존 SOS를 확장한 S-SOS를 제안한다.

본 논문의 구성은 다음과 같다. 제 2장에서는 객체의 공간 질의를 표현하기 위한 GML과 XML질의어, 그리고 SWE의 발전 과정과 이미 존재하는 SOS들의 종류와 특징에 대해 검토한다. 제 3장에서는 고정 및 이동 객체를 위한 공간 연산을 확장한 S-SOS의 정의와 시스템 구조와 제안된 S-SOS를 기반으로 공간 연산을 위한 질의 표현에 대해 설명한다. 제 4장에서는 S-SOS에 대한 성능평가와 제 5장에서는 결론 및 향후 연구에 대해 기술한다.

II. 관련 연구

2.1 GML과 XML 질의

2006년 OGC에서 공간과 비공간 정보를 포함한 지리정보를 표현하기 위한 XML 기반의 인코딩 표준인 Geography Markup Language (GML)[4]을 발표하였다. OGC는 개방형 지리정보 시스템 환경을 위해 지리정보 데이터와 응용프로그램간의 표준 인터페이스를 위해 2000년 GML v1.0을 제안했으며 현재는 2007년 3월 제안된 GML v3.2.1이 최신 표준 규격이다. GML v2.x에서는 XML DTD(Document Type Definition)을 기반으로 프로파일을 제공하는 GML v1.x를 포함하고, eXtensible Markup Language (XML) 스키마를 기반으로 복잡한 피쳐(feature)와 이들 간의 관계를 인코딩 할 수 있는 스키마를 제공한다. GML v3.0은 기존의 GML과의 호환성을 제공하며, 모듈화, 복합 지오메트리(geometry), 시공간 참조 시스템, 위상, 메타데이터, 그리드 데이터, 측정 단위 등이 추가 되었고 이후 버전에서도 GML은 늘어나는 요구에 따라 지속적으로 확장되고 있다. 특히 GML v3.0이상의 버전에서는 동적 객체 스키마를 통해 위치, 시간, 그리고 위상의 개념 등을 포함하여 위치기반서비스에서 효과적으로 관리되어야 하는 이동 객체의 정보를 표현하는데 활용 가능하다.

GML v1.0이 발표되고 GML을 확장하여 공간 질의어를

제안하기 위한 시도들이 있었다. 첫 번째로 URL 형식을 사용하고 Web Map Server (WMS)와 GML을 기반으로 상호운용적인 웹 매핑 시스템을 제안한 Shekhar[5]이 있다. 두 번째로 SQL 구문을 확장하여 GML에 대한 공간 질의어를 명세했으나 GML 문서에 대한 질의 수행을 위해 제안 질의어를 XML 표준 질의어에 맞게 변형해야하는 문제가 있었던 Corcoles[6]가 있었다. 마지막으로 기존의 XML 질의어를 확장하여 질의어를 제안했지만 제안 질의어에 대한 데이터 모델 및 질의 대수가 정의되지 않았으며 이동 객체에 대한 질의를 언급하지 않은 Vatsavai[7]가 존재한다.

XML질의어는 데이터베이스와 같이 XML 문서나 XML 문서의 컬렉션에 접근이 가능하다. W3C XML Query 워킹 그룹은 웹에서 XML 문서로부터 데이터를 추출하기 위한 유연한 질의 기능을 제공하기 위한 목적으로 XQuery를 제공한다. XQuery는 XML-QL[8], X-Path[9], XQL[10] 등의 XML 질의어 개념과 SQL의 개념을 추가하여 확장한 질의 언어로, 다양한 형태의 XML 데이터 소스에 포괄적으로 사용될 수 있도록 설계되어 XML 질의 언어의 표준으로 자리 잡고 있다.

2.2 서비스로써 SWE의 진화

2007년 OGC에서는 기존 센서 네트워크들의 단점이었던 비표준화로 인한 문제를 해결하기 위해 SWE v1.0을 발표했다. SWE는 크게 SensorML[11], Observation&Measurement Schema (O&M)[12-14], TransducerML (TML)[15], Sensor Observation Service (SOS)[16], Sensor Alert Service (SAS)[17], Sensor Planning Service (SPS)[18] 등으로 구성되며 이들 각각을 구현함으로써 센서 웹을 실현할 수 있도록 한다. SWE는 또한 XML을 기반으로 표준화된 센서의 표현, 데이터 교환을 위한 형식과 인터페이스들을 정의한다. 이를 위해 SWE v1.0은 다음 7개의 OpenGIS 규격을 포함한다.

- O&M은 센서가 관측 또는 측정된 센싱 정보를 인코딩하는 XML 기반의 표준 모델로서 특정 센서 또는 특정 단체에 종속되는 데이터 포맷으로만 해석되는 문제를 제거한다.
- SensorML은 모든 센서들을 추상화하기 위한 XML 기반의 표준 모델로서, 센서에 대한 사전 지식 없이도 센서를 발견하고 센싱 정보를 해석할 수 있도록 센서에 대한 메타 정보를 제공한다.
- TML은 센서와 구동장비를 포함한 트랜듀서에 관한 정보를 모델링하는 함수와 메시지 포맷으로서 트랜듀서의

데이터를 획득하고 저장 및 전달하는 공통 포맷을 제공한다.

- SOS는 다양한 센서 또는 센서 시스템으로부터 관측된 데이터에 대한 접근을 제공하는 표준 인터페이스로서, 센서를 사용하는 사용자들 사이에 발생할 수 있는 용어 및 관점 차이를 제거한다.
- SPS는 클라이언트가 웹을 통해 연결되어 있는 센서를 제어할 수 있도록 하는 표준 인터페이스이다.
- SAS는 센서에서 측정된 데이터가 특정 값을 넘는 경우나 특정 상황이 발생한 경우를 이벤트로 정의하고 해당 이벤트에 대한 정보 메시지를 클라이언트에게 전달한다.
- WNS는 HTTP와 같은 동기적 및 비동기적 알람을 통해 SAS가 클라이언트에게 이메일, SMS, HTTP, 팩스 등을 전달할 수 있도록 하는 표준 인터페이스이다.

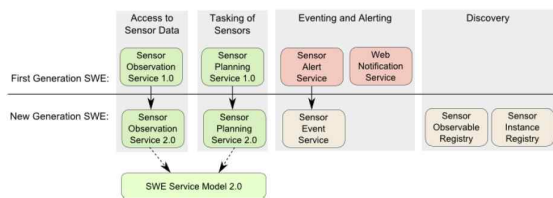


그림 1. SWE v1.0과 v2.0의 설계도
Fig. 1. Architecture of SWE v1.0 and v2.0

하지만 그림1을 통해 2011년 1월에 발표된 SWE v2.0에서는 인터페이스의 설계가 변경된 것을 발견할 수 있다. SOS v2.0과 SPS v2.0을 하나의 서비스로 묶어 SWE Service Model 2.0이란 서비스로 정의하였다. 또한, WNS가 사라지고 SAS가 Sensor Event Service로 변경되었고 센서의 시멘틱적 발견을 위한 Sensor Observable Registry[19]와 메타 데이터 관리를 위한 Sensor Instance Registry 규격들이 추가된 것을 볼 수 있다. 이번 SWE v2.0에서 SWE는 기존 다양한 인터페이스들의 조합을 벗어나 하나의 서비스로 탈바꿈하고 있다.

2.3 기존 SOS들의 특징 분석

OGC 웹사이트의 자료에 따르면 현재까지 52 North를 비롯해 SOS를 구현한 기관 및 단체는 공식적으로 12개에 이르며 이들은 각자의 목적에 따라 SOS를 구현하였다. 그 중에 대표적인 SOS 네 가지는 표 1과 같다.

- 52 North SOS는 레퍼런스 SOS로써 OGC의 SOS 규격을 가장 많이 구현하였으며 산림 화재부터 수질 모니터링까지 다양한 분야에 적용되어 있다. 52 North SOS를 적용한 가장 좋은 사례는 남아프리카에서 수행

되었던 Council for Scientific and Industrial Research (CSRI)의 Advanced Fire Information System (AFIS)가 있다.

- GeoBliki SOS는 다양한 웹 기술의 결합으로 탄생했다. 초기부터 GeoBliki 팀은 비용의 절감과 많은 기여자들의 참여를 위해 오픈소스 방식을 채택했고 빠른 프로토타이핑 속도, Model-View-Controller 접근, MySQL 지원, 내장 테스트 프레임워크를 위해 Ruby-on-Rails를 사용했다. GeoBliki의 응용으로는 Earth Observing-1 (EO-1) Sensor Web 프로젝트가 있다.
- VisAnalysis Systems Technologies (VAST)는 Alabama 대학교의 Earth System Science Center에서 과학적 분석을 위해 개발되었다. VAST는 다음 다섯 가지 타입의 SOS들을 포함한다: Satellite Orbital Elements SOS, Satellite Nadir Track SOS, Satellite Sensor Footprints SOS, Airdas unmanned aerial vehicle SOS, Weather station SOS.
- Degree SOS는 OGC SOS v0.9를 기반으로 개발되었으며 PostGIS, Oracle Spatial를 비롯한 다양한 DBMS들을 JDBC를 통해 지원하는 것이 특징이다. 따라서 이미 구축된 DBMS를 SOS에 바로 연결할 수 있다. iGeoPortal은 웹기반의 포털 프레임워크로 표준 웹브라우저를 통해 위치 데이터의 시각화를 제공한다 [20].

이외에도 SOS의 스트림 데이터 처리 문제를 해결한 SANY IP (<http://www.sany-ip.ed>)의 SOS-X[21]도 존재한다. 하지만 위에서 살펴본 SOS들은 특정 목적에 초점을 맞추고 있으며 고정 및 이동 GeoSensor에 대한 공간 연산을 지원할 수 없다는 공통점이 있다. 따라서 본 논문에서는 급격하게 증가하는 위치기반 서비스의 요구에 따라 함께 증가하는 GeoSensor의 공간연산이 가능한 S-SOS를 제안한다.

III. 공간연산을 확장한 S-SOS

3.1 정의

지역적 특성을 갖는 고정 및 이동형 GeoSensor는 위치에 기반한 센싱 데이터를 생성한다. 이 데이터들은 지역별 통계 혹은 실시간 모니터링 등의 다양한 응용에서 사용될 수 있다.

표 1. SOS 구현 비교표

Table 1. SOS implementation comparison

Feature	Type			
	520 North	GeoBliki	VAST	Degree
Functions				
GetCapabilities	O	O	O	O
GetObservation	O	O	O	O
DescribeSensor	O	O	O	O
GetFeatureOfInterest	O			
InsertObservation				
RegisterSensor				
GetResult	O			
GetFeatureOfInterestTime	O			
DescribeFeatureOfInterest	O			
DescribeObservationType	O			
DescribeResultModel				
DataBase	PostgreSQL	MySQL	NONE	MySQL
Program language	Java	Perl	Java	Java
Service	Servlet	PHP	Servlet	Servlet
User Interface	HTML	Ajax	HTML	HTML
DCP request	GET/POST	POST	GET/POST	GET
Sensor type	In-situ	Remote	Remote	In-site
Client	Ox-Framework		STT	iGeoPortal
Typical application	AFIS	EO-1	Weather station	

O : Supported

제한된 공간 연산을 통해 원하는 센서의 데이터 획득을 위해 센서의 개수만큼 DescribeSensor와 GetObservation 질의를 했었지만 S-SOS는 질의의 횟수가 1/n으로 감소된다. 이는 단 한 번의 질의로 원하는 공간의 GeoSensor 정보를 획득함으로써 빠른 응답성과 네트워크 비용의 감소를 가져올 수 있다. 또한, 기존 SOS의 용이성을 향상시키고 클라이언트의 공간 연산에 대한 부하를 제거한 S-SOS를 제안한다.

동작하거나 InsertObservation 메시지에 state, city, national code, zipcode 등의 정보를 미리 입력해 해당 필드들을 통해 검색하는 두 가지의 방법이 존재한다.

3.2 SOS 아키텍처

다양한 센서나 센서 시스템으로부터 관측된 데이터에 대한 표준 접근을 제공하는 SOS를 최소한의 수정을 통해 고정 및 이동형 GeoSensor를 위한 S-SOS로 확장할 수 있다. 이것은 기존 구축된 SOS를 모두 수정하거나 재개발할 필요 없으며 필요에 따라 추가적으로 도입하면 되기 때문에 매우 유연하며 소프트웨어 공학적 측면에서 매우 효율적이라고 할 수 있다.

그림 2는 S-SOS의 전체 아키텍처를 나타낸다. 추가된 두 가지의 공간 연산들은 SOS의 필수 항목이 아닌 선택적 항목이다. spatialFilter 연산은 고정 및 이동형 GeoSensor들을 위한 연산으로 SOS를 잘 모르는 클라이언트의 용이성 향상과 고급 클라이언트들을 위한 더욱 정밀한 공간 연산 기능을 제공한다. 해당 연산은 기본적으로 도, 시, 우편번호, 사용자 지정 지점들의 네 가지 선택 사항들을 통해 해당 지역의 위치를 접근하기 때문에 기존의 구축된 지도 데이터를 기반으로

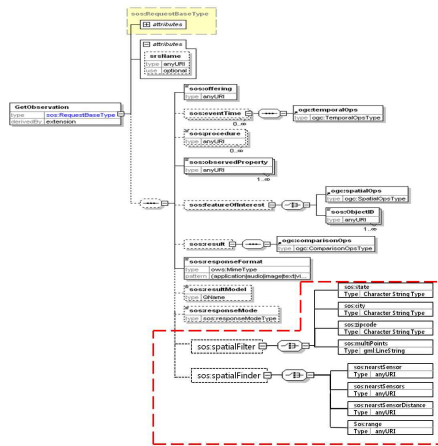


그림 2. S-SOS 아키텍처

Fig. 2. Architecture of S-SOS

spatialFinder 연산은 고정 및 이동형 GeoSensor를 위해 사용할 수 있지만 이동형 GeoSensor에 더욱 적합하다. spatialFinder 연산은 사용자가 정의한 특정 센서를 중심으로 주변의 다른 센서들을 찾는 연산이다. 예를 들면, 가장 가

까운 주변 친구를 찾거나 상점 주변에 있는 고객들에게 세일 광고를 푸시하는 등의 위치기반 서비스에 적합하다.

공간연산을 위해 확장된 spatialFilter와 spatialFinder에 대한 질의 표현은 다음 절에서 다룬다.

3.3 질의 표현

S-SOS에서 새롭게 제공되는 공간 연산은 크게 둘로 나뉜다. 첫 번째 spatialFilter 연산은 2011년 새롭게 제안된 SOS v2.0의 단순히 4개의 x,y 좌표를 통한 사각형 공간 필터의 원하는 센서에 대한 필터링 정확도와 SOS의 용이성을 크게 향상시킬 수 있다. 두 번째 spatialFinder 연산은 이동형 GeoSensor를 위한 연산으로 현재 센서의 위치를 기준으로 주변 센서들의 정보를 획득한다.

3.3.1 spatialFilter 연산

- zipcode : zipcode 연산은 city 보다 더 규모가 작으며 해당 우편번호에 포함되는 지역의 GeoSensor 데이터만을 획득 가능하다. 질의에 대한 표현은 아래의 표2와 같다.

표 2. zipcode 연산 예제
Table 2. Example of zipcode operation

Request message example
<pre><?xml version="1.0" encoding="UTF-8"?> <GetObservation> <spatialFilter> <nationalCode>82</nationalCode> <zipcode>110-230</zipcode> </spatialFilter> </GetObservation></pre>

- state : state 연산은 클라이언트가 원하는 주 혹은 도 단위의 공간 필터 연산을 제공한다. 따라서 클라이언트는 SOS에 대한 지식 없이도 원하는 지역의 국가 코드와 영문 이름만 있다면 연결된 SOS 서버에서 관리하는 해당 지역의 GeoSensor 데이터만을 획득할 수 있다. 질의에 대한 표현은 아래의 표3과 같다.

표 3. state 연산 예제
Table 3. Example of state operation

Request message example
<pre><?xml version="1.0" encoding="UTF-8"?> <GetObservation xmlns="http://www.opengis.net/sos/1.0" xmlns:ows="http://www.opengis.net/ows/1.1" xmlns:gml="http://www.opengis.net/gml"</pre>

```
xmlns:ogc="http://www.opengis.net/ogc"
xmlns:om="http://www.opengis.net/om/1.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://www.opengis.net/sos/1.0
http://schemas.opengis.net/sos/1.0.0/sosGetObservation.xsd"
service="SOS" version="1.0.0"
srsName="urn:ogc:def:crs:EPSG:4326">
<offering>GAUGE_HEIGHT</offering>
<observedProperty>urn:ogc:def:phenomenon:OGC:1.0.30:waterlevel</observedProperty>
<responseFormat>text/xml:subtype="om/1.0.0"&quot;</responseFormat>
<spatialFilter>
  <nationalCode>1</nationalCode>
  <state>Illinois</state>
</spatialFilter>
```

- city : city 연산은 state 보다 규모가 작은 시 단위별 공간 필터 연산을 제공한다. 내용은 state와 같으며 질의에 대한 표현은 아래의 표3과 같다.

표 4. city 연산 예제
Table 4. Example of city operation

Request message example
<pre><?xml version="1.0" encoding="UTF-8"?> <GetObservation> <spatialFilter> <nationalCode>82</nationalCode> <city>seoul</city> </spatialFilter> </GetObservation></pre>

- multiPoints : multiPoints 연산은 클라이언트로부터 입력받은 다수의 위도와 경도 좌표들을 통해 클라이언트가 원하는 특정 지역의 GeoSensor에 대한 데이터를 획득할 수 있다. 예를 들어 규모가 크고 지역의 형태가 불규칙한 공업단지부터 작은 섬과 같은 형태까지 크기나 형태와 관계없이 사용자 정의 지역의 GeoSensor 데이터만 획득함으로써 불필요한 데이터의 크기를 줄인다. 질의에 대한 표현은 아래의 표5와 같다.

표 5. multiPoints 연산 예제
Table 5. Example of multiPoints operation

Request message example
<pre><?xml version="1.0" encoding="UTF-8"?> <GetObservation> <spatialFilter> <multiPoints></pre>

```

<gml:LineString gml:id="b1_s1"
  srsName="urn:ogc:def:crs:EPSG:6.8:4347">
  <gml:pos>37.270662, 126.390981</gml:pos>
  <gml:pos>37.270748, 126.385405</gml:pos>
  <gml:pos>37.265518, 126.390023</gml:pos>
  <gml:pos>37.264878, 126.391559</gml:pos>
  <gml:pos>37.265507, 126.391953</gml:pos>
  <gml:pos>37.264861, 126.393596</gml:pos>
  <gml:pos>37.265981, 126.393844</gml:pos>
  <gml:pos>37.270662, 126.390981</gml:pos>
</gml:LineString>
</multiPoints>
</spatialFilter>
</GetObservation>
    
```

3.3.2 spatialFinder 연산

- nearestSensor : 사용자에게 의해 정의된 GeoSensor로부터 가장 가까운 GeoSensor를 찾아낸다. 질의에 대한 표현은 아래의 표6과 같다.

표 6. nearestSensor 연산 예제
Table 6. Example of nearestSensor operation

Request message example
<pre> <?xml version="1.0" encoding="UTF-8"?> <GetObservation ...> <nearestSensor> <procedure>urn:ogc:object:Sensor:MyOrg:12 34</procedure> </nearestSensor> </GetObservation> </pre>

- nearestSensors : 사용자에게 의해 정의된 GeoSensor로부터 가장 가까운 n개의 GeoSensor 리스트를 획득 가능하다. 질의에 대한 다음 예제는 센서에서 가장 가까운 5개의 센서에 대한 정보를 반환하며 그에 대한 표현은 아래의 표7과 같다.

표 7. nearestSensors 연산 예제
Table 7. Example of nearestSensors operation

Request message example
<pre> <?xml version="1.0" encoding="UTF-8"?> <GetObservation ...> <nearestSensors> <procedure>urn:ogc:object:Sensor:MyOrg:12 34</procedure> <number>5</number> </nearestSensors> </GetObservation> </pre>

- nearestSensorDistance : 사용자에게 의해 정의된 GeoSensor와 가장 가까운 GeoSensor 간의 거리를

획득 가능하다. 질의에 대한 표현은 아래의 표8과 같다.

표 8. nearestSensorDistance 연산 예제
Table 8. Example of nearestSensorDistance operation

Request message example
<pre> <?xml version="1.0" encoding="UTF-8"?> <GetObservation ...> <NearestSensorDistance> <procedure>urn:ogc:object:Sensor:MyOrg:12 34</procedure> </NearestSensorDistance> </GetObservation> </pre>

- range : 사용자에게 의해 정의된 GeoSensor로부터 지정된 범위에 해당하는 GeoSensor들의 리스트를 획득 가능하다. 아래의 예제는 지정된 센서 주변 100m의 센서에 대한 정보를 반환하며 질의에 대한 표현은 아래의 표9와 같다.

표 9. range 연산 예제
Table 9. Example of range operation

Request message example
<pre> <?xml version="1.0" encoding="UTF-8"?> <GetObservation ...> <range> <procedure>urn:ogc:object:Sensor:MyOrg:12 34</procedure> <distanceInMeter>100</distanceInMeter> </range> </GetObservation> </pre>

IV. 성능평가

4.1 평가환경

본 실험은 Microsoft Windows 7 Ultimate K를 기반으로 수행되었으며, 시스템 환경은 AMD Phenom(tm) II X4 945 3.00 GHz이고 메인 메모리는 4GB인 단일 서버 환경이다. S-SOS는 52 North의 SOS를 확장했으며 Java로 구현되었다. 실험에 사용된 GeoSensor의 데이터 셋은 센서 시뮬레이터 (Sensor Simulator)로 대체하였다. 센서 시뮬레이터는 제안 기법의 성능을 평가하기 위해 개발된 응용프로그램으로 시스템 관리 툴의 일종이다. 센서 시뮬레이터는 고정 및 이동형 GeoSensor를 선택하면 데이터의 사이즈 및 발생 주

기 등 몇 가지 옵션을 설정할 수 있다. GIS 데이터는 2009 Census TIGER/Line® Shapefiles 데이터로 Oracle 10g에 구축하여 본 실험에 사용한다.

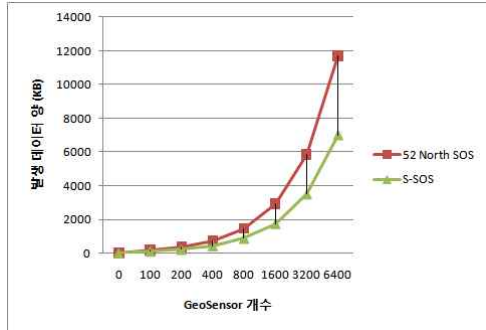


그림 3. 통신 데이터 감소량 평가
Fig. 3. Evaluation of communication data decrement

4.2 성능평가

본 논문에서 제안하는 시스템의 우수성을 증명하기 위해 기존 52 North SOS와 S-SOS의 공간연산을 위한 GetObservation 응답 메시지 크기를 비교하여 통신 데이터 크기의 차이를 보인다. S-SOS의 공간연산을 통해 감소된 클라이언트의 시스템 부하는 생략한다.

표 10. 센서의 증가에 따른 통신 데이터 발생 양 비교
Table 10. Difference of communication cost as the number of sensors increases

센서의 수	0	100	200	400	800	1600	3200	6400
52 North SOS	0	182.62	365.2	730.4	1460.8	2921.6	5843.2	11686.4
S-SOS	0	116.78	219.63	438.53	876.3	1751.93	3503.13	7005.53

단위 : KB

표 10은 52 North SOS를 통해 n개의 GeoSensor로부터 n번 호출된 GetObservation 메시지 크기의 합과 S-SOS로부터 공간연산이 수행된 후 전송된 GetObservation 메시지의 크기를 나타낸다. 그림 3에서 볼 수 있듯이 실험을 통해 기존 52 North SOS는 GetObservation 메시지의 호출이 늘어날수록 메시지의 크기가 급증하지만 그에 반해 S-SOS는 매우 적은 폭의 증가를 나타낸다. 6400개의 센서를 통해 실험된 결과에서 GetObservation 메시지의 크기는 59.94% 감소된 결과를 보였으며 GeoSensor의 수가 증가할수록 기존 SOS 대비 S-SOS의 메시지 크기는 더욱 큰 폭으로 감소한다.

V. 결론

기존 센서 네트워크들의 비 표준화로 인한 단점을 해결하기 위해 2007년 OGC에서 SWE를 발표했다. 이는 XML을 기반으로 센서와 센서 시스템에 대한 표준화된 접근 및 조작 인터페이스를 제공한다. SWE가 갖고 있는 프로토콜 중 센서 데이터의 접근을 위한 SOS에 대한 다양한 시도들이 있어왔다. 하지만 고정 및 이동형 GeoSensor를 위한 공간 연산을 지원하는 SOS에 대한 연구는 진행된 적이 없었다.

본 논문에서는 기존 SOS의 공간 연산이 확장된 S-SOS를 제안한다. S-SOS는 적은 컴퓨팅 리소스를 사용하는 클라이언트 측에서 구현 및 실행했던 공간 연산을 SOS 서버에서 제공함으로써 클라이언트 측의 공간 연산에 따른 시스템 부하를 감소시켰으며 그 결과 SOS 서버와 클라이언트의 통신 데이터양도 큰 폭으로 감소할 수 있었다.

향후 연구로써 수많은 클라이언트로부터 공간 연산 및 다양한 센서 정보를 요청받는 SOS 서버에 대한 성능 고도화를 위해 R트리 기반 적용된 SOS와 이에 적합한 새로운 캐싱 기법에 대하여 연구한다.

참고문헌

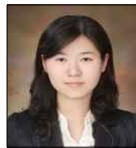
- [1] Botts Mike, and Alex Robin, "Bringing the Sensor Web Together," Geosciences, pp. 46-53, October 2007
- [2] The Open Geospatial Consortium (OGC) is an international industry consortium of 419 companies, government agencies and universities participating in a consensus process to develop publicly available

- interface standards
<http://www.opengeospatial.org>
- [3] OGC Implementations by Specifications
<http://www.opengeospatial.org/resource/products/byspec/?specid=289>
- [4] Open Geospatial Consortium Inc., "OpenGIS® Geography Markup Language (GML) Encoding Standard," Reference Number : OGC 07-036, August 2007
- [5] S. Shekhar and et al., "WMS and GML based interoperable Web Mapping System," ACM CIKM 2001, pp. 106-111, November 2001
- [6] J. E. Corcoles and P. Gonzalez, "A Specification of a Spatial Query Language over GML," ACM CIKM 2011, pp. 112-117, November 2011
- [7] R. R. Vatsavai, "GML-QL: A Spatial Query Language Specification for GML,"
<http://www.cobblestoneconcepts.com/ucgis2summer2002/vatsavai/vatsavai.htm>, 2002
- [8] A. Deutsch and et al., "XML-QL: A Query Language for XML," <http://www.w3.org/TR/NOTE-xml-ql/>, August 1998
- [9] Open Geospatial Consortium Inc., "Web Map Server Interface Specification," Revision 1.0, 2000
- [10] J. Robie, "The design of XML,"
<http://www.w3.org/style/XSL/Group/1998/09/XQL-design.html>, September 1998
- [11] Open Geospatial Consortium Inc., "OpenGIS® Sensor Model Language (SensorML) Implementation Specification," Reference Number : OGC® 07-000, July 2007
- [12] Open Geospatial Consortium Inc., "Observations and Measurements - Part 1 - Observation schema," Reference Number : OGC 01-022r1, December 2007
- [13] Open Geospatial Consortium Inc., "Observations and Measurements - Part 2 - Sampling Features," Reference Number : OGC 07-002r3, December 2007
- [14] Open Geospatial Consortium Inc., "Observations and Measurements - XML Implementation," Reference Number : OGC 10-025r1, March 2011
- [15] Open Geospatial Consortium Inc., "OpenGIS® Transducer Markup Language (TML) Implementation Specification," Reference Number : OGC 06-010r6, July 2007
- [16] Open Geospatial Consortium Inc., "Sensor Observation Service," Reference Number : OGC 06-009r6, October 2007
- [17] Open Geospatial Consortium Inc., "OGC® Sensor Alert Service Candidate Implementation Specification," Reference Number : OGC 06-028r3, May 2006
- [18] Open Geospatial Consortium Inc., "OGC® Sensor Planning Service Implementation Specification," Reference Number : OGC® 07-014r3, August 2007
- [19] Simon Jirka, Arne Bröring, Theodor Foerster, "Handling the Semantics of Sensor Observables within SWE Discovery Solutions," Collaborative Technologies and Systems (CTS), pp. 322-329, May 2010
- [20] Nengcheng Chen and et al., "A flexible geospatial sensor observation service for diverse sensor data based on Web service," ISPRS Journal of Photogrammetry and Remote Sensing, Vol. 64, Issue 2, pp. 234-242, 2009
- [21] Denis Havlik, Thomas Bleier and Gerald Schimak, "Sharing Sensor Data with SensorSA and Cascading Sensor Observation Service," Sensors, Vol.9, Issue 7, July 2009

저 자 소개



이 혁
2010 : 인하대학교 컴퓨터정보공학과 공학사.
현 재 : 인하대학교 컴퓨터정보공학과 석사과정
관심분야 : 데이터 스트림, 클라우드 컴퓨팅, 위치기반 서비스
Email : hyuklee@dblab.inha.ac.kr



이 연
2006 : 중국 중경우전대학교 지리정보시스템학과 이학사.
2008 : 인하대학교 컴퓨터정보공학과 공학석사.
현 재 : 인하대학교 컴퓨터정보공학과 박사과정
관심분야 : 공간 데이터베이스, 공간 데이터 웨어하우스, 지리정보시스템, USN, 스트림 데이터 시스템
Email : leeyeon@dblab.inha.ac.kr



정 원 일
1988 : 인하대학교 전자계산학과 공학사.
2004 : 인하대학교 컴퓨터정보공학과 공학박사.
2004-2006 : 한국전자통신연구원 선임 연구원
현 재 : 호서대학교 정보보호학과 조교수
관심분야 : 데이터스트림, 이동객체, 데이터베이스
Email : wnchung@hoseo.edu



배 혜 영
1974 : 인하대학교 응용물리학과 공학사.
1978 : 연세대학교 전자계산학과 공학석사.
1989 : 숭실대학교 전자계산학과 공학박사
1985 : Univ. of Houston 객원교수
1992 - 1994 : 인하대학교 전자계산소장
2004 - 2006 : 인하대학교 정보통신대학원장
2006-2009 : 인하대학교 대학원장
현 재 : 인하대학교 컴퓨터정보공학과 교수 지능형 GIS연구센터 센터장 중국 중경우전대학교 대학원 명예교수
관심분야 : 데이터베이스, 공간 데이터베이스, 멀티미디어 지리정보 시스템, 데이터베이스
Email : hybae@inha.ac.kr