

상황 센서 스트림 처리 시스템의 설계

권오제^o, 이기준
부산대학교 정보컴퓨터공학부
kwonoj@isel.cs.pusan.ac.kr, lik@pnu.edu

Design of Contextual Sensor Data Stream Management System

Kwon Oje^o, Li Ki-Joune
Department of Computer Science & Engineering, Pusan National University

요 약

본 논문에서는 이종의 센서들로부터 획득되는 스트림 데이터를 분석하여 상황 인지를 위한 데이터를 제공하는 상황 스트림 처리 시스템을 설계하였다. 상황 스트림 처리 시스템은 각 센서 스트림의 카탈로그 정보를 관리하고 이를 이용하여 센서 스트림 데이터를 상황인지데이터로 변환한다. 뿐만 아니라 상황인지 데이터에 대한 연속적인 질의를 정의하고 질의 처리를 위한 다양한 연산자들을 제공한다. 기존의 DSMS들은 단순히 센서 데이터를 처리하고 전달하는 반면 본 논문에서 제안하는 상황 스트림 처리 시스템은 이종의 센서 데이터와 상황정보를 결합하여 다양한 상황인지데이터를 제공하여 다양한 응용 분야에 활용될 수 있다.

1. 서 론

센서 네트워크의 발전으로 센서로부터 획득되는 다양한 상황에 관한 정보들을 활용하여 다양한 서비스들을 제공할 수 있다. 특히 이종의 센서 데이터로부터 상황을 추론하는 기술은 다양한 응용 분야에 매우 중요하게 활용될 수 있다. 하지만 센서로부터 획득되는 상황정보를 처리하는 기술은 그 필요성에 비해 아직 많은 연구가 진행되지 않았다. 기존의 데이터 스트림 처리 시스템들은 단순히 센서의 데이터를 처리하고 전달해주는 역할에 초점을 맞추고 있다. 하지만 상황 정보를 분석하여 서비스하기 위해서는 이종의 센서간의 데이터 분석 뿐만 아니라 상황정보와 센서 정보간의 결합(binding) 역시 매우 중요하다. 따라서 본 논문에서는 센서로부터 획득되는 상황 스트림 데이터를 분석하고 사용자의 연속적인 질의를 처리하는 상황 스트림 처리 시스템을 설계하였다.

상황 스트림 처리 시스템은 센서 네트워크로부터 전달되는 이종의 센서 스트림을 상황인지데이터로 변환하여 저장소에 저장한다. 뿐만 아니라 센서 데이터를 상황인지데이터로 변환하기 위해서는 각 센서 스트림의 정보와 상황정보간의 결합이 필요한데 이를 위해 상황 스트림 처리 시스템은 센서 스트림간의 카탈로그 정보를 관리한다. 또한 상황인지데이터를 검색하기 위한 연속적인 질의를 불린(boolean) 함수 형태로 정의하고 이를 처리하기 위한 다양한 연산자들을 제공한다.

본 논문은 다음과 같이 구성된다. 2장에서는 관련 연구로 센서로부터 전달되는 데이터들을 정의하기 위한 모델링 방법과 기존의 데이터 스트림 처리 시스템들에 대해 알아본다. 3장에서는 본 논문에서 제안하는 상황 스트림 처리 시스템의 구조와 센서 카탈로그 관리 및 연속적인

질의 처리 과정에 대해 살펴보고 4장에서 결론 및 향후 연구를 서술한다.

2. 관련 연구

본 장에서는 관련 연구로 센서 스트림 데이터를 정의하기 위한 모델링 방법들과 기존의 데이터 스트림 처리 시스템들에 대해 살펴본다.

2.1. 센서 데이터의 모델링

센서 및 디바이스들로부터 다양한 상황 정보가 스트림 형태로 획득되는데 이러한 센서 스트림은 센서의 종류에 따라 다양한 형태로 표현될 수 있다. 따라서 센서로부터 획득된 정보를 모델링하기 위한 다양한 방법들이 연구되고 있다. 대표적인 작업으로 Open Geospatial Consortium의 Sensor Web Enhancement Working Group의 표준인 Sensor Modeling Language (SensorML)[1], Transducer Markup Language (TML)[2] 그리고 Sensor Observation Service (SOS)[3]가 있다.

SensorML은 센서에서 생성되는 데이터를 표준화하여 호환되어 이용될 수 있도록 센서의 위치 정보를 포함하여 다양한 종류의 값들을 XML 형태로 정의하고 있다. TML은 이종의 센서들이 주변 상황의 정보를 인지하여 값으로 변환하는 과정의 기능적 표준을 정의하고 있다. SOS는 센서나 센서로부터 획득된 데이터를 다루기 위한 다양한 API를 제공한다.

2.2. 데이터 스트림 처리 시스템 (DSMS : Data Stream

Management System)

센서로부터 획득되는 데이터는 시간에 연속적인 스트림의 형태를 지닌다. 따라서 스트림 데이터에 대한 연속적인 질의 처리를 위한 데이터 스트림 처리 시스템에 대한 연구가 활발하게 진행되었다. 대표적인 DSMS 기술로는 스탠포드 대학의 STREAM[4], 버클리 대학의 TelegraphCQ[5], 위스콘신 매디슨 대학의 NiagaraCQ[6] 그리고 브라운 대학과 M.I.T의 Aurora[7] 등이 있다.

STREAM은 센서 스트림의 연속적인 질의를 처리하기 위해 기존의 DBMS의 질의어인 SQL을 확장하여 CQL (Continuous Query Language)를 정의하고 관계형 연산자들을 연속적인 질의에 맞게 확장하였다. TelegraphCQ는 다량의 스트림을 동시에 처리할 때 발생할 수 있는 성능 저하를 개선하기 위해 자원 공유 기반의 질의 처리 방법을 제안하였다. 뿐만 아니라 동적인 스트림 데이터와 정적인 데이터베이스 간의 연산을 위한 push 및 pull API 집합인 Fjords[8]를 제공한다. NiagaraCQ는 분산 환경에서 XML을 이용하여 센서 스트림의 연속 질의를 처리하기 위한 방법을 제공한다. Aurora는 센서 스트림의 전송 속도가 가변적일 때 실시간적인 연속 질의를 처리하기 위한 메모리 연산 스케줄링 방법을 제공한다.

3. 상황 스트림 처리 시스템

상황 스트림 처리 시스템은 이종의 센서로부터 전달받은 저레벨의 센서 스트림 데이터를 분석하고 가공하여 상황인지데이터로 변환하는 역할을 수행한다. 뿐만 아니라 사용자 또는 다른 시스템에서 상황인지 데이터를 필요로 할 경우 이를 위한 연속적인 질의를 제공한다. 이를 위해 상황 스트림 처리 시스템은 이종의 센서에 대한 카탈로그 정보 및 사용자의 연속적인 질의를 수행하기 위한 질의 정보를 관리한다. 그림 1은

상황 스트림 처리 시스템의 개념적인 수행 과정을 나타낸다.



그림 1 상황 스트림 처리 시스템의 개념적인 수행 과정

3.1. 상황 스트림 처리 시스템의 구조

상황 스트림 처리 시스템은 그림 2와 같이 크게 4가지의 세부 모듈로 구성된다.

- **센서 스트림 카탈로그 관리기(Stream Catalog Manager)**

센서 스트림 카탈로그 관리기는 센서 스트림 데이터를 상황인지데이터로 변환하기 위한 센서의 기본 정보를 관리한다. 센서 카탈로그 정보는 테이블 형태로 정의되는데 센서의 기본 정보(센서 id, 종류, 값의 상하한 값, etc.) 뿐만 아니라 사용자의 연속적인 질의를 수행하기 위한 전처리 필터 정보들을 포함한다.

- **센서 스트림 분석기(Stream Analyzer)**

센서 스트림 분석기는 센서 네트워크로부터 전달되는 이종의 센서 스트림을 분석하여 센싱 값을 기준으로 분류하여 각각의 부분 스트림으로 나누는 역할을 수행한다.

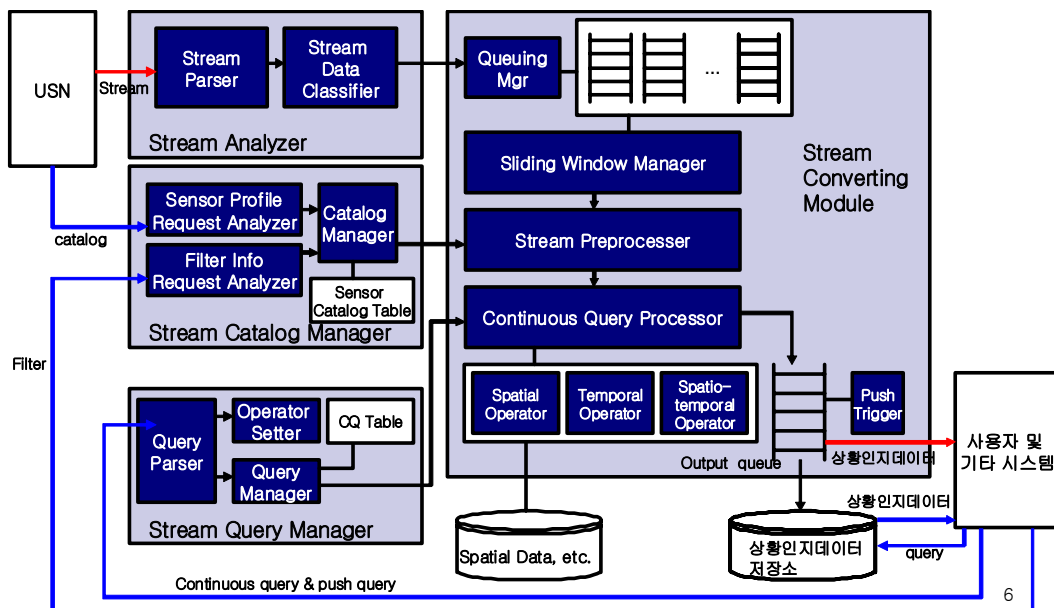


그림 2 상황 스트림 처리 시스템의 구조

- **센서 스트림 연속 질의 관리기(Stream Query Manager)**

센서 스트림 연속 질의 관리기는 사용자로부터 상황인지데이터를 검색하기 위해 질의되는 연속 질의를 관리한다. 연속 질의는 상황 인지 연산자들로 구성된 불린 함수(Boolean function) 형태로 표현되는데 이는 3.3절에서 자세히 살펴본다.

- **센서 스트림 변환기(Stream Converting Module)**

센서 스트림 변환기는 센서 스트림의 정보와 연속적인 질의 정보를 바탕으로 센서 스트림 데이터를 상황인지데이터로 변환하고 저장하는 역할을 수행한다. 뿐만 아니라 사용자의 연속 질의를 처리하고 그 결과를 사용자에게 전달해준다.

3.2. 센서 스트림의 카탈로그 관리

기존의 DSMS들은 주로 센서 데이터를 처리하고 사용자에게 전달해주는 단순한 기능을 수행하였다. 하지만 상황 분석을 위해서는 단순히 센서의 정보만을 전달하는 것이 아니라 이를 위한 센서 간의 결합(binding)이 수행되어야 한다. 예를 들어 "현재 건물 이동 통로에 있는 인원 수를 알려달라." 라는 상황의 질의가 주어진다고 가정하자. 이를 처리하기 위해서는 각 이동 통로(복도, 계단, 엘리베이터 등)에 존재하는 센서들로부터 획득된 정보들을 통합하여 분석해야 한다. 따라서 그림 3과 같이 센서와 상황정보 사이의 결합작업이 필요하다. 본 논문의 상황 스트림 처리 엔진은 이종의 센서들의 카탈로그 정보를 관리하고 이를 이용하여 센서 간의 결합을 통한 질의 처리를 수행한다.

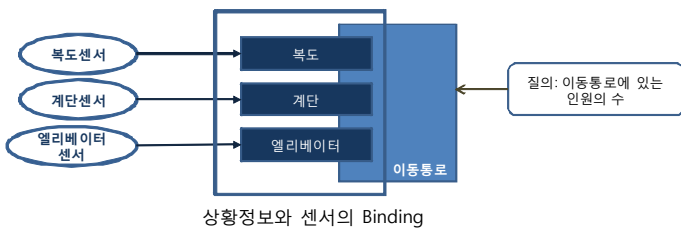


그림 3 상황정보와 센서의 결합

3.3. 센서 스트림의 연속적인 질의 처리

상황 판단을 위한 연산자들은 시스템의 응용 분야에 의해 결정된다. 하지만 상황을 인지하기 위해서는 데이터의 공간, 시간 그리고 시공간적인 특징을 고려해야 한다. 본 논문에서는 시스템의 응용 분야와 독립적으로 상황 인지를 위해 센서 스트림 데이터가 가지는 공간, 시간 그리고 시공간적인 특징에 대해 표

1과 같은 연산자 집단을 정의하였다.

표 1 연속적인 상황 질의를 처리하기 위한 연산자의 예

분류	연산자
공간연산자	INSIDE, OUTSIDE, NEAR
시간연산자	DURING
시공간연산자	ENTER, LEAVE, PASS, UP, DOWN
단순 연산자	ON, OFF, HIGH, LOW
조인연산자	MULTIJOIN

사용자 또는 다른 응용 프로그램들은 표 1의 연산자들을 이용하여 상황인지데이터에 대한 연속적인 질의를 요청할 수 있다. 연속적인 질의는 위의 연산자의 조건을 만족하는지의 유무에 의해 불린 함수 형태로 표현된다. 표 1의 연산자를 바탕으로 연속적인 상황 질의를 불린 함수 형태로 표현하면 표 2와 같다.

표 2 연속적인 상황 질의의 예

질의 1. "강의실 410호에 철수(sensor1) 또는 영희(sensor2) 중 한 사람만 존재할 경우 알려 달라."

```
SELECT c.context_id
FROM Context c, Stream s1, Stream s2
WHERE (INSIDE(s1,room("410")) = True and
INSIDE(s2, room("410")) = False) or
(INSIDE(s1,room("410")) = True and INSIDE(s2,
room("410")) = False)
```

질의 2. "강의실 410호의 온도가 30도 보다 높은 상황이 1분 이상 지속될 경우 알려 달라."

```
SELECT c.context_id
FROM Context c, TemperatureStream t
WHERE DURING(HIGH(room("410"), 30)) > 60
```

연속적인 질의 처리 과정은 그림 4와 같이 수행된다. 먼저 사용자로부터 상황 정보에 대한 질의를 요청받으면 센서 스트림 연속 질의 관리기에서 표 2의 예와 같은 연속 질의를 분석한다. 질의 관련 정보와 센서의 카탈로그 정보를 바탕으로 센서 스트림 변환기는 센서 스트림을 상황인지데이터로 변환하여 저장소에 저장한다. 이 때 질의의 조건을 만족하는 상황 데이터가 생성이 되면 상황 스트림 변환기는 트리거(trigger) 메시지를 사용자에게 전달하고 사용자는 저장소로부터 질의에 해당하는 상황인지데이터를 제공받게 된다.

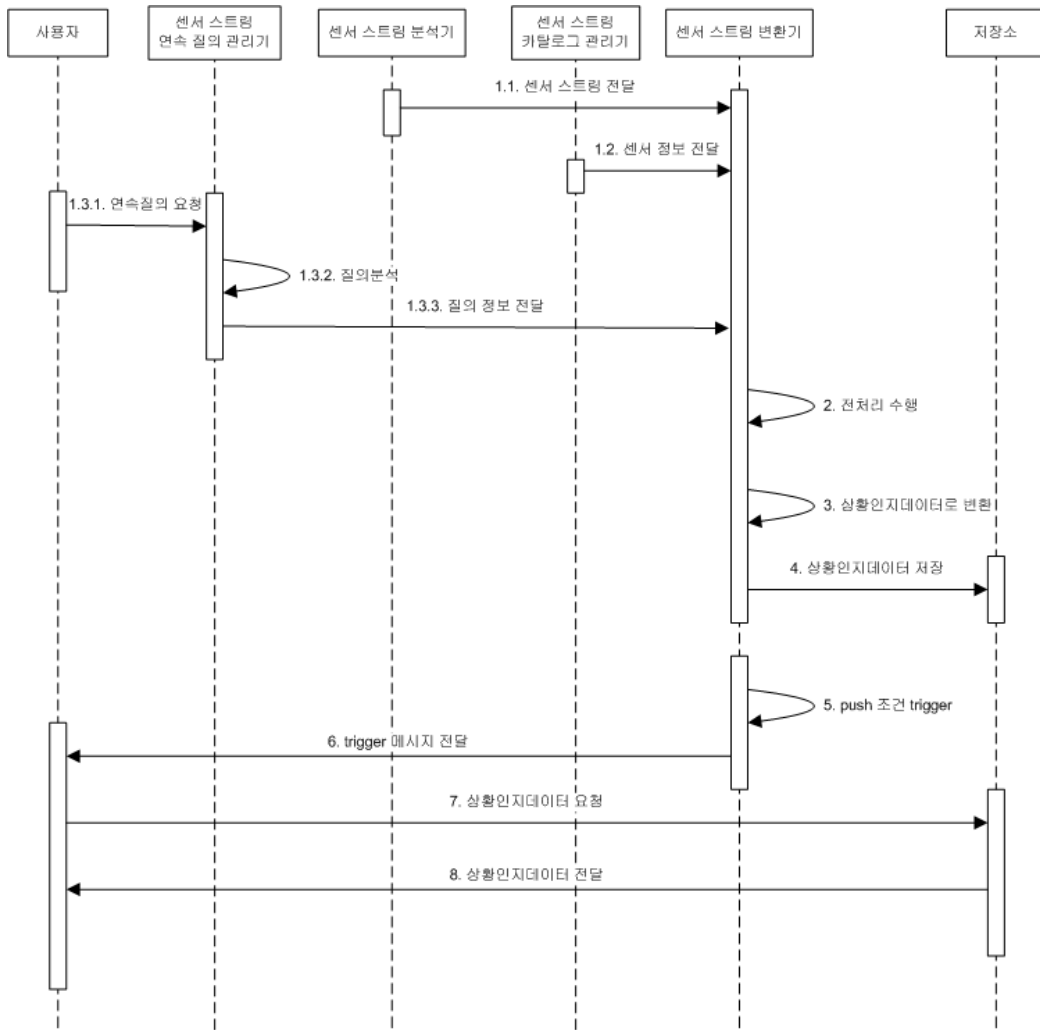


그림 4 센서 스트림의 연속 질의 처리 과정

4. 결론 및 향후 연구

본 논문에서는 이종의 센서 스트림을 분석하여 상황 추론을 위한 상황인지데이터로 변환하고 저장하는 상황 스트림 처리 시스템을 설계하였다. 본 논문의 상황 스트림 처리 시스템은

- 상황 정보에 대한 효율적인 질의를 처리하기 위해 상황정보와 센서정보간의 결합을 위한 센서 카탈로그 정보를 관리하고,
- 사용자의 연속적인 질의 처리를 위한 불린 함수 형태의 연속 질의를 정의하고 이를 위한 연산자들을 정의하였다.

본 연구는 지식경제부 및 정보통신연구진흥원의 IT핵심기술개발사업의 일환으로 수행하였음. [2008-S-018-01, u-City Service용 개방형 SW 플랫폼 개발]

본 논문은 2단계 두뇌한국21사업에 의하여 지원되었음

참고문헌

- [1] OpenGIS Sensor Model Language(SensorML), <http://www.opengeospatial.org/standards/sensorml>
- [2] OpenGIS Transducer Markup Language, <http://www.opengeospatial.org/standards/tml>
- [3] OpenGIS Sensor Observation Service(SOS), <http://www.opengeospatial.org/standards/sos>
- [4] A. Arasu, B. Babcock, S. Babu, J. Cieslewicz, M. Datar, K. Ito, R. Motwani, U. Srivastava and J. Widom, "Stream: The Stanford Data Stream Management System", in IEEE Data Engineering Bulletin, vol. 4(1), 2003
- [5] S. Chandrasekaran, O. Copper, A. Deshpande, M. J. Franklin, H. Joseph M, W. Hong, S. Krishnamurthy, S. Madden, V. Raman, F. Reiss and M. Shah, "TelegraphCQ: Continuous Dataflow Processing for an Uncertain World", in Proceedings

of the Conference on Innovative Data Systems Research, pp.11–18, 2003

[6] Jianjun Chen, David J. DeWitt, Feng Tian and Yuan Wang, "NiagaraCQ: A Scalable Continuous Query Stream for Internet Databases", in ACM SIGMOD Record, pp.379–390, 2000

[7] Don Carney, Uğur Cetintemel, Mitch Cherniack, Christian Convey, Sangdon Lee, Grag Seidman, Michael Stonebraker, Nesime Tatbul and Stan Zdonik, "Monitoring Streams– A New Class of Data Management Applications", in Proceedings of the 28th International Conference on Very Large Data Bases, pp 215–226, 2002

[8] S. Madden and M. J. Franklin, "Fjording the Stream: An Architecture for Queries over Streaming Sensor Data" in Proceedings of the Conference on Innovative Data Systems Research, pp.11–18, 2003