

논문 2010-3-19

유비쿼터스 환경을 위한 시공간 질의 처리 시스템

Spatio-temporal Query Processing Systems for Ubiquitous Environments

이기영*, 임명재**, 김규호***, 김정준****

Ki-Young Lee, Myung-Jae Lim, Kyu-Ho Kim, Joung-Joon Kim

요 약 최근 유비쿼터스 컴퓨팅 기술의 발전과 함께 센서 및 RFID에 관련한 정보 인식, 위치 측위와 같은 다양한 유비쿼터스 영역에 대한 관심과 연구가 증대되고 있다. 특히, 시공간 데이터를 다루는 실시간 위치 추적 서비스 기술이 대두되고 있다. 이에, ISO/IEC에서는 실시간 위치 추적 서비스의 데이터 호환성과 상호 운용성을 위해 RTLS 표준 명세를 제시하였다. 본 논문에서는 이동 객체의 시공간 데이터 스트림에 대한 효율적인 관리와 검색을 지원하는 시공간 질의 처리 시스템을 설계 및 구현하였다. 시공간 질의 처리 시스템의 시공간 미들웨어는 끊임없이 연속으로 들어오는 시공간 데이터 스트림을 실시간으로 처리하고 시공간 DBMS간의 양방향 동기화를 지원한다. 웹 서버는 서버와 클라이언트간의 상호 운용성을 위하여 SOAP(Simple Object Access Protocol) 메시지를 사용하며, 클라이언트의 SOAP 메시지를 분석하여 시공간 미들웨어의 CQL(Continuous Query Language)로 변환하는 기능을 지원한다. 마지막으로, 본 논문에서는 실시간 위치 추적 서비스에 적용해 봄으로써 시공간 질의 처리 시스템의 효용성을 입증하였다.

Abstract With the recent development of the ubiquitous computing technology, there are increasing interest and research in technologies such as sensors and RFID related to information recognition and location positioning in various ubiquitous fields. Especially, RTLS(Real-Time Locating Services) dealing with spatio-temporal data is emerging as a promising technology. For these reasons, the ISO/IEC published the RTLS standard specification for compatibility and interoperability in RTLS. Therefore, in this paper, we designed and implemented Spatio-temporal Query Processing Systems for efficiently managing and searching the incoming Spatio-temporal data stream of moving objects. Spatio-temporal Query Processing Systems's spatio-temporal middleware maintains interoperability among heterogeneous devices and guarantees data integrity in query processing through real time processing of unceasing spatio-temporal data streams and two way synchronization of spatio-temporal DBMSs. Web Server uses the SOAP(Simple Object Access Protocol) message between client and server for interoperability and translates client's SOAP message into CQL(Continuous Query Language) of the spatio-temporal middleware. Finally, this thesis proved the utility of the system by applying the spatio-temporal Query Processing Systems to a real-time Locating Services.

Key Words : Ubiquitous, Spatio-temporal Data Stream, Spatio-temporal Query Processing Systems, CQL

I. 서 론

최근 유비쿼터스 컴퓨팅 환경에서 시공간 및 위치 정보를 제공하는 u-GIS 공간 정보 기술이 미래의 핵심 기술로 대두되고 있다. 특히, RFID(Radio Frequency Identification)나 GPS(Global Positioning System) 등과 같은 위치 측위 기술의 발달로 무선 통신 기술을 이용한

*중신회원, 을지대학교 의료IT마케팅학과

**중신회원, 을지대학교 의료IT마케팅학과(교신저자)

***정회원, 을지대학교 의료IT마케팅학과

****정회원, 건국대학교 컴퓨터공학과

접수일자 2010.6.05 수정일자 2010.6.17

위치 추적 서비스가 증가함에 따라 이동 객체의 위치 정보를 실시간으로 서비스하기 위한 실시간 위치 추적 서비스 기술 연구가 활발히 진행되고 있다. 위치 추적 서비스는 RFID 태그를 부착한 이동 객체들의 위치 정보를 수집하여 어플리케이션에 제공하는 실시간 위치 추적 서비스로 최근 물류/유통 관리, 의료/보건, 자산 관리, 기업 보안, 공공 안전 등의 다양한 분야에서 널리 활용되고 있다^[1].

실시간 위치 추적 서비스에서는 효율적인 서비스 제공을 위해 RFID 태그로부터 끊임없이 입력되는 데이터 스트림에 대한 안정적인 처리가 필요하다. 그러나 기존의 실시간 위치 추적 서비스는 주로 응용 서비스를 위한 기능에만 중점을 두고 있기 때문에 이러한 데이터 스트림으로 인해 시스템 부하가 크게 증가한다^[2]. 따라서, 데이터 스트림을 효율적이고 안정적으로 처리할 수 있는 데이터 스트림 처리 기능이 필요하다. 그리고 기존 실시간 위치 추적 서비스에서는 다양한 위치 추적 서비스 제공을 위해 필요한 공간 연산 처리가 미흡하기 때문에 공간 데이터 처리 시 많은 연산 비용이 발생한다. 따라서, 실시간 위치 추적 서비스를 효율적으로 제공하기 위해서는 다양한 시공간 연산자 지원이 필요하다^[3]. 본 논문에서는 이동 객체의 시공간 데이터 스트림에 대한 효율적인 관리와 검색을 지원하는 시공간 질의 처리 시스템을 설계 및 구현하였다. 시공간 질의 처리 시스템의 시공간 미들웨어는 끊임없이 연속으로 들어오는 시공간 데이터 스트림을 실시간으로 처리하고 시공간 DBMS간의 양방향 동기화를 지원한다. 또한, OGC(Open Geospatial Consortium)에서 제시한 “Simple Feature Specification for SQL”을 확장한 시공간 데이터 타입을 지원한다. 웹 서버는 서버와 클라이언트간의 상호 운용성을 위하여 SOAP(Simple Object Access Protocol) 메시지를 사용하며, 클라이언트의 SOAP 메시지를 분석하여 시공간 미들웨어의 CQL(Continuous Query Language)로 변환한다.

마지막으로, 시공간 질의 처리 시스템을 실시간 위치 추적 서비스에 적용해 봄으로써 시스템의 효율성을 실험하였다.

II. 관련 연구

본 장에서는 ISO/IEC에서 제공한 실시간 위치 추적 서비스 표준 명세와 OGC에서 제공한 Simple Feature

Specification for SQL에 대하여 설명한다.

1. 실시간 위치 추적 서비스 표준 명세

국제 표준화 단체인 ISO/IEC에서는 지정된 공간에서 자산 또는 사람 등에 RFID 태그를 부착하여 실시간으로 각 이동객체의 위치와 상태 데이터를 모니터링할 수 있는 시스템을 설계하고 상호 기기종 플랫폼 상의 데이터 운용성을 보장할 수 있도록 실시간 위치 추적 서비스 표준 명세를 제시하였다^[4].

Query 스키마에 정의된 QueryName 엘리먼트는 RTLS에 질의 이름을 지정하고 FilterBy 엘리먼트는 검색 조건을 지정한다. Fields 엘리먼트는 TagBlink 필드의 이름을 지정하여 RTLS로 입력되는 데이터 스트림에서 필요한 필드만을 검색하여 읽어 올 수 있도록 하고, SortBy 엘리먼트는 검색된 TagBlink의 정렬 순서를 지정하며 Fields 엘리먼트에서 선택한 필드를 기준으로 정렬 순서를 결정한다.

OpenSession은 RTLS에 새로운 세션을 생성하고, 검색한 이동 객체의 상태 및 위치 데이터를 세션에 저장하며, 생성한 세션을 클라이언트로 전달하는 세션 기반 프로시저를 말한다. 마지막으로 CloseSession은 생성된 세션이 더 이상 필요하지 않을 경우에 해당 세션의 모든 리소스를 해제하는 세션 기반 프로시저를 말한다.

2. OGC SQL 스키마

OGC에서 제안한 SQL 스키마인 “Simple Feature Specification for SQL”은 ODBC API를 경유하는 심플 공간 피쳐 집합의 저장, 검색, 질의, 그리고 갱신을 지원하는 표준 SQL 스키마를 정의하고 있다^[5]. “Simple Feature Specification for SQL”은 그림 1과 같은 계층 구조로 구성되어 있다.

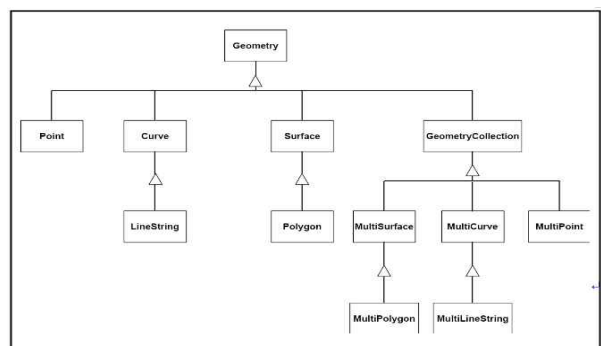


그림 1. SQL Geometry 타입의 계층 구조
Fig 1. Hierarchy of SQL Geometry Type

III. 시스템 설계

본 장에서는 시공간 질의 처리 시스템의 전체 구조 및 상세 설계에 대해서 자세히 설명한다.

1. 시스템 전체 구조

그림 2는 본 논문에서 개발한 유비쿼터스 환경을 위한 시공간 질의 처리 시스템 전체 구조를 보여준다.

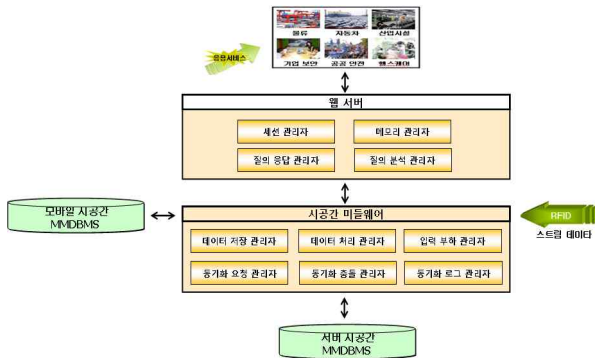


그림 2. 시스템 전체 구조
Fig 2. Structure of System

그림 2를 살펴보면 실시간 위치 추적 시스템은 시공간 미들웨어와 웹서버로 구성된다.

2. 시공간 미들웨어

가. 입력 부하 관리자

입력 부하 관리자는 입력 데이터 스트림의 양을 줄여 과부하 문제를 해결하기 위해 시공간 객체를 여러 저장소에서 공유할 수 있도록 시공간 객체 아이디와 참조 카운터를 관리함으로써 시공간 객체의 처리 효율성을 향상시킨다. 그림 3은 시공간 객체가 여러 저장소에서 공유되는 예를 보여준다.

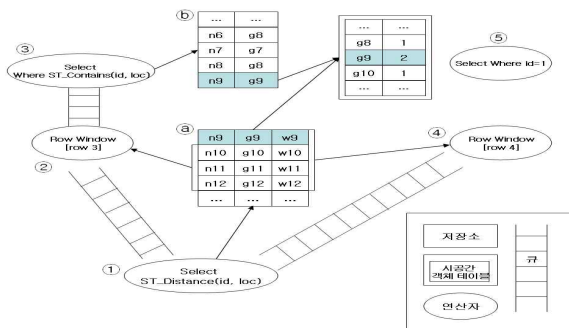


그림 3. 시공간 객체 공유 예
Fig 3. Example of Spatial-temporal Object Sharing

그림 3을 살펴보면 연산자 1이 ST_Distance 연산자를 통해 시공간 객체를 생성한 뒤 이것을 저장소 a에 저장해 두면 연산자 2와 4가 공유하게 된다. 이때, 연산자 2는 저장소 a에 있는 9번째 튜플을 처리하고 슬라이딩 윈도우를 10부터 12번 튜플로 옮긴 뒤 연산자 3이 이를 저장소 b에 저장하지만, 연산자 4가 9번째 튜플을 아직 처리하지 않아서 저장소 a에는 9번째 튜플이 남아있게 된다. 이러한 경우 저장소 a와 저장소 b 모두에 시공간 객체 g9가 존재하게 되며 g9에 대한 참조 카운터를 2로 늘린다. 연산자 4가 9번째 튜플을 처리하면 저장소 a에서 9번째 튜플이 삭제되는데, 이때 입력 부하 관리자는 Heap 영역에 있는 시공간 객체를 바로 삭제하지 않고 참조 카운터만 1로 줄인 후, 저장소 b에서도 9번째 튜플이 삭제되면 참조 카운터가 0이 되면서 g9를 삭제한다.

나. 데이터 처리 관리자

데이터 처리 관리자는 등록된 질의를 분석하여 실행 계획을 생성하고 연속 질의를 처리하는 기능을 수행하며, 질의에 시공간 연산자가 포함되어 있을 경우 이를 처리한다. 표 1은 시공간 미들웨어에서 지원하는 시공간 데이터 타입을 보여준다.

표 1. 시공간 데이터 타입
Table 1. Spatio-temporal Data Type

형식	설명	타입
ST_POINT	시공간 점 객체를 표현한다.	double
ST_LINESTRING	시공간 선 객체를 표현하며, 각 선들은 여러개의 점으로 구성된다.	double
ST_POLYGON	시공간 다각형 객체를 표현하며, 각 다각형들은 여러개의 점으로 구성된다.	double
ST_MULTIPoint	다중 시공간 점 객체를 표현한다.	double
ST_MULTILINESTRING	다중 시공간 선 객체를 표현한다.	double
ST_MULTIPOLYGON	다중 시공간 다각형 객체를 표현한다.	double

표 2는 시공간 미들웨어에서 지원하는 시공간 연산자의 종류와 이름 및 설명을 보여준다.

표 2. 시공간 연산자

Table 2. Spatio-temporal Operator

종류	이름	설명
시공간 관계 함수	ST_Equals	각 단위 시간마다 두 시공간 객체가 동일하면 참을 반환한다.
	ST_Disjoint	각 단위 시간마다 두 시공간 객체가 교집합이 없으면 참을 반환한다.
	ST_Touches	각 단위 시간마다 두 시공간 객체의 경계선만 닿았으면 참을 반환한다.
	ST_Within	각 단위 시간마다 첫 번째 시공간 객체가 두 번째 시공간 객체를 포함하면 참을 반환한다.
	ST_Overlaps	각 단위 시간마다 두 시공간 객체가 같은 차원이면서 겹쳐 있으면 참을 반환한다.
	ST_Crosses	각 단위 시간마다 두 시공간 객체가 교차하면서 교차된 영역의 차원이 같거나 줄어들면 참을 반환한다.
	ST_Intersects	각 단위 시간마다 두 시공간 객체가 교차하면 참을 반환한다.
	ST_Contains	각 단위 시간마다 두 번째 시공간 객체를 포함하면 참을 반환한다.
	ST_Relate	각 단위 시간마다 두 시공간 객체의 위상관계가 주어진 조건(9-Intersection Matrix)을 만족하는지 여부 반환한다.
시공간 분석 함수	ST_Distance	두 시공간 객체간의 거리 변화를 시간 순으로 추출하여 반환한다.
	ST_Intersection	두 시공간 객체가 겹치는 부분을 시간 순으로 추출하여 반환한다.
	ST_Difference	첫 번째 시공간 객체에서 두 번째 시공간 객체를 뺀 부분을 시간 순으로 추출하여 반환함.
	ST_Union	두 시공간 객체를 합친 시공간 객체를 시간 순으로 추출하여 반환한다.
	ST_Buffer	시공간 객체 주변을 주어진 크기만큼 버퍼링 해서 시간 순으로 계산, 추출하여 반환한다.

표 2를 살펴보면 시공간 미들웨어는 시공간 관계 연산자 9개와 시공간 분석 연산자 5개를 지원한다.

다. 데이터 저장 관리자

데이터 저장 관리자는 위치 측위 시스템에서 변환되는 동질 또는 이질의 데이터간에 요구되는 데이터의 교환을 위하여 DML의 구성을 변경하고 위치 측위 시스템에서 입력되는 데이터 스트림을 서버 시공간 MMDBMS에 저장하는 기능을 제공한다. 데이터 스트림은 실시간으로 처리되기 때문에 DSMS와 같은 시스템에서는 데이터 스트림의 저장이 이루어지지 않는다. 하지만 위치 추적 서비스에서 이동체의 위치 동선을 통해 이동경로를 파악하는 경우 과거의 이력 데이터가 필요하다.

라. 요청 관리자

요청 관리자는 데이터 동기화를 위하여 모바일 시공간 DBMS와 서버 시공간 MMDBMS로부터 들어오는 다수의 동기화 요청을 받아 동기화를 수행하는 기능을 제공한다. 그림 4는 모바일 시공간 DBMS와 서버 시공간 MMDBMS간의 동기화가 진행되는 프로토콜을 보여준다.

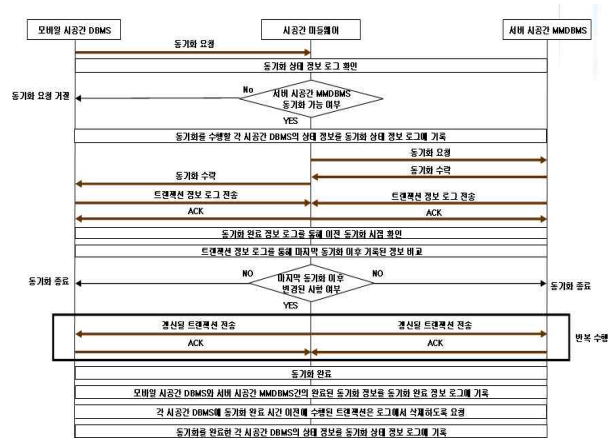


그림 4. 동기화 프로토콜
Fig 4. Synchronization Protocol

그림 4를 살펴보면 동기화가 진행되는 일련의 과정은 다음과 같다. 먼저, 모바일 시공간 DBMS에서 동기화를 요청한 후 동기화 수행이 가능하면, 동기화 완료 정보 로그를 통해 동기화를 수행할 각 시공간 DBMS의 최근 동기화 완료 시간을 확인한 후 각 시공간 DBMS들로부터 트랜잭션 정보 로그를 전송받아 비교한다. 이후, 각 시공간 DBMS에 반영할 질의를 생성하여 전송하고, 동기화 완료 정보 로그에 진행된 동기화 정보를 기록한다. 마지막으로 동기화 완료 후 동기화 상태 정보 로그에 해당 시공간 DBMS의 동기화 상태를 동기화 수행 가능 상태로 변경함으로써 동기화 과정이 끝나게 된다.

마. 충돌 관리자

충돌 관리자는 동기화 수행시 데이터의 충돌 여부를 확인하고 해결하는 기능을 제공한다. 데이터 충돌 발생시 정해진 정책에 따라 데이터의 충돌 종류를 확인하고 이를 처리한다. 표 3은 데이터 충돌 발생 상황에 따른 해결 정책을 보여준다.

표 3. 충돌 해결 정책
Table 3. Collision Solving Policy

데이터 충돌 종류	충돌 발생 상황	해결 정책
Duplication 충돌	최근 데이터 우선 방식으로 동기화시, 동일한 고유 ID, 시간 값, 트랜잭션 수행 시간을 가진 데이터가 존재할 경우	서버 시공간 MMDBMS 우선 방식
Insert 충돌	동일한 고유 ID, 시간 값을 가진 데이터를 삽입할 때 이미 다른 값으로 삽입된 경우	모바일 시공간 MMDBMS 우선 방식 서버 시공간 MMDBMS 우선 방식
Delete 충돌	동일한 고유 ID, 시간 값을 가진 데이터가 한쪽만 삭제된 경우	
Update 충돌	동일한 고유 ID, 시간 값을 가진 데이터가 다른 값으로 변경된 경우	최근 데이터 우선 방식

표 3에서 보는 바와 같이 최근 데이터 우선 방식으로 동기화 수행하는 도중 Duplication 충돌 발생시에는 서버 시공간 MMDBMS를 우선시하여 서버 시공간 MMDBMS 우선 방식으로 해결한다. 그리고 Insert, Delete, Update 충돌 발생시에는 요청된 동기화 수행 방식에 따라 데이터의 우선순위를 판단하여 이를 해결하게 된다.

바. 로그 관리자

로그 관리자는 동기화를 수행할 수 있는지의 상태 정보와 동기화 수행 완료시 해당 정보를 로그에 기록하고 관리하는 기능을 제공한다. 각 시공간 DBMS의 동기화 재요청시 로그를 이용하여 마지막 동기화 이후에 변화된 부분만 갱신하는 기법을 제공하는데 이를 위해 로그 관리자는 동기화를 수행할 수 있는지 여부의 정보를 저장하는 동기화 상태 정보 로그, 동기화 완료시 동기화를 수행한 각 시공간 DBMS의 정보를 저장하는 동기화 완료 정보 로그, 각 시공간 DBMS에서 수행된 트랜잭션 정보를 저장하는 트랜잭션 정보 로그를 관리한다.

3. 웹 서버

가. 질의 처리 관리 관리자

질의 처리 관리 관리자는 SOAP 메시지의 검색 조건에 맞는 TagBlink를 검색하여 수집하고, 그 결과를 질의 응답 관리 관리자로 전달하는 작업을 담당한다. 또한, 질의 처리 관리 관리자는 SOAP 메시지를 CQL로 변환하는 매핑 규칙을 적용해서 생성한 CQL 문을 공간 DSMS로 전달한다. 그림 5는 SOAP 메시지의 CQL 매핑 규칙을 보여준다.

SELECT Fields FROM RTLS	① Fields 엘리먼트는 “,”로 구분되어 CQL의 SELECT 절로 매핑된다.
[WHERE FilterBy]	② FilterBy 엘리먼트는 CQL의 WHERE 절로 매핑된다.
[ORDER BY SortBy]	③ SortBy 엘리먼트는 CQL의 Order By 절로 매핑된다.

그림 5. SOAP 메시지의 CQL 매핑 규칙
Fig 5. CQL Mapping Rules of SOAP Message

그림 5의 CQL 구문에서 Fields 절 부분은 생략될 수 없는 필수 부분이며, 대괄호로 쌓여있는 WHERE 절과 ORDER BY 절은 클라이언트가 SOAP 메시지를 RTLS로 전송할 때 FilterBy 또는 SortBy 엘리먼트를 생략하

여 전송하였을 경우 생략될 수 있다. 그림 6은 연산자간의 매핑 규칙을 보여준다.

RTLS	CQL
<, >	<, >
<=, >=	<=, >=
<>, =	!=, =

그림 6. 연산자 매핑 규칙
Fig 6. Operator Mapping Rules

그림 7은 질의 처리 관리 관리자로 전달된 Query 프로시저를 호출하는 SOAP 메시지와 CQL 예를 보여준다.

```

<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope">
  <env:Body>
    <Query xmlns="http://www.autoid.org/iso24730-1/RTLS-schema">
      <QueryName>RTLS_Blinks</QueryName>
      <FilterBy>
        <TagID><![CDATA[<1234]]></TagID>
        <Location>
          <<<![CDATA[>567]]></><<<![CDATA[<=1000]]></>
          <><><![CDATA[<1000]]></><><><![CDATA[>300]]></>
        </Location>
        <BatteryLow>=true</BatteryLow>
      </FilterBy>
      <Fields>TagID Location BatteryLow</Fields>
      <SortBy><Order>asc</Order><Field>TagID</Field></SortBy>
    </Query>
  </env:Body>
</env:Envelope>
    
```

```

SELECT TagID, Location, BatteryLow FROM RTLS
WHERE TagID != 1234 and X > 567 and X <= 1000 and Y < 1000 and Y > 300 and BatteryLow =true
ORDER BY TagID asc
    
```

그림 7. SOAP 메시지와 CQL 예
Fig 7. Example of SOAP Message and CQL

그림 7에서 보는 바와 같이 변환된 SOAP 메시지의 Fields 엘리먼트는 CQL 문의 SELECT TagID, Location, BatteryLow로 변환되었으며, FilterBy 엘리먼트는 CQL 문의 WHERE TagID != 1234 and X > 567 and X <= 1000 and Y < 1000 and Y > 300 and BatteryLow =true로 변환되었다. 그리고 SOAP 메시지의 SortBy 엘리먼트는 CQL 문의 ORDER BY TagID asc로 변환되었다.

나. 질의 응답 관리 관리자

질의 응답 관리 관리자는 RTLS의 질의 처리 관리 관리자와 세션 관리 관리자로 부터 데이터 값을 전달받아 클라이언트로 SOAP 메시지를 작성하여 응답을 하는 작업을 담당한다.

Query 프로시저를 호출하는 경우에는 질의 처리 관리

관리자에서 전달 받은 TagBlink 데이터를 SOAP 메시지로 작성하여 클라이언트로 응답하고, OpenSession 프로시저를 호출하는 경우에는 세션 관리 관리자로부터 전달 받은 세션 식별자 번호를 SOAP 메시지로 작성하여 클라이언트로 응답한다. 또한, QuerySession 프로시저를 호출하는 경우에는 세션 관리 관리자에서 전달 받은 TagBlink 데이터를 SOAP 메시지로 작성하여 클라이언트로 응답하고, CloseSession 프로시저를 호출하는 경우에는 세션이 삭제되었다는 SOAP 메시지를 작성하여 클라이언트로 응답한다.

다. 세션 관리 관리자

세션 관리 관리자는 클라이언트의 연결 정보 및 질의 관련 정보를 저장하기 위해 세션을 관리하는 작업을 담당한다. 세션에는 클라이언트의 검색 조건과 질의 응답 관리 관리자로부터 입력된 TagBlink 데이터가 저장되고, 세션 관리 관리자는 세션에 필요한 메모리 영역을 메모리 관리 관리자에게 요청한다.

또한, 세션 관리 관리자는 세션의 유효 시간이 초과될 때까지 생성된 세션을 클라이언트가 사용하지 않을 경우 해당 세션에 할당된 메모리 영역을 메모리 관리 관리자가 해제하도록 하고, 이와 반대로 세션의 유효 시간이 초과되기 전에 생성된 세션을 클라이언트가 사용하였을 경우 세션 관리 관리자는 메모리 관리 관리자의 벡터 사용 시간 변수에 새로운 세션 유효 시간을 다시 할당한다.

라. 메모리 관리 관리자

메모리 관리 관리자는 세션 관리 관리자에서 세션에 클라이언트의 검색 조건과 TagBlink 데이터를 저장하기 위해 필요한 메모리를 관리하는 작업을 담당한다. 메모리 관리 관리자에서 메모리 관리를 위해 사용하는 벡터 풀은 벡터 배열과 벡터 배열의 벡터 사용 유무 및 벡터 사용 시간을 저장하기 위한 구조체로 구성되어 있다.

메모리 관리 관리자는 세션 관리 관리자가 할당받았던 벡터를 사용한 후 반환하면 반환된 벡터 객체의 해당 벡터 사용 유무 변수를 사용하지 않음으로 할당하고 벡터를 벡터 풀로 회수한다. 또한, 세션 관리 관리자에서 생성한 세션의 유효 시간이 초과되면 세션이 더 이상 사용되지 않는 것으로 간주하고 세션에 할당했던 벡터를 강제로 회수하기 위해 벡터 사용 유무 변수를 사용하지 않음으로 할당하고 벡터를 벡터 풀로 회수한다.

IV. 시스템 구현 및 검증

본 장에서는 시스템의 구현 환경을 살펴보고 시공간 질의 처리 시스템의 효용성을 검증하기 위하여 실시한 위치 추적 시스템에 적용한 내용에 대해서 기술한다.

1. 시스템 구현

본 논문에서는 시공간 질의 처리 시스템을 구현하기 위하여 운영체제는 Fedora 4(Linux 2.6.15) 리눅스 환경을 기반으로 하였고, 개발 도구는 리눅스에서 제공하는 g++ 3.2.3 버전을 사용하였다. 클라이언트는 Windows XP Professional 환경을 기반으로 하였고, 개발 도구는 Java 1.5 버전을 사용하였다. 표 4는 시공간 질의 처리를 실험하기 위해 클라이언트에 등록된 스트림 목록을 보여준다.

표 4. 스트림 목록
Table 4. Stream List

이름	타입	속성
Person	스트림	id Int, t Int, location Char(30)

표 4를 살펴보면 Person은 이동체의 정보를 담고 있는 스트림으로서 이동체의 고유 아이디를 나타내는 id, 시간을 나타내는 time, 그리고 위치 좌표를 나타내는 location 속성으로 구성된다. 표 5는 시공간 질의 목록을 보여준다.

표 5. 시공간 질의 목록
Table 5 Spatio-temporal Query List

	예시
시공간 질의	select Person.id, Building.id, ST_Distance(GeomFromText(Building.location, 0), GeomFromText(Person.location, 0)) From Person, Building;

표 5를 살펴보면 시공간 질의는 동일한 시간에 이동체가 건물에서 떨어져 있는 거리를 반환하는 질의이다.

2. 시스템 검증

실시간 위치 추적 시스템은 주로 근거리 및 실내와 같은 제한된 공간에서의 위치 확인 및 위치 추적 시스템이다. 이러한 실시간 위치 추적 시스템을 통해 시공간 질의 처리 시스템의 효용성을 입증하기 위한 시나리오는 다음

과 같다. 사무실 직원과 방문자의 위치 데이터는 실시간에 데이터 스트림으로 입력되며 급작스럽게 발생할 수 있는 회사의 기밀 정보 유출에 빠르게 대응하기 위해 보안구역이 관심의 대상이 된다. 본 논문에서는 실시간으로 입력되는 사무실 직원의 위치 데이터를 모니터링하여 회사의 기밀 정보 유출에 신속하게 대응하는 시나리오를 적용해 보았다. 그림 8은 실시간 위치 추적 시스템의 화면 구성을 보여준다.

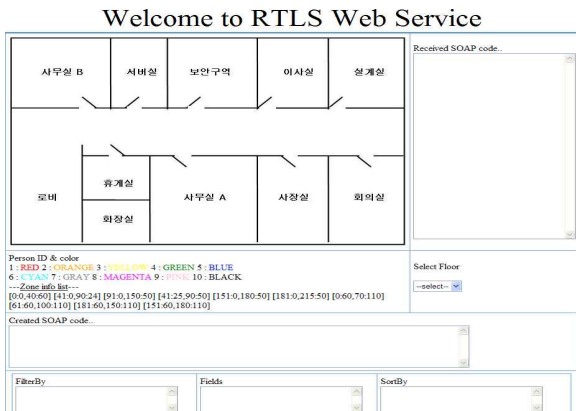


그림 8. 실시간 위치 추적 시스템 화면
Fig 8. Screen of Real-time locating Systems

사용자가 질의를 입력하기 위해 FilterBy 영역을 통해 사무실 영역의 좌하단, 우상단 좌표값을 입력하면 실시간 위치 추적 시스템은 이를 SOAP 메시지로 작성한다. SOAP 메시지는 연속 질의인 CQL로 변환되어 시공간 미들웨어에 전달되며 질의가 처리된 후 반환되는 질의 결과를 Java Applet 영역에 보여주게 된다. 그림 9는 실시간 위치 추적 시스템의 결과 화면을 보여준다.

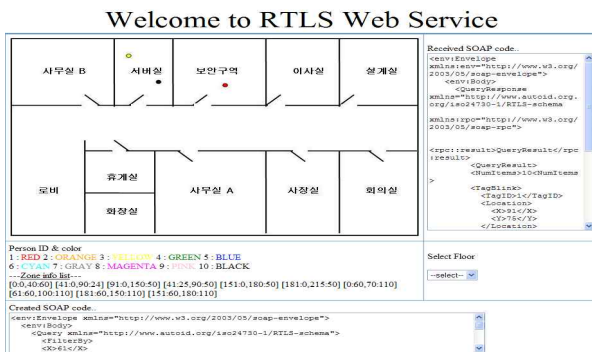


그림 9. 실시간 위치 추적 시스템 결과 화면
Fig 9. Result Screen of Real-time Locating Systems

그림 9를 살펴보면 입력된 좌표에 따른 검색영역이 '서버실'과 '보안구역'이므로 현재 해당 영역에 위치한 이동체들을 화면에 보여주고 있다.

V. 결 론

본 논문에서는 이동 객체의 시공간 데이터 스트림에 대한 효율적인 관리와 검색을 지원하는 시공간 질의 처리 시스템을 설계 및 구현하였다. 시공간 질의 처리 시스템의 시공간 미들웨어는 끊임없이 연속으로 들어오는 시공간 데이터 스트림을 실시간으로 처리하고 시공간 DBMS간의 양방향 동기화를 지원한다. 웹 서버는 서버와 클라이언트간의 상호 운용성을 위하여 SOAP 메시지를 사용하며, 클라이언트의 SOAP 메시지를 분석하여 시공간 미들웨어의 CQL로 변환하는 기능을 지원한다.

참 고 문 헌

- [1] 이충호, 안경환, 이문수, 김주완, "u-GIS 공간정보 기술 동향," 전자통신동향분석, 제22권, 제3호, 110-123쪽, 2007년.
- [2] 정혜선, 정창성, "u-city를 위한 유비쿼터스 컴퓨팅 미들웨어," 인터넷정보학회지, 제7권, 제2호, 38-45쪽, 2006년.
- [3] 원종호, 이미영, 김명준, "유비쿼터스 컴퓨팅 환경을 위한 RFID 기반 센서 데이터 처리 미들웨어 기술 동향," 전자통신동향분석, 제19권, 제5호, 21-29쪽, 2004년.
- [4] ISO/IEC 24730-1, Real Time Locating Systems(RTLS) - Part 1: Application Programming Interface(API), 2003.
- [5] Open Geospatial Consortium Inc., OpenGIS Implementation Specification for Geographic information - Simple feature access - Part 2: SQL option, 2005.

저자 소개

이 기 영(중신회원)



- 제 10 권 1호 참조
- 2009년~현재 한국인터넷방송통신학회 이사
- 1991년~현재 을지대학교 의료IT마케팅학과 교수
- <주관심분야 : u-Healthcare, 공간 데이터베이스, GIS, LBS, USN, 텔레매틱스 등>

임 명 재(중신회원) : 교신저자



- 제 10 권 2호 참조
- 2009년~현재 한국인터넷방송통신학회 이사
- 1992년~현재 을지대학교 의료IT마케팅학과 교수
- <주관심분야 : SE개발방법,, CBD방법론, HCI, u-Healthcare 등>

김 규 호(정회원)



- 제 10 권 2호 참조
- 1992년~현재 : 을지대학교 의료IT마케팅학과 교수
- 2007년~현재 을지대학교 ERIC(지역혁신센터) 부소장
- <주관심분야 : u-Healthcare, 유비쿼터스, USN 등>

김 정 준(정회원)



- 제 9 권 5호 참조
- 2005년~현재 건국대학교 대학원 컴퓨터공학과 박사 수료
- 2010년 9월 건국대학교 대학원 컴퓨터공학과 박사 취득예정
- <주관심분야 : 공간 MM 데이터베이스, GIS, LBS, USN, 텔레매틱스 등>