모바일 환경에서 시공간 데이터의 동기화를 위한 시간 모델 설계

김홍기, 조대수
동서대학교

A Design of Time Model for Synchronization of sptio-temporal data on a Mobile Environments

Hong-Ki Kim, Dae-Soo Cho
Dongseo University
E-mail : inthestream@nate.com, dscho@dongseo.ac.kr

요약

모바일 환경의 동기화 시스템에서 시공간 데이터들의 시간 정보는 여러 종류 시간 데이터가 적용될 수 있다. 그러나 시공간 데이터의 시간 데이터를 데이터 등록 시간을 기준으로 하여 데이터를 저장하면 동기화 작업의 진행에 따라 과거의 데이터를 동기화하는 문제가 발생 한다. 이 논문에서는 시공간 데이터를 위해 시간 데이터로 사용될 수 있는 시간들을 분류 한다. 그리고 시공간 데이터의 동기화에 적합한 시간 데이터의 모델을 정의하고 이를 이용한 동기화 시스템을 제안한다.

ABSTRACT

The time information of sptio-temporal data in synchronization system of a mobile environment can be applied according to time data of various kinds. However, that previous data is synchronized related with progress of synchronization operation is a problem when saving the time data of sptio-temporal data depends on the registration time. In this paper, classifies the time used as time data for synchronization data, defines time data model suitable for synchronization of sptio-temporal data and suggests a synchronization system which uses time data model.

키워드

동기화, 시공간 데이터, 모바일, 시간 모델

Ⅰ. 서 론

유비쿼터스 환경의 GIS 서비스는 모바일 기기를 이용하여 언제 어디서나 사용자에게 공간정보를 제공한다[1]. 모바일 기기는 서버와 항상 연결되어 서비스될 수 없기 때문에 서버의 시공간 데이터를 복사하여 단절된 상태에서도 복사된 시공간 데이터를 이용하여 서비스를 제공하고 서버와 단절된 기간 동안 서버나 모바일 기기에서 변경한 데이터가 있으면 서버와 연결되는 시간에 동기화를 통하여 시공간 데이터의 일관성을 유지한다. 그러나 모바일 기기는 서버와 연결되어 있는 경우보다 단절되어 작업되는 시간이 많다. 모바일 기기가 단절 상태에서 변경한 시공간 데이터를 동기화할 때 상황에 따라 동기화하는 데이터가 최신의 데이터라고 아닌 경우가 발생한다.

예를 들어 시공간 객체 O_1에 대하여 모바일 클라이언트 M_1과 M_2가 각각 T_1시간과 T_2시간에 O_1에 대하여 변경을 하였다고 가정하자. M_2는 T_3시간에 동기화를 하고 M_1은 T_4시간에 동기화를 하였다. T_3시간의 동기화로 인해 서버는 O_1에 대한 최신 데이터로 T_2시간에 변경된 데이터를 가지게 된다. 이후 T_4시간에 동기화 하는 M_1은 O_1에 대하여 과거 데이터인 T_1시간의 데이터를 동기화하는 문제가 발생한다. 이러한 문제의 해결을 위해 단절 상태에서 변경된 시공간 데이터의 동기화를 위한 신뢰할 수 있는 시간 모델이 필요하다.

이 논문에서는 모바일 클라이언트와 서버의 양방향 동기화 작업의 시간 모델을 제안하고, 제안한 시간 모델에서 신뢰할 수 있는 시간 데이터를 분류하고 이를 이용한 동기화 기법을 제시한다.

이 논문의 구성은 다음과 같다. 2장에서는 관련

* 본 연구는 국토해양부 첨단도시기술개발사업 - 지능형국토정보기술혁신 사업과제의 연구비지원(07국토정보C05)에 의해 수행 되었습니다.

연구를 기술하고, 3장에서는 시공간 데이터의 동기화를 위한 시간 모델을 설명한다. 4장에서는 제안한 시간 모델을 이용한 동기화 기법을 소개한다. 마지막으로 5장에서 결론 및 향후 연구에 대하여 기술한다.

II. 관련 연구

ActMAP 시스템[2]은 유럽 자동차 회사들과 디지털 지도 제작 업체와의 프로젝트 팀에 의해 2007년도에 제안된 네비게이션 지원 시스템으로 자동차에 탑재된 네비게이션 기기에 무선 네트워크를 이용하여 지도의 최신 변경사항을 반영할 수 있도록 지원하는 시스템이다. ActMAP은 유선에 비해 대역폭의 제한을 많이 받는 무선 네트워크상에서 서버와 모바일 단말기사이에서 고용량 지리정보의 효율적인 동기화를 위하여 부분 점진적 갱신(Partial Incremental Update)을 지원하며, 부분 점진적 갱신을 위한 방법을 제시한다. ActMAP은 실세계의 변화된 데이터를 즉각적으로 업데이트 하기위한 시간 모델을 제안한다. ActMAP에서 제안하는 시간 모델은 실세계에서 변화가 발생한 시점을 시작으로 실세계의 변화가 끝나는 시점 그리고 ActMAP에서 업데이트를 제공하는 시점과 제공을 중단하는 시점에 대한 시간 모델을 라이프 사이클을 정의한다. 또한 ActMAP은 실세계의 변화가 등록되기까지의 시간과 등록된 실세계의 변화가 사용자에게 제공되기 위해 처리되는 시간, 처리된 변화가 사용자에게 제공 가능한 시점, 실제 사용자에게 제공되기까지의 시간, 사용자에게 업데이트가 제공되는 시점을 정의한 시간 모델을 제공한다.

마지막으로 실세계의 변화가 사라졌는지에 대한 확인과 실세계의 변화가 사라졌을 때 업데이트되는 작업에 대한 시간 모델을 정의하고 있다.

III. 시공간 데이터 동기화 시간 모델

이 장에서 사용되는 시간 모델의 용어는 다음과 같다.

- Temporary registration time
  - 실세계의 변화를 클라이언트가 수정, 등록 시점
- Registration delay
  - 실세계에서 발생한 변화가 클라이언트 또는 서버의 시공간DB에 등록되기까지의 기간
- Processing delay
  - 실세계의 변화 또는 클라이언트의 수정사항을 서버의 시공간DB에 갱신하거나 Delta reCord생성까지의 기간.
- Availability time
  - 클라이언트에게 제공될 Delta reCord가 생성된 시점
- Available delay
  - 업데이트가 서버에서 등록된 후 클라이언트가 접속하기 전까지의 기간
- Provision Time
  - 클라이언트에게 Delta reCord Set이 제공되는 시점.
- Provision delay
  - Availability time에서 클라이언트가 서버로부터 업데이트를 전송 받기까지의 기간

모바일 클라이언트-서버 환경에서 양방향 동기화 시스템[3][4][5]은 아래 그림 1,2,3과 같이 3가지 형태의 동기화 작업이 발생한다.

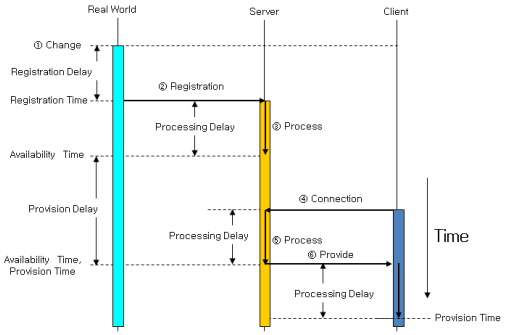

그림 1 서버->클라이언트 단방향 동기화

그림 1은 실세계의 변경이 서버에 직접 갱신되는 경우의 동기화 진행 과정이다. 그림 1에서 시공간 데이터에 사용될 수 있는 시간데이터는 다음과 같다.

- 실세계의 변경이 발생한 시간
- 실세계의 변경을 서버에 등록하는 시간
- 실세계의 변경을 서버에 등록을 완료하는 시간
- 클라이언트가 서버에 접속하여 동기화를 시작하는 시간
- 클라이언트와 동기화를 완료한 시간

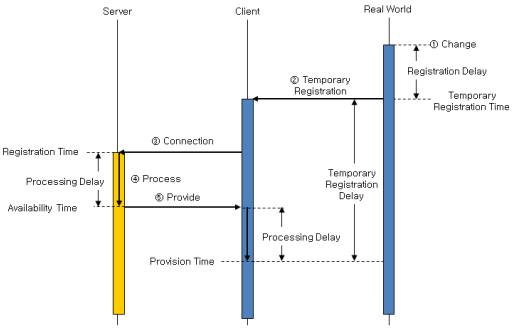

그림 2 클라이언트->서버 단방향 동기화

그림 2는 실세계의 변경을 모바일 클라이언트에서 갱신하는 경우의 동기화 진행 과정이다. 그림 2에서 시공간 데이터에 사용될 수 있는 시간데이터는 다음과 같다.

- 실세계의 변경이 발생한 시간
- 실세계의 변경을 모바일 클라이언트에 등록을 시작하는 시간
- 실세계의 변경이 모바일 클라이언트에 등록을 완료한 시간
- 모바일 클라이언트에서 등록한 변경을 서버에 접속하여 동기화를 시작하는 시간
- 모바일 클라이언트에서 등록한 변경에 대하

여 서버와 동기화를 완료한 시간

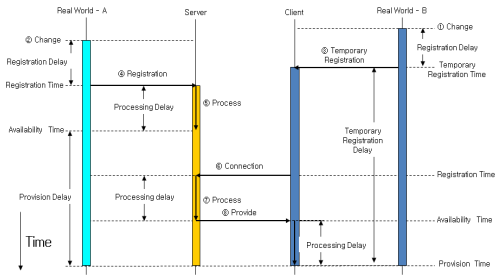

그림 3 양방향 동기화

그림 3은 모바일 클라이언트가 서버와 단절되어 있는 동안 서버에도 시공간 데이터의 변경이 발생하고, 모바일 클라이언트에도 발생한 경우이다. 그림 3에서 시공간 데이터에 사용될 수 있는 시간데이터는 다음과 같다.

- 실세계의 변경이 발생한 시간
- 실세계의 변경을 서버에 등록을 시작하는 시간
- 실세계의 변경을 서버에 등록을 완료한 시간
- 실세계의 변경을 모바일 클라이언트에 등록을 시작하는 시간
- 모바일 클라이언트에서 등록한 변경을 서버에 접속하여 등록을 시작하는 시간(동기화 시작 시간)
- 모바일 클라이언트에서 등록한 변경을 서버에 등록을 완료하는 시간(클라이언트의 변경을 동기화 완료한 시간)
- 서버에 등록된 변경을 모바일 클라이언트로 동기화를 완료한 시간

그림 1,2,3을 통해 분류된 시공 데이터의 적용 가능성의 분석결과는 다음과 같다.

적용 시 문제가 발생하는 시간 데이터:

- 실세계의 변경이 발생한 시간 : 적확한 변경 시간을 알 수 없다.
- 실세계의 변경을 서버에 등록을 시작하는 시간 : 등록을 처리하는 과정에서 시스템의 오류 및 에러 발생으로 변경될 수 있다.
- 실세계의 변경을 모바일 클라이언트에 등록을 시작하는 시간 : 서버와 단절된 상태에서 모바일 클라이언트의 시스템 시간이 서버의 시스템 시간과 다른 시간 값을 가지는 경우가 발생할 수 있다.
- 모바일 클라이언트에서 등록한 변경을 서버에 접속하여 등록을 시작하는 시간(동기화 시작 시간) : 등록을 처리하는 과정에서 시스템의 오류 및 에러 발생으로 변경될 수 있다.

적용 가능한 시간 데이터:

- 실세계의 변경을 서버에 등록을 완료한 시간
- 모바일 클라이언트에서 등록한 변경을 서버에 등록을 완료한 시간(클라이언트의 변경을 동기화 완료한 시간)
- 서버에 등록된 변경을 모바일 클라이언트로 동기화를 완료한 시간

적용 가능한 시간 데이터를 간단히 정의하면, 서버의 시공간 DB에 변경 데이터가 갱신 완료되는 시간이다.

Ⅳ. 시간 모델을 이용한 동기화 기법

4.1 동기화를 위한 타임스탬프

이장에서는 모바일 환경에서 시공간 데이터를 동기화할 수 있는 타임스탬프를 정의하고 이를 이용한 동기화 기법을 제안한다. 제안한 동기화 기법은 타임스탬프는 다음과 같이 정의된다.

- 최종 갱신 시간(Last Update Time, LUT) : 서버 시공간 데이터 분할 영역에 포함된 시공간 객체들 중 마지막으로 갱신된 객체의 시간 데이터(실세계의 변경을 서버에 등록을 완료한 시간, 모바일 클라이언트에서 등록한 변경을 서버에 등록을 완료한 시간). 각 분할 영역들은 하나의 LUT를 가진다.

- 최종 동기화 시간(Last Sync Time, LST) : 모바일 클라이언트 시공간 데이터의 각 분할 영역이 서버와 마지막으로 동기화를 완료한 시간(서버에 등록된 변경을 모바일 클라이언트로 동기화를 완료한 시간). 각 분할 영역들은 하나의 LST를 가지며 동기화를 통해 갱신되는 시공간 데이터가 없어도 LST는 마지막으로 동기화한 시간을 가진다.

위에서 정의한 타임스탬프는 모바일 클라이언트-서버간 동기화를 진행할 때 모바일 클라이언트에서 변경한 시공간 데이터의 변경충돌을 검사에 이용된다. 또한 서버에서 모바일 클라이언트로 동기화를 해야 하는 시공간 데이터와 클라이언트에서 서버로 동기화를 해야 하는 시공간 데이터를 정의하는데 사용한다.

타임스탬프를 이용한 동기화 데이터의 분류는 다음과 같다.

- 모바일 클라이언트 : 모바일 클라이언트의 각 분할영역의 LST보다 시간데이터의 값이 큰 시공간 데이터들을 서버로 동기화할 데이터로 분류한다.
- 서버 : 동기화를 요청한 모바일 클라이언트의 각 분할영역의 LST보다 LUT가 큰 영역에 대하

여 각 영역에 대하여 LST보다 시간데이터의 값이 큰 시공간 데이터들을 모바일 클라이언트로 동기화할 데이터로 분류한다.

4.2 타임스탬프를 이용한 변경충돌 검사

모바일 환경의 클라이언트-서버 양방향 동기화는 분산된 다수의 클라이언트로부터 수집 변경되는 시공간 데이터의 일관성을 유지해야한다. 이를 위해 서버는 동기화를 요청한 모바일 클라이언트의 변경 데이터에 대하여 변경충돌 검사가 필요하다.

이 논문에서는 타임스탬프를 이용한 변경충돌 기법을 제시한다.

변경충돌 검사 과정:

① 변경 데이터가 존재하는 모바일 클라이언트의 분할영역의 LST와 서버의 LUT 비교한다.

② LUT의 시간이 크면 변경충돌이 발생 가능한 영역으로 정의한다. LST가 큰 경우는 변경충돌이 없는 영역으로 정의하고, 다른 영역에 대하여 변경충돌 검사를 시작한다.

③ 변경충돌이 발생 가능한 영역은 서버의 영역에 대하여 LST보다 큰 시간데이터를 가진 객체를 추출한다.

④ 서버에서 추출된 객체와 클라이언트의 변경 객체와 동일한 객체가 있는지 비교한다.

⑤ 동일한 객체가 있으면 변경충돌 발생으로 동기화 작업을 취소한다. 없는 경우는 다른 영역에 대하여 변경충돌 검사를 시작한다.

⑥ 변경데이터가 있는 모든 클라이언트의 분할 영역에 대하여 동일한 작업을 반복진행.

타임스탬프를 이용한 변경충돌 검사는 대용량의 시공간 데이터에 대하여 변경 객체를 모두 검사하는 경우보다 검사 시간을 단축할 수 있다.

Ⅴ. 결론

이 논문에서는 모바일 클라이언트-서버 환경에서 시공간 데이터의 양방향 동기화 작업에 대한 시간 모델을 정의한다. 정의된 시간 모델에서 시공간 데이터의 양방향 동시화 시 신뢰할 수 있는 시간 데이터로 트랜잭션 완료 시간을 적용하였다. 또한 양방향 동기화를 위한 타임스탬프를 정의하고 이를 이용하여 변경충돌 검사 및 클라이언트-서버간 동기화 작업 시 동기화할 데이터를 구분한다. 향후, 트랜잭션 완료 시간뿐 아니라 실세계의 변경시간 및 등록 시간을 이용하여 변경충돌이 발생한 동기화 데이터의 작업을 무조건 취소만 하는 것이 아니라 변경충돌의 해결에 관한 연구가 필요하다.

참고문헌

[1] 이은경, 하원규, "유비쿼터스 컴퓨팅 비전과 주요국의 연구 동향", 전자통신 동향분석, 제 17권, 제 6호, 2002년 12월

[2] "ActMAP White Paper and Interfaces to the FeedMAP framework," white paper, 2007

[3] 김홍기, 임창우, 이상신, 조대수, 김동현 "모바일 GIS DB를 위한 양방향 동기화 프로토콜 설계," 한국해양정보통신학회, 한국해양정보통신학회 춘계종합학술대회, 2008년도 춘계종합학술대회 Vol. 12 No. 1, 2008. 5, pp. 183~186.

[4] 이혜진, 김진석, "모바일 환경에서 공간데이터 동기화 시스템 설계," 한국정보과학회, 2004년 가을 학술발표논문집 제31권 제2호 (Ⅱ), 2004. 10, pp. 184~186

[5] 최우영, 이경아, 염태진, 진성일 "내장형 DBMS를 위한 동기화 서버 시스템," 한국정보과학회, 한국정보과학회 학술발표논문집 한국정보과학회 2004년도 봄 학술발표논문집 제31권 제1호(B), 2004. 4, pp. 127 ~ 129.