

다차원 GIS에서 과거데이터의 무결성을 위한 연산의 설계

박 동 선[†]·김 재 홍^{††}·배 해 영^{†††}

요 약

지난 90년대 초반부터 시간차원을 포함하는 다차원 지리정보시스템에 대한 연구가 활발히 진행되고 있으며, 디수의 시간 지원 공간 연산과 시간지원 공간 질의어에 대한 연구 결과가 발표되었다. 그러나, 이들 연구들은 대부분 현재시점에 유효한 데이터를 대상으로 하는 연산에 치중하고 있으며, 과거데이터의 오류를 수정하기 위한 연산에 대한 연구는 미비하다. 본 논문에서는 시간을 지원하는 다차원 GIS의 과거데이터에서 발생할 수 있는 오류의 수정이나, 누락된 정보를 삽입하여 과거데이터의 무결성을 유지할 수 있는 연산들을 설계하며, 설계하는 연산이 효율적으로 수행될 수 있는 새로운 데이터모델을 개입한다.

Design of Operation for Integrity of Past data in Multi-dimensional GIS

Dong-Seon Park[†] · Jae-Hong Kim^{††} · Hae-Young Bae^{†††}

ABSTRACT

In the last decade, considerable research effort has been studied to multi-dimensional GIS and many spatiotemporal operations and query languages have been proposed. Many of them have focused on designing of operation, which process with data that is valid at current point in time. But, research that studied to update operation that updates or inserts the past data is very rare. This paper designs an update operation and an insertion operation for integrity of past data of multi-dimensional GIS, and proposes a new data model for efficient processing of the operations.

1. 서 론

지난 90년대 초반부터 시간차원을 포함하는 다차원 GIS에 대한 연구가 진행되고 있으며, 특히 시간차원을 포함하는 다차원 데이터모델[1-3]과 시간지원 공간 연산 및 질의어에 대한 연구[4, 5]가 활발히 진행중이다. 그러나 이들 연구에서 지원하는 갱신 연산들은 대부분

현재시점에 유효한 데이터들을 대상으로 하는 연산과 질의어에 치중하고 있으며, 잘못 입력된 과거데이터의 수정이나, 누락되어 데이터베이스에 입력되지 않은 과거데이터의 삽입을 수행할 수 있는 방법에 대한 연구는 문자기반 데이터베이스 시스템에서는 연구되었으나 [6-8] 시간지원 GIS시스템에서는 이에 대한 연구가 미비하다.

본 논문에서는 기존의 연구에 대한 문제점을 알아보고, 시간차원을 포함하는 다차원 GIS에서 과거데이터의 오류를 수정할 수 있는 연산과 누락된 과거데이터를 삽입할 수 있는 연산을 설계하여 과거데이터의 무

※ 본 연구는 정보통신부의 대학 S/W연구센터 지원사업에 의하여 수행되었음
† 준 회원 인하대학교 대학원 전자계산공학과 박사과정
†† 정 회원 영동대학교 컴퓨터공학과 교수
††† 중신회원 인하대학교 전자계산공학과 교수
논문접수 : 2000년 4월 17일, 심사완료 : 2000년 6월 19일

결성을 보장할 수 있도록 하며, 설계한 연산들이 효율적으로 수행될 수 있는 데이터모델을 설계한다. 본 논문에서 설계하는 데이터모델은 [9]에서 설계한 시간지원 다차원 지라정보 데이터모델을 확장하여 과거데이터에 대한 연산을 지원할 수 있도록 한다.

본 논문의 구성은 다음과 같다. 2장에서는 기존의 시간지원 시스템에서 지원하는 연산의 특징 및 본 연구의 필요성을 살펴보고, 3장에서는 과거데이터에서 발생할 수 있는 연산들을 정의한다. 4장에서는 3장에서 정의한 연산의 설계와 이들을 효율적으로 지원할 수 있는 데이터모델을 설계하고, 마지막 5장에서는 결론과 함께 향후 연구과제를 제시한다.

2. 연구 배경 및 관련 연구

2.1 연구 배경

시간지원 데이터베이스에서 거래시간 이외의 대부분의 데이터들이 데이터베이스 사용자들에 의해서 입력 작업이 수행되기 때문에 사용자의 실수를 비롯한 여러 가지 요인에 의하여 그 데이터를 입력하는 시점에서 잘못된 정보가 입력되거나 누락된 데이터가 존재할 가능성은 언제나 존재한다. 따라서 향후 과거데이터에 존재하는 오류나 누락된 과거데이터가 발견되었을 때 오류를 수정하거나 누락된 데이터를 과거 데이터베이스에 추가할 수 있는 연산을 수행하여 과거데이터의 무결성을 유지하여야 한다.

객체_ID	공간정보	Vs	Ve	Ts	Te
100	...	5	10	5	10
100	...	10	15	10	15
100	...	15	20	15	20
100	...	20	now	20	now

(a) 예제 테이블

객체_ID	공간정보	Vs	Ve	Ts	Te
100	...	5	8	5	10
100	...	8	13	10	15
100	...	13	20	15	20
100	...	20	now	20	now

(b) 유효시간 수정후의 테이블

객체_ID	공간정보	Vs	Ve	Ts	Te
100	...	5	10	5	10
100	...	10	14	10	15
*100	...	14	17	10	15
100	...	17	20	15	20
100	...	20	now	20	now

(c) 누락된 튜플 삽입후의 테이블

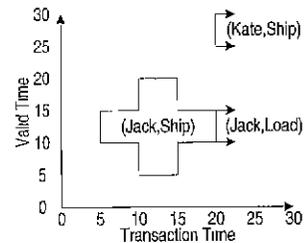
(그림 1) 오류수정과 누락삽입의 예

(그림 1)은 시간지원 시스템에서 발생할 수 있는 오

류 및 누락을 설명하기 위한 그림이다. 오류에 의한 수정은 '두 번째 튜플의 유효시간 범위가 (8-13)인데 사용자의 실수로 인하여 (10-15)로 입력되었다'와 같은 경우에 필요하며, 실제계에 존재하는 공간객체는 유효시간의 범위에 공백이 발생할 수 없기 때문에 (그림 1(a))의 두 번째 튜플의 유효시간을 수정할 경우 첫 번째 튜플과 세 번째 튜플의 유효시간범위도 수정하여야 한다. 또한, 누락된 정보의 삽입은 '유효시간의 범위가 (14-17)인 정보가 데이터베이스에 저장되지 않았다'와 같은 경우에 필요하며, 실제계의 공간객체는 동일한 시간에 서로 다른 정보를 가질 수 없기 때문에 두 번째 튜플과 세 번째 튜플의 유효시간도 수정하여야 한다.

2.2 관련 연구

Snodgrass는 [6]에서 오류수정에 대하여 언급하고 있으며, 여섯 가지의 시간지원 데이터모델에 적용한 데이터의 표현 방법을 보여주고 있다. (그림 2)는 Snodgrass가 오류수정 연산의 필요성을 설명하기 위하여 제시한 예이다.



①

Emp	Dept	Ts	Te	Vs	Ve
Jake	Ship	5	UC	10	15

②

Emp	Dept	Ts	Te	Vs	Ve
Jake	Ship	5	9	10	15
Jake	Ship	10	UC	5	20

③

Emp	Dept	Ts	Te	Vs	Ve
Jake	Ship	5	9	10	15
Jake	Ship	10	14	5	20
Jake	Ship	15	UC	10	15

④

Emp	Dept	Ts	Te	Vs	Ve
Jake	Ship	5	9	10	15
Jake	Ship	10	14	5	20
Jake	Ship	15	19	10	15
Jake	Load	20	UC	10	15

■ UC Until Changed

(그림 2) Snodgrass의 오류수정 예

2) 논문에서의 표현을 간결하게 하기 위하여 향후 '오류수정'이나 '오류수정연산'으로 표기한다.

3) 향후 '누락삽입'이나 '누락삽입연산'으로 표기한다.

Snodgrass가 언급한 오류수정연산의 대상 데이터들은 유효시간의 관점에서는 비록 과거데이터이긴 하지만, 거래시간의 관점에서는 아직 유효한 데이터들을 연산의 대상으로 하고 있다. 즉, 수정하는 데이터에 의하여 영향을 받는 다른 튜플이 존재하지 않으며, 이는 단순한 논리 삭제 연산에 의하여 수행될 수 있다 또한, (그림 1)과 같은 경우의 오류수정과 누락삽입에 대해서는 언급하지 않고 있다

[7]에서는 (그림 3)과 같이 주식과 관련된 데이터를 이용하여 오류를 수정하는 예를 보여주고 있다.

OWN

Client	Stock	Share	From	To
Melanie	DEC	20	7-15-1992 3 00 PM	forever

range of O is OWN

replace O("Melanie", "DEC", O Shares)

valid during [12PM July 15, 1992, 3PM July 15, 1992]

where O.Client = "Melanie" and O.Stock = "DEC"

when O overlap [3PM July 15, 1992]

(그림 3) Snodgrass의 TQUEL을 이용한 과거데이터의 갱신

[7]에서는 자세한 수정 연산의 수행방법에 대하여 언급하지 않고 있으며, 연산 대상의 데이터들이 (그림 1)에서 설명한 데이터들과는 다른 특징을 가지고 있다. 즉, (그림 1)에서는 어느 객체의 정보에 대한 유효시간이 중복되거나 공백이 발생할 수 없지만, (그림 3)의 데이터들은 유효시간의 중복과 공백이 발생할 수 있으며, 수정되는 데이터에 의하여 영향을 받는 다른 튜플의 수정에 대하여 언급하지 않고 있다.

Degree

name	degree	valid time	transaction time
Jim	PhD	(85,now)	(90,until_changed)

Position

name	position	valid time	transaction time
Jim	Professor	(91,now)	(91,until_changed)

(a) 수정 전의 테이블

Degree

name	degree	valid time	transaction time
Jim	PhD	(85,92)	(90,92)
Jim	Master	(85,now)	(92,until_changed)

Position

name	position	valid time	transaction time
Jim	Professor	(91,now)	(91,until_changed)

(b) 수정 후의 테이블

(그림 4) [8]의 과거데이터 수정 예

[8]에서는 비전모델을 이용한 시간지원 데이터베이스

스에서 과거데이터의 수정에 대하여 언급하고 있다 (그림 4)에서는 과거데이터를 수정하였을 경우 '박사학위를 가진 사람만 교수가 될 수 있다'라는 무결성을 위배하는 결과를 보여준다.

이와 같이 [8]은 과거데이터의 변경에 의하여 데이터베이스에서 정의한 무결성이 위배되는 경우를 처리하기 위한 방법과, 실제세계에서 발생한 데이터의 변경에 의하여 생성된 버전간의 일관성을 유지하기 위한 무결성 규칙을 관리하는 방법에 대하여 언급하고 있으며, 이는 본 논문에서 다루는 과거데이터의 무결성과는 다른 개념이다.

2.3 MBNF기법을 이용한 현재 및 과거데이터의 저장

본 절에서는 [9]에서 제안한 분리된 저장구조 및 MBNF기법에 대하여 간략히 소개한다

[9]에서는 객체 정보의 유효 시간을 기준으로 현재 시점에 유효한 데이터는 현재 데이터베이스에 저장하고, 과거시점에 유효한 데이터는 과거 데이터베이스에 저장하도록 하는 분리된 저장구조를 설계하였으며, 과거데이터의 중복을 최소화 하고, 현재데이터 및 과거데이터에 대한 검색 연산을 빠르게 수행할 수 있는 저장기법인 MBNF기법을 설계하였다.

현재 데이터베이스는 공간객체의 상태를 표현하는 모든 공간정보를 저장하는 현재데이터블로 구성되며, 과거 데이터베이스는 공간 객체의 상태를 표현하는 데이터들을 저장하는 과거데이터블과, 각 공간 객체의 기준상태(base-state)에 대한 정보를 저장하는 기준상태 데이터블로 구성된다. 과거데이터블에 저장되는 과거 데이터는 기준상태와 비교하여 추가되거나 삭제된 데이터를 저장하여 데이터베이스의 크기를 줄일 수 있도록 하였다.

3. 과거데이터의 무결성 보장을 위한 연산의 정의

본 장에서는 2장에서 살펴본 오류수정과 누락삽입을 수행하기 위한 오류수정연산과 누락삽입연산을 정의한다

본 논문에서 사용되는 용어의 의미는 다음과 같다. t_n , t_{n-1} 는 과거 데이터베이스나 현재 데이터베이스에 저장된 튜플이고, t_n , t_{n-1} 는 튜플 t_n 에 저장된 유효시작시간을 의미하며, t_n , t_{n-1} 는 튜플 t_n 에 저장된 유효종료시간을 의미한다.

- 선행시간인접(PTN : preceding temporally neighboring)류플과 후행시간인접(STN : succeeding temporally neighboring)류플 : $t_1.ve = t_2.vs$ 이면 류플 t_1 은 류플 t_2 의 PTN류플이고, 류플 t_2 는 류플 t_1 의 STN류플이다.
- 시간선행(TP : temporally preceding)류플과 시간후행(TS : temporally succeeding)류플 : $t_1.ve < t_2.vs$ 이면 류플 t_1 은 류플 t_2 의 TP류플이고, 류플 t_2 는 류플 t_1 의 TS류플이다.

(그림 5)는 [9]에서 설계한 분리된 저장구조를 이용하여 저장한 현재 테이블과 과거 테이블의 예이며, 표현을 단순히 하기 위하여 시간을 정수로 표현하였다. 또한 류플 번호는 설명을 위한 번호이다

O_ID	Node	VS	TS
100	...	27	27

(a) 현재 데이터베이스, 현재 테이블

No.	O_ID	UNode	DNode	VS	VE	TS	TE
1	100	3	5	2	5
2	100	5	11	5	11
3	100	11	16	11	16
4	100	16	22	16	22
5	100	22	27	22	27

(b) 과거 데이터베이스, 과거 테이블

(그림 5) 분리된 저장구조를 이용하여 저장한 데이터베이스의 예

3.1 오류수정연산

3.1.1 유효시간의 수정

시간 차원을 포함하는 공간객체의 정보는 (그림 5)와 같이 유효시간을 기준으로 여러 개의 류플에 분산되어 저장되며, 이들은 유효시간의 순서에 의하여 논리적으로 연속된 류플에 저장되었다고 할 수 있다. 따라서 어느 한 류플의 유효시간을 수정할 경우, 이 류플의 PTN(또는 STN)류플과 TP(또는 TS)류플의 유효시간도 수정되어야 한다. 유효시간의 수정은 유효시작시간의 수정과 유효종료시간이 수정되는 경우가 있으며, 각 경우에 발생할 수 있는 유효시간 수정은 다음과 같다.

• 유효시작시간의 수정

- ① 수정하는 류플의 유효시작시간과 함께 PTN류플의 유효종료시간을 수정해야 하는 경우.

- ② 수정하는 류플의 유효시작시간과 함께 TP류플의 유효종료시간을 수정해야 하는 경우
- ③ 수정하는 류플의 유효시작시간이 과거 데이터베이스에 최초로 저장된 류플의 유효시작시간보다 앞서는 경우
- ④ 수정하는 류플의 PTN류플이 없는 경우

• 유효종료시간의 수정

- ⑤ 수정하는 류플의 유효종료시간과 함께 STN류플의 유효시작시간을 수정해야 하는 경우.
- ⑥ 수정하는 류플의 유효종료시간과 함께 TS류플의 유효시작시간을 수정해야 하는 경우.
- ⑦ STN류플이나 TS류플이 현재테이블에 있는 경우.

(그림 6)과 (그림 7)은 유효시간 수정의 예이며 수정연산이 수행된 시스템 시간은 30으로 가정하였다. (그림 6)은 PTN(또는 STN)류플과 TP(또는 TS)류플이 과거테이블에 존재하는 경우로 각각 류플-4의 유효시작시간과 류플-2의 유효종료시간이 수정된 예이다.

No	O_ID	UNode	DNode	VS	VE	TS	TE
1	100	3	5	2	5
2	100	5	11	5	11
3	100	11	16	11	16
4	100	16	22	16	22
5	100	22	27	22	27
6	100	11	13	30	30
7	100	13	22	30	30

(a) 유효시작시간의 수정-①

No.	O_ID	UNode	DNode	VS	VE	TS	TE
1	100	3	5	2	5
2	100	5	11	5	11
3	100	11	16	11	16
4	100	16	22	16	22
5	100	22	27	22	27
6	100	5	18	30	30
7	100	18	22	30	30

(a) 유효종료시간의 수정-⑥

- 유효시간이 수정되는 류플
- PTN 류플
- TS 류플

(그림 6) 유효시간의 수정

(그림 7)은 STN류플이나 TS류플이 현재테이블에 존재하는 경우로 류플-2의 유효종료시간이 수정된 예이다.

No.	O_ID	UNode	DNode	VS	VE	TS	TE
1	100	3	5	2	5
2	100	5	11	5	11
3	100	11	16	11	16
4	100	16	22	16	22
5	100	22	27	22	27
6	100	5	28	30	30
7	100	27	28	27	30

(a) 과거 테이블

O_ID	Node	VS	TS
100	...	27	27
100	...	28	30

(b) 현재 테이블

□ 유효시간이 수정되는 튜플 ▨ 시간후행튜플

(그림 7) 유효 종료시간의 수정-①

(그림 6(a))의 튜플-3,4와 (그림 6(b))의 튜플-2,3,4 그리고 (그림 7(a))의 튜플-2,3,4,5와 튜플-7은 모두 실제의 공간객체 상태를 표현하는 정보로서의 가치는 상실하지만 시스템에서 수행된 작업에 대한 이력정보로서의 가치를 지니고 있기 때문에 이들 튜플에 대한 물리적인 삭제의 허용 여부는 시스템의 응용분야에 따라서 결정되어야 할 것이다. 즉, 시스템에서 발생한 작업에 대한 이력정보를 유지할 필요성이 있는 분야에서는 물리적인 삭제는 허용하지 않고 논리적인 삭제만 허용할 수 있다. 또한 거래시간을 기록하지 않는 데이터에서 수정연산이 발생할 경우는 4장에서 설계한 알고리즘의 논리적인 삭제가 물리적인 삭제로 대체될 수 있다 단, 본 논문에서 사용하는 과거데이터에는 MBNF 기법을 이용하여 공간정보를 저장하기 때문에 논리적으로 삭제되는 튜플이 기준상태일 경우는 물리적인 삭제는 허용하지 않는다.

기존의 시간지원 시스템에서 논리적으로 삭제되는 튜플은 객체의 상태를 표현하는 정보로서의 가치를 가지고 있지만, (그림 6)과 (그림 7)에서 논리적으로 삭제되는 튜플은 객체의 상태를 표현하는 정보로서의 가치를 상실하게 된다. 또한 삭제 대상 튜플이 기존의 시간지원 시스템에서는 삭제 연산이 수행되는 시점에 유효한 튜플이기 때문에 단지 거래종료시간을 명확히 함으로써 논리적으로 삭제되었음을 표현할 수 있지만, (그림 6)과 (그림 7)에서의 삭제 대상 튜플은 이미 거래종료시간이 명확하게 입력되어 있는 튜플들이다. 따라서 이들 튜플이 또 다시 논리적으로 삭제되었음을 표현하기 위한 방법이 필요하며 이에 대해서는 4장에서 연구한다.

3.1.2 공간정보의 수정

[9]에서는 과거데이터에 저장되는 공간정보들은 기준상태와 비교하여 변경된 데이터만 저장하도록 하였다. 따라서 과거데이터에 저장되어 있는 튜플의 공간정보를 수정할 경우 수정되는 튜플과 연관된 기준상태 정보를 고려한 수정 연산의 설계가 필요하다.

(그림 8)은 MBNF기법을 이용하여 저장한 과거 데이터베이스의 예이며, 표현을 간결하게 하기 위하여 공간정보와 시간값을 정수로 표현하였다. (그림 8(a))에서 튜플-1,4는 기준상태로써 모든 공간 정보가 UNode에 저장되어 있으며, 나머지 튜플들은 기준상태와 비교하여 변경된 정보가 각각 UNode와 DNode에 저장되어 있다

No.	O_ID	UNode	DNode	VS	VE	TS	TE
1	100	1,2,3,4,5,6,7,8,9,10,11		1	3	1	3
2	100	12	2	3	8	3	8
3	100	13,14	2,3	8	14	8	14
4	100	1,13,15,16,9,10,17		14	20	14	20
5	100	18	16	20	23	20	23

(a) 과거 테이블

O_ID	VS	TS	VS	TS
100	1	14	1	14
100	14	now	14	now

(b) 기준상태 테이블

(그림 8) MBNF기법을 이용한 과거 데이터베이스의 예

공간정보의 수정은 다음과 같이 두 가지 경우로 분류할 수 있다.

- ① 수정되는 공간정보가 기준상태인 경우 · 수정되는 튜플을 기준상태로 적용하여 지칭된 모든 튜플의 공간 정보를 수정하여야 한다. 따라서, 수정된 기준상태를 계속 유지할 수 있다. 망뮴을 고려하여 수정 연산을 설계하여야 한다.
- ② 수정되는 공간정보가 기준상태가 아닌 경우 · 수정되는 공간정보가 MBNF기법의 기준에 의하여 새로운 기준상태가 될 수 있으며, 이 경우 수정된 튜플의 STN튜플과 TS튜플의 공간정보도 변경하여야 한다. 따라서, 수정되는 공간정보가 새로운 기준상태가 되는 경우에도 이를 새로운 기준상태로 하지 않고 기존의 기준상태를 이용하여 변경된 느드만 저장할 수 있도록 수정 연산을 설계하여야 한다.

공간정보가 수정되는 경우에도 공간 정보의 유효성

이 상실되는 튜플이 발생하며, 이러한 튜플들에 대한 논리적인 삭제를 표현할 수 있는 방법이 필요하다.

3.2 누락삽입연산의 정의

과거데이터블에 누락된 정보가 삽입되는 경우는 다음과 같이 여섯 가지의 경우가 있다.

- ① 누락된 정보의 유효시간 범위가 한 개의 튜플의 유효시간과 관련되는 경우
- ② 누락된 정보의 유효시간 범위가 두 개의 연속된 튜플의 유효시간과 관련되는 경우.
- ③ 누락된 정보의 유효시간 범위가 세 개 이상의 연속된 튜플의 유효시간과 관련되는 경우
- ④ 누락된 정보의 유효시작시간이 과거 데이터베이스에 최초로 입력된 튜플의 유효시작시간보다 앞서는 경우.
- ⑤ 누락된 정보의 유효종료시간이 과거 데이터베이스에 마지막으로 입력된 튜플의 유효종료시간보다 큰 경우.
- ⑥ 누락된 정보의 유효종료시간이 과거 데이터베이스에 처음 입력된 튜플의 유효시작시간과 같은 경우.

유효시간 수정의 경우와 동일하게, 삽입되는 튜플의 시간인접튜플을 검색하여 유효시간을 수정하여야 하며, 수정된 시간인접튜플을 논리적이거나 물리적으로 삭제하여야 한다. 또한 공간정보가 수정되는 경우와 같이 삽입되는 공간정보가 새로운 기준상태가 될 수 있으며, 이를 위한 효율적인 공간정보의 저장방법이 고려되어야 한다.

(그림 9)는 누락삽입의 예이다. 튜플-2,5가 삽입되는 튜플의 시간인접튜플이며 각각 유효종료시간과 유효시작시간이 수정되어 튜플-6과 8로 갱신되었다

No.	O_ID	UNode	DNode	VS	VE	TS	TE
1	100	3	5	2	5
2	100	5	11	5	11
3	100	11	16	11	16
4	100	16	22	16	22
5	100	22	27	22	27
6	100	5	8	30	30
7	100	8	25	30	30
8	100	25	27	30	30

□ 누락삽입된 튜플 ▨ 시간인접튜플

(그림 9) 누락삽입-③

4. 데이터모델의 확장 및 연산의 설계

4.1 데이터모델의 확장

(그림 10)은 논리적으로 삭제된 튜플의 표현과 오류 수정 및 누락삽입 연산을 효율적으로 수행할 수 있도록 [12]에서 설계한 데이터모델을 확장한 과거데이터블의 구조이다. 논리적으로 삭제된 튜플임을 표현하기 위하여 L_DEL 속성을 추가하였으며, 유효시작시간이 수정될 경우 시간인접튜플을 효율적으로 검색하기 위하여 객체ID와 유효종료시간의 복합키를 이용한 색인인 'Past_Idx2'를 추가하였다

```
CREATE TABLE Past
( O_ID CHAR NOT NULL,
  UNode NODE, DNode NODE,
  VS TIME NOT NULL,
  VE TIME NOT NULL,
  TS TIME NOT NULL,
  TE TIME NOT NULL,
  L_DEL CHAR,
  PRIMARY KEY (O_ID, VS) ),
CREATE INDEX Past_Idx2
ON Past (O_ID, VE),
```

(그림 10) 확장된 과거데이터블 스키마

4.2 연산의 설계

본 절에서는 3장에서 살펴본 오류수정 및 누락삽입에 대한 연산을 설계한다. MBNF기법과 (그림 10)의 스키마를 이용하여 저장된 데이터블을 연산 대상으로 한다

```
입력 : 수정할 튜플의 키 K=(O_ID, VS).
      수정할 유효시작시간값 New_VS
출력 : 수정된 과거데이터블
01 : 입력된 키를 이용하여 과거데이터블로부터 수정할 튜플 검색
02 : VS ← K에 포함된 VS
03 : 과거데이터블로부터 (O_ID+VE)=(O_ID+VS)인 튜플 검색
04 : if(단계-03에서 검색된 튜플이 없으면)
05 :   단계-01에서 검색된 튜플 삭제
06 :   삭제된 튜플의 유효시작시간을 수정하여 과거데이터블에 삽입
07 :   알고리즘 종료
08 : else if(단계-03에서 검색된 튜플의 VS ≥ New_VS)
09 :   VS ← 단계-03에서 검색된 튜플의 VS
10 :   단계-03에서 검색된 튜플 삭제
11 :   단계-03으로 이동
12 : end if
13 : 단계-01과 단계-03에서 검색된 튜플 삭제
14 : 삭제된 튜플의 유효시간을 수정하여 과거데이터블에 삽입
```

(그림 11) 유효시작시간 수정 알고리즘

(그림 11)과 (그림 12)는 유효시간을 수정하는 경우의 알고리즘이며, 각각 수정되는 튜플의 키와 수정할 유효시간을 입력으로 하며, 수정된 과거데이터와 현재 데이터를 출력한다. 알고리즘의 가독성을 높이기 위하여 유효시작시간이 수정되는 경우와 유효종료시간이 수정되는 경우로 분류하여 알고리즘을 기술하며, 알고리즘에의 삭제는 논리적 삭제를 의미한다.

```

입력 : 수정할 튜플의 키 K = (O_ID + VS),
      수정할 유효종료시간값 New_VE
출력 : 수정된 과거데이터와 현재데이터
01 : K를 이용하여 과거데이터로부터 수정할 튜플 검색
02 : VE ← 검색된 튜플의 VE
03 : 과거데이터에서 (O_ID,VS) = (O_ID,VE)인 튜플 검색
04 : if(검색된 튜플이 없으면)
05 :   단계-01에서 검색된 튜플 삭제
06 : K에 포함된 O_ID를 이용하여 현재데이터로부터 튜플 검색
07 : if(검색된 튜플이 없으면)
08 :   삭제된 튜플의 유효종료시간을 수정하여 과거데이터에 삽입
09 : else
10 :   단계-06에서 검색된 튜플을 과거데이터로 이동후 삭제
11 : end if
12 : 알고리즘 종료
13 : else
14 :   if(단계-03에서 검색된 튜플의 VE ≤ New_VE)
15 :     VE ← 단계-03에서 검색된 튜플의 VE
16 :     단계-03에서 검색된 튜플 삭제
17 :     단계-03으로 이동
18 :   end if
19 :   단계-01과 단계-03에서 검색된 튜플 삭제
20 :   삭제된 튜플의 유효시간을 수정하여 과거데이터에 삽입
21 : end if
    
```

(그림 12) 유효종료시간 수정 알고리즘

위의 알고리즘에서 수정되는 튜플이 기준상태일 경우 과거 데이터베이스의 기준상태 데이터의 내용을 수정해야 하며, 또한 유효시간의 수정에 의하여 (그림 10)에서 생성한 기본 색인과 Past_Idx2의 색인을 수정할 필요가 있지만 이는 본 논문에서 설계하는 알고리즘의 핵심과 관련성이 없기 때문에 알고리즘에는 포함하지 않았다.

(그림 13)은 공간정보를 수정하기 위한 알고리즘으로 알고리즘의 목적이 과거데이터의 무결성 유지를 위한 연산이기 때문에 현재데이터에 저장되어있는 공간정보의 수정은 고려하지 않았다. 공간정보를 수정할 튜플의 키와 수정되는 공간정보가 입력 값이며, 수정된 과거데이터를 출력한다. 공간정보의 수정에서는 기

준상태 정보의 변경이나 색인의 변경은 발생하지 않는다.

```

입력 : 수정할 튜플의 키 K = (O_ID + VS),
      새로운 공간정보 New_Node
출력 : 수정된 과거데이터
01 : 기준상태 데이터를 이용하여 수정할 튜플의 공간정보 기준상태인지 검색
02 : if(수정할 튜플이 기준상태인 경우)
03 :   기존의 기준상태 튜플을 삭제
04 :   삭제된 튜플의 UNode에 New_Node의 내용을 저장하여 과거데이터에 삽입
05 : else
06 :   과거데이터로부터 수정할 튜플 검색
07 :   검색된 튜플 삭제
08 :   기준상태 정보를 이용하여 과거데이터로부터 기준상태 검색
09 :   검색된 기준상태와 New_Node를 비교하여 추가(삭제)된 공간정보 추출
10 :   단계-7에서 삭제된 튜플에 단계-9의 결과를 저장하여 과거데이터에 삽입
11 : end if
    
```

(그림 13) 공간정보 수정 알고리즘

(그림 14)는 누락삽입 연산을 위한 알고리즘이다. 삽입할 튜플의 공간정보와 유효시간을 입력으로 하며, 수정된 과거데이터와 현재데이터를 출력한다.

```

입력 : 삽입되는 객체의 ID
      공간 정보 Node와 유효기간 VS와 VE
출력 : 수정된 과거데이터와 현재데이터
01 : S_TN ← 과거데이터로부터 알고리즘에 입력된 VE를 포함하는 튜플 검색
02 : if(S_TN != NULL)
03 :   단계-01에서 검색된 튜플 삭제
04 :   if(S_TN의 VS < 알고리즘에 입력된 VS)
05 :     단계-03에서 삭제된 튜플의 유효종료시간을 수정하여 과거데이터에 삽입
06 :     단계-29로 이동
07 :   end if
08 :   T_VE ← S_TN의 VS
09 : else
10 :   과거데이터로부터 (O_ID,VS)=(ID,VE)인 튜플 검색
11 :   if(검색된 튜플이 없으면)
12 :     현재데이터로부터 ID의 튜플 검색
13 :     검색된 튜플을 과거데이터로 이동 후 삭제
14 :     T_VE ← 단계-12에서 검색된 튜플의 VS
15 :   else
16 :     알고리즘에 입력된 정보를 과거데이터에 삽입
17 :   end if
18 : end if
19 : P_TN ← 과거데이터로부터 (O_ID,VE)=(ID,T_VE)인 튜플 검색
20 : if(P_TN != NULL)
21 :   단계-19에서 검색된 튜플 삭제
    
```

```

22: if(P_TN의 VS ≤ 알고리즘에 입력된 VS)
23:   식체된 튜플의 유효시간을 수정하여 과거데이터에
      삽입
24: else
25:   T_VE←P_TN의 VS
26   단계-19로 이동
27   end if
28: end if
29: 알고리즘에 입력된 정보를 과거데이터에 삽입
    
```

(그림 14) 누락삽입 알고리즘

5. 결 론

시간지원 시스템에서는 객체의 현재상태와 함께 과거의 상태도 관리한다. 이러한 시스템의 데이터베이스에 저장된 각 객체의 속성데이터나 공간데이터, 유효시간들은 사용자에게 의하여 입력되는 정보들이기 때문에 입력과정의 오류 등에 의하여 잘못된 데이터가 데이터베이스에 저장될 수 있으며, 누락되어 입력되지 않은 데이터들이 존재할 수 있다. 본 논문에서는 이러한 오류와 누락의 다양한 경우를 알아보았으며, 이들을 수정할 수 있는 오류수정연산과 누락삽입 연산을 설계하였고, 기존의 데이터모델을 확장하여 이들 연산이 효율적으로 수행될 수 있도록 함으로써 과거데이터의 무결성을 보장할 수 있도록 하였다.

본 연구와 관련하여 향후 연구할 과제는 다음과 같다. 현재 시간지원 지리정보시스템 분야에서 시간과 공간을 모두 포함하는 색인에 대한 연구가 진행되고 있지만, 이들 연구들은 대부분 본문에서 살펴본 거래시간의 특징을 고려하고 있다. 따라서 유효시간이 포함된 시공간질의 효율적인 수행을 위한 시공간 색인에 대한 연구가 필요하다.

참 고 문 헌

[1] Sudha Ram, J. S. Park, "Modeling Spatial and Temporal Semantics in a Large Heterogeneous GIS Database Environment." Proc. of the 2nd Americas Conf. on Information Systems(AIS '96), pp 683-685, 1996.

[2] Erwig, M., Guting, R.H., Schneider, M., Vazirgiannis, M., "Spatio-Temporal Data Types : An Approach to Modeling and Querying Moving Objects in Databases," GeoInformatica, 1999

[3] C. Claramunt, Theiault, M., "Toward semantics for modeling Spatio-Temporal processes withun GIS," SDH, 1996

[4] T. S. Cheng, S. K. Gadia, "A pattern matching language for spatio-temporal databases," Proc. of the 3rd Int. Conf. on Information and Knowledge management, pp.288-295, 1994.

[5] R. Edsall, D. Peuquet, "Graphical query techniques for temporal GIS." 13th Auto-carto Conf., 1997.

[6] R. T. Snodgrass, "The TSQL2 Temporal Query Language," Kluwer Academic Publishers, pp.186-219, 1995.

[7] A. Tansel, J. Clifford, S.K. Gadia, S. Jajodia, A. Segev, R. Snodgrass(des.), "Temporal Databases . Theory, Design, and Implementation," Benjamin/Cummings, pp.141-182, 1993

[8] A. Doucet, M-C. Fauvet, S. Gancarski, G. Jomier, S. Monties, "Using Database Versions to Implement Temporal Integrity Constramts," Second Int Workshop on Constraint Database Systems, CDB '97, LNCS 1191, pp.219-233, 1997

[9] 박동선, 배해영, "다차원 지리정보시스템을 위한 저장기법 및 분리된 저장구조", 정보처리논문지, Vol.7, No 1. 2000



박 동 선

e-mail : dseon@netian.com

1992년 인하대학교 전자계산공학과 졸업(공학사)

1994년 인하대학교 대학원 전자계산공학과 졸업(공학석사)

1996년~현재 인하대학교 대학원 전자계산공학과 박사과정

관심분야 : 데이터베이스, 지리정보시스템, 시간지원 지리정보시스템



김재홍

e-mail . jhong@kachi.ytt.ac.kr

1985년 인하대학교 전자계산학과

(B.S)

1990년 인하대학교 전자계산학과

석사

1994년 인하대학교 전자계산학과 박사

1995년~현재 영동대학교 컴퓨터공학과 부교수

관심분야 : Multimedia DBMS, GIS, ATM networks



배해영

e-mail . hybae@dragon.inha.ac.kr

1974년 인하대학교 응용물리학과
졸업(공학사)

1978년 연세대학교 대학원 건국
계산학과 졸업(공학석사)

1989년 숭실대학교 대학원 전자
계산학과 졸업(공학박사)

1985년 Univ. of Houston 객원교수

1992년~1994년 인하대학교 전자계산소 소장

1982년~현재 인하대학교 전자계산공학과 교수

관심분야 : 데이터베이스, 멀티미디어 데이터베이스, 지
리정보시스템, 실시간 데이터베이스