

# 토지대장/지적도 이력관리 시스템을 위한 시공간 연산자의 설계 및 구현

## Design and Implementation of Spatiotemporal Operators for History Management System of Cadastre and Cadastral Map

김성룡\*, 김상호\*\*, 이화종\*\*\*, 류근호\*\*\*\*

Sung Ryoung Kim, Sang Ho Kim, Hwa Jong Lee, Keun Ho Ryu

**요약** 지리정보 시스템은 토지와 관련한 속성 데이터와 공간 데이터의 통합 정보를 저장하고 저장된 데이터를 분석할 수 있다. 대부분의 지리정보 시스템은 토지의 현재 정보만을 지원한다. 그러나 토지에 대한 소유권 분쟁이나 토지 정책 수립에 필요한 통계 정보는 현재 정보뿐만 아니라 과거의 이력 정보를 필요로 한다. 따라서, 이 논문에서는 효율적으로 이력 정보를 관리할 수 있는 시공간 연산자를 설계 및 구현하였다. 이 연산자들은 GIS 기반 토지대장/지적도 이력관리 시스템에서 사용된다. 이력 정보는 시공간 연산자의 공간 연산에 의해 발생되며, 구현된 연산자는 이력 정보를 저장하고, 관리할 수 있는 기능을 제공한다.

**ABSTRACT** GIS Tools is able to not only store the integrated information between attribute data and spatial data, but also analyze them. Most of GIS Tools support current information of land. However, lawsuit of right of possession as well as statistic of past data for land policy may need historical information. Therefore, in the paper, we designed the spatiotemporal operators what can handle the historical information, and implemented them. The spatiotemporal operators are used in the history management system of Cadastre and Cadastral map based on GIS. The historical information is generated in spatial operation of spatiotemporal operators and then the implemented operators provide a function to store the historical information, and handle them.

**키워드** : 이력관리(History Management), 토지대장(Cadastre), 지적도(Cadastral), 시공간 데이터베이스(Spatiotemporal Database)

### 1. 서론

많은 국가들은 토지에 관련된 문제를 해결하고 이를 관리하기 위하여 최근 지적의 역할을 더욱 중요하게 인식하게 되었다. 토지는 다른 일반 재화와 달리 토지가 갖는 특성으로 인하여 상품으로서의 가치가 높고 있다. 특히, 인구의 증가와 함께 토지의 배분과 이용에 있어서 보다 많은 관심과 노력이 요구되고 있기 때문에 이미 오래 전부터 지적의 중요성이 강조되어

왔고 이를 위한 많은 연구들이 수행되었다. 이와 같이 토지와 관련된 정보를 저장 및 관리하는 토지정보 시스템은 사회의 발전과 사용자 요구사항이 증대함에 따라 토지의 소유뿐만 아니라 토지와 관련된 다양한 정보를 기록하고 관리하는 것이 필요하며, 축적된 정보를 이용하여 사용자의 요구에 대해 적절한 결과를 제시해야 한다.

최근에는 지리정보 시스템(Geographic Information Systems)의 등장으로 토지와 관련된 일반 속성 정보

\* 한국통신 정보기술

\*\* 충북대학교 전자계산학과 박사과정

\*\*\* (주) 이니텍

\*\*\*\* 충북대학교 전기전자 및 컴퓨터공학부 교수

E-Mail : srkim@kkit.com.

E-Mail : shkim@dblab.chungbuk.ac.kr

E-Mail : hjlee@initech.com.

E-Mail : khryu@dblab.chungbuk.ac.kr

및 공간 정보를 통합하여 저장, 분석, 관리할 수 있게 되었다. 이로 인해 기존에 수작업으로 수행하던 도면판리의 업무에서 지리정보 시스템을 이용한 효율적인 업무 수행 방향으로 바뀌어가고 있는 추세이다. 그런데, 토지정보는 시간이 경과함에 따라 소유주, 소유주의 주소, 지번 등이 변경되는 일반 속성 변화와 필지의 분할, 합병, 경계변경과 같은 공간정보의 변화가 발생하게 된다. 이처럼 지적 데이터는 수시로 변경될 수 있기 때문에 현 상태를 정확하게 유지해야 함은 물론 토지에 대한 소유권 분쟁이나 토지 정책 수립에 필요한 통계 정보 등을 제공하기 위해 과거의 정보도 현 상태와 연결되어 정확하게 유지되어야 할 필요성이 있다.

그러나, 지금까지 토지정보를 관리하는데 사용된 아크인포(ArcInfo), 스몰월드(Smallworld), 지오윈(Geowin) 등과 같은 지리정보 시스템은 토지의 현재 상태에 대한 정보만을 기록하고 관리하는 특성을 지니고 있다. 따라서, 공간 및 비공간 속성의 변경에 대한 이력 정보를 유지하지 못하므로 토지의 분할, 합병, 영역 경계 변경, 소유주 변경 등과 같은 수정이 발생할 경우 이전 상태에 대한 정보는 삭제되고, 사용자는 현재의 토지 상태에 대한 정보만을 확인 할 수 있으며 과거 정보와 관련된 정보는 전혀 획득 할 수 없는 문제점이 발생하게 된다.

따라서, 이 논문에서는 기존의 지리정보 시스템을 이용하여 토지정보 시스템을 구현할 경우 효율적으로 지원할 수 없는 이력 관리 기능을 제공하기 위해, 기존의 지리정보 시스템에 시간 속성을 추가하고, 공간 연산자에 시간 개념을 확장한 시공간 연산자를 설계 및 구현한다. 이 시공간 연산자는 분할, 합병, 영역 경계 변경, 소유주 변경 등의 토지 속성 변경 처리에 있어 이력 정보를 발생시키고, 발생한 이력 정보를 현재 정보와 이력 포인터로 연결함으로써 토지의 현재 상태는 물론 과거 상태에 대한 질의를 수행 할 수 있는 특징을 가진다.

이 논문의 전체적인 구성은 다음과 같다. 먼저, 2장에서는 시공간 데이터베이스의 개념 및 이 논문에서 사용될 스몰월드 지리정보 시스템에 대하여 기술한다. 3장에서는 토지대장/지적도의 관련 업무 및 토지대장/지적도 이력관리 시스템에 관해 서술한다. 4장에서는 시공간 연산자의 기능 및 알고리즘을 제시한다. 5장에서는 구현된 토지대장/지적도 이력관리 시스템을 이용한 시공간 연산자의 사용 결과를 보이고, 마지막으로 6장에서는 결론을 맺는다.

## 2. 관련연구

이 장에서는 시공간 데이터베이스와 스몰월드 지리정보 시스템에 대해 설명한다. 시공간 데이터베이스는 시간에 따라 공간 속성이 변하는 시공간 객체들을 저장 및 관리하기 위해 기존의 공간 데이터베이스를 시간차원으로 확장한 것이며 이를 위한 시간의 개념에 대해 설명한다. 또한 이 논문의 구현에 사용되는 스몰월드 지리정보 시스템의 특징과 구조에 대해 설명한다.

### 2.1 시공간 데이터베이스

시간은 끊임없이 발전하는 실세계에 대한 필수적인 정보의 한 부분이며 사실이나 자료는 시간과 관련지어 해석할 필요가 있다. 의사결정지원시스템에서의 경향 분석과 같은 과거 상태에 대한 이력질의 처리, 소급 또는 사전행동에 의한 변화를 표현하기 위해 시간지원 데이터베이스가 필요하다[8]. 이러한 시간의 구조는 선형(Linear), 분기(Branching), 주기적(Periodic) 시간으로 표현되고, 선형시간의 경우 시작시간과 끝시간으로 기술된다. 또한 시간의 밀도는 시간선이 정수 형태인 이산(Discrete)구조, 유리수 형태인 조밀(Dense)구조, 실수 형태인 연속(Continuous) 구조로 구분된다[7]. 시간의 단위(Granularity)는 시간선의 분할을 의미하며 세기(Century), 년(Year), 월(Month), 일(Day), 시간(Hour), 분(Minute), 초(Second), 마이크로초(Micro second) 등이 있다. 이 연구에서는 선형이며 이산적이고 시작시간과 끝시간으로 표현되는 시간 범위(Bound)를 지원하는 시간 모델을 사용한다.

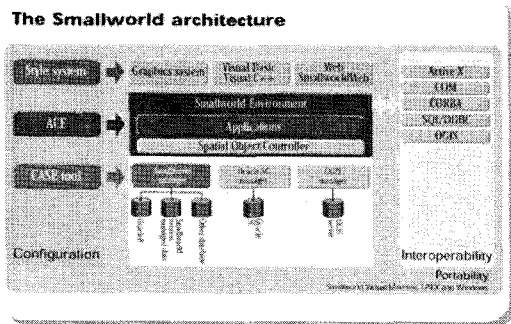
시간지원 데이터베이스에서 사용하는 시간은 데이터베이스에 자료가 저장된 시간을 표현하는 거래시간, 실세계에서 사건의 발생 시간을 나타내는 유효시간이 있다. 유효시간은 사용자에게 의해 결정되며 그 기간에 유효한 데이터의 수정이 가능하다. 거래시간(Transaction time)은 어떤 사건이 데이터베이스에 수록된 시간으로 데이터베이스 연산의 발생과 동시에 DBMS에 의해 자동으로 시스템 클럭의 시간으로 기록되며, 어떤 값이 데이터베이스에 수록되기 시작한 거래의 시작시간과 그 종료시간으로 표현된다[9].

시공간 데이터 모델은 실세계에서 발생하는 데이터에 대한 시간 적용과 그 범주에 따라 스냅샷 데이터 모델[2], 시공간 복합 데이터 모델[3], 사건 지향 모델[5], 이력 그래프 모델[4,6,7], Three-domain 모델[11], 객체지향 모델[12] 등이 있다. 이들은 모두

시공간 데이터를 모델링 하는 방법이지만, 각각 나름대로의 장점과 단점을 가지고 있다. 스냅샷 모델은 쉽고 단순하지만 자료 중복에 대한 문제(5)가 있고 시공간 복합 데이터 모델은 자료 중복의 문제를 어느 정도 해결하였지만 동일한 이력을 갖는 공간 객체의 구분이 불가능하다[13]. 또한 사건 지향 또는 사건 기반 데이터 모델에서는 발생 이력들이 기본 도면에 연결 리스트로 연결되어 있기 때문에 최근 상태를 얻기 위해서는 기본 도면으로부터의 변화상태를 조합해야 한다는 단점을 가진다. 시공간 객체 지향 모델은 시공간 객체를 쉽게 표현할 수 있지만 구현이 복잡하고 어렵다는 문제점이 있다. 그러므로 시공간 데이터 모델은 응용에 적합한 모델을 선택해야 한다.

## 2.2 스몰월드 지리정보 시스템

스몰월드는 영국의 Smallworld사가 개발한 공간자료 관리 시스템으로 객체지향 기술을 이용하여 공간자료와 비공간 자료를 하나의 객체로 표현 할 수 있다. 스몰월드는 자체 데이터베이스나 기존의 상용 DBMS 모두를 지원하며 응용 시스템은 Magik 언어를 이용하여 데이터의 저장, 검색 또는 공간 및 속성 자료의 처리를 수행 할 수 있다. 또한, Case Tool을 이용하여 데이터 모델을 설계 할 수 있다[10].



〈그림 1〉 스몰월드 구조

그림 1은 스몰월드 시스템의 구조를 보여준다. Smallworld에서 객체(Object)는 실제계의 지형을 표현하기 위해 사용되는 기본적인 공간요소이며 지원하는 객체 타입(Type)으로는 점(Point), 폴리곤(Polygon), 면(Area)이 있다. 스몰월드의 위상 데이터 모델에서 객체들은 점(Point, 0차원)으로 표현되며, 체인(Chain, 1차원)이나 면(Area, 2차원) 객체들은 하나의 노드

(Node)나 하나 이상의 링크(Link), 하나 이상의 폴리곤(Polygon)들의 집합으로 구성된다. 노드는 링크간의 연결과 링크와 점 객체간의 연결을 나타내는데 사용한다. 연결된 링크들은 최소한 하나의 공통된 노드를 갖는다. 링크는 라인스트링(Line String)이라고도 하며, 좌표들의 연속으로 표현된다. 링크 중의 일부는 곡선(Curve)이나 호(Circular arc)로 사용될 수 있으며 여러 개의 폴리곤과 체인 사이에서 공유된다. 폴리곤은 하나 이상의 링크를 포함하고, 링크는 하나 이상의 폴리곤에 포함되며 각각의 폴리곤은 다른 폴리곤들과 홀(Hole)관계를 가질 수도 있다. 점은 0차원 객체를 나타낸다. 점들은 실제로 2개의 좌표를 갖는데 하나는 실제계에서 객체의 위치를 나타내는 좌표이며, 또 다른 하나는 도면상에서의 위치를 나타내는 좌표이다. 이러한 점들은 공유되지 않는다. 체인은 링크의 집합으로 구성되며 강이나 도로의 중앙선과 같은 선형 객체를 나타내는데 사용된다. 면은 폴리곤들의 집합이고 그림(Picture)은 링크의 집합으로 내장되어 있는 위상구조에 의해 표현되므로 복잡한 그래픽 특성을 가진 객체를 표현하는 데에 사용된다.

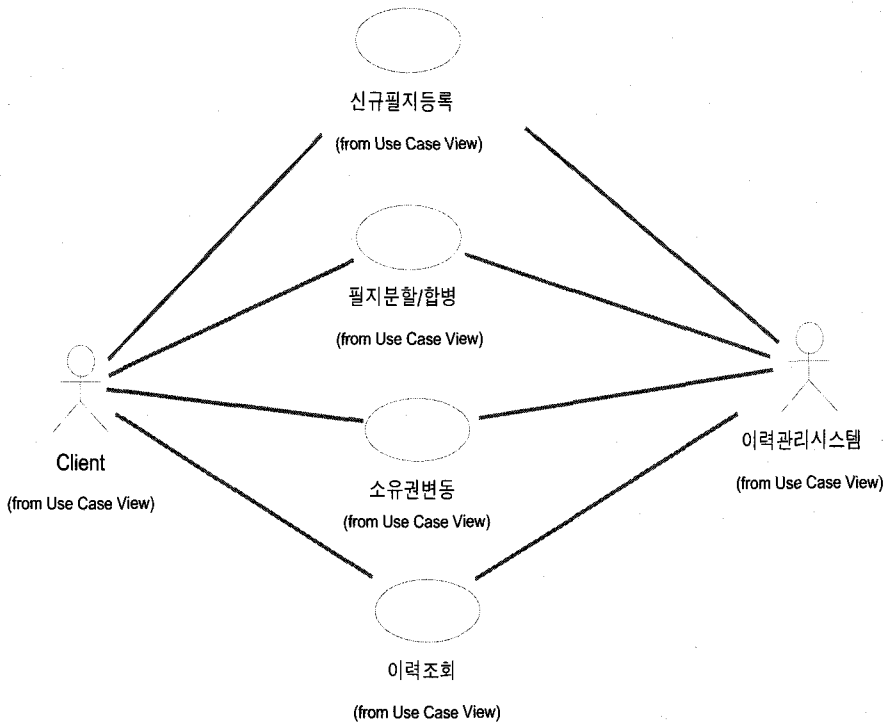
## 3. 토지대장/지적도의 이력관리

이 장에서는 토지대장/지적도의 이력관리 및 시공간 연산자의 개발을 위해 관련 업무를 분석하고, 토지정보 시스템 구현을 위한 지리정보 시스템의 시공간 확장에 대해 기술한다.

### 3.1 토지대장/지적도 업무 분석

현행 토지정보 시스템은 국토의 균형 있는 개발과 활용을 위해 1979년도부터 정부에 의해 각시도별로 토지 기록관리 전산화를 추진하여 현재까지 이르고 있다. 토지 기록관리는 크게 신규 필지 등록, 분할, 합병과 같은 토지 이동으로 발생하는 모든 변동자료를 관리하는 토지 이동관리 업무와 소유권 보존, 소유권 이전, 주소변경, 성명변경과 같은 소유권 변동 원인에 의한 변동 사항을 관리하는 소유권 변동 관리 업무가 있다. 이외에도 등급 변동이나 지적 통계에 관한 업무가 있다[14].

이 논문에서는 토지정보 시스템의 핵심업무인 토지 기록 관리측면에서의 이력관리에 주목적을 두었으며 변경이 발생하여 그 이력을 관리할 필요가 있는 토지 이동 관리업무에 대한 시공간 연산자를 설계 및 구현한다. 그리고, 향후 이들 연산자를 바탕으로 통계업무를 비롯한 소유권 변동 관리 업무 등 다른 업무로의 연계가 가능하다.



〈그림 2〉 토지이동 관리업무의 Use Case Diagram

그림 2는 토지이동 관리업무의 Use Case Diagram이다. 신규등록은 토지를 신규로 지적공부에 등록하는 것을 말한다. 따라서, 토지 기본사항, 토지 표시사항, 소유권 표시사항 등 토지의 모든 사항을 입력한다. 토지 기본사항으로는 토지가 위치하고 있는 행정구역을 말하는 토지소재, 이동사유구분, 소유구분, 주소, 지번, 지목, 면적 등이 있다. 다음의 그림 3은 신규필지 등록처리 흐름을 Sequence diagram으로 표현한 것이다. 사용자는 Land Editor라는 객체로부터 메뉴를 활성화하고 기본사항을 입력한 뒤 입력 메시지를 전달하면 Land Editor 객체는 Owner 지속객체에 사용자 입력사항을 저장하며 유효시간을 설정한다.

토지 분할은 지적공부에 등록된 1필지를 2필지 이상으로 분할하여 등록하는 것을 말한다. 기본 등록사항으로는 토지 고유번호, 지번, 지목, 행정구역 코드, 면적 등이 있다. 그림 4는 이러한 분할의 처리 순서를 보여준다. 여기서 GEO Editor는 위상(Topology)을 처리하는 스몰월드 Topo Engine 객체를 상속하며 분할의 처리는 split area와 split link 같은 메소드

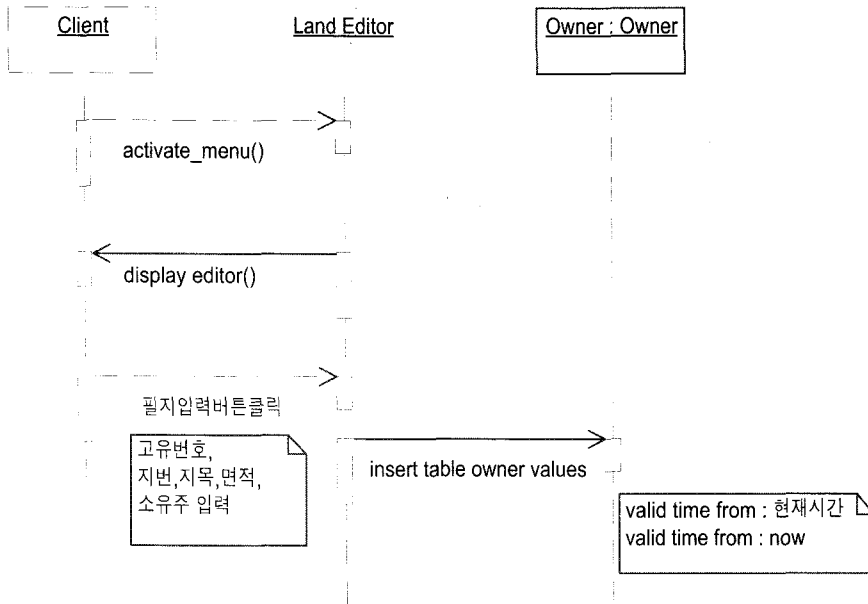
의 호출로 수행된다.

토지 합병은 지적공부에 등록된 2필지 이상을 1필지로 등록하는 것을 말한다. 실제 합병을 처리할 때는 분할 연산과 반대로 merge\_link와 merge\_area 메소드를 호출하여 처리한다. 합병된 필지들은 합병이전의 필지와 포인터로 연결된다.

### 3.2 스몰월드 지리정보 시스템의 시간 확장

지리정보 시스템을 이용하여 지적객체의 이력을 관리하기 위해서는 시간 데이터를 저장하고 관리 할 수 있도록 공간 데이터베이스에 시간차원을 확장해야 하며, 시간 데이터들을 처리할 수 있는 시간 위상 관계 연산자를 제공해야 한다. 따라서, 본 논문에서 사용되는 스몰월드 지리정보 시스템의 시간 확장 및 시간 위상 관계 연산에 대하여 서술한다.

스키마 설계에 있어 각각의 이력 버전은 비공간 속성 및 공간 속성 외에 유효시간을 표현하는 타임스탬프(Timestamp), 객체 단위 이력 지원을 위해 이전 이력과 이후 이력을 연결하기 위한 포인터, 발생한 사



〈그림 3〉 신규필지 등록 Sequence Diagram

건을 표현하기 위한 사건 속성을 추가로 가지게 된다 [15].

$$\begin{aligned}
 O^{ST} = & \{ D_{attribute} \cup D_{spatial} \cup \\
 & D_{time} \cup D_{predecessor} \cup D_{successor} \} \\
 D_{time} = & \langle VTs, VTe \rangle \\
 (\text{단, } & VTs \leq VTe) \\
 D_{predecessor} : & \text{Set}\langle Oid \rangle \\
 D_{successor} : & \text{Set}\langle Oid \rangle
 \end{aligned}$$

시공간 객체 OST는 속성정보 Dattribute, 공간정

보 Dspatial, 시간정보 Dtime, 이전 이후 이력을 위한 선행자 Dpredecessor와 후행자 Dsuccessor를 갖는다. 시간정보 Dtime은 객체가 실제 세계에서 효력을 발생하는 시간인 유효시간을 나타내며 선행자와 후행자는 객체식별자를 값으로 하는 집합으로 표현된다. 이력 검색을 위한 시간 위상 관계 연산자에 대해서는 Allen[1]과 Snodgrass[9]가 제안한 시간 위상관계 연산자를 참고하여 다음의 표 1과 같이 시간 위상 관계 연산자를 정의한다.

〈표1〉 시간 위상 관계 연산자

연산구분	피연산자	시간 비교 연산자	피연산자	시간 조건식
점 시간 연산	Time Point a	Start	Time Point b	a = b (a start at b)
	Time Point a	End	Time Point b	a = b (a end at b)
간격 시간 연산	{a,b}	During	{c,d}	(a < c or a = c) and (b > d or b = d)
	{a,b}	Equal	{c,d}	a = c and b = d
	{a,b}	Start	{c,d}	a ∈ [c,d)
	{a,b}	Finish	{c,d}	b ∈ [c,d)
	{a,b}	Overlap	{c,d}	{∃t   t ∈ {a,b} ∧ t ∈ {c,d}}

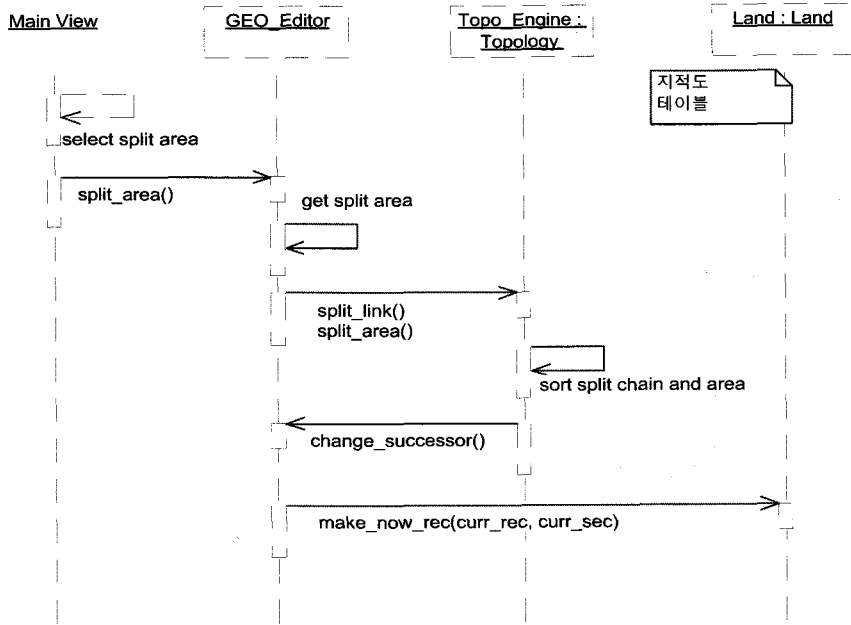


그림 4. 필지 분할 Sequence Diagram

표 1에서 제안한 시간 위상 관계 연산자의 구현은 시간 연산자 클래스인 time\_exemplar에서 시간 연산자에 대한 정의를 하고 응용 프로그램에서는 이를 상속받아 사용한다. 점 시간(Time point) 연산은 비교되는 시간 속성이 하나의 시간 값인 특정한 점(Point)으로 표현된다. 이를 위한 연산 유형으로는 특정시간에 발생한 사건에 관한 질의인 시작(Start)과 종료된 사건에 관한 질의인 종료(Finish)가 있고 수행 방식은 다음 그림 5와 같다.

간격 시간(Time interval) 연산은 비교되는 시간 속성이 두 개의 시간 값인 시간의 간격(Interval)으로 표현된다. 이를 위한 연산 유형은 During, Equal,

```

value << time_operand
type << time_operator

_if type = Start
_then
_return
predicate.eq(:v_time_from, value)
_elif type = Finish
_then
_return
predicate.eq(:v_time_to, value)
_endif
    
```

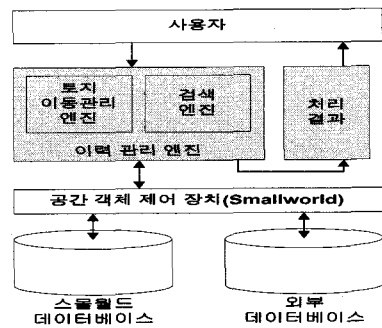
<그림 5> 점 시간 연산 알고리즘

Start, Finish, Overlap 등이 있고 수행 방식은 다음 그림 6과 같다.

그림 6은 이력 객체가 유효한 시간을 나타내는 시작시간과 종료시간 및 발생한 사건의 유형을 피연산자로 하여 predicate을 구성한다.

### 3.3 토지대장/지적도 이력관리 시스템

토지대장/지적도 이력관리 시스템의 구조는 그림 7과 같으며 스몰월드 시스템을 기반으로 이력관리 엔



<그림 7> 시스템 구조도

```

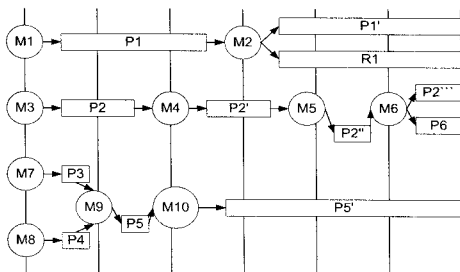
s_value    << start_time_value
f_value    << finishe_time_value
type << time_operator
result << time_predecate

_if type = During
    result << (predicate.gt(:v_time_from, s_value) _or predicate.eq(:v_time_from,
s_value))
                _and (predicate.lt(:v_time_to, f_value) _or predicate.eq(:v_time_to,
f_value))
    _elif type = Equal
        result << (predicate.eq(:v_time_from, s_value) _and predicate.eq(:v_time_to,
f_value))
    _elif type = Start
        result << (predicate.gt(:v_time_from, s_value) _or predicate.eq(:v_time_from,
s_value))
                _and (predicate.lt(:v_time_from, f_value) _or predicate.eq(:v_time_from,
f_value))
    _elif type = Finish
        result << (predicate.gt(:v_time_to, s_value) _or predicate.eq(:v_time_to,
s_value))
                _and (predicate.lt(:v_time_to, f_value) _or predicate.eq(:v_time_to,
f_value))
    _elif type = Overlap
        result << ((predicate.gt(:v_time_to, s_value) _or predicate.eq(:v_time_to,
s_value))
                _and (predicate.lt(:v_time_to, f_value) _or predicate.eq(:v_time_to,
f_value)))
                _or ((predicate.lt(:v_time_from, f_value) _or predicate.eq(:v_time_from,
f_value))
                _and (predicate.gt(:v_time_from, s_value) _or predicate.eq(:v_time_from,
s_value)))
    _endif
return result
    
```

〈그림 6〉 간격 시간 연산 알고리즘

진을 추가하여 토지의 이력을 관리한다. 이력 관리 엔진은 토지이동 관리엔진과 검색엔진으로 구성되며 사

용자의 변경 작업에 대해 이력을 생성, 관리하고 현재 객체와 이력 객체 모두에 대한 검색을 수행한다.



〈그림 8〉 토지정보 시스템을 위한 이력 그래프

토지대장/지적도 이력관리 시스템에서는 토지의 이력관리를 위해 이력 그래프 데이터 모델을 사용하였다. 이력 그래프 데이터 모델은 사건 지향 데이터 모델의 일종으로 실세계에는 시간의 구간을 가지는 변경들이 계속해서 일어나며, 이는 시간적 사건(Behavior)의 영역에 나타난다고 보고 있다. 그 사건의 유형은 계속적으로 변경되는 객체(7)와 기본적으로는 정적이지만 구간을 가지는 사건에 의해 변경되는 객체(6) 그리고 항상 정적이지만 갑작스런 사건으로 변경되는 객체(4)의 세 가지 유형으로 분류된다.

그림 8에서 원은 사건, 사각형은 공간객체를 나타

〈표 2〉 토지대장/지적도 릴레이션

Field Name	Data Type	etc
id	sys_id	System에서 관리 Object ID(Key)
일반 속성	native (String, Integer)	토지명, 지번/지목, 면적, 소유주 등의 토지가 가지는 특성들을 나타냄
coverage	geometry (area)	Smallworld GIS에서 제공, 필지 영역 경계 표시
event	string	발생 이력의 종류
VT-Start	date	유효 시작 시간
VT-End	date	유효 종료 시간
pred	set of sys_id	선행 이력 id(Set)
succ	set of sys_id	후행 이력 id(Set)

내며 시공간 객체의 분할과 합병으로 발생하는 이력을 표현한다.

각각의 시공간 객체는 생성이라는 사건에 의해 시공간 상에 나타나게 되며 분할, 합병, 소유변경, 지번변경 등의 사건에 의해 속성이 변경되어 이력을 생성한다. 시공간 데이터베이스에서 객체는 선행 객체와 후행 객체를 가리키는 포인터로 연결되어 있다. 생성,

분할, 합병과 같은 사건의 발생은 실제계의 유효시간을 반영하며 이력은 포인터를 통해 관리되어진다. 이력 그래프 모델을 기반으로 하는 토지대장/지적도 이력관리 시스템의 릴레이션 구성은 표 2와 같다.

표 2에서 event, VT-Start, VT-End, pred, succ 등의 필드들은 이력의 생성 및 유지관리에 사용되는 것일 뿐 필드의 속성을 표현하지는 않는다. 토지대장/지적도의 이력 레코드와 현재 레코드는 동일한 릴레이션 안에 저장되고 관리된다.

#### 4. 시공간 연산자 설계

시간의 흐름에 따른 공간의 변화를 나타낼 때 사용되는 시공간 연산은 크기가 변하는 경우(축소, 확대), 존재가 변하는 경우(나타남, 사라짐), 위치가 변하는 경우(회전, 이동), 모양이 변하는 경우, 변화하지 않는 경우, 분할하는 경우, 합병하는 경우로 나누어 질 수 있다. 일반적인 시공간 연산자에 대해 정의하고, 토지대장/지적도 이력관리시스템에서 주로 사용되는 시공간 연산인 분할 연산, 합병 연산과 검색 연산에 대한 설계를 한다.

##### 4.1 시공간 연산자

시공간 연산은 시간에 흐름에 따른 공간 객체에 변화를 표현할 수 있게 한다. 객체의 크기변화, 존재변화, 위치변화, 모양변화, 불변화와 분할 및 합병을 시간에 흐름에 따라 표현 할 수 있게 한다. 시공간 연산자에 대한 정의는 [정의 1]과 같다.

##### 【정의 1】 시공간 연산자

시공간 연산자는 주어진 시간 간격 <from, to>와 Rspatiotemporal의 관계에 있는 객체를 추출해 내는 연산자이다. 단 from < to이며 Vfrom < Vto 이다.

Rspatiotemporal = {OPcontract , OPexpand , OPappear , OPdisappear , OProtate , OPdisplace , OPtransformate , OPstable.}

OPcontract(from, to) = ((∃fi, 1 ≤ fi < n (from ≤ OSTh(fi).Vfrom < to)) & (∃ti, fi < ti ≤ n (from ≤ OSTh(ti).Vto ≤ to)) & (OSTh(fi).area(geom) > OSTh(ti).area(geom)))

OPexpand(from, to) = ((∃fi, 1 ≤ fi < n (from ≤ OSTh(fi).Vfrom < to)) & (∃ti, fi < ti ≤ n (from ≤ OSTh(ti).Vto ≤ to)) & (OSTh(fi).area(geom) < OSTh(ti).area(geom)))

OPappear(from, to) = (from ≤ OSTh(1).Vfrom ≤ to)

OPdisappear(from, to) = (from ≤ OSTh(1).



〈표 3〉 시공간 연산자 분류

구분	연산자	의미	그림상의 정의
크기변화	축소(contract) 확대(expand)	객체의 모양은 그대로 유지되면서 그 크기만 축소/확대되는 경우	
존재변화	나타남(appear) 사라짐(diappear)	객체 자신의 존재변화 즉, 객체가 생겨나고, 객체가 사라지는 연산	
위치변화	회전(rotate) 이동(displace)	그 위치가 변화되는 경우로서 회전에 의한 위치변화와 이동연산이 포함됨	
모양변화	모양변형 (transformate)	크기는 유지되면서 모양이 변화되는 경우	
불변화	유지(stable)	시간이 흘러도 공간적인 변화가 없는 경우	
분할	분할(split)	시간이 흐르면서 한 객체가 두개로 나누어지는 경우	
합병	합병(union)	시간이 흐르면서 두 객체가 한개로 합쳐지는 경우	

$V_{to} \leq t_o$   
 $OP_{stable}(from, to) =$   
 $(\exists fi, \exists ti, 1 \leq fi < n, fi < ti \leq n (from \leq$   
 $OSTh(fi).V_{from} \& (OSTh(ti).V_{to} \leq to))$   
 (같은 방법으로  $V_{from}/V_{to}$ 에  $T_{start}/T_{stop}$ 을 대응시킬 수 있다)

위에서 정의한 시공간 연산자를 분류해보면 표 3과 같다. 시공간 연산이 일어나는 경우는 크기가 변하는 경우(축소, 확대), 존재가 변하는 경우(나타남, 사라

짐), 위치가 변하는 경우(회전, 이동), 모양이 변하는 경우, 변화하지 않는 경우, 분할하는 경우, 합병하는 경우로 나누어질 수 있다.

위와 같이 구분된 시공간 연산자 중에서 토지 이동관리 연산에 주로 사용되는 연산은 분할연산과 합병연산이다.

#### 4.2 토지 이동관리 연산

공간 객체의 생성 즉, 토지의 생성은 도시계획 등에 의해 새롭게 지적공부에 필지를 등록하는 것을 의미하

OID	지번	지적도	VT-Start	VT-End	pred	succ
50	100	L100	97-3-1	now	{}	{}
60	101	L101	97-3-1	now	{}	{}

OID	지번	지적도	VT-Start	VT-End	pred	succ
50	100	L100	97-3-1	98-5-6	{}	{70, 80}
60	101	L101	97-3-1	now	{}	{}
70	100-1	L102	97-5-7	now	{50}	{}
80	100	L103	97-5-7	now	{50}	{}

〈그림 9〉 시공간 데이터베이스 변화(분할)

며 생성된 공간객체는 시공간 데이터베이스에서 지적 정보와 속성 정보의 등록이 필요하다. 표 4에서 보여주는 바와 같이 생성된 공간객체는 새로운 객체 식별자(OID)를 시스템으로부터 부여받으며 관련 정보는 사용자로부터 입력받는다. 객체 식별자가 60인 새로운 필지는 지번이 101이고 유효시간 ('97-3-1', 'NOW')를 가지고 시공간 데이터베이스에 저장된다.

〈표 4〉 시공간 데이터베이스 생성

OID	지번	지적도	VT-Start	VT-End	pred	succ
50	100	L100	97-3-1	NOW	{}	{}
60	101	L101	97-3-1	NOW	{}	{}

분할(Split) 연산은 한 필지가 2개 이상의 필지로 분할되어 2개 이상의 새로운 필지를 생성함을 의미한다. 분할 객체는 시공간 데이터베이스에서 새로운 공간 객체의 생성을 의미하며, 분할되기 이전의 필지는 과거의 이력 정보로서 저장된다. 또한 분할되어 새로이 생성된 필지는 새로운 객체 식별자를 가지고 등록되며, 새로 등록된 필지의 이력 정보는 이전 이력 객체의 객체 식별자를 저장한다.

그림 9은 50번 필지가 98년 5월 6일 분할이 되어 70, 80번 필지로 분할된 경우이다. 새로 생성된 70번과 80번 필지는 50번 필지에서 분할되었다는 의미로 선행자에 객체 식별자 50을 저장한다. 또한 50번 필지는 유효시간의 끝이 98년 5월 6일이며 분할되었음을 기록하기 위하여 후행자에 70번과 80번의 객체 식별자를 저장한다. 이후 이력 검색에서 이러한 선행자

와 후행자를 검색함으로써 효과적인 이력 검색을 할 수 있게된다. 그림 10은 분할 연산의 수행 알고리즘

을 기술한 것이다.

```

## selected_obj를 new_obj1과
new_obj2로 분할한다
temp_obj << selected_obj
## Historical Value Setting.
selected_obj.event << "Split"
selected_obj.VTe << date.now()
(new_obj1, new_obj2)
<< split(selected_obj)
## Historical Relationship Setting
selected_obj.predecessor
<< new_obj1.id + new_obj2.id
new_obj1.successor
<< selected_obj.id
new_obj2.successor
<< selected_obj.id
    
```

〈그림 10〉 분할 연산 알고리즘

합병은 분할 연산의 반대 개념으로 2개 이상의 필지를 합병하여 하나의 필지를 생성하는 것이다. 다음의 그림 11과 그림 12는 합병 연산의 수행에 의한 시공간 데이터베이스의 상태변화와 수행 알고리즘을 기술한 것이다.

그림 11은 50번과 60번 필지가 98년 5월 6일 합병이 되어 80번 필지가 되었음을 의미한다.

분할 및 합병과 같은 연산을 통해 필지는 이력을 갖게되며, 그림 13과 같은 이력 그래프로 분할이나 합병에 대한 필지 이력을 파악할 수 있다.

그림 9의 이력을 가지는 시공간 객체들에 대해, "1960년에서 1970년 사이에 합병된 객체를 검색하라"는 질의가 있다고 한다. 현재 시간이 1980년이면 현재에 유효한 객체들인 101, 102, 103, 202번 객체로부터 검색이 시작된다. 즉, 현재 유효한 객체들의

OID	지번	지적도	VT-Start	VT-End	pred	succ
50	100	L100	97-3-1	now	{}	{}
60	100-1	L101	97-3-1	now	{}	{}
70	101	L102	97-3-1	now	{}	{}

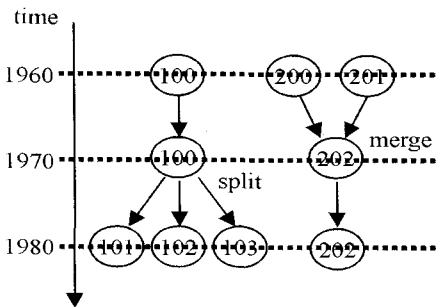
OID	지번	지적도	VT-Start	VT-End	pred	succ
50	100	L100	97-3-1	98-5-6	{}	{80}
60	100-1	L101	97-3-1	98-5-6	{}	{80}
70	101	L102	97-3-1	now	{}	{}
80	100	L103	97-5-7	now	{50, 60}	{}

〈그림 11〉 시공간 데이터베이스 변화(합병)

```

## target_obj를 selected_obj로 합병 new_obj
## 생성한다
T_obj_1 << selected_obj, T_obj_2
<< target_obj
## Historical value Setting.
selected_obj.event << "Merge"
selected_obj.VTe << date.now()
target_obj.event << "Merge"
target_obj.VTe << date.now()
## Do Merge
new_obj << merge(target_obj,
selected_obj)
## Historical Relationship Setting
new_obj.predecessor << selected_obj.id
+ target_obj.id
selected_obj.successor << new_obj.id
target_obj.successor << new_obj.id
    
```

〈그림 12〉 합병 연산 알고리즘



〈그림 13〉 객체의 이력 그래프

predecessor를 검색하여 객체식별자가 2개 이상이면 합병이 이루어진 것이므로 해당 객체의 predecessor를 반환한다. 그림 9의 이력 그래프에서 보면 주어진

시간에 합병이 이루어진 객체는 200번과 201번이다. 따라서 합병이 이루어진 202번 객체의 predecessor를 통해 합병된 객체가 200번과 201번임을 알 수 있다. 분할은 predecessor를 통해 이력 그래프상의 이전 객체를 찾아내고 그 객체의 successor의 객체식별자의 개수가 2개 이상이면 분할이 이루어진 것이므로 predecessor가 지시하는 이전 객체의 successor가 저장하고 있는 객체식별자를 통해 분할된 객체를 검색할 수 있다. 이러한 이력 검색을 위해 후입 선출형 자료구조인 스택(Stack)이 사용된다.

### 4.3 이력 검색 연산

이력 검색 연산은 3.2절에서 기술한 시간 연산자를 기반으로 주어진 조건을 만족하는 이력 객체들을 데이터베이스 내에서 검색하는 연산이다. 전체이력 검색 연산은 현재 상태의 지적 객체에 대한 모든 이력을 검색하여 출력하는 연산으로, “메인 윈도우에서 선택된 객체와 관련된 모든 이력을 검색하라”와 같은 질의가 가능하고, 검색 결과로써 이력 상태의 필드 정보를 출력한다.

전체 이력에 대한 검색에서 분할, 합병 연산의 경우 이전 또는 이후 이력이 복수개가 될 수 있다. 이럴 경우 탐색은 한쪽 방향으로만 진행되고 나머지 이력들은 후입 선출형 자료구조인 스택(Stack)에 저장되며 진행 방향으로의 탐색이 종료되면 스택에 있는 이력 객체를 찾아 다시 탐색을 시작한다. 그러므로 이력의 선 후관계와 출력된 순서와는 일치하지 않을 수 있다. 구현 알고리즘은 그림 14와 같다.

분할 및 합병으로 발생한 이력은 이전, 이후 이력이 2개 이상 존재 할 수 있다. 따라서 현재 객체로부터 이력을 검색할 때 여러 이력 객체로부터 특정 한 객체

```

## 스택이 비고 다음 이력이 없으면 종료.
_loop
_if result_tab.exist(current_rec)
_then _continue
_else stack.save(current_rec) _endif

_if current_rec.successor.size() != 0
_then
_if current_rec.successor.size() == 1
_then
    current_rec
    << current_rec.successor.getFirst()
_endif
_if current_rec.successor.size() == 2
_then
    current_rec
    << current_rec.successor.getFirst()
stack.push(current_rec.successor.getLast())
_else
    _if stack.isEmpty() = TRUE _then
        _end _endif
        current_rec << 스택.pop()
_endif
_endloop
    
```

〈그림 14〉 이력 검색 연산 알고리즘

```

Spatial_predicate
    << predicate.eq(event, "Split")
Start_predicate
    << predicate.gt(Start_Time, 1999/07/10)
End_predicate
    << predicate.lt(End_Time, 1999/07/20)
Time_predicate
    << Start_predicate _and End_predicate
result_collections
    << land_table.select(Spatial_predicate
        _and Time_predicate)
    
```

〈그림 15〉 사건 검색 연산 알고리즘

를 선택하여 검색을 하고 나머지 이력들은 스택에 해당 객체의 객체식별자를 push한다. 한 객체의 이력 검색을 다 마치고 나면 스택에서 다음 객체를 가져와 그 객체에 대한 이력 검색을 한다.

사건기반 검색연산은 주어진 사건의 종류와 시간의 범위를 만족하는 이력을 가진 객체의 현재 상태를 검색하는 것으로 검색 연산의 예로써, "1999년 7월 10일부터 1999년 7월 20일 사이에 분할이 발생했던 필지를 출력하라"와 같은 질의가 가능하다. 사건기반 검색연산은 검색 결과로써 이력이 아닌 현재 상태의 필지를 출력한다.

그림 15에서와 같이 사건기반 검색 연산은 공간상의 사건 유형과 사건이 발생한 시점을 피연산자로서 공간 predicate과 시간 predicate 구로 표현된다.

## 5. 시공간 연산자 적용

이 장에서는 4장에서 설계된 시공간 연산자를 실제 구현하여 시스템에 적용하고 각각의 시공간 연산자를 토지 관리에 적용한 결과를 설명한다. 토지이동 관리 연산과 이력 검색 연산에 대해서 기술하며 토지이동 관리연산의 경우 대표적 연산인 분할/합병 연산에 대해서만 기술한다.

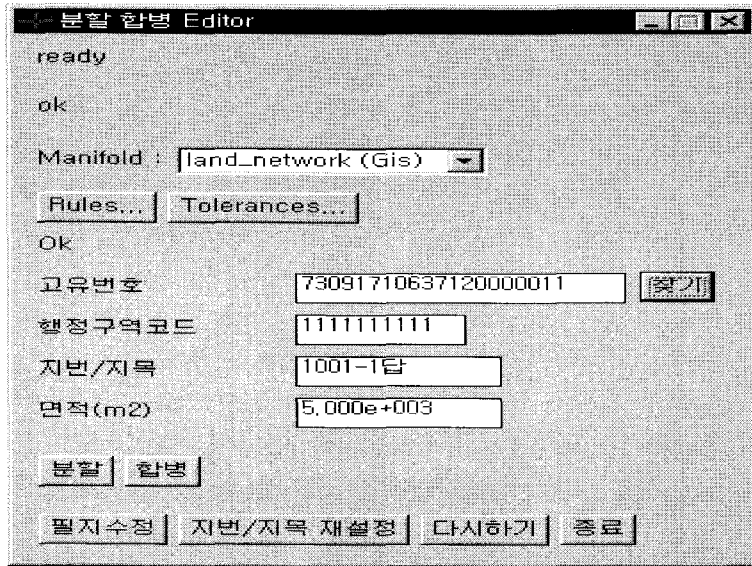
분할 및 합병 연산은 그림 16의 분할/합병 Editor에 의해 수행된다. 토지의 분할/합병에 의한 비공간 속성변화는 분할/합병 연산의 일부로 처리되므로, 분할/합병 Editor에서 비공간 속성변화를 처리할 수 있는 기능을 추가하였다.

다음의 그림 17은 분할 연산의 수행 과정을 나타낸다. 분할을 원하는 토지를 선택하고 트레일(Trail)로 분할 경계를 설정한 후에 분할을 실시하면 분할 경계에 따라 토지가 분할되며, 분할 이전의 토지는 이력으로 시공간 데이터베이스에 저장된다. 트레일은 스몰월드 시스템에서 공간상에 점, 선, 면 등의 공간객체를 작성하기 위해 제공되는 기능이다.

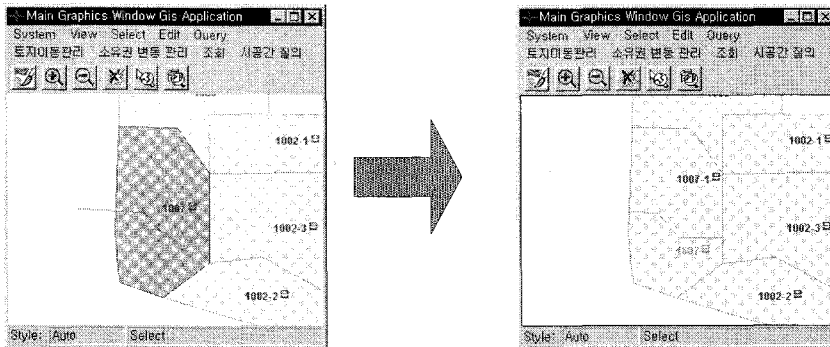
다음의 그림 18는 합병 연산의 수행 과정을 나타낸다. 기준 필지를 선택하고 기준 필지에서 대상 필지로 트레일을 그린 후에 합병을 수행하면 기준 필지에서 대상 필지로 합병이 수행되며 합병 이전의 토지는 이력으로 시공간 데이터베이스에 저장된다. 기준 필지는 합병 연산 수행 후에 비공간 속성 정보를 설정하는 기준이 되는 필지이다.

지금까지 설명한 토지이동 관리연산에 의해 발생한 이력들을 검색하여 사용자에게 출력하는 이력 검색 연산에는 사건기반 검색연산과 전체이력 검색연산이 있다. 먼저, 사건기반 검색연산은 특정 시점에 특정 이력을 가지는 토지객체를 검색하는 연산으로 연산의 결과는 현재 객체이다. 그림 19은 "1999년 7월 10일에서 1999년 7월 20일 사이에 분할이 발생했던 토지를 검색하라"는 질의를 작성한 화면이며 검색 결과는 그림 20의 형태로 사용자에게 출력된다.

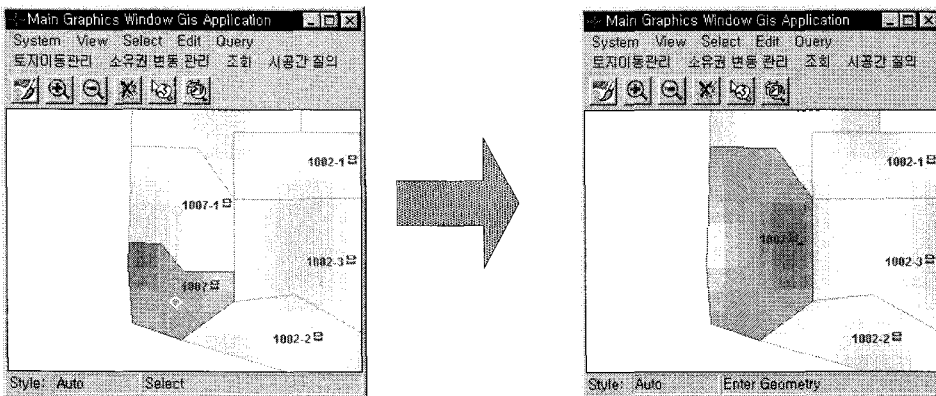
전체이력 검색연산은 특정 토지가 가지는 전체 이력 정보를 검색하는 연산으로 연산의 결과는 이력 객체이다. 그림 21는 "메인 윈도우에 선택된 토지의 전체 이력을 검색하라"는 질의를 수행하기 위한 화면이며 토지 고유번호를 입력해서 수행 할 수도 있다. 검색 결과는 그림 22의 형태로 사용자에게 출력된다.



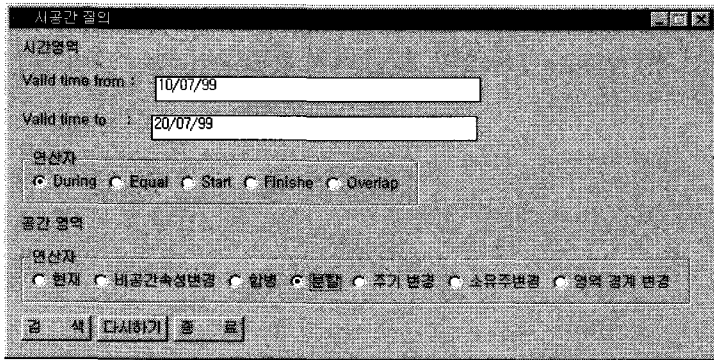
〈그림 16〉 분할/합병 Editor



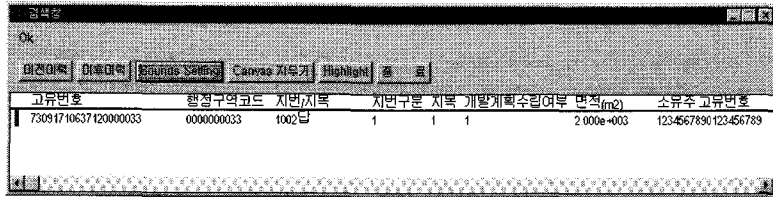
〈그림 17〉 분할 수행 결과



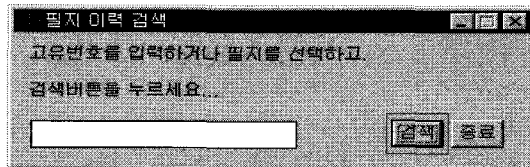
〈그림 18〉 합병 수행 결과



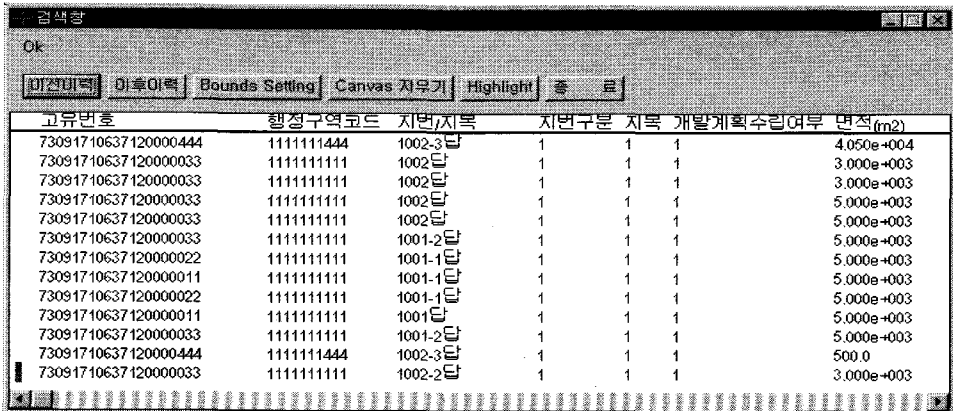
<그림 19> 사건기반 이력질의



<그림 20> 사건기반 이력질의 결과



<그림 21> 전체이력 검색질의



<그림 22> 전체이력 검색질의 결과

객체별 이력검색 연산은 그림 20와 그림 22에서 볼 수 있는 질의결과 출력화면의 기본 기능으로 제공된다. 이들은 검색 조건에 대한 결과 집합(Result Set)을 전달받아 출력하며 검색 결과 중에서 사용자가 선택한 토지정보에 대해 이전 이력이나 이후 이력을 조회하고 그들이 가지는 공간 정보를 출력할 수 있는 기능을 제공한다. 이를 통해 특정 객체의 시간 순서에 따른 이력정보들을 조회할 수 있다.

## 6. 결론

토지정보 시스템에서 토지는 시간이 경과함에 따라 소유주 변경 또는 소유주의 주소 변경, 필지의 지번 변경과 같은 비공간 속성 변화 및 토지의 분할, 합병, 경계변경과 같은 공간 속성 변화에 의해 토지객체의 이력정보가 발생한다. 그러나, 기존의 지리정보 시스템은 단순히 현재 존재하는 토지의 공간 또는 비공간 속성 정보만을 유지, 관리할 수 있는 기능을 제공한다. 그러므로 정보의 변경이 발생하면 기존의 정보는 데이터베이스 안에서 삭제되고 변경된 정보만이 기록된다. 이러한 방법은 토지에 대한 과거의 이력 정보를 유지하지 못하기 때문에 소유권 분쟁과 같은 문제 해결을 위한 증거 자료나 토지 정책 수립에 필요한 통계 정보의 제공 등과 같은 과거 상태를 필요로 하는 사용자 질의에 대해 정보를 제공하지 못한다.

이 논문에서는 토지대장 및 지적도의 이력관리를 위한 시공간 연산자를 제안하고 구현하였다. 이를 위해 스몰월드 지리정보 시스템의 공간 데이터베이스를 시간 차원으로 확장하였으며, 시간 및 시공간 연산자를 구현하였다. 또한, 발생한 이력들을 검색하여 사용자에게 적절한 결과를 보여줄 수 있는 이력 검색 연산자를 구현하였다. 이러한 이력 검색 연산은 사용자에게 다양한 검색방법을 제공하기 위해 사건기반 검색연산과 전체이력 검색연산으로 나누어 구현하였으며 각각의 검색 결과에 대해 이전 이력과 이후 이력을 검색할 수 있는 객체별 이력검색 기능도 지원하였다.

이 논문에서는 지적 객체의 이력관리를 위한 시간 표현에 있어 단순히 유효시간만을 제공하였지만 향후에는 거래시간을 포함한 이원시간을 지원함으로써 시간에 대한 더욱 다양한 표현이 가능하도록 하는 연구가 진행될 것이다.

## 참고문헌

[1] J. F. Allen, "Maintaining Knowledge about

temporal intervals", *Communication of the ACM*, Vol. 26, Nov. 1983.

- [2] M. P. Armstrong, "Temporality in Spatial Databases", *Proceedings: GIS/LIS*, 1988.
- [3] G. Langran, "A Framework for Temporal Geographic Information Systems", *Cartographica*, 25(3), 1998.
- [4] A. Montani and B. Pernici, "Temporal Reasoning", In *Temporal Databases: Theory, Design and Implementation*, The Benjamin Cummings Publishing Company Inc., 1993.
- [5] Donna J. Peuquet and Miu Duan, "An event-based spatiotemporal data model(ESTDM) for temporal analysis of geographic data", *International Journal of Geographic Information Systems*, 9(1), 1995.
- [6] A. Renolen, "The Spatio-temporal Object Model: A Conceptual Model Based on Ontology", article submitted to *Geoinformatica*, 1997.
- [7] A. Segev and A. Shoshani, "A Temporal Data Model Based on Time Sequences", In *Temporal Databases, Theory, Design, and Implementation Benjamin/Cummings Publishing Company Inc.*, 1993.
- [8] Richard T. Snodgrass, and Ahn, I., "Temporal databases", *IEEE Computer* Vol. 19, No. 9, Sep. 1986.
- [9] Richard T. Snodgrass, "The TSQL2 Temporal Query language", *Kluwer Academic Publishers*, 1995.
- [10] Smallworld, "Smallworld 3 개요", *Smallworld 3 Overview*, page 5~7, 1998.
- [11] M. Yuan, "Use of a Three-Domain Representation to Enhance GIS Support for Complex Spatio-Temporal Queries", In *Press by Transactions in GIS*, 1997.
- [12] M. F. Worboys, H. M. Hearnshaw and D. J. Maguire, "Object-Oriented Modelling for Spatial Databases", *International Journal of GIS*, 4(4), 1990.
- [13] 김동호, "시공간 데이터베이스 질의처리 시스템", 충북대학교 전자계산학과, 박사학위논문, 1999

년 2월.

- [14] 내무부, "제 3장 지적정리", 내무부 지적사무전 산처리규정, page 8~15, 1999년.
- [15] 이화중, "인터넷기반 토지대장/지적도의 이력관리 시스템의 설계 및 구현", 충북대학교 전자계산학과, 석사학위논문, 1999년 2월.
- [16] 배종철, 박경현, 이화중, 류근호, "토지대장/지적도를 위한 이력관리 시스템 설계 및 구현", KDBC2000(Korean Database Conference 2000), page 82~91, 2000년.



**김성통**

1980년 광운대학교 무선통신공학과 졸업  
 1988년 한양대학교 대학원 전자공학 석사  
 1996년-현재 충북대학교 대학원 전산학과 박사과정  
 1999년-현재 한국통신 정보기술㈜ 대표이사  
 관심분야 : 공간 데이터베이스, 지리 정보시스템, Web GIS 등



**김상호**

1997년 충북대학교 컴퓨터과학과 졸업  
 1999년 충북대학교 대학원 전산학과 석사  
 1999년-현재 충북대학교 대학원 전산학과 박사과정  
 관심분야 : 시공간 데이터베이스, Web Visualization, 시공간 뷰 등



**이화중**

1996년 충북대학교 컴퓨터과학과 (이학사)  
 2000년 충북대학교 대학원 전산학과 이학석사  
 2000년-현재 이니텍㈜ 근무  
 관심분야 : 시간 데이터베이스, 시공간 데이터베이스, 객체지향 데이터베이스, GIS



**류근호**

1976년 숭실대 전산과 졸업  
 1980년 연세대학교 산업대학원 전산전공 (공학석사)  
 1988년 연세대 대학원 전산전공(공학박사)  
 1976-1986년 육군군수지원사전산실 (ROTC장교), 한국전자통신연구소(연구원), 한국방송통신대, 전산학과(조교수)  
 1989-1991년 Univ. of Arizona 연구원 (TempIS 연구원, Temporal DB)  
 1986년-현재 충북대학교 전기전자 및 컴퓨터공학부 교수  
 관심분야: 시간 데이터베이스, 시공간 데이터베이스, Temporal GIS, 객체 및 지식기반 시스템, 지식 기반 정보검색시스템, 데이터 마이닝, Bio-Informatics 등