

공간데이터 일반화의 파급을 처리하기 위한 규칙

Rules for Control Propagation of Geospatial Data Generalization

강혜경(姜惠卿)* · 이기준(李基準)**

He-Gyoung Kang , Ki-Joune Li

요약 공간데이터의 일반화는 기존에 구축된 공간 데이터베이스로부터 새로운 소속적 데이터베이스를 유도할 수 있는 중요한 GIS 기법이다. 공간데이터의 일반화는 공간데이터의 기하 및 속성데이터를 변형(3, 15)시킬 뿐만 아니라, 데이터 모델의 관계를 따라서 연결되어 있는 다른 공간데이터도 변형(8-10,14)시킨다. 이것을 공간데이터 일반화의 파급이라고 한다. 이 파급을 처리하지 않은 채 일반화를 계속 진행하면, 일관성 혹은 원시데이터베이스 정보 중의 일부가 손실된 채 새로운 데이터베이스가 생성될 수 있다. 그럼에도 불구하고 일반화에 관한 기존 연구들은 공간데이터의 상호관계를 무시한 채 독립된 하나의 공간데이터에 대한 유도를 위해서 방법들을 제시해 왔다. 그리고 그 결과 공간데이터의 기하 및 속성을 변형시키는 많은 일반화 연산자들이 제시되어왔다. 본 연구는 이 일반화 연산자들이 어떤 공간데이터에 적용되었을 때 그와 관련된 다른 공간데이터에도 파급 적용될 수 있도록, 일반화 연산자를 확장을 시킬 것이다. 이 일반화 파급을 처리하기 위해서, 본 연구는 일반화 과정에서 반드시 고려될 필요가 있는 규칙들을 제시한다. 그리고 일반화 연산자들이 반드시 준수해야 하는 규칙들을 기술한다. 이 규칙들은 관계대수로서 표현될 수 있으므로, SQL로 쉽게 전환할 수 있다. 이 확장된 일반화 연산자들의 적합성을 검토하기 위해서 간단한 프로토타입을 구현하였다.

ABSTRACT The generalization of geospatial data is an important way in deriving a new database from an original one. The generalization of a geospatial object changes not only its geometric and aspatial attributes but also results in propagation to other objects along their relationship. We call it generalization propagation of geospatial databases. Without proper handling of the propagation, it brings about an inconsistent database or loss of semantics. Nevertheless, previous studies in the generalization have focused on the derivation of an object by isolating it from others. And they have proposed a set of generalization operators, which were intended to change the geometric and aspatial attributes of an object. In this paper we extend the definition of generalization operators to cover the propagation from an object to others. In order to capture the propagation, we discover a set of rules or constraints that must be taken into account during generalization procedure. Each generalization operator with constraints is expressed in relational algebra and it can be converted to SQL statements with ease. A prototype system was developed to verify the correctness of extended operators.

키워드 : 지도 일반화, 모델 일반화(데이터 베이스일반화)

1. 서론

공간데이터 일반화란 공간데이터를 보다 덜 자세한 형태로 변형시키는 것이다. 이 변형은 크게 두 가지 작업으로 구성되는데, 공간데이터의 기하적 변형과 데이터 모델의 변형이 해당한다. 일반화가 진행되는 중에 새로운 클래스나 관계가 생성되는 것이 모델 변형

에 해당한다. 기존 연구들은 이 일반화를 사용하는 목적을 기존에 구축된 공간데이터를 축척에 적합한 형태로 변형함으로써 새로운 정보를 생성하기 위한 것(2, 13)과, 기존에 구축된 공간 데이터베이스로부터 풍부한 정보를 유도하는 것(3, 6-8, 12)으로 정의하고 있다(15). 이를 위하여 많은 일반화 연산자들이 제안되었다.

그러나 이 기존 연구들은 일반화가 파급된다는 사실

* 미국ESRI 연구원

** 부산대학교 정보컴퓨터공학부 부교수

hkang@esri.com

lik@hyowon.cc.pusan.ac.kr

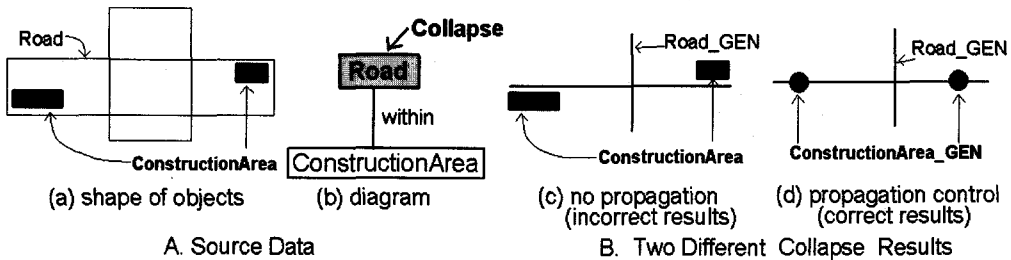
을 고려하지 않고 있다. 즉, 어떤 공간데이터의 변형은 다른 공간데이터의 변형을 초래[10,12]하며, 이것은 데이터 모델 안에서 관계를 따라서 파급된다. 이것을 본 연구에서는 일반화 파급이라고 한다. 이 파급의 처리하지 않을 경우, 새 데이터베이스는 일관성이 손실된 채 생성될 수 있다. 그러나 이 파급 처리에 관한 기존 일반화 연구들은 아직 미흡한 실정이다. 그러므로 본 연구는 일반화 파급 처리를 위한 방법을 제시하고자 한다. 다음 예는 일반화 파급을 처리하지 않을 경우 발생할 수 있는 문제점과 본 연구의 필요성을 동시에 보여주고 있다.

1:500 축척의 공간 데이터베이스 GD500안에 두개의 클래스 도로(Road)와 도로공사지역(ConstructionArea)이 있다. 도로 클래스는 도로의 이름과 도로경계를 면으로 표현하고 있다. 그리고 도로공사지역 클래스는 공사중인 도로의 일부를 면으로 표현하고 있으며, 반드시 도로의 내부(within)에 존재한다. 이때 우리는 도로의 면을 선으로 축소(collapse)시킴으로서 1:1000 축척의 도로를 유도하려고 한다. 그리고 다음 그림1은 원시 데이터베이스 GD500에 저장된 데이터 형태와 두 가지 서로 다른 일반화 결과를 보여준다.

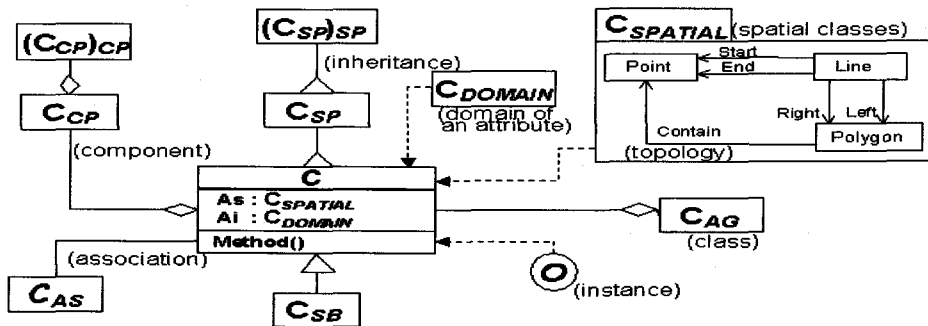
그림 1-(c)는 도로클래스가 선으로 축소되었음에도

불구하고, 도로에 의해서 포함되어 있던 면형 공간데이터는 선으로 축소되지 않은 것을 보여준다. 반면그림 1-(d)는 도로공사지역이 점으로 축소됨으로서, 여전히 도로에 의함 '포함' 위상관계가 유지되고 있는 것을 보여준다. 즉, 첫 번째 일반화 결과는 도로와 도로공사지역의 위상관계가 틀리게 변했지만, 두 번째 일반화 결과는 위상관계가 보호되고 있다. 그러므로, 일반화 결과의 무결성을 보호하기 위해서는 일반화 파급을 고려할 필요가 있다.

본 연구는 다음과 같이 구성되어 있다. 먼저 2장에서는 공간데이터 모델을 정형화 한다. 그리고 기존 연구들로부터 제안된 일반화 연산자들 중에서, 본 연구에서 사용할 여섯 가지 일반화 연산자를 선정한다. 3장에서는 일반화 파급을 처리하기 위한 규칙들을 제안한다. 이를 위해, 먼저 일반화 연산자가 지켜야 할 조건(constraint)을 정의한다. 그리고 일반화 연산자들이 각 조건들을 위반하지 않도록 규칙을 제시한다. 이 규칙들은 관계대수에 의해서 표현될 수 있다. 또 SQL로도 쉽게 표현될 수 있는데, 4장에서 이것에 대한 간단한 예를 기술한다. 마지막으로 본 연구에서 개발한 프로토타입을 설명하고 이 논문을 결론짓겠다.



<그림 1> 공간데이터 일반화 예



<그림 2> 일반 공간 데이터 모델 다이어그램

2. 공간데이터 모델과 일반화 연산자

2.1 일반 공간데이터 모델

일반화의 파급은 연관, 상속, 집합과 같은 데이터 모델의 관계를 따라서 발생할 수 있다. 그래서 본 연구에서 사용할 공간데이터 모델의 일반적인 형태를 먼저 정의할 필요가 있다. 다음 클래스 다이어그램은 본 연구에서 사용할 일반 공간데이터 모델을 보여준다. 이 모델은 객체지향[2]과 지형지물[7] 개념을 바탕으로 정의된 것이다. 이 모델의 공간 데이터는 2차원 벡터 공간으로 제한한다.

2.2 일반화 연산자

일반화 연산자들은 일반화 과정에서 발생하는 다양한 기하적 변형을 표현하기 위해서 기존 연구들[1,4,6,8-9]에 의해서 제안되어져 왔다. 이 일반화 연산자들은 표 2에서 나타나는 것처럼, 연산자들의 특징적인 주요 기능을 기준으로 몇 가지 그룹으로 분류할 수 있다. 이 중에서 우리는 정의가 가장 명확하고 단순한 [6]에서 제시한 일반화 연산자들을 선택하였다. 여섯 가지 연산자들의 정의와 예는 표3과 같다.

3. 일반화 파급 제어를 위한 규칙

1장의 예에서 보여준 것처럼, 데이터 모델안에서 공간데이터의 일반화는 관계를 따라서 파급된다. 이 장에서는 일반화 연산자들에 의해서 파급될 수 있는 위반유형을 조사하여 이를 보호하기 위한 조건을 설정한다. 그리고 이 조건들을 준수하는 일반화 규칙들을 제안한다. 그러면 먼저 반드시 통제될 필요가 있는 일반화 파급과 선택적으로 통제할 필요가 있는 일반화 파급을 기술하겠다.

3.1 일반화의 필수 및 선택 조건(constraint)

일반화 과정에서 일반화 연산자는 공간데이터를 선택/삭제/분류하거나 공간형태를 변형시킨다. 이러한 작용이 이루어지는 동안, 일반화 연산자는 공간데이터의 일관성(consistency)이나 적합성(correctness)을 위반할 수 있다. 또한 이 연산자들은 공간데이터간의 관계를 무시하고 어떤 한 공간데이터만을 독립적으로 처리함으로써, 원시 데이터 베이스에 있는 일부 정보들을 손실할 수 있다.

일반화 파급에 의해 일반화 결과의 일관성 및 적합성이 손상되는 경우는 반드시 보호되어야 한다. 이것은 다음처럼 일반화 연산자에 대한 무결성 조건들을 정의함으로써 가능하다. 여기서 사용되어지는 데이터 모델 관련 기호들은 표 1에 기술되어 있다.

〈표 1〉 데이터 모델 관련 기호 및 의미

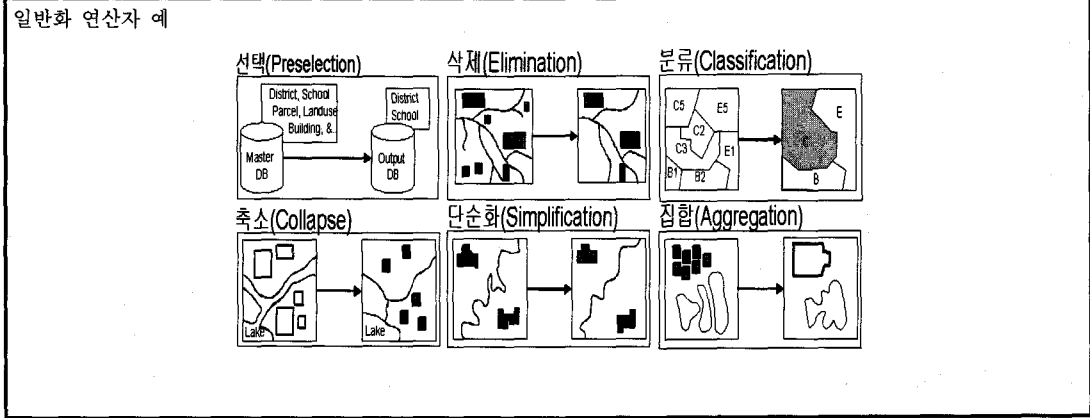
기 호	의 의	기 호	의 의
<i>C</i>	주어진 클래스	<i>O</i>	<i>C</i> 의 객체 혹은 인스턴스
<i>C_{SP}</i>	<i>C</i> 의 상위(super) 클래스	<i>C_{SB}</i>	<i>C</i> 의 하위(sub) 클래스
<i>C_{CP}</i>	<i>C</i> 의 부분(component) 클래스	<i>C_{AG}</i>	<i>C</i> 의 집합(agggregation)클래스
<i>C_{AS}</i>	<i>C</i> 의 연관(association)클래스	<i>C_{DOMAIN}</i>	<i>C</i> 의 속성의 도메인 클래스
<i>C_{SPATIAL}</i>	<i>C</i> 의 공간 도메인	<i>dom(C)</i>	<i>C</i> 의 도메인

〈표 2〉 일반화 연산자의 분류

기능	[1]	[6]	[8]	[9]	[10]
선택	Select	Preselection	Selection	Selection	Selection
삭제	Omit	Elimination	-	Deletion	-
단순화	Coarsen	Simplification	-	-	Simplification
축소	Collapse	Collapse	-	Collapse	Symbolisation
분류	Classfy	Classification	Generalization	Reclassification	-
집합	Combine	Aggregation	-	Aggregation Hogenization Universalization	Aggregation

〈표 3〉 일반화 연산자의 정의

일반화 연산자	정의
선택(preselection)	원시 데이터베이스로부터 일부 클래스들을 선택.
삭제(elimination)	불필요한 공간데이터를 선택적으로 삭제.
분류(classification)	동일한 속성값을 가지는 인접 공간객체들을 하나로 통합.
축소(collapse)	공간객체의 공간차원을 감소시킴.
단순화(simplification)	공간객체의 차원은 유지한 채, 정확도를 감소시킴.
집합(aggregation)	근접 추정 연산에 의해 공간객체들을 하나로 통합.



조건 1 (도메인 무결성)

어떤 클래스 C 가 선택되면, C 의 도메인 클래스도 또한 반드시 선택되어야 한다. 새로운 클래스 C_{GEN} 이 C 의 하위 셋이면, C_{GEN} 의 도메인 $dom(C_{GEN})$ 도 C 의 도메인 $dom(C)$ 의 하위 셋이다($C_{GEN} \subset C$ iff $dom(C_{GEN}) \subseteq dom(C)$). 선택(preselection)이나 분류(classification)와 같이, 공간데이터를 선택하거나 새로운 공간데이터를 생성하는 일반화 연산자들은 이 조건을 반드시 준수해야 한다.

조건 2 (참조 무결성)

어떤 클래스 C 에 있는 공간데이터 O 가 C 와 관계가 있는 클래스 C_0 안에 있는 O_0 를 참조할 때, 만약 O_0 가 삭제되면, O 의 참조 값도 반드시 삭제되어야 한다. 이것은 관계 데이터베이스에서 참조 무결성과 유사하다. 이 조건은 삭제(elimination)처럼, 공간데이터를 삭제하는 일반화 연산자가 반드시 준수해야 한다.

조건 3 (위상 무결성)

어떤 클래스 C 에 있는 공간데이터 O 의 공간형태가

변형되더라도, 어떤 클래스 C 에 있는 공간데이터 및 다른 클래스 C_0 에 있는 공간데이터 간의 위상은 반드시 유지되어야 한다 ($topology(C) \Leftrightarrow topology(C_{GEN})$). 이 조건은 축소(collapse), 단순화(simplification), 분류(classification) 혹은 집합(aggregation)처럼, 공간데이터의 공간적 형태를 변형시키는 연산자들이 반드시 지켜야 한다.

다음으로 공간데이터를 관계를 무시한 채, 독립적으로 처리함으로써 발생하는 정보의 손실은 공간데이터 간의 관계를 고려함으로써 방지할 수 있다. 데이터 모델에서 관계는 의미(semantic) 및 구조(schematic) 상관성을 표현한다. 예를 들어, 어떤 클래스 C 는 C_{SP} 로부터 상속을 받아 생성된 클래스이다. 이것은 C 의 스키마가 C_{SP} 없이는 불완전함을, C 의 의미(semantic)는 C_{SP} 의 의미에 포함됨을 나타낸다. 따라서 C 를 선택할 때는 C_{SP} 도 함께 선택해야 한다. 이를 위해 다음과 같은 조건들을 정의할 수 있다. 이 조건들은 선택(preselection), 분류(classification), 집합(aggregation)처럼, 원시 데이터베이스로부터 클래스를 선택하거나 새로운 정보를 생성시키는 일반화 연산자들과 함께 사용된다.

조건 4 (스키마 완전성)

위에서 기술한 것 처럼, C의 스키마는 상위 클래스 C_{SP}의 스키마 없이는 불완전해지므로, 어떤 클래스 C가 선택되면, C의 상위 클래스CSP도 또한 반드시 선택되어야 한다(스키마(C) 스키마(C_{SP})). 이 조건은 선택(preselection)처럼, 클래스를 선택하는 일반화 연산자가 반드시 지켜야 한다.

조건 5 (구조 및 의미적 포함)

어떤 클래스 C_{SP}에 일반화 연산자가 적용되면, C_{SP}의 하위 클래스 C에도 일반화 연산자가 파급 적용되어야 한다. 왜냐하면, 객체지향 개념에 의해서 하위 클래스 C는 C_{SP}에 의해 의미적(semanticly) 그리고 스키마적(schematically)으로 포함되기 때문이다 (C ⊂ C_{SP}).

조건 6 (의미적 완전성)

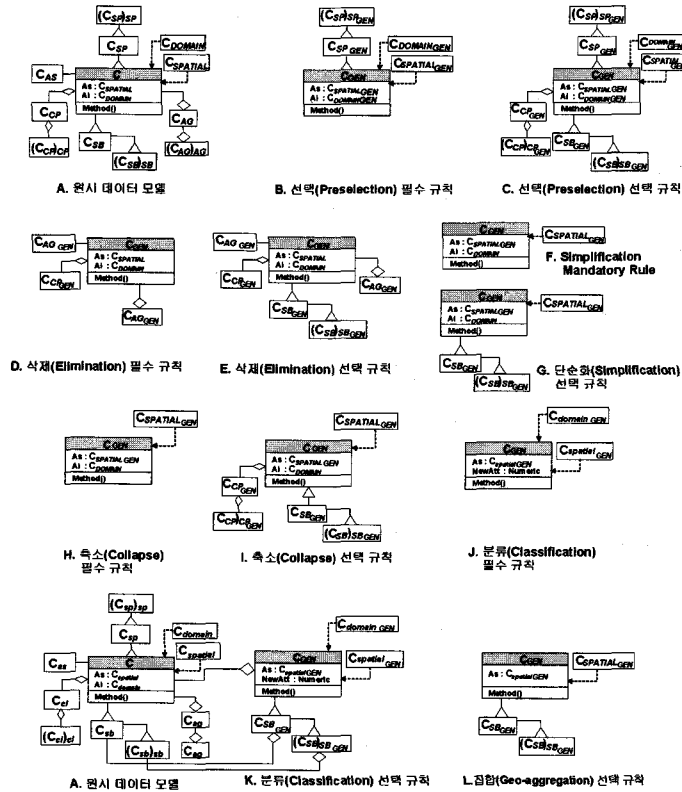
어떤 클래스 C가 선택되어 졌을 때, C_{CP} 없이는 C가 의미적으로 불완전하다. 객체지향개념에서 C_{CP}를C의 '부분(part of)'이라고 하고, C는 C_{CP}로 '구성(consist of)' 된다고 한다. 그러므로 C와 C_{CP}의 관계를 의미(C) ⊃ 의미(C_{CP})라고 표현할 수 있다. 그러 므로 C를 선택했을 때 의미적으로 완전해지기 위해서는, C_{CP}도 함께 선택되어야 한다. 이 조건은 공간데이터를 선택하거나 새로 생성하는 일반화 연산자들, 즉 선택(preselection), 축소(collapse), 집합(aggregation) 혹은 분류(classification)와 같은 연산자와 함께 사용된다.

이상과 같은 조건들은 일반화 연산자들이 원시 데이터베이스로부터 불완전한 정보를 유도하거나, 정보를

〈표 4〉 일반화 연산자와 규칙들

연산자	규칙	설명	Cnst.	M/O	관계대수
선택	R1	$C_{DOMAIN}, C_S, (C_{SP})_{SP}$	C1, C4	M	$- C_{GEN}, \leftarrow II(C) \cup C_{GEN}, II(C_\theta)$
	R2	$C_{SB}, (C_{SB})_{SB}$	C5	O	
	R3	$C_{CP}, (C_{CP})_{CP}$	C6	O	
삭제	R4	C_{CP}, C_{AG}, C_{AS}	C2	M	<ul style="list-style-type: none"> - $C_{GEN}, \leftarrow \sigma_{o(A_i)} \ast \alpha (C)$ - Modify $O_\theta[R]$ to null where $O_\theta[R]$ refer to eliminated objects $O' [A]$ and $O_\theta \in C_\theta$. - $C_{GEN}, \leftarrow \sigma (C_\theta)$ where C_θ is already modified.
	R5	$C_{SB}, (C_{SB})_{SB}$	C5	O	
단순	R6	-	C3	M	
	R7	$C_{SB}, (C_{SB})_{SB}$	C5	O	
축소	R8	-	C3	M	
	R9	$C_{SB}, (C_{SB})_{SB}$	C5	O	
	R10	$C_{CP}, (C_{CP})_{CP}$	C6	O	
분류	R11	C_{DOMAIN}	C1,C3	M	<ul style="list-style-type: none"> - $C_{GEN}, \leftarrow A_i \uparrow A_s, A_i, A_c(C)$ - $C_{GEN}, \leftarrow A_i \uparrow A_s, A_i, A_c (C_\theta)$ where C_{GEN} are inherited from C_{GEN}.
	R12	$C_{SB}, (C_{SB})_{SB}$	C5	O	
	R13	$A_{compuatable}, A_{condition}, A_s$		O	
집합	R14	$C_{SB}, (C_{SB})_{SB}$	C5	O	<ul style="list-style-type: none"> - $C_{GEN}, \leftarrow v \uparrow A_s (C)$ - $C_{GEN}, \leftarrow v \uparrow A_s (C)$ where C_{GEN} are inherited from C_{GEN}.

Cnst. : 제약조건, M/O: 필수(M) 선택(O), θ : relationship, C : related classes, \ast : elimination condition, $O(A_i)$: 객체 O의 속성 A_i 의 값, \uparrow : 집합 함수, A_s :공간속성, $A_{compuatable}$:계산가능 속성, v : 거리추정을 위해 주어진 조건 값



(그림 3) 일반화 연산자와 규칙들-다이어그램

손실한 채 새로운 데이터베이스를 생성할 때 제한자로서 작용한다. 이 조건들을 만족시키기 위해서 다음과 같은 일반화 규칙들이 정의될 필요가 있다.

3.2 일반화 연산자와 규칙

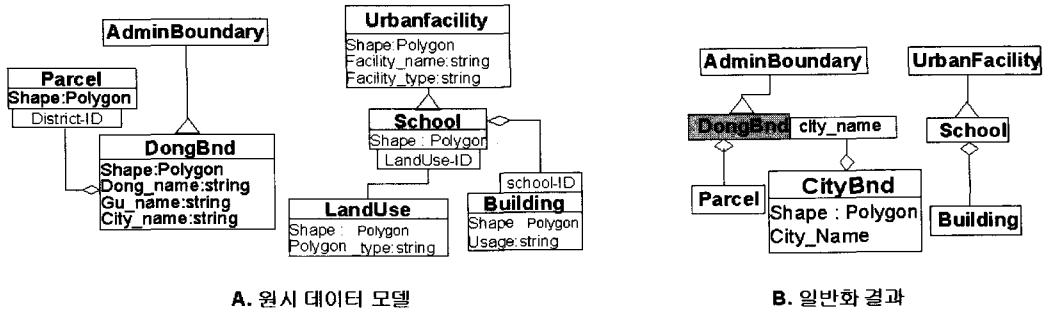
여기서는 여섯 가지 일반화 연산자에 대해서 규칙들을 제시한다. 어떤 규칙들은 반드시 일반화 연산자와 함께 사용되어야 하며, 어떤 규칙들은 선택적으로 사용될 수 있다. 이 규칙들을 표 5에 정리하였다. 표에서, '설명' 열은 주어진 클래스 C와 함께 일반화 연산자가 적용되어야 할 C의 연관 클래스들을 보여준다. 이 클래스들은 다음 열 'Cnst.'에 명시된 조건들에 의해서 선택되어진다. 다음 열인 'M/O'에서 M은 반드시 일반화 연산자와 함께 사용되어야 할 규칙을, O는 선택적으로 사용 가능한 규칙을 의미한다. 마지막 열 'RelationalAlgebra'은 규칙을 관계대수로 표현한 것이다. 이 관계대수는 데이터 모델의 변형을 표현한 것으로, 공간데이터의 기하적 변형은 사용자가 구현한

함수로서 표현한다. 그림 3은 원시 데이터 모델에 이 규칙들을 적용한 결과를 보여준다. 예를들어, 그림 3-B는, 선택(preselection)연산자를 어떤 클래스 C에 적용했을 때, 이 연산자를 C의 상위 클래스에도 적용해야 하며, 도메인 클래스에도 파급 적용해야 함을 의미한다.

4. 사례 연구

표 4에서 우리는 규칙들을 관계대수로 표현하였다. 이 장에서는 관계대수를 SQL 구문으로 표현하는 것을 간단한 예를 이용해서 보여준다. 사례에 사용될 원시 데이터 모델은 그림 4와 같다. 이 데이터 모델은 국토연구원의 1:5000 도지 데이터베이스중의 일부이다. 이 데이터베이스로부터 1:1000 데이터베이스를 유도하기 위해서, 다음과 같은 일반화 초기 조건들을 가정하였다.

- 조건 1 : 새로운 데이터베이스는 시경계와 학교



〈그림 4〉 원시 데이터 모델과 일반화 결과(예상)

를 가진다. 시경계는 동경계의 시이름 속성을 재 분류함으로써 생성된다.

- 조건 2 : 시경계와 학교의 단순화를 위해 허용 오차를 100으로 설정한다.
- 조건 3 : 수원시에 속하는 동이 아니면, 동경계를 삭제한다.

이 조건들을 만족하는 1:1000 데이터베이스를 유도하기 위해서, 일반화 연산자들과 몇몇 규칙들이 함께 사용되는 예를 이후부터 SQL로 기술하겠다. 본 연구에서는 연산자 순서에 따른 일반화 결과의 차이는 무시하며, 일반화 연산자는 선택, 삭제, 단순화, 분류 순서로 사용한다. 원시 데이터 모델은 다음 그림 4-(A)와 같으며, 일반화 과정의 결과로서 기대되는 데이터 모델은 그림 4-(B)와 같다.

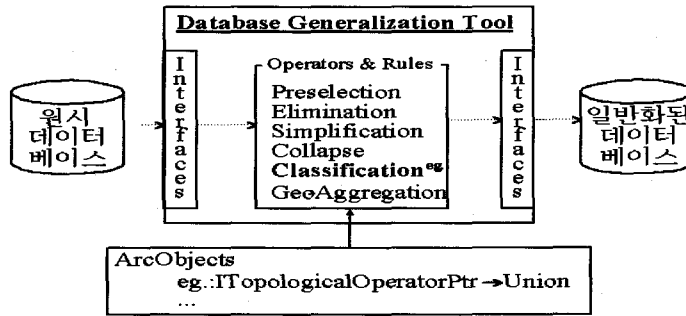
- 단계 1 (선택 연산자와 규칙 1, 3) : 선택 (preselection) 연산자를 학교와 동경계에 적용
- SELECT school.* INTO schoolPRE FROM school;
- SELECT urbanfacility.* INTO urbanfacilityPRE FROM urbanfacility; %(R1 적용)
- SELECT building.* INTO buildingPRE FROM building; %(R3 적용)
- SELECT dongbnd.* INTO dongPRE FROM district;
- SELECT adminboundary.* INTO adminboundary PRE FROM urbanfacility; %(R1 적용)
- SELECT parcel.* INTO parcelPRE FROM building; %(R3 적용)
- 단계 2 (삭제 연산자와 규칙 4) : 삭제 (elimination) 연산자를 1단계 결과인 동경계에 적용.

- SELECT dongPRE: INTO dongELI FROM dongPRE WHERE city_name = 'suown' ;
- SELECT parcelPRE: INTO parcel ELI FORM parcelPRE;
- UPDATE parcelELI SET code = null WHERE NOT EXISTS (SELECT dong_name FROM dongELI dELI WHERE code = dELI.dong_name); %(R4 적용)
- 단계 3(단순 연산자와 규칙 6) : 단순 (simplification) 연산자를 2단계 결과인 동경계에 적용.
- SELECT simplifiatio(dELI:shape), dELI.dong_name, dELI.dong_name, dELI.gu_name INTO dongSIM FROM dongELI dELI; %(simplification 적용)
- 단계 4(분류 연산자와 규칙 13) : 분류 (classification) 연산자를 3단계 결과인 동경계에 적용.
- SELECT classifcation(dSIM:spahe), dSIM.city_name, INTO citybnd FROM dongSIM dSIM GROUP BY city_name; %(classification, R13 적용)

단계 2와 3은 일반화 결과의 무결성을 위해서 반드시 지켜져야 하는 규칙인 반면, 단계 1과 4는 원시 데이터베이스에 있는 의미적 정보를 보다 많이 유도하기 위해서 규칙이 사용되었다.

5. 프로토타입

본 연구는 일반화가 데이터 모델상에서 관계를 따라



〈그림 5〉 프로토타입 구조

〈그림 6〉 일반화 결과

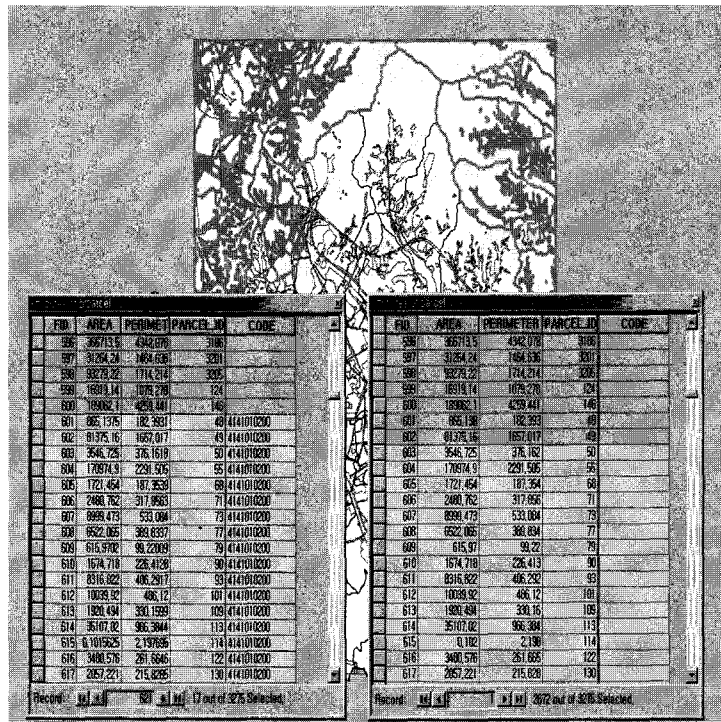


그림 6-(A) 삭제(elimination)와 규칙 4가 적용된 예

파급되는 것을 제어하기 위하여, 기존의 일반화 연산자들에 제약조건을 부여하고, 일반화 연산자들과 함께 사용해야 할 규칙들을 제안하였다. 이 규칙들과 일반화 연산자들이 함께 사용된 일반화 결과를 검토하기 위하여 간단한 프로토타입을 개발하였다. 이 프로토타입은 Arcobjects와 VC++을 이용해서 구현되었으며, 그 구조는 그림 5와 같다.

이 프로토타입 개발은, 먼저 ArcObjects에서 제공하는 기본적인 함수들을 조합하여 기존에 제안된 6가지 일반화 연산자들을 구현하였다. 여기에 본 연구에서 제안한 규칙들을 통합시킴으로써, 6가지 일반화 연산자들을 확장하였다. 그림 5에서 분류(classification)의 경우, Arcobjects의 ITopologicalOperatorPtr의 Union()을 이용해서 구현되었음을 보여준다.

다음 그림들은 이 프로토타입을 이용해서 생성한 4장의 단계 2, 3, 4에 해당하는 일반화 결과들이다. 그림 6-(A)는 수원시에 포함되지 않는 동경계를 삭제(elimination)했을 때, 연관관계를 가지고 있는 지적(parcel) 클래스의 참조 무결성을 위해 규칙 4가 적용되었으며, 그 결과 지적(parcel)클래스의 code속성이 null로 대체된 것을 보여준다. 그림 6-(B)는 단순(simplification) 연산자와 규칙 6을 동경계에 적용한 결과인데, 규칙 6에 의해 위상관계가 그대로 유지되고 있음을 보여준다. 그림 6-(C)는 동경계로부터 시경계를 새로 생성한 결과인데, 분류(classification)연산자와 규칙 13이 함께 사용된 결과 새로운 속성이 생성되었다.

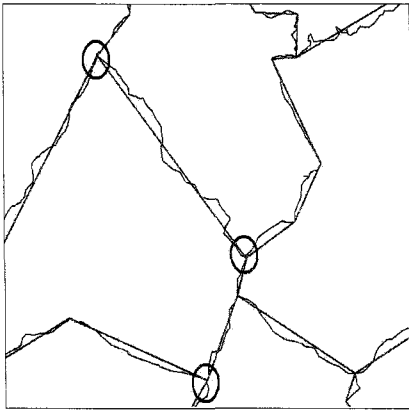


그림 6-(B) 단순(simplification)과 규칙 6이 적용된 예

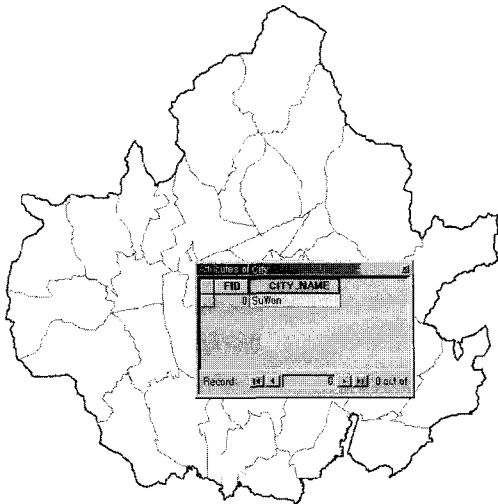


그림 6-(C) 분류(classification)과 규칙 13이 적용된 예

6. 결론

기존의 일반화 연구들은 일반화를 공간데이터를 축척에 적합한 공간데이터로 변형시키는 과정이라고 정의하였다. 이때 어떤 공간데이터의 변형은 다른 공간데이터에 영향을 미치며, 이것은 데이터 모델 안에서 관계를 따라 파급되어진다. 이 일반화의 파급을 고려하지 않을 경우, 이것은 일관성이 깨지거나 정보가 손실된 결과를 초래하게 된다. 그럼에도 불구하고, 기존 연구들은 이 파급이 발생하는 것을 간과한 채, 공간데이터의 기하적 변형에 초점을 맞추어 왔다. 그리고 이들의 연구결과는 일반화 연산자의 형태로 나타나고 있다. 그래서 본 연구에서는 이 기존 연구들에 의해 제안된 일반화 연산자들을 파급을 처리할 수 있도록 확장할 것을 제안하였다. 이를 위해 일반화 제약조건과 규칙들을 제안하였다. 이 규칙들은 일반화가 진행되는 동안 데이터의 일관성과 무결성을 보호하며, 원시 데이터베이스의 의미적 정보도 유도했다. 그 결과 일반화 결과의 질도 향상되었으며, 보다 많은 정보를 유도할 수 있었다. 이러한 본 연구의 기여는 Arcobjects와 VC++을 이용해서 개발된 프로토타입을 통해서 검증할 수 있었다.

참고문헌

- [1] K. Beard, and W. Mackness, "Generalization Operations and Supporting Structure", Proc. Auto-Carto, Vol 10, pp.35-39, 1991.
- [2] J. Rumbaugh et al, Object-Oriented Modeling And Design: Object Modeling, Advanced Object Modeling, Prentice-Hall Inc., 1991.
- [3] R. B. McMaster, and K. S. Shea, Generalization in Digital Cartography, The Association of American Geographers, 1992.
- [4] J. P. Lagrange, and A. Ruas, "Geographic Information Modelling: GIS and Generalisation", Proc. SDH, pp.1100-1103, 1994.
- [5] D. E. Richardson, "Generalization of Spatial and Thematic data using inheritance and classification and aggregation hierarchies", Proc. SDH, pp.957-972, 1994.

[6] ESRI, Automation of Map Generalization, 1996, ESRI.

[7] A. Y. Tang, T. M. Adams, and E. L. Usery, "A spatial data model design for feature-based geographical information systems", IJGIS 10(5) pp.643-659, 1996.

[8] W. N. John, and V. Smaalen, "A hierarchic rule model for geographic information abstraction", Proc. SDH, pp.215-225 1996.

[9] W. Peng, and K. Tempfli, "An Object-Oriented Design for Automated Database Generalization", Proc. SDH, pp.199-213, 1996.

[10] G. Dettori, and Enrico Puppo, "How Generalization interacts with the topological and metric structure of maps", Proc. SDH 1996.

[11] P. Atzeni, and R. Torlone, "Management of Multiple Models in an Extensible Database Design Tool", RT-DIA-27-1997, Technical reports at Dept. of CS and Automation, Roma TRE, 1997.

[12] V. Delis, and T. Hadzilacos, "On the Assessment of Generalization Consistency", Proc. SSD, pp.321-335, 1997.

[13] M. Sester, K. H. Anders, and V. walter, "Linking Objects of different spatial data sets by Integration and aggregation", Journal of GeoInformatica 2(4) pp.335-358, 1998

[14] J. G. Stell, and M. F. Worboys, "Generalizing Graphs Using Amalgamation and Selection", Proc. SSD, pp.19-32, 1999.

[15] C. A. Davis Jr and A. H. F. Laender, "Multiple Representation in GIS : Materialization Through Map Generalization, Geometric, and Spatial Analysis Operations", Proc. ACM-GIS, pp.60-65, 1999.



이 기 준

84. 2 서울대학교 전자계산학 학사
 86. 2 서울대학교 전자계산학 석사
 92. 7 프랑스로립응용과학원
 전자계산학 박사

부산대학교 정보컴퓨터공학부 부교수

강 혜 경

95. 2 창원대학교 행정학 학사
 98. 2 부산대학교 지리정보협동과정 석사
 98. 2 부산대학교 지리정보협동과정 박사수료
 02. 10- 현재 미국 ESRI 연구원