

다중 축척 수치 지도를 위한 공간 데이터 모델링에 관한 연구

정규상*, 박상미, 이기준, 최병남†
부산대학교 전자계산학과
‡국토개발연구원

Research on Spatial Data Modeling for Multi-Scale Digital Map

Gyu-Sang Jeong*, Sang-Mi Park, Ki-Joune Li and Byong-Nam Choe
Department of Computer Science, Pusan National University

‡Korea Research Institute for Human Settlements

e-mail : {gsjeong, smpark, lik}@chronos.cs.pusan.ac.kr, bnchoe@av400.krihs.re.kr

요약

현재 대부분의 상용 GIS 패키지는 축척을 임의로 조정할 수 있는 기능을 제공하고 있다. 그러나 축척을 임의로 조정한다고 보다는 다른 표현 공간에 같은 내용을 나타내는 것으로 마치 축소 혹은 확대 복사와 같은 것이다. 따라서 축척의 변화에 따른 객체의 표현 형태나 모델이 변하는 것은 아니다. 또한 이는 응용 시스템 수준이지 데이터베이스 수준은 아니다.

이 논문에서는 응용 목적에 따라 축척이 달라질 경우, 응용 프로그래머가 기본 데이터 도면과 모델을 이용해서 그 축척에 맞는 새로운 데이터 도면과 모델을 만들기 위한 방안을 제시하고자 한다. 즉 사용자가 일반화(generalization)를 편집기를 이용해서 작성한 일반화 룰을 기본 데이터 모델에 적용시켜 응용 데이터 모델을 생성한다. 생성된 응용 데이터 모델에 기본 데이터 도면을 적용시켜 응용 도면을 생성한다.

1. 서론

GIS에서는 공간 자료(spatial data)와 속성 자료(attribute data)를 다루어야 한다. 공간 자료는 객체의 공간적인 위치를 표현하는 좌표로 시각적으로는 점, 선, 면으로 표현된다. 이 기호적인 표현에 의해서 객체의 생김새가 추상적으로 표현되기도 한다. 그리고 속성 데이터는 그 객체가 무엇인지를 나타내는 자료로 문자나 숫자를 사용하여 표현한다.

사용자에 따라 데이터를 사용하는 목적이 다르다. 목적이 다르기 때문에 그 목적에 적합한 다른 데이터 추상화 수준의 데이터가 요구된다. 동일한 객체에 대해서 데이터 추상화 수준이 달라지면 객체의 표현 방법과 항목 그리고 데이터의 자세한 정도가 달라진다. 지도를 예로 들면 지도를 사용하는 목적에 따라 대축척(1/500, 1/1000)에서 소축척(1/250,000, 1/500,000)에 이르기까지 다양한 축척을 사용한다. 지도의 축척이 달라지면 공간 자료의 표현 방법과 표현되는 항목이 다르며 또한 데이터의 자세한 정도가 달라진다. 이러한 필요성으로 현실 객체에 대해 다른 수준의 의미론적 그리고 기하학적인 표현을 요구한다.

데이터 추상화 수준이 다른 데이터를 확보하는 방법으로 두 가지가 있다.

하나는 사용자의 목적에 적절한 추상화 수준의 데이터를 입력하는 방법이다. 같은 객체에 대해서 사용 목적에 따라 다른 추상화 수준의 데이터를 입력하는 경우 다음과 같은 문제가 발생한다.

- ① 데이터 불일치가 발생한다. 즉, 동일한 객체에 대해서 공간 자료가 달라진다.
- ② 같은 객체에 대해서 데이터 추상화 수준에 따른 중복 입력으로 과도한 데이터의 입력 비용과 유지 관리 비용이 발생한다.
- ③ 시간이 흐름에 따라 데이터의 변화가 일어나는데 변화에 대한 관리, 갱신, 추가, 삭제가 어렵다. 이는 데이터 불일치를 발생시키는 원인이 될 것이다.
- ④ 사용 목적에 따라 다른 데이터베이스에 접근해야 하기 때문에 사용자가 사용이 불편하다.

다른 하나는 공간 데이터베이스에 저장되어 있는 기본 데

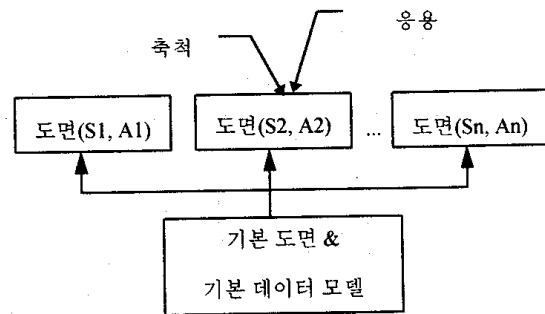
이터 모델을 이용해서 사용자가 요구하는 추상화 수준에 적절한 데이터 모델을 제공하는 방법이다. 이는 추상화 수준에 관계없이 데이터의 일관성을 유지하는 경제적인 방법이라고 생각되나 이러한 공간 데이터베이스를 구축하는 방법이 필요하다.

이 논문에서는 이러한 공간 데이터베이스를 구축하는 방안을 제시하고자 한다. 먼저 다중 축척을 위한 모델 생성기의 전체적인 구조를 살펴본 후, 다중 축척을 위한 모델 생성기에 영향을 미치는 일반화 연산자를 살펴본다. 세 번째로 일반화 연산자를 적용했을 때 미치는 영향 즉, 모델링상의 변화를 살펴본 후 결론을 내리겠다.

2. 다중 축척을 위한 모델 생성기의 설계

2.1 다중 축척을 위한 모델 생성기의 설계

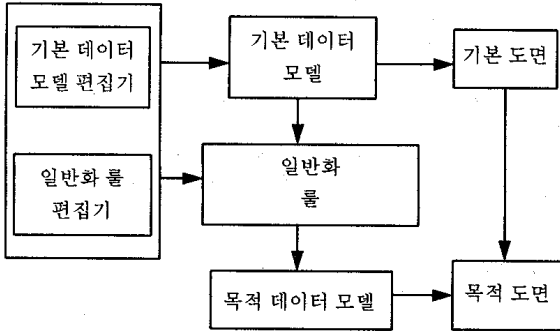
다중 축척 도면 생성의 기본 개념도는 [그림 1]과 같다. 애플리케이션 프로그래머가 목적에 맞는 해당 축척 도면을 생성하기를 원할 경우, [그림 1]과 같이 기본 도면과 기본 데이터 모델을 이용해서 해당하는 각각의 도면과 데이터 모델을 생성한다. 즉, 해당 응용의 축척에 맞는 새로운 모델을 생성한다. 각각의 도면들은 다 같은 지역을 나타내고 있고 단지 축척에 따라서 모델과 도면 표현 방식이 달라지게 된다. 여기서는 이러한 해당 축척에 맞는 모델을 생성하는 방법에 대해서 소개하고자 한다.



[그림 1] 다중 축척 도면 생성기

2.2 다중 축척을 위한 모델 생성기의 설계

다중 축척 도면 생성기의 전체 구성은 [그림 2]와 같다. 먼저 사용자 즉, 응용 프로그래머가 기본 데이터 모델 편집기와 일반화 룰 편집기를 사용해서 각각 기본 데이터 모델과 일반화 룰을 작성한다. 이 일반화 룰은 사용자가 원하는 축척에 따라 달리 적용될 것이다. 이 생성된 기본 데이터 모델에 일반화 룰을 적용시켜 해당 축척에 맞는 목적 데이터 모델을 만든다. 이 목적 데이터 모델이 실제 응용 프로그래머가 구현하는 응용 프로그램의 데이터 모델이 될 것이다. 그리고 기본 데이터 모델로부터 생성된 기본 도면을 이 목적 데이터 모델에 적용시켜 응용 프로그래머가 원하는 최종 목적 도면을 생성한다. 따라서 응용 프로그래머는 기본 데이터 모델 편집기와 일반화 룰 편집기를 이용해서 기본 데이터 모델과 일반화 룰을 작성하는 과정이 필요하다.



[그림 2] 다중 축척 도면 생성기

3. 일반화 연산자

일반화 연산자란 임의의 도면을 일반화시킬 때 적용되는 관련 연산자로서, 크게 두 범주로 나눌 수 있다. [표 1]을 참조하면 먼저 모델 지향적 일반화와 표현 지향적 일반화로 나눌 수 있다[7]. 표현 지향적 일반화는 전통적으로 지도 제작자들에 의해 행해진 그래픽 표현에 대한 연속적인 변형으로 이루어진다. 모델 지향적 일반화는 GIS의 경우에서 볼 수 있듯이, 그래픽 표현에만 제한되지 않고 모델에 영향을 미치는 일반화를 말한다. 이 논문에서는 다중 축척을 위한 모델링에 관한 측면을 다루므로 모델 지향적 일반화 연산자에 대해 언급하고 모델에 미치는 영향에 대해서 기술하고자 한다.

모델 지향적 일반화	표현 지향적 일반화
simplification, selection, collapse, spatial aggregation	exaggeration, displacement

[표 1] 일반화 연산자의 분류

3.1 일반화 연산자

일반화 연산자는 공간 자료를 대축척에서 소축척으로 도면을 일반화 할 경우에 적용되는 연산자로서 위 [표 1]에서 언급한 연산자들이 존재한다. 각 연산자들에 대해서 살펴보면 아래와 같다.

- **simplification** : 가장 많이 행해지는 일반화 연산자이다. 복잡한 형태의 폴리곤을 간단한 형태의 폴리곤으로 만들거나 복잡한 폴리라인을 간단한 폴리라인으로 만든다. 즉 소축척으로의 도면 일반화가 이루어질 경우, 폴리곤이나 폴리라인을 구성하는 포인트를 삭제하여 구성하는 포인트 수를 줄이게 된다[3, 6, 7].
- **selection** : 더 이상 고려될 필요 없는 즉 너무 작거나 의

미적으로 부적절한 객체나 객체의 그룹을 삭제하거나, 어떤 기본 데이터 모델에서 원하는 객체들의 그룹을 선택하는 일반화 연산자이다[7].

- **collapse** : 고차원의 객체들이 저차원의 객체로 일반화가 이루어지는 경우이다. 즉 폴리곤이 폴리라인이나 포인트로 변환이 되거나 폴리라인이 포인트로 변환이 되는 경우이다. 가령 학교를 나타내는 복잡한 폴리곤이 간단한 폴리라인으로 나타내어지거나 또는 단지 포인트만으로 나타내어질 경우이다.
- **spatial aggregation** : 여러 종류의 객체들이 한 종류의 객체로 통합이 될 경우이다. 이때 공간 데이터의 차원은 같다. 가령 아파트라는 폴리곤들이 소축척에서 아파트 단지라는 폴리곤으로 나타내어지거나 동이라는 폴리곤들이 소축척에서 구라는 폴리곤으로 나타내어지는 경우이다[3, 6, 7].
- **exaggeration** : 의미적으로는 중요한 객체가 너무 작아서 잘 표현이 되지 않은 경우 그 객체를 강조하는 일반화 연산자이다. 따라서 일반화 된 후의 객체는 이전의 기하학적인 특성이 달라질 수도 있다. 가령 대축척에서의 고속도로가 소축척으로 일반화되면서 표현이 희미해질 경우, 그 고속도로 객체를 강조하기 위해서 굵은 선으로 뚜렷이 나타내는 경우이다. 그러나 그 고속도로가 대축척에서 접해있던 도시나 그 고속도로가 지나온 지역들을 소축척에서도 그대로 보존시켜주어야 한다. 그렇지 않으면 그 지도의 위상학적인 관계나 의미적인 관계가 이상해질 것이다[7].
- **displacement** : 객체를 강조하기 위한 또 다른 일반화 연산자이다. 이 연산자는 가령 대축척에서 인도, 차도, 중심선이 소축척에서 서로 중첩이 되거나 서로 엇갈려 분별이 힘들게 될 때, 이 인도, 차도, 중심선들을 서로 중첩되지 않게 명확히 구분하기 위해 각 객체들의 위치를 약간 옮기는 연산자이다[7].

여기서는 모델에 영향을 미치는 모델 지향적 일반화 연산자인 simplification, selection, spatial aggregation, collapse를 적용했을 때 모델이 어떻게 변화되는지 알아보겠다.

4. 일반화 연산자의 적용시 모델링상의 변화

공간 데이터의 모델링은 보통 객체 지향 방식을 취하고 있다. 따라서 일반화 연산자를 기본 데이터 모델에 적용했을 때의 모델링상의 변화를 객체 지향 방식과 관련된 다음 세 가지 측면에서 고려하겠다.

- **class hierarchy** : 기본 데이터 모델의 각 클래스들은 부모 클래스와 이를 상속 받아 새로 형성된 자식 클래스들의 관계를 이루고 있다. 이를 클래스 계층이라고 한다. 가령 동물 클래스와 사람 클래스가 있다면 동물 클래스는 부모 클래스가 되고 이 부모 클래스를 상속 받은 사람 클래스는 자식 클래스가 된다[9].
- **aggregation** : 여러 종류의 구성 요소 클래스들이 모여서 한 복합체 클래스를 형성할 때 이 관계를 aggregation이라고 한다. 가령 자동차 클래스가 복합체 클래스라면 이를 형성하는 구성 요소 클래스들은 바퀴 클래스, 핸들 클래스, 엔진 클래스 등이 있을 것이다[9].
- **association** : 클래스와 클래스간의 관계를 association이라고 한다. 가령 교수 클래스와 학생 클래스가 존재한다면 이 두 클래스 간에는 사제지간이라는 association이 존재할 것이다[9].

그리고 도면 일반화룰 시킬 때 각 일반화 연산자가 모델에 영향을 미치는 경우를 Rule1 부터 Rule10 까지 나열하였다.

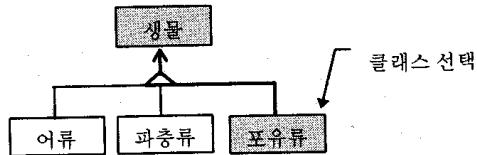
1) Selection

가. Class hierarchy

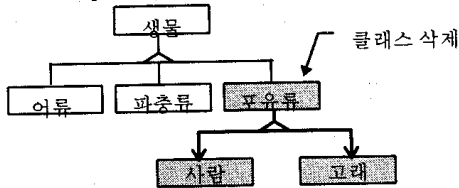
Rule 1: 만약 임의의 클래스가 선택이 되면 그 클래스의 상위 클래스들 또한 반드시 선택이 되어야 한다.

Rule 1': 만약 임의의 클래스가 삭제되면 그 클래스의 하위 클래스들 또한 반드시 삭제가 되어야 한다.

[그림 2]는 포유류 클래스를 선택했을 때, 그 클래스의 상위 클래스인 생물 클래스가 선택이 되는 경우이고, [그림 3]은 포유류 클래스를 삭제했을 때, 그 클래스의 하위 클래스인 사람 클래스와 고래 클래스가 삭제되는 경우이다.



[그림 2] Selection



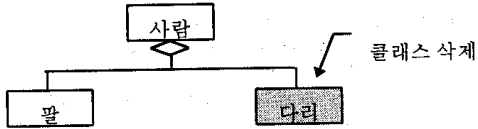
[그림 3] Elimination

나. Aggregation

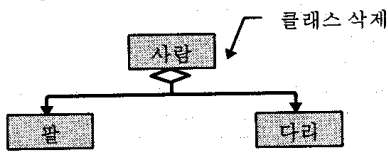
Rule 2: 복합체 클래스 삭제 없이 구성요소 클래스 삭제가 가능하다.

Rule 3: 복합체 클래스가 삭제 된다면 그것의 구성요소 클래스들은 전부 삭제가 되어야 한다.

[그림 4]는 구성 요소 클래스인 다리 클래스를 삭제할 경우, 복합체 클래스인 사람 클래스에 관계없이 단독으로 삭제되는 경우이고, [그림 5]는 복합체 클래스인 사람 클래스를 삭제하는 경우, 그 구성요소 클래스인 팔 클래스와 다리 클래스가 삭제되는 경우이다.



[그림 4] 구성요소 클래스 삭제



[그림 5] 복합체 클래스 삭제

다. Association

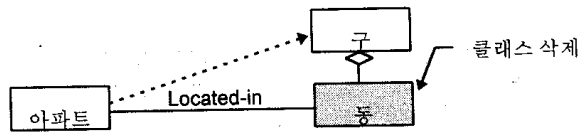
임의의 클래스와 Aggregation 관계를 가지고 있는 다른 클래스와의 Association 과, 임의의 클래스와 Class hierarchy 관계를 가지고 있는 다른 클래스와의 Association 두 측면에서 고려하였다.

① Aggregation 과 Association 의 관계

Rule 4: 만약 association 관계를 가지는 클래스가 삭제가 된다면 association 은 삭제된 클래스의 복합체 클래스와 관계지어져야 한다.

Rule 4': 사용자가 원한다면 association 은 삭제가 될 수 있다.

[그림 6]은 아파트와 동 클래스가 Located-in 이라는 association 관계를 가지고 있을 경우 동이라는 클래스를 삭제했을 때, 아파트 클래스가 동 클래스의 복합체 클래스인 구 클래스와 association 이 설정되는 경우이다.

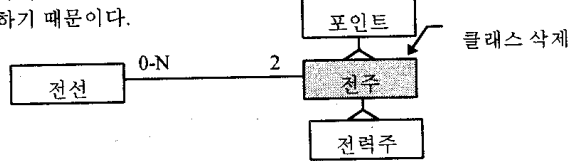


[그림 6] Aggregation 과 Association 의 관계

② Class hierarchy 와 Association 의 관계

Rule 5: 클래스 삭제시, 삭제되는 클래스와 association 관계를 가지고 있는 클래스가 혼자서 의미를 가지지 못할 경우에는 그 클래스를 삭제한다.

[그림 7]은 전선 클래스와 전주 클래스가 association 관계를 가지고 있을 경우 전주 클래스를 삭제했을 때, 전선 클래스도 삭제되는 경우이다. 이는 전선 클래스가 전주 클래스에 의존하기 때문이다.



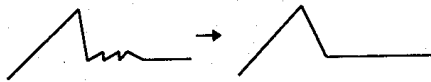
[그림 7] Aggregation 과 Class hierarchy 의 관계

2) Simplification

임의의 클래스를 Simplification 하게 되면, 그 클래스의 구조가 바뀌게 되고 따라서 모델에 영향을 미치게 된다.

① 폴리라인 형태를 가지는 클래스

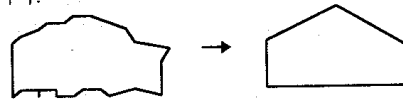
[그림 8]은 복잡한 폴리라인 형태를 가지고 있는 강 클래스를 간단한 폴리라인의 형태를 가지고 있는 강 클래스로 도면 일반화 시킨 경우이다. 따라서 강 클래스를 구성하는 포인트 개수의 변화가 생긴다.



[그림 8] 강 클래스에 대한 Simplification

② 폴리곤 형태를 가지는 클래스

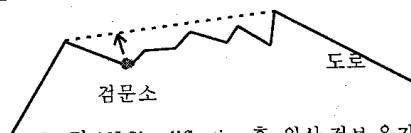
[그림 9]는 복잡한 폴리곤 형태를 가지고 있는 건물 클래스를 간단한 폴리곤 형태를 가지고 있는 건물 클래스로 도면 일반화 시킨 경우이다.



[그림 9] 건물 클래스에 대한 Simplification

Rule 6: simplification 이후, 어떤 위상적인 정보가 달라져서는 안된다.

[그림 10]은 도로 클래스와 검문소 클래스가 인접해 있을 때 복잡한 폴리라인 형태를 가지고 있는 도로 클래스를 간단한 폴리라인 형태를 가지고 있는 도로 클래스로 일반화 시켰을 경우, 검문소 클래스와의 인접성이라는 위상 정보를 유지시키기 위해 검문소 클래스의 위치를 이동하는 모습을 나타내었다.



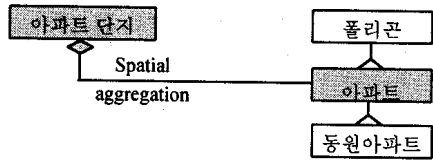
[그림 10] Simplification 후, 위상 정보 유지

3) Spatial Aggregation

가. Class hierarchy

Rule 7: 클래스 계층상에 있는 임의의 클래스가 Spatial aggregation 되면 그것에 해당하는 새로운 클래스를 만든다.

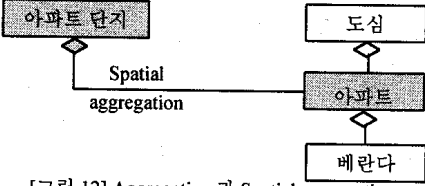
[그림 11]은 클래스 계층상에 있는 아파트 클래스를 Spatial aggregation 했을 경우, 아파트 단지라는 새로운 클래스가 생성되는 경우를 보여주고 있다.



[그림 11] Class hierarchy 와 Spatial aggregation 나. Aggregation

Rule 8 : Aggregation 상에 있는 임의의 클래스가 spatial aggregation 되면 그것에 해당하는 새로운 클래스를 만든다.

[그림 12]는 Aggregation 상에 있는 아파트라는 클래스를 Spatial aggregation 했을 경우, 아파트 단지라는 새로운 클래스가 생성되는 경우를 보여주고 있다.



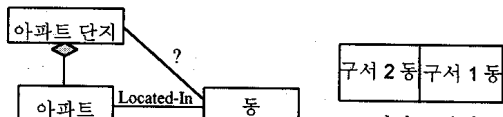
[그림 12] Aggregation 과 Spatial aggregation

다. Association

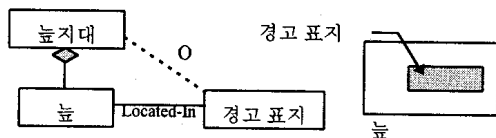
Rule 9 : Spatial aggregation 되는 클래스가 Association 관계를 맺는 클래스에 기하학적으로 포함되면 Spatial aggregation 된 클래스는 Association 관계를 알 수가 없다. 사용자가 결정해야 한다.

Rule 9' : Spatial Aggregation 되는 클래스가 Association 관계를 맺는 클래스를 기하학적으로 포함하면 Spatial aggregation 된 클래스는 그 Association 관계를 그대로 유지한다.

[그림 13]은 아파트 클래스와 동 클래스가 Association 관계를 가지고 있을 경우 아파트 클래스를 Spatial aggregation 하여 아파트 단지라는 새로운 클래스를 생성했을 때, 아파트 클래스가 기하학적으로 동 클래스 안에 포함되므로 아파트 단지와 동 클래스간의 Association 은 모호하게 된다. 따라서 이런 경우에는 사용자가 결정을 해야 한다. 그리고 [그림 14]는 높이라는 클래스가 경고 표지라는 클래스를 포함하기 때문에 생성된 높지대 클래스와 경고 표지 클래스 사이에는 Located-In 이라는 Association 관계가 그대로 유지된다.



[그림 13] Association 과 Spatial aggregation 1.



[그림 14] Association 과 Spatial aggregation 2.

4) Collapse

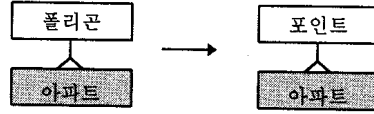
Rule 10 : Collapse 후, 어떤 위상적인 변화가 생겨서는 안된다.

[그림 15]는 폴리곤 형태를 가지는 아파트 클래스가 Collapse 된 후 포인트 형태를 가지게 된 경우에도 도로 클래스와 인접이라는 위상 정보를 유지하는 있는 모습을 나타내었다.



[그림 15] Collapse 후, 위상 정보 유지

[그림 16]은 아파트 클래스가 폴리곤 클래스로부터 상속 받았으나, Collapse 이후 포인트 클래스로 상속을 받게 되는 모델링상의 변화를 나타내었다.



[그림 16] Collapse 후, 모델링상의 변화

5. 결론

사용자에 따라 데이터를 사용하는 목적이 다르기 때문에 그 목적에 적합한 다른 데이터 추상화 수준의 데이터가 요구된다. 이는 데이터 추상화 수준이 달라지면 객체의 표현 방법과 항목 그리고 데이터의 자세한 정도가 달라지기 때문이다.

이 논문에서는 응용 목적에 따라 축척이 달라질 경우, 응용 프로그래머가 기본 도면과 데이터 모델을 이용해서 그 축척에 맞는 새로운 도면과 데이터 모델을 만들기 위한 방안을 제시했다.

먼저 다중 축척을 위한 모델 생성기의 전체적인 구조와 다중 축척을 위한 모델 생성기에 영향을 미치는 일반화 연산자를 살펴보았다. 그리고 일반화 연산자가 모델에 미치는 영향을 살펴보았다.

앞으로 살펴 볼 향후 연구 계획은 이 공간 데이터 모델링을 토대로 실제 다중 축척을 위한 모델 생성기를 구현하는 일이다. 즉 객체 지향 방식을 사용한 모델링 생성기를 구현하여 실제 객체 지향 데이터베이스를 위한 스키마 코드를 생성하는 일이다. 객체 지향형 스키마 코드로 선택한 언어는 C++이다. 그리고 도면 일반화에 미치는 연산자로서 selection, spatial aggregation, simplification, collapse 를 살펴보았는데 이외에도 영향을 미치는 다른 일반화 연산자를 고려할 예정이다.

참고문헌

[1] Nectaria Tryfona and Thanasis Hadzilacos, "Geographic Applications Development: Models and Tools for the Conceptual Level," ACM-GIS'95 Proceedings of the 3rd ACM International Workshop on Advances in GIS, pp 19-28
 [2] Michael F. Worboys, Hilary M. Hearshaw and David J. Maguire, "Object-oriented data modelling for spatial databases," INT. J. Geographical Information Systems, Vol. 4, No. 4, 1990, pp 369-383.
 [3] Anne Ruas and Corinne Plazanet, "Strategies for Automated Generalization," 7th International Symposium on Spatial Data Handling, Vol. 1, 1996.
 [4] Thomas Devogele, Jenny Trevisan and Laurent Raynal, "Building a Multi-scale Database with Scale-transition Relationships," 7th International Symposium on Spatial Data Handling, Vol. 1, 1996.
 [5] Serge Abiteboul and Richard Hull, "IFO: A Formal Semantic Database Model," ACM Transactions on Database Systems, Vol. 12, No. 4, December 1987, pp 525-565.
 [6] Vasilis Delis and Thanasis Hadzilacos, "On the Assessment of Generalisation Consistency," 5th International Symposium, SSD '97 Berlin, Germany, July 1997 Proceedings, pp 321-338.
 [7] Giuliana Dettori and Enrico Puppo, "How Generalization Interacts with the Topological and Metric Structure of Maps," 7th International Symposium on Spatial Data Handling, Vol. 1, 1996.
 [8] Michael F. Worboys, "Object-oriented approaches to georeferenced information," INT. J. Geographical Information System, Vol. 8, No. 4, 1996, pp 385-399.
 [9] James Rumbaugh, Michael Blaha, William Premerlani, Frederick Eddy and William Lorensen, Object-Oriented Modeling and Design, Prentice-Hall International.