

한국 GIS 학회 1996 년 춘계 학술 발표회

발표 요약문

지하시설물의 3 차원 표현을 위한 공간 데이터

변환 시스템의 객체 지향적 설계 및 구현

오 승 * · 강병익** · 정정화***

The Object-Oriented Design & Implementation of Spatial Data Transformation System for the 3-D Representation of Underground Utilities

Seung Oh · Byung-Ik Kang · Jong-Wha Chong

* 한양대학교 환경과학대학원 환경계획학과 석사과정 (Department of Environmental Planning, Hanyang University, 17 Hangdang-Dong, Sungdong-Gu, Seoul, Korea, 133-791, Tel. (02) 290-1519)

** 건양대학교 컴퓨터공학과 교수(Department of Computer Engineering, Keonyang University, San 30 Nae-Dong, Nonsan, ChungNam, Korea, 320-800, Tel. (0461)735-1702)

*** 한양대학교 전자공학과 교수(Department of Electronics Engineering, Hanyang University, 17 Hangdang-Dong, Sungdong-Gu, Seoul, Korea, 133-791, Tel. (02) 290-0356)

**** 본 연구는 건설 교통부 및 한국건설기술연구원의 건설교통기술연구 개발사업(1995)의 지원으로 이루어진 것임.

지하시설물의 3차원 표현을 위한 공간 데이터

변환 시스템의 객체 지향적 설계 및 구현

오 승* · 강병익** · 정정화***

**The Object-Oriented Design & Implementation of Spatial
Data Transformation System for the 3-D Representation of
Underground Utilities**

Seung Oh · Byung-Ik Kang · Jong-Wha Chong

요 약

본 연구에서는 상용 GIS 소프트웨어인 ARC/INFO 에 구축되어 있는 지하 시설물 데이터를 3차원 공간 객체로 변환하기 위하여 3차원 공간 객체의 형식과 객체 저장 구조를 정의하고, 2차원의 지하 시설물 데이터를 정의된 3차원 공간 객체로 변환하는 공간 데이터 변환 시스템을 객체 지향 개발 기법인 Booch 의 방법을 사용하여 개발하였다. 이를 위하여 ARC/INFO 의 관계형 데이터 모델과 저장 구조를 분석하고, 지하 시설물의 3차원 시각화를 위한 요구 조건들을 분석하였다. 그리고 ARC/INFO 에 저장된 2차원 지하 시설물 데이터는 내부적인 전처리 과정을 통하여 ASCII 파일로 변환 후, 변환 시스템의 입력 자료로 사용하였고, 변환 시스템의 개발을 위한 개발 방법으로 Booch 의 방법을 선정하여 시스템의 분석과 설계 과정에 적용하였다. 마지막으로 분석과 설계의 결과를 ARC/INFO 의 매크로 언어인 AML 과 객체 지향 프로그래밍 언어인 C++를 사용하여 구현하였다. 본 연구의 결과, 지하 공간에 존재하는 2차원 시설물들을 3차원으로 시각화 할 수 있는 3차원 공간 객체를 얻을 수 있었으며, 시스템의 개발 과정에 객체 지향 개발 기법을 적용함으로써 효율적인 개발을 수행할 수 있었다.

ABSTRACT : In order to transform the underground utility data into the 3-D spatial objects, this thesis defined the type of the 3-D spatial objects and the storage structure of objects, and developed the spatial data transformation system that transforms the 2-D underground utility data into the pre-defined 3-D spatial objects through the Booch Method, an object-oriented development method. For this sake, the relational data model of ARC/INFO and the storage structure are analyzed, as well as the pre-requisites, required for the 3-D visualization of the underground utilities. Also, the 2-D underground utility data, saved in ARC/INFO, were transformed into the ASCII files through the internal pre-processing procedure, then used as the input data of the transformation system. In addition, to develop the transformation system, the object-oriented development methods are studied first and, then, applied to the system analysis and the design procedure with the Booch Method as a development method. Finally, the results of analysis and design procedure are implemented through AML, a ARC/INFO macro language, and C++, an object-oriented programming language. As a result of this study, the 3-D spatial objects that can visualize the 2-D utilities in 3-D, are acquired, and the adaptation of object-oriented development method to the system development procedure enabled an effective development procedure.

서론

도시 지역 지하 공간을 효율적으로 개발하고 이용하기 위한 계획 작성에 있어서는, 대상으로 하는 지하공간이 지표공간과는 달리, 본질적으로 불균질이며 또한 불가시 공간인 것에 유의할 필요가 있다. 이 불균질성을 3 차원 공간상에서 바르게 평가함과 동시에, 그 결과를 가시화하여 객관적으로 표시·도화하는 새로운 기술을 개발하는 것이 요구되고 있다. 그러나 현업에서는 지하의 상황을 문자로 대장에 기록하고, 그 위치를 지도상에 표현하여 관리하고 있다. 여기서 지도상에 선으로 나타내는 방법이 바로 2

차원 시각화(visualization) 방법인데, 2 차원적인 시각화는 지하의 복잡한 현황을 표현하기에는 한계가 있어 수집된 정보가 사용자에게 정확히 전달되지 못하고 있는 실정이다. 따라서 지하시설물의 설계 내용의 적합성 및 안전성에 대한 평가, 공사에 대한 감리·감독 업무 등이 효율적으로 이루어지지 못하고 있어 지하의 상황을 보다 정확히 파악할 수 있는 지하 시설물의 3 차원 표현 도구가 필요하다.

본 연구에서는 기존의 상용 GIS 소프트웨어인 ARC/INFO 에 구축되어 활용되고 있는 지하 시설물 데이터를 3 차원 객체로 변환하기 위한 공간 데이터 변환 시스템을 객체 지향 분석 및 설계 기법인 Booch method 를 사용하여 개발하고, 변환된 3 차원 객체를 효율적으로 저장 및 관리할 수 있는 객체 저장 구조를 설계하는데 그 목적이 있다.

따라서 본 연구에서는 첫째, 객체 지향 개념의 핵심 이론과 데이터 모델을 이론적으로 고찰하고, 상용 GIS 소프트웨어인 ARC/INFO 의 운용을 통하여 기존의 관계형 데이터 모델을 분석하고 이들의 차이점을 정리하였다. 둘째, 지하 시설물의 3 차원 시각화를 위한 요구 조건들을 분석하여, 공간 데이터 변환 시스템을 통해 변환될 3 차원 공간 객체의 형식 및 이들 객체들의 저장 구조를 정의하였다. 셋째, 객체 지향 개발 방법인 Booch method 를 사용하여 공간 데이터 변환 시스템을 설계하고, 이를 ARC/INFO 의 매크로 언어인 AML 과 객체 지향 언어인 C++를 사용하여 구현하였다. 이를 위해 서울특별시 중구의 주거·도심 지역을 대상지로 선정하였으며, 대상지 내의 지하 시설물 데이터로는 (주)한진지리정보에서 ARC/INFO 를 사용하여 구축한 상수도관, 하수도관, 가스관, 전기관, 전화관 데이터를 사용하여 연구를 수행하였다.

객체 지향 이론의 핵심 개념

객체와 객체 식별자 : 실세계에서 임의의 한 개체(entity)는 상태(state)

와 행위(behavior)를 갖는 하나의 객체로 모델화 될 수 있다. 상태는 객체의 애트리뷰트들이 갖는 값으로 표현되며, 행위는 메소드(method)에 의해 정의되어 해당 연산이 호출될 때 객체의 상태에 대한 동작을 수행하게 된다. 또한 각각의 객체는 하나의 객체 분류 시스템에서 유일한 식별자(identifier)를 갖는데, 이 객체 식별자는 객체가 갖는 애트리뷰트(attribute)의 값과는 독립적이다.

애트리뷰트와 메소드 : 한 객체는 하나 이상의 애트리뷰트를 가지며, 애트리뷰트값 상에서 작동할 수 있는 하나 이상의 메소드가 존재한다. 애트리뷰트란 객체와 같이 자기 스스로 추상화 되어진 모든 실체에 의해서 소유된 단 하나의 특성에 대한 추상물이라 정의할 수 있다. 한 객체의 애트리뷰트에 대한 값 또한 객체이다.

캡슐화(encapsulation)와 메시지 전달 : 어떤 객체의 애트리뷰트값과 그 애트리뷰트에 속한 메소드들에 접근하기 위해서는 그 객체로 메시지를 보내야 한다. 각 객체에 명시된 공용 인터페이스를 통하지 않고는 그 객체에 접근할 수 없다. 객체의 상태는 해당 연산이 메소드들을 호출함으로써 조작된다.

클래스(class)와 실체(instance) : 애트리뷰트와 메소드를 공유하는 모든 유사한 객체들은 클래스로 그룹화 된다. 이때 객체는 그 객체의 애트리뷰트와 메소드를 정의한 클래스에 속하는 하나의 실체이다. 따라서 한 클래스로부터 만들어진 모든 객체들은 애트리뷰트와 메소드에 대한 정의를 클래스를 통해 공유하고 있다. 클래스도 역시 객체가 될 수 있다.

계승(inheritance) : 계승은 한 클래스를 하나 또는 그 이상의 다른 일반적인 클래스들로 정의하기 위한 방법을 제공한다. 새로이 정의된 객체는 원 객체(original object)의 성질을 계승 받을 수 있고, 계승된 메소드를 재정의할 수 있으며, 새로운 메소드를 추가하고, 계승된 객체에 새로운 성질과 메소드를 추가할 수 있다. 계승은 정보의 중복성을 제거하고 통합성을 유지하도록 한다.

데이터 모델의 비교

데이터 모델은 데이터, 데이터의 관계, 데이터의 의미 및 일관성 제약 조건 등을 기술하기 위한 개념적 도구들의 모임이며, 실세계에서의 현상이나 객체가 어떻게 표현되는가를 결정하고 이들 객체들에 대한 속성들이 데이터베이스에 어떻게 구축되는가를 정의하는 것이다.

관계형 데이터 모델 : 관계형 데이터 모델은 데이터와 데이터간의 관계를 테이블의 모임으로 나타내며, 각 테이블은 유일한 이름을 가진 여러 개의 열로 구성된다. 본 연구에서 사용하는 지하 시설물의 2차원 공간 데이터는 ARC/INFO의 INFO 데이터베이스에 저장되어 있는데, ARC/INFO의 데이터 모델은 지도 데이터베이스에 기반을 둔 혼합 지리 관계형(hybrid georelational) 데이터 모델이다. 데이터는 한 집단의 커버리지(coverage)들에 위상적 데이터 구조를 사용하여 저장된다. 커버리지들은 벡터 데이터를 저장하기 위한 기본 단위이고, 몇 가지 종류의 지형 요소(geographic features)들을 표현할 수 있다.

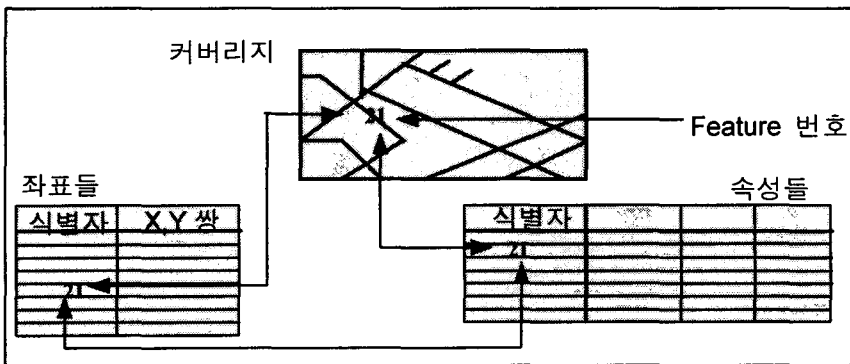


Fig. 1 The Structure of ARC/INFO georelational data model

커버리지는 feature 들의 위치(location)와 속성(attribute)을 기술하는 관련된 파일들을 담고 있는 하나의 디렉토리에 저장된다. 한 커버리지 내에 포

함된 각각의 feature 에는 유일한 수치 식별자가 할당되고 유일한 위치(공간 데이터)와 속성들의 집합(기술 데이터)에 의해 특징 지워진다. 커버리지 내의 각 feature 에 대한 유일한 번호인 식별자는 좌표 파일과 속성 파일 모두에 저장된다. 이를 그림으로 나타내면 Fig. 1 과 같다.

객체 지향 데이터 모델 : 객체 지향 데이터 모델이란 간단히 말하면 데이터를 모델링하기 위해 유용한 객체 지향 개념을 잘 모아서 정의해 놓은 집합이다. 객체 지향 데이터 모델은 relation-entity 관계와 같은 초기의 방법론보다 더 의미(구문)적인 내용을 갖는 모델을 구축하기 위하여 클래스화(classification), 일반화(generalization), 연관성(association), 그리고 집단화(aggregation) 같은 추상화 개념을 활용한다. 비록 객체 지향 데이터 모델에 대한 명확한 정의가 없지만, 이 모델은 관계형 모델로부터 초기의 의미적 모델을 거쳐 객체 지향 모델까지 이어지는 연결고리가 있음을 일반적으로 인식하고 있다.

동일한 타입의 값과 메소드를 포함하는 객체들은 클래스라는 곳으로 함께 넣어진다. 클래스는 객체에 대한 타입 정의로서 나타낼 수 있다. 타입 정의 내부의 코드와 데이터의 조합은 프로그래밍 언어의 추상 데이터 타입(abstract data type) 개념과 비슷하다. 한 객체에서 다른 객체에 있는 데이터를 액세스할 수 있는 유일한 방법은 다른 객체의 메소드를 호출하는 방법으로 이루어진다. 이것을 객체에 대한 메시지 전달이라 한다. 때문에 어떤 객체의 메소드의 호출 인터페이스는 완전히 가시적인 부분으로 정의한다. 그러나 객체의 내부는 외부에 노출되지 않는다. 결과적으로 데이터의 추상화가 2 단계로 이루어지는 것이다.

두 모델의 비교 : 객체 모델은 데이터, 데이터 타입, 절차(procedure), 처리 방법들을 통합적으로 관리할 수 있도록 하는 데이터 타입과 관련된 처리 모델이다. 반면에 관계형 모델은 데이터가 어떻게 구축되는지를 기술하고, 그 모델의 한계 내에서 잘 정의된 연산자들의 집합에 접근할 수 있

도록 해주는 하나의 데이터 구조 모델이다.

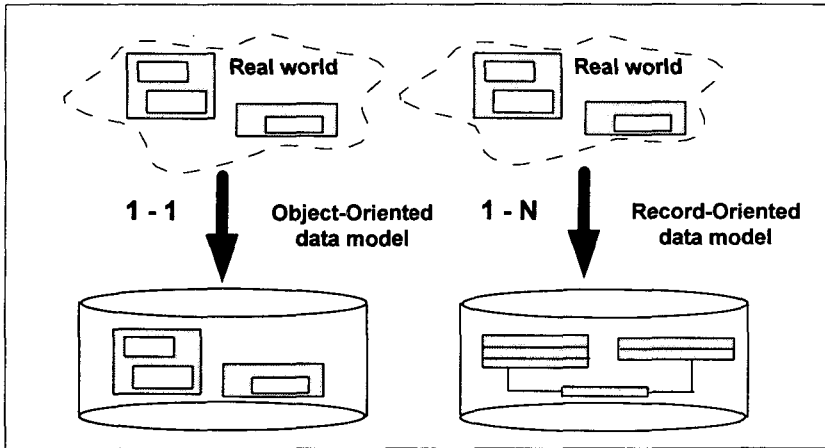


Fig. 2 The difference between the relational and object-oriented model

관계형 데이터 모델의 한가지 주요한 결점은 그 정보가 분산된다는 것이다. 사실상, 정규화 조건의 구현은 데이터를 작은 관계들에 할당하기 때문에 특정 개체에 관한 모든 정보를 분할하게 만든다. 예를 들어, 사람이나 건물에 포함된 정보는 분리된 몇 개의 테이블로 분산된다. 결과적으로 질의가 복잡해지게 되는 것이다. 이러한 문제점을 해결하기 위하여, 객체 지향 데이터 모델은 객체에 관련된 모든 데이터를 그룹으로 묶게 된다. 이러한 특징은 하나의 단일 객체에 관한 검색을 위한 간단한 기준을 제공한다. Fig. 2는 관계형과 객체 지향 데이터 모델간의 차이점을 보여 주고 있다.

공간 데이터 변환 시스템의 개요와 구조

시스템 개요: 본 연구에서 설계하고자 하는 공간 데이터 변환 시스템의 개요는 Fig. 4와 같다. 이를 간략히 설명하면, 변환하고자 하는 지하 시설물 데이터는 ARC/INFO의 INFO 데이터베이스에 저장되어 있는데 이를

전처리 과정을 통하여 ASCII 파일로 변환하게 된다. 변환된 ASCII 파일은 본 연구에서 설계하여 구현된 공간 데이터 변환기를 거쳐 3차원 시각화에 사용될 3차원 공간 객체들로 변환된다. 마지막으로, 이들 3차원 공간 객체들은 객체 형식의 파일 구조로 보조 기억 장치에 저장된다.

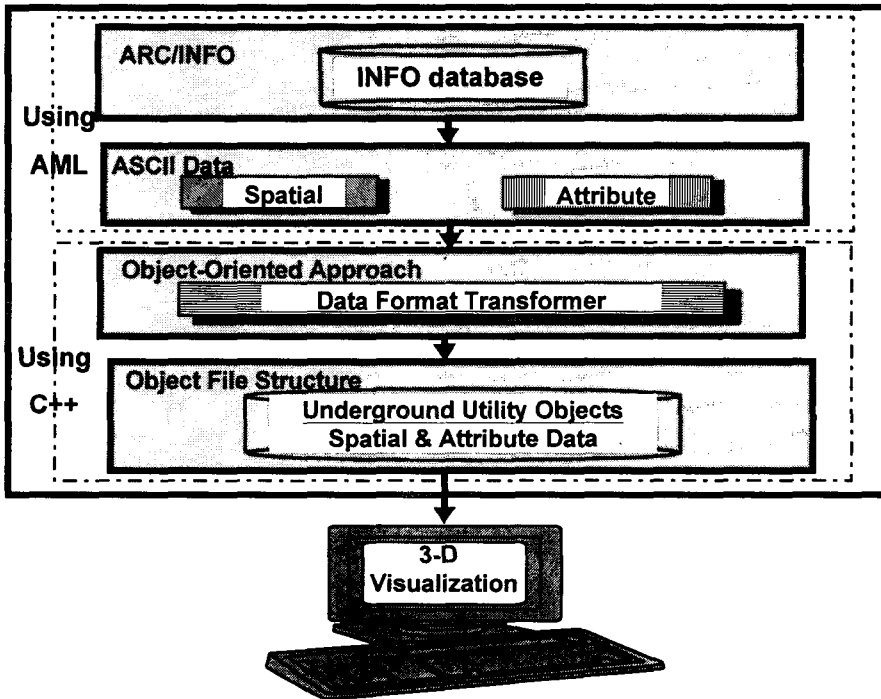


Fig. 4 The outline of spatial data transformation system

전처리 과정: 공간 데이터 변환 시스템에서 전처리 과정은 ARC/INFO 소프트웨어에 저장되어 있는 지하 시설물들의 공간 데이터 및 속성 데이터를 내부적으로 처리하여 ASCII 데이터 형식의 파일로 export하기까지의 과정을 의미한다. 전처리 과정의 작업 흐름도는 Fig. 5와 같다.

Table. 1은 상수도관 커버리지의 속성 테이블(waterline.aat)이다. Table. 1의 속성 테이블에는 상수도관의 아크(즉, 라인)에 대한 정보들을 시작 노드와

끝 노드 아이템에 담고 있다. 여기에서 각각의 아크들에 붙여진 시작 노드와 끝 노드의 번호를 1부터 시작하여 순차적으로 다시 번호를 붙여주어야 한다. 그 이유는 모든 노드들의 내부 번호가 순차적으로 되어 있게 하고 영(zero)이 아닌 값을 갖도록 하는 것이 바람직하기 때문이다. 또한 만일 두 개 이상의 노드들이 동일한 x, y 좌표를 공유한다면, 그것들은 동일한 내부 노드 번호들에 의해 식별되어야만 하기 때문이다. 이것은 ARC/INFO의 RENODE 명령어에 의해 이루어진다(Arc: RENODE waterline.aat).

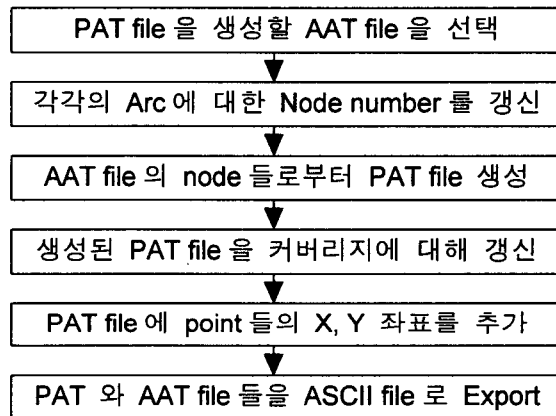


Fig. 5 The flow chart of preprocessing stage

다음은 갱신된 AAT 파일의 노드들로부터 새로운 속성 테이블을 생성하게 된다. 즉, 각각의 노드에 대하여 point attribute table(PAT)을 생성하는 것이다. 이것은 ARC/INFO의 NODEPOINT 명령어를 이용한다(Arc : NODEPOINT waterline.aat waterpoint.pat).

NODEPOINT 명령을 실행하게 되면, waterpoint.pat 내 각 노드의 사용자-ID는 대응하는 waterline.aat내 노드의 내부 번호와 일치하게 된다. 이 때 생성된 PAT 파일은 Table. 2와 같은 정보만을 담고 있다. 이렇게 생성된 PAT 파일은 ARC/INFO의 BUILD 명령어를 사용하여 저장된다(Arc : BUILD

waterpoint.pat {POINT}).

Table. 1 The attribute table of the water pipe coverage(waterline.aat)

| COL | ITEM NAME | WID | OUT | TYPE | N.DEC | ALT. NAME | Remark |
|-----|---------------|-----|-----|------|-------|-----------|-----------|
| 1 | FNODE# | 4 | 5 | B | - | - | 시작 노드 번호 |
| 5 | TNODE# | 4 | 5 | B | - | - | 끝 노드 번호 |
| 9 | LPOLY# | 4 | 5 | B | - | - | 좌측 폴리곤 번호 |
| 13 | RPOLY# | 4 | 5 | B | - | - | 우측 폴리곤 번호 |
| 17 | LENGTH | 8 | 18 | F | 5 | - | 길이 |
| 25 | WATER-LINE# | 4 | 5 | B | - | - | 시스템 ID |
| 29 | WATER-LINE-ID | 4 | 5 | B | - | - | 사용자 ID |
| 33 | DXF-LAYER | 16 | 16 | C | - | - | DXF 레이어 |
| 49 | SURYNG | 2 | 2 | I | - | - | 수량 |
| 51 | KNKYNG | 4 | 4 | I | - | - | 관경 |
| 55 | KNGILI | 6 | 6 | N | 2 | - | 관 길이 |
| 61 | SSIMDO | 5 | 5 | N | 2 | - | 시작 깊이 |
| 66 | ESIMDO | 5 | 5 | N | 2 | - | 끝 깊이 |
| 71 | KNJONG | 6 | 6 | C | - | - | 관 재질 |
| 77 | SLILJA | 6 | 6 | C | - | - | 설치 일자 |
| 83 | SYMBOL | 3 | 3 | I | - | - | 심볼 |
| 86 | DORONO | 9 | 9 | C | - | - | 도로 번호 |

Table. 2 The attribute table produced by NODEPOINT command(waterpoint.pat)

| COL | ITEM NAME | WID | OUT | TYPE | N.DEC | ALT. NAME | Remark |
|-----|---------------|-----|-----|------|-------|-----------|--------|
| 1 | AREA | 8 | 18 | F | 5 | - | 면적 |
| 9 | PERIMETER | 8 | 18 | F | 5 | - | 둘레 |
| 17 | WATERPOINT# | 4 | 5 | B | - | - | 시스템-ID |
| 21 | WATERPOINT-ID | 4 | 5 | B | - | - | 사용자-ID |

다음은 생성된 PAT파일에 각 노드들의 X, Y 좌표를 추가시켜 주어야 한다. 즉 각 노드의 공간 좌표값을 PAT 파일에 속성으로서 추가한다는 의미이다. 이것은 ARC/INFO의 ADDXY 명령어를 통하여 이루어진다. 그 결과는 Table. 3과 같다(Arc : ADDXY waterpoint.pat {POINT}).

이러한 과정들을 통하여 생성·갱신된 AAT와 PAT 파일은 ARC/INFO의 TABLES 세션에서 UNLOAD 명령어를 사용하여 ASCII 파일로 export한다.

ARC/INFO의 외부로 export된 두개의 파일은 waterline.asc와 waterpoint.asc이다. 이때, waterline.asc 파일은 상수도관의 관 번호, 그 관의 시작

과 끝 노드 번호, 환경, 그리고 재질 등과 같은 공간 정보와 속성 정보를 포함하고 있고, waterpoint.asc 파일은 각 노드들의 공간 좌표 정보를 담고 있다. 지금까지 설명한 모든 과정들은 ARC/INFO의 매크로 언어인 AML을 이용하여 구현하였다.

Table. 3 The attribute table updated by ADDXY command (waterpoint.pat)

| COL | ITEM NAME | WID | OUT | TYPE | N.DEC | ALT. NAME | Remark |
|-----|---------------|-----|-----|------|-------|-----------|--------|
| 1 | AREA | 8 | 18 | F | 5 | - | 면적 |
| 9 | PERIMETER | 8 | 18 | F | 5 | | 둘레 |
| 17 | WATERPOINT# | 4 | 5 | B | - | - | 시스템-ID |
| 21 | WATERPOINT-ID | 4 | 5 | B | - | - | 사용자-ID |
| 25 | X-COORD | 8 | 18 | F | 5 | | X-좌표값 |
| 33 | Y-COORD | 8 | 18 | F | 5 | | Y-좌표값 |

객체 식별자 및 공간 객체 저장 구조의 정의

객체 식별자의 정의 : 본 연구에서는 타입 식별자(타입 ID)와 객체 식별자의 일부를 가지는 방식으로 객체 식별자를 정의하였다. 즉 변환하고자 하는 지하 시설물 데이터들을 각각 상수도관 클래스, 하수도관 클래스, 가스관 클래스, 전력관 클래스, 전기관 클래스로 정의하고 이들을 별개의 파일에 저장하였다. 그래서 각 클래스의 객체들에는 객체 식별자의 앞부분에 그들의 클래스 이름의 초기 문자를 사용하였다. 그리고 그 뒤에는 입력 데이터의 관 번호를 사용하여 입력되는 2차원 지하 시설물 데이터와 본 연구의 최종 결과인 3차원 공간 데이터 사이의 상호 참조가 가능하도록 정의하였다. 따라서 본 연구에서 정의된 각 클래스의 객체 식별자는 Fig. 6과 같이 정의된다.

공간 객체 저장 구조의 정의 : 현재까지 제시된 OODBMS에서의 저장 기법은 크게 다음의 두가지 방식으로 구분된다.

| | | |
|---|----------|------------|
| 상수도관 클래스 (Waterline class)의 253 번 객체 : | W | 253 |
| 하수도관 클래스 (Sewerline class) 의 253 번 객체 : | S | 253 |
| 가스관 클래스 (Gasline class) 의 253 번 객체 : | G | 253 |
| 전력관 클래스 (Powerline class) 의 253 번 객체 : | P | 253 |
| 전화관 클래스 (Teleline class) 의 253 번 객체 : | T | 253 |

Fig. 6 The definition of the identifier regarding the 3-D spatial objects

1) 직접 모델 : 직접 모델에서는 객체들은 개념 스키마에 정의된 것과 같은 방식으로 저장되는 데 즉, 저장 단위가 의미상의 단위와 동일하다. 자세히 말하면 동일한 클래스에 속하는 객체들은 동일한 파일에 저장되며 파일의 각 레코드는 그 클래스에서의 각 인스턴스이다. 직접 모델은 객체들의 애트리뷰트들을 저장하는 가장 간단한 방법이며 관계 DBMS에서 사용하는 것과 동일하다.

2) 정규화 모델 : 이 모델에서 객체들은 여러 개의 부분으로 분해되어 다른 파일에 저장된다. 각 부분들의 관계는 OID(객체 식별자)들을 사용하여 유지된다.

본 연구에서는 이러한 객체 지향 데이터베이스의 저장 기법 중 직접 모델을 적용하였다. 따라서 Fig. 7과 같은 3차원 공간 객체의 저장 구조를 정의할 수 있다. 또한 이러한 공간 객체들의 집합인 각 지하 시설물 클래스들을 각각의 파일에 객체 타입으로 정의하였다. Fig. 8은 상수도관 클래스를 저장하는 파일 구조를 예를 들어 나타낸 것이다.

| | | | | | | | | | |
|-----------|---------------|---------------|---------------|--------------|--------------|--------------|---------|---------|-----|
| 객체 식별자 | 시작 X 좌표 | 시작 Y 좌표 | 시작 Z 깊이 | 끝 X 좌표 | 끝 Y 좌표 | 끝 Z 깊이 | 속성 1 | 속성 2 | ... |
| | | | | | | | | | ... |

Fig. 7 The definition of the 3-D spatial object

| 객체 식별자 | 시작 X 좌표 | 시작 Y 좌표 | 시작 Z 깊이 | 끝 X 좌표 | 끝 Y 좌표 | 끝 Z 깊이 | 속성 1 | 속성 2 | 속성 ... |
|-----------|---------------|---------------|---------------|--------------|--------------|--------------|---------|---------|-----------|
| | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| W1 | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| W2 | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| W3 | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |

Fig. 8 An example of the file storage structure of water pipe class

공간 데이터 변환 시스템의 설계 및 구현

개념적 단계 : Booch의 방법을 이용한 객체 지향 개발의 첫번째 단계는 사용자가 시스템이 무슨 일을 하기를 원하는가를 결정하는 과정이다. 이 단계에서는 시스템이 수행하는 핵심 기능을 식별하고 그 시스템이 지원할 영역(domain)의 범위를 정의한다.

공간 데이터 변환 시스템의 핵심 기능에 대한 문제 서술서를 변환기의 기능들에 대해 요약하면 다음과 같다.

첫째, 전처리 과정의 결과인 2차원 지하 시설물의 공간 데이터(waterpoint.asc)와 속성 데이터(waterline.asc)를 변환기로 읽어 들인다.

둘째, 변환기에 읽혀진 속성 데이터의 각 아이템들을 화면에 출력하여 사용자가 속성 정보를 파악하게 하고, 3차원 공간 객체로 변환할 때 포함하고자 하는 속성 아이템들을 선택할 수 있도록 한다.

셋째, 3차원 공간 객체에 포함될 아이템의 선택이 끝나면, 변환 후 생성될 3차원 공간 객체의 클래스 정의를 화면에 출력하고, 디스크에 저장한다.

넷째, 사용자가 변환을 지시하면, 각 관의 공간 데이터(시작과 끝 노드의 x,y 좌표)와 선택된 속성 데이터를 3차원 공간 객체로 변환한다.

다섯째, 변환이 끝나면, 변환된 결과를 디스크에 저장한다.

이렇게 개발하고자 하는 시스템의 영역이 정의가 되면, 이 단계의 결과물인 문제 서술서를 가지고 분석 단계로 넘어가게 된다.

분석 단계 : 이 단계는 개발하고자 하는 시스템과 관련된 실세계 영역의 일부를 정확하고, 간결하며 객체 지향적인 모델로 정의하는 과정이다.

1) 클래스와 객체의 선정 및 의미 식별

분석 단계에서 제일 먼저 해야 할 일은 공간 데이터 변환 시스템에 존재하는 핵심 클래스들을 정의하는 것이다. 여기에서 선정된 초기 후보 클래스들은 다음과 같다.

* 시설물 데이터(UtilityData)

* 공간 데이터(SpatialData)

* 속성 데이터(AttributeData)

* 변환기(Transformer)

* 3 차원 공간 객체(3DSpatialObjects)

* 관(Pipe)

위와 같이 후보 클래스들을 선정한 후에는 이들의 의미(semantics)를 식별하는 단계로 넘어간다. 즉, 선정된 후보 클래스들의 의미를 명확히 정의하고, 그것이 클래스로서 정의가 되면 Fig. 9와 같은 클래스 명세서에 포함시키게 된다.

2) 클래스와 객체들의 관계 정의

클래스들은 독립적으로 존재하지는 않는다. 이들은 다양한 방법으로 문제 영역의 클래스 구조(structure)를 형성하도록 연관되어 있다. 따라서 이들의 관계(relationship)를 정의함으로써, 다른 클래스에 대한 특정 클래스의 의미와 종속성을 식별할 수 있게 된다.

Fig. 10은 변환기와 지하 시설물 데이터 그리고 3 차원 공간 객체 클래스들의 연관성 관계를 표현한 것이다. 여기에서, 변환기는 시설물 데이터를 읽어 들이고, 시설물 데이터는 변환기에 의해 읽히는 의미적 종속성 관계가 존재한다.

- 클래스 이름 : 시설물 데이터(**UtilityData**)
 - 정의 : 시설물 데이터는 공간 데이터 변환 시스템의 전처리 과정을 통해 생성된 AS-CII 파일들로서, 상수도관 데이터, 하수도관 데이터, 가스관 데이터, 전력관 데이터, 전화관 데이터를 포함한다. 이들은 각각 공간 데이터와 속성 데이터로 이루어져 있다.
- 클래스 이름 : 공간 데이터(**SpatialData**)
 - 정의 : 공간 데이터는 시설물 데이터의 공간 좌표를 담고 있는 데이터이다. 여기서 공간 좌표란, 각 관의 시작과 끝 지점의 x, y 좌표를 말한다.
- 클래스 이름 : 속성 데이터(**AttributeData**)
 - 정의 : 속성 데이터는 각 시설물의 성질을 설명해 주는 데이터이다. 여기에는 수량, 관경, 관의 길이, 시작 깊이, 끝 깊이, 관의 종류, 설치 일자 등이 포함되어 있다.
- 클래스 이름 : 변환기(**Transformer**)
 - 정의 : 변환기는 2 차원 시설물 데이터를 3 차원 공간 객체로 변환해 주는 공간 데이터 변환 시스템의 핵심 클래스이다.
- 클래스 이름 : 3 차원 공간 객체(**3DSpatialObjects**)
 - 정의 : 3 차원 공간 객체는 변환기를 통해 변환된 3 차원 시설물 데이터로서, 시설물 데이터의 공간 데이터 부분에서 x, y 좌표를, 속성 데이터 부분에서 시작 깊이와 끝 깊이를 사용하여 객체의 공간 속성(z 값)을 정의하고, 사용자가 선택한 시설물의 속성 아이템을 객체의 속성으로 정의하게 된다.
- 클래스 이름 : 관(**Pipe**)
 - 정의 : 관은 지하 시설물의 가장 기본적인 특성이다. 모든 지하 시설물은 모두 관으로 구성되어 있으며, 어떤 용도로 사용되느냐에 따라 상수도관, 하수도관 등으로 구분된다.

Fig. 9 The class specification of the classes defined on the analysis stage

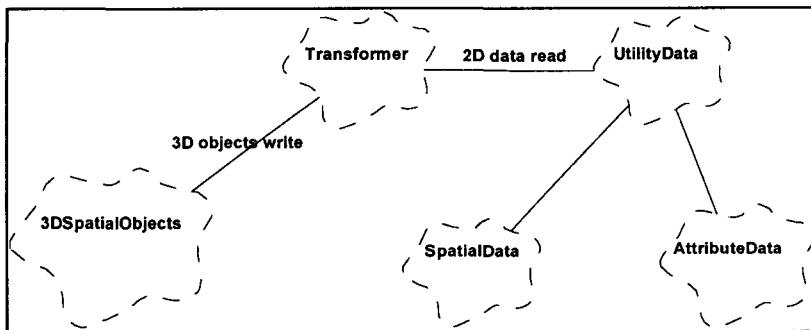


Fig. 10 The relationships among the Transformer, UtilityData, and 3DSpatialObjects

Fig. 11 은 시설물 데이터 클래스와 공간, 속성 데이터 클래스 사이의 집단화 관계를 정의한 것이다. 시설물 데이터는 논리적으로 공간 데이터와 속

성 데이터를 각각 공간 요소와 기술적 요소의 관계로 포함하고 있음을 보여 주고 있다.

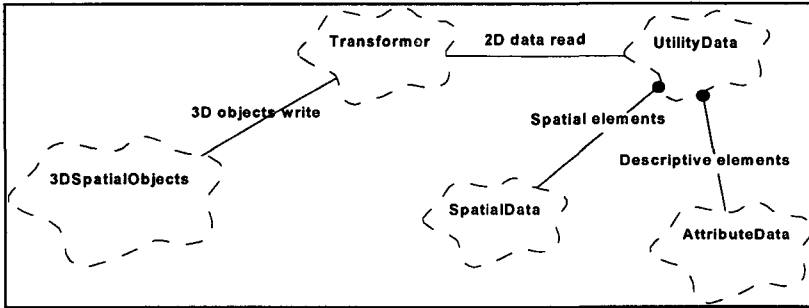


Fig. 11 The Aggregate relationships among the UtilityData, SpatialData, and AttributeData

다음은 이들 관계 사이의 cardinality 를 정의해 주어야 한다. 관계의 cardinality 는 한 클래스의 관계에 참여하는 실체들의 수를 의미한다. Fig. 12 는 Fig. 11 에서 나타난 클래스들 간의 cardinality 를 정의하고 있다.

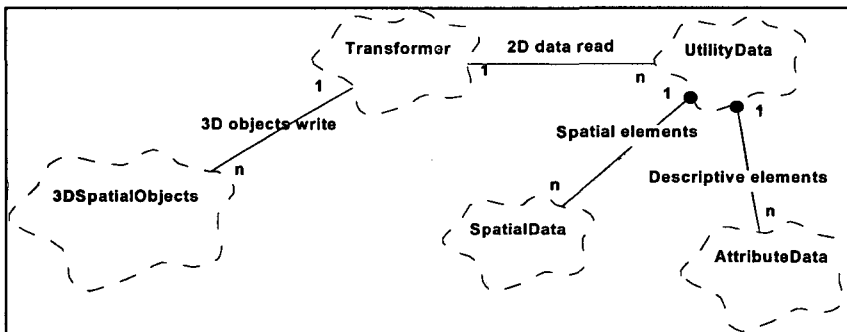


Fig. 12 The cardinality of the relations among the classes

3) 클래스의 메소드 및 애트리뷰트의 정의

이 과정에서는 클래스 구조와 시스템 기능을 지원하기 위해 필요한 주요 메소드들을 정의하고, 그 클래스들을 기술하는 성질들인 애트리뷰트들

을 결정하게 된다.

본 연구에서는 각각의 use-case 에 대한 시나리오를 정의하고, 정의된 시나리오를 수행하기 위해 필요로 하는 메소드를 결정하는 방법을 사용하여 메소드를 결정하였다.

문제 서술서에서 공간 데이터 변환 시스템의 핵심인 변환기의 use-case 들을 상세한 시나리오로 확장한 결과는 Fig. 13 과 같고, 이것을 객체-시나리오 다이어그램으로 표현하면 Fig. 14 와 같다. 객체-시나리오 다이어그램은 하나의 use-case 를 실현하기 위해 객체들이 어떻게 상호 협력하는지를 기술한다. 또한, 그것은 한 시나리오의 실행을 추적한다.

2 차원 시설물 데이터를 3 차원 공간 객체로 변환

1. 2 차원 지하 시설물의 공간 데이터와 속성 데이터를 변환기로 읽음.
2. 3 차원 공간 객체로 변환할 때 포함하고자 하는 속성 아이템들을 선택.
3. 각 관의 공간 데이터와 선택된 속성 데이터를 3 차원 공간 객체로 변환.
4. 변환된 결과를 디스크에 저장.

Fig. 13 The scenario of the Transformer class

Fig. 14 에서 화살표는 객체들간에 전달되는 메시지를 나타내고, 화살표 위에는 메소드의 이름이 메시지가 전달되는 순서를 나타내는 번호와 함께 표현되어 있다. 여기에서 변환기 클래스에 대해 정의되는 주요 메소드는 4 가지가 정의됨을 알 수 있다. 그것들은 시설물 데이터 파일들을 읽기 전용으로 열기 위한 OnOpen(), 화면에 출력된 속성 아이템들을 선택하기 위한 OnSelitem(), 정의된 공간 객체의 형식으로 변환하고, 내부적 계산을 위한 OnCalculation(), 그리고 변환된 공간 객체를 디스크에 쓰는 OnWrite()이다.

각 클래스에 필요한 메소드들의 정의가 끝나면, 그 클래스를 기술하는 성질인 애트리뷰트들을 정의하여야 한다. 애트리뷰트를 정의하기 위해 사용되는 일반적인 방법은 클래스를 하나씩 선정하여 그것의 성질들을 나열

하고, 나열된 성질들을 애트리뷰트로 결정하는 방법이다.

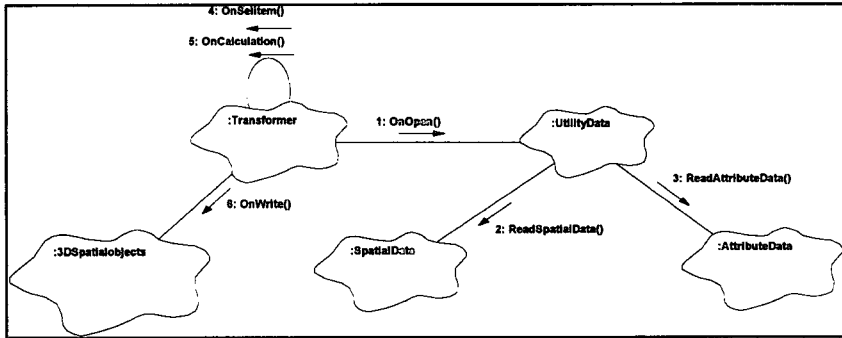


Fig. 14 The object-scenario diagram of the spatial data transformation system

예를 들어, 시설물 데이터의 공간 데이터는 Table 3 과 같은 아이템들을 포함하고 있는데, 이 아이템들 중에서 X-좌표값(x-coord)과 Y-좌표값(y-coord)은 공간 데이터를 기술하기 위한 핵심 성질들이라 할 수 있으므로, 공간 데이터 클래스의 애트리뷰트로 정의할 수 있다. 그리고, 시설물 데이터의 속성 데이터는 Table. 1 에 포함된 아이템들을 속성 데이터 클래스의 애트리뷰트로 정의하였다. 위의 방법들을 이용하여 정의된 주요 메소드와 애트리뷰트들은 Fig. 15 에 나타난 것처럼 클래스 다이어그램에 추가된다.

4) 클래스들 사이의 계승 관계의 정의

클래스들 사이의 계승 관계란 어떤 특정 클래스가 그것의 상위 클래스가 가지는 모든 특성들 뿐만 아니라 자신의 특성들도 가질 때 성립하는 관계이다. 이것은 동일한 영역 내의 일반화(generalization)와 특수화(specialization)를 발견하는 단계라 할 수 있다. 의미적으로 정확한 계승 구조를 정의하는 것은 상위 클래스의 메소드와 애트리뷰트들을 하위 클래스에서 재정의하지 않아도 되기 때문에 재사용이라는 측면에서 매우 유용하다.

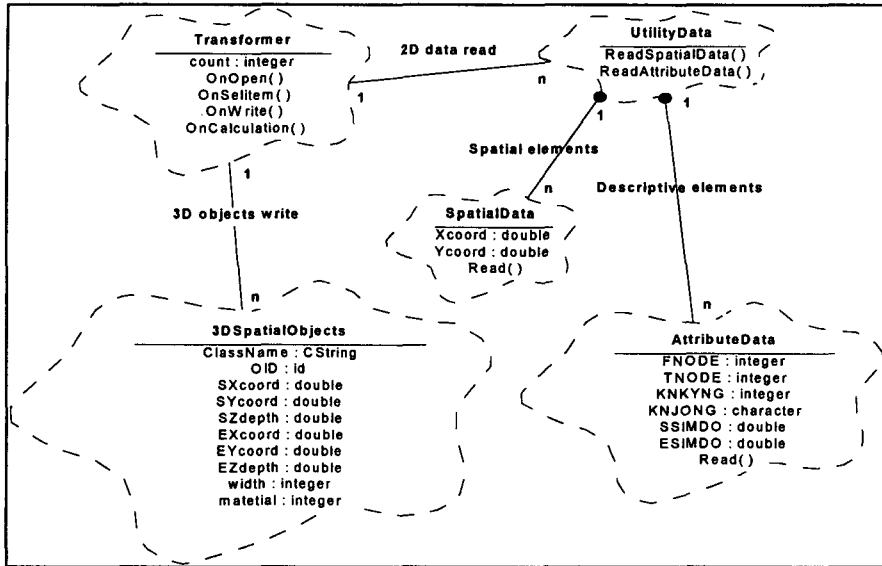


Fig. 15 The class diagram including the defined methods and attributes

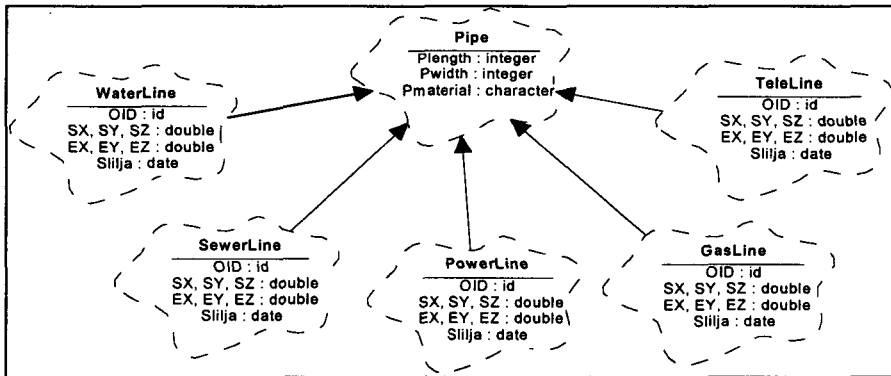


Fig. 16 The inheritance relationships among the pipe classes

Fig. 9의 클래스 명세서에서 관(pipe) 클래스를 정의할 때, 관이 어떤 용도로 사용되느냐에 따라 상수도관, 하수도관, 가스관, 전력관, 그리고 전화관으로 나뉘어지는 것을 발견했다. 그러므로 관 클래스와 5개의 용도별

관 클래스들은 그들의 애트리뷰트들을 정의하면, 이들의 상위·하위 클래스 관계를 식별할 수 있게 된다. Fig. 16은 관 클래스와 상수도관, 하수도관, 가스관, 전력관, 그리고 전화관 클래스들의 계승 관계를 표현한 클래스 다이어그램이다.

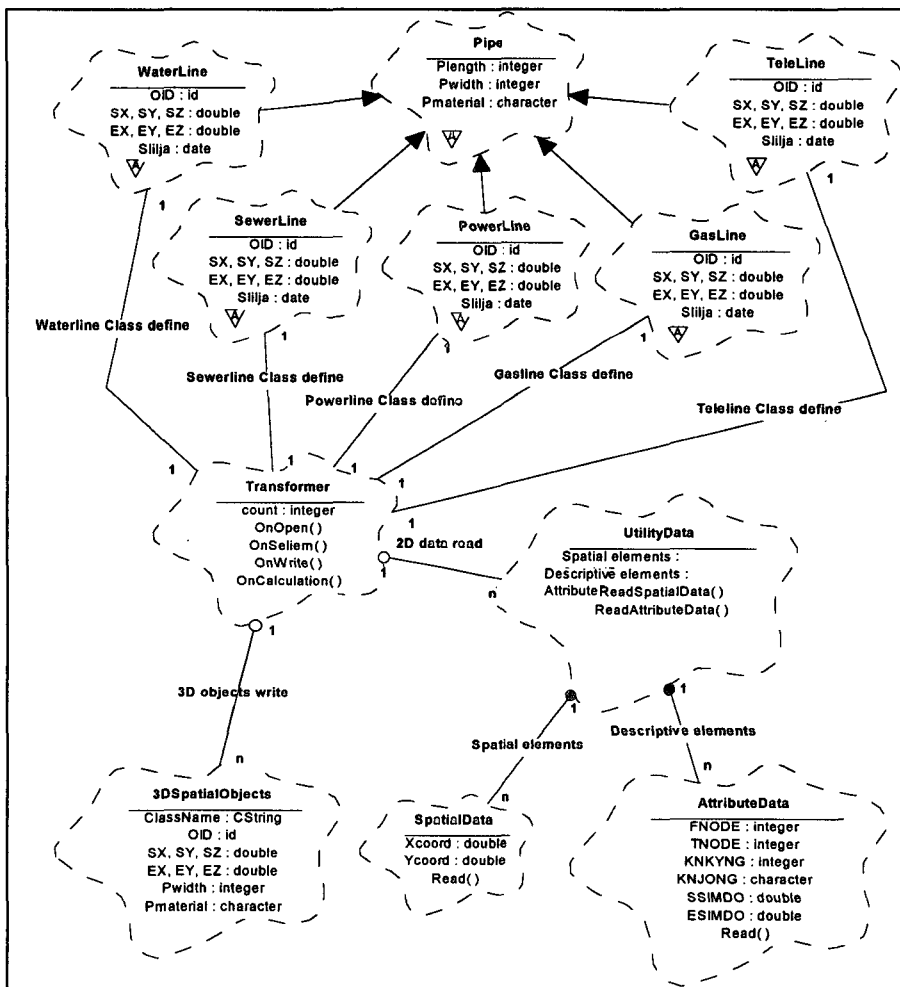


Fig. 17 The final class diagram on the analysis stage

5) 분석 단계의 결과물

지금까지 공간 데이터 변환 시스템을 설계하고 구현하기 위하여 시스템에 대한 분석을 수행하였다. 순차적인 단계를 통해 분석 과정을 설명하였지만, 실제 실험 과정에서는 이들 단계들을 앞,뒤로 오가며 최종적인 결과를 얻을 때까지 반복적으로 수행하게 된다. 분석 단계가 끝난 후 3가지 결과물을 얻을 수 있었다. 그것은 클래스 다이어그램과 클래스 명세서, 그리고 객체-시나리오 다이어그램이다. 이들은 설계 단계의 기본 자료가 되며, 설계 단계가 진행됨에 따라 추가되고 갱신될 수 있다. Fig. 17은 분석 단계에서 얻은 최종 클래스 다이어그램이고, 클래스 명세서는 Fig. 9에 수록하였으며, 객체-시나리오 다이어그램은 Fig. 14에 나타내었다.

설계 단계 : 객체 지향 개발 과정의 세번째 단계인 설계 단계는 분석 단계에서 정의된 데이터를 저장하고 기능을 수행하기 위한 효과적이고 능률적인 구현을 결정하는 과정이다. 분석 단계에서는 개발 될 시스템 영역의 이해에 중점을 두었지만, 이 단계에서는 그 영역의 요구 조건들을 어떻게 구현할 수 있는가에 중점을 둔다. 또한, 시스템의 개발을 위해 필요한, 적절한 하드웨어, 소프트웨어, 운영체제, 그리고 데이터 저장 전략 등이 선정된다.

1) 시스템 구조(architecture)의 정의

구조를 정의하는 과정에는 시스템을 구현하기 위해 사용되는 하드웨어와 소프트웨어, GUI(Graphic User Interface), 그리고 객체 저장 등과 같은 전략적(strategic) 결정을 하는 과정과, 시스템 클래스들을 그룹으로 분할하는 과정이 포함된다.

공간 데이터 변환 시스템을 구현하기 위해 사용되는 하드웨어와 소프트웨어, 그리고 운영 체제의 명세는 Table.4와 같다. 이들을 설계 단계의 초기에 결정하는 것은 이후에 결정될 설계 결정들에 중요한 영향을 주기 때문에 신중히 고려되어야 할 과정이다.

Table. 4 The strategic decision items and the contents on the design stage

| 전략적 결정 사항 | 결정 내용 | 비 고 |
|-----------|--|-----------------|
| 운영체제 | ● Microsoft Windows NT 3.5 | |
| GUI | ● 윈도우 환경 | |
| 하드웨어 | ● Intel Pentium Processor(100Mhz)를 갖춘 컴퓨터 ● 사용자 인터페이스를 위한 모니터 ● 키보드 ● 마우스 ● 하드 디스크와 메인 메모리 | |
| 소프트웨어 | ● Microsoft Visual C++ 4.0 | 프로그 램 구현용 |
| 객체 저장 | ● 객체 형식의 파일 구조로 디스크에 저장([그림 8] 참조) | |

설계 단계의 전략적 결정이 끝나면, 시스템 내의 클래스들을 그룹으로 분할하는 과정을 수행하게 된다. 여기서, 하나의 그룹으로 분할된 클래스 단위를 클래스 범주(category)라 한다. 클래스 범주는 일반적으로 주요한 구조적 기능들을 중심으로 묶여진다. 따라서, 공간 데이터 변환 시스템에서는 다음과 같은 클래스 범주들이 선정되었다(Fig. 18).

- 객체_데이터(Object_Data) 클래스 범주 : 3 차원 공간 객체와 2 차원 지하 시설물 데이터 관리 기능 부분
- 클래스_정의(Class_Definition) 클래스 범주 : 관(pipe)을 비롯한 상수도관, 하수도관, 가스관, 전력관, 그리고 전화관 클래스의 정의 기능 부분
- 변환기(Transformer) 클래스 범주 : 공간 데이터 변환 시스템의 핵심인 변환기 클래스의 기능 부분
- GUI_Library 클래스 범주 : 시스템이 구동하는 운영 체제에 종속적인 부분으로, 운영체제의 변경이 발생시 이 부분만을 수정하면 다른 운영체제에도 이식이 가능하도록 하기 위해 선정한 부분

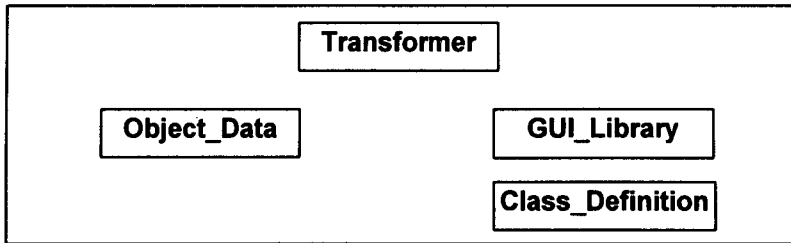


Fig. 18 The class categories of the spatial data transformation system

클래스 범주의 선정이 끝난 후에는 이들 사이의 가시성(visibility)을 결정한다. 가시성이란 한 클래스 범주의 내용을 다른 클래스 범주에 드러내는 것을 뜻하는 용어이다. 예를 들어, “변환기” 클래스 범주는 데이터의 변환을 위하여 “객체_데이터” 클래스 범주에 들어 있는 데이터를 사용해야 하므로 이들 사이에는 가시성이 존재해야 한다. 이것은 클래스의 “사용(using) 관계”를 통해 클래스-범주 다이어그램에 표현할 수 있다. Fig. 19에서 “클래스_정의” 클래스 범주는 데이터 변환 시스템 내의 모든 클래스 범주들에 의해 사용되어지므로 전역적(global)으로 정의 되었다.

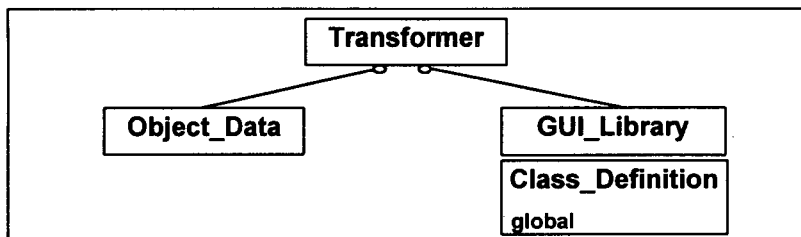


Fig. 19 The class-category diagram for the spatial data transformation system

다음은 시스템의 물리적 설계 단계로서 모듈에 클래스와 객체들을 할당하는데, 이것은 모듈 다이어그램으로 나타낼 수 있다. 설계 단계가 끝난 후 구현 단계에서는 모듈 다이어그램을 참조하여 실제 코드를 작성하게 된다. 공간 데이터 변환 시스템에서 사용되는 클래스와 객체들을 할당한 모듈 다이어그램은 Fig. 20 과 같다.

Fig. 20 에서 “Transformer”모듈은 본 시스템의 메인 프로그램이며, “interface”는 사용자 인터페이스 부분을 정의한 모듈이다. “Transform”은 실제 변환을 수행하는 모듈이고, “Classdef”는 사용자에 의해 정의된 클래스 정의를 담고 있는 모듈이다. 화살표가 나타내는 의미는 컴파일시의 종속성이다. 즉, “Transformer”모듈이 컴파일 되기 위해서는 “interface” 모듈이 꼭 포함(include)되어 있어야 한다.

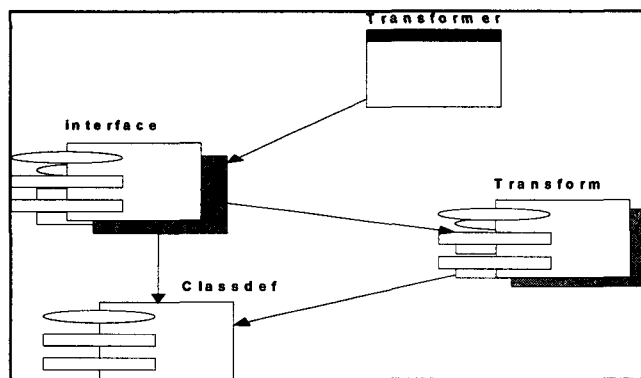


Fig. 20 The module diagram for the spatial data transformation system

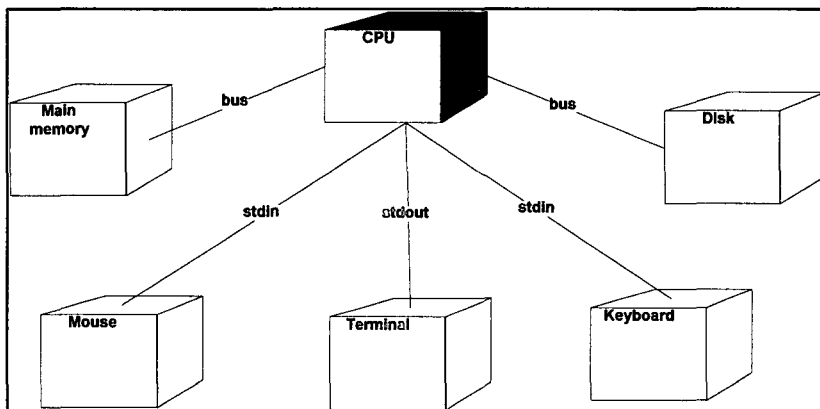


Fig. 21 The process diagram for the spatial data transformation system

시스템 정의 과정에서 프로세스 다이어그램은 프로세서에 프로세스를

할당하는 것을 나타낸다. 이것은 시스템을 개발하는 동안, 개발된 시스템의 실행을 위한 플랫폼으로서의 역할을 하는 프로세서들과 장비들의 물리적 집합을 나타내 준다. 공간 데이터 변환 시스템을 위한 프로세스 다이어그램은 Fig. 21 과 같다.

2) 실행 가능한 배포물(executable release)의 개발

시스템 설계 단계에서는 그 시스템의 하위 레벨의 상세함에 초점을 두는데, 실행 가능한 배포물이 만들어지면 분석 모델에서는 포함되지 않았지만 구현을 위하여 필요한 새로운 클래스들이 추가된다. 왜냐하면 분석 단계에서는 시스템의 사용자 관점에 중점을 두지만, 설계 단계에서는 그것이 어떻게 구현되느냐에 중점을 두기 때문이다. 공간 데이터 변환기를 위한 실행 가능한 배포물은 Fig. 22 와 같다.

Executable Release : Spatial Data Transformer

목적 : ARC/INFO 에서 export 된 지하 시설물 데이터를 3 차원 공간 객체로 변환하기 위한 메시지의 전달 경로와 변환 로직(logic)을 검토.

구현되어야 할 클래스들 :

Transformer, UtilityData, SpatialData, AttributeData, Pipe 와 그들 사이의 관계

구현되어야 할 use-case 들 :

- ✓ 2 차원 지하 시설물 데이터 읽기
- ✓ 읽혀진 데이터의 속성 정보 아이템 출력 및 필요한 아이템 선택
- ✓ 정의된 클래스 정보 출력 및 저장
- ✓ 3 차원 공간 객체의 공간 좌표값 결정 및 속성 정의
- ✓ 변환 및 저장

입력 : 사용자가 선택한 2 개의 데이터 파일 (*line.asc 와 *point.asc)

출력 : [그림 7]의 형식을 갖는 공간 객체

Fig. 22 The executable releases for the spatial data transformation system

3) 설계 단계의 결과물

이 단계의 결과물은 시스템의 전략적 결정 사항 및 내용을 포함하는 시스템 구조 기술서(Table 4), 작업의 기능 단위로 분할한 결과와 그들 사

이의 관계를 설명하는 클래스-범주 다이어그램(Fig. 19), 구현 단계에서 구현할 모듈들에 클래스와 객체들을 할한 내용을 보여주는 모듈 다이어그램(Fig. 20), 시스템을 구성하는 프로세서들과 장비들의 집합을 나타내는 프로세스 다이어그램(Fig. 21), 그리고 시스템의 작업 개요와 기능들을 제공하는 실행 가능한 배포물(Fig. 22) 등이 있다.

구현 및 결과 : 이 단계는 지금까지 수행한 분석과 설계 단계의 결과들을 바탕으로 실제 컴퓨터 상에서 실행이 가능한 프로그램을 작성하는 단계이다. 개발 환경은 Windows NT 3.5 버전의 운영 체제하에서 C++ 언어를 사용하여 프로그램 하였고, 컴파일러는 Microsoft Visual C++ 4.0 버전을 사용하였다.

Fig. 23 은 공간 데이터 변환을 수행하는 변환기를 나타낸 것이다. 여기서 “open” 버튼을 누르면, 변환에 사용될 2 차원 데이터를 선택할 수 있는 Fig. 24 와 같은 다이얼로그 박스가 화면에 나타난다. Fig. 24 에서는 변환할 공간 데이터와 속성 데이터로 상수도관 데이터를 선택한 경우이다.

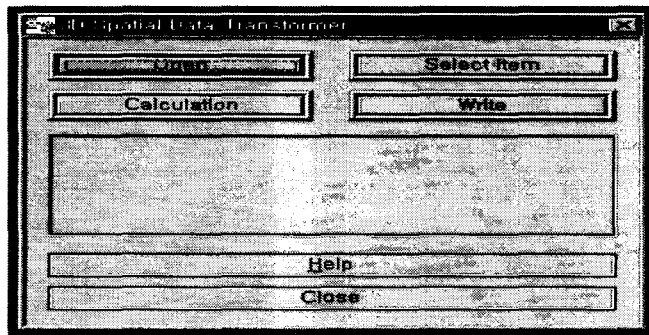


Fig. 23 The 3-D spatial data transformer implemented through C++ language

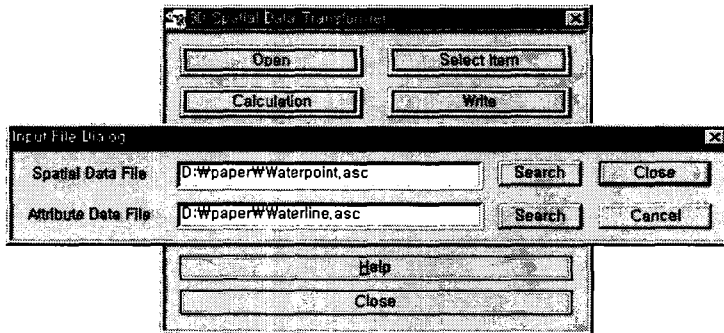


Fig. 24 The 2-D utility data selection screen

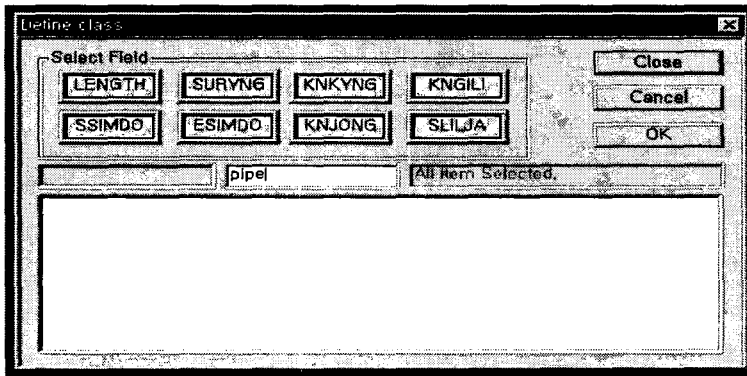


Fig.25 The selection screen of the attribute items to be included in transformation

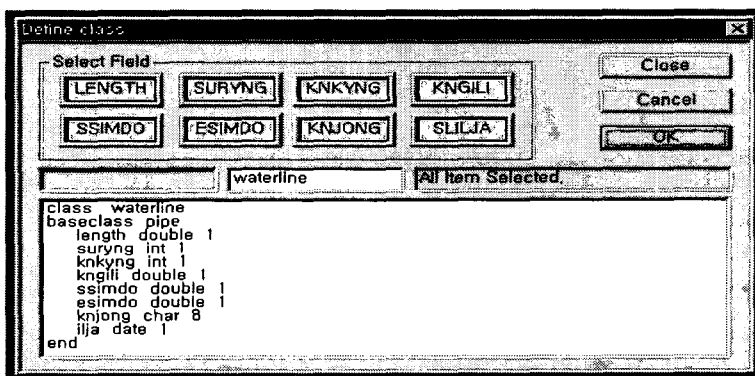


Fig. 26 The screen showing the name and information of the defined class

파일의 선택이 끝난 후 “Select Item” 버튼은 누르면, 선택한 속성 데이터에서 3 차원 공간 객체에 포함될 속성 아이템을 선택하는 다이얼로그 박스가 나타난다(Fig. 25). 초기에는 모든 아이템이 선택되어 있으므로, 필요하지 않은 아이템을 선택하면, 그 아이템은 변환에 포함되지 않는다. 선택이 끝난 후, 선택된 정보를 저장할 클래스의 이름을 입력하고 “OK” 버튼을 누르면 클래스 정보가 Fig. 26의 화면에 출력되면서, 그 내용이 디스크에 저장된다. 그리고, 출력된 클래스 정보를 확인한 후에 “Close” 버튼은 누르면, 다시 Fig. 23의 화면이 나타난다. Fig. 23에서 2 차원 데이터를 3 차원 공간 객체로 변환하는 것은 “Calculation” 버튼이 눌러졌을 때이다. 마지막으로 “Write” 버튼을 누르면, 변환된 3 차원 공간 객체를 디스크에 저장하게 된다.

결 론

본 연구에서는 상용 GIS 소프트웨어인 ARC/INFO에 구축되어 있는 지하 시설물 데이터를 3 차원으로 표현해 주기 위해 필요한 3 차원 공간 객체의 형식과 저장 구조를 정의하고, 2 차원의 지하 시설물 데이터를 정의된 3 차원 공간 객체로 변환해 주는 공간 데이터 변환 시스템을 개발하였다.

이것의 결과로서, 지하 공간에 존재하는 시설물들을 3 차원으로 표현해 줄 수 있는 3 차원 공간 객체를 생성할 수 있었다. 지하 시설물 데이터의 모델링에 기존의 관계형 모델을 적용하지 않고, 객체 지향 모델을 적용하여 공간 데이터와 속성 데이터를 하나의 객체로 묶음으로써, 통합적인 관리가 가능한 효율적인 공간 객체를 생성할 수 있었다. 또한 시스템의 개발 과정에 객체 지향 개발 기법을 적용함으로써, 시스템의 분석과 설계, 그리고 구현의 모든 과정이 분리되지 않고 연속적으로 연결되어, 능률적인 개발을 수행할 수 있었다.

본 연구의 결과로 얻을 수 있는 기대 효과로는 기존의 2차원적인 지하 시설물 관리에서 직관적인 3차원 관리가 가능한 도구가 개발됨으로써, GIS 데이터베이스 구축 시 검증 틀로서 활용되어 정확한 자료가 입력되는 것을 확인할 수 있으므로 부정확한 정보의 입력을 막을 수 있다는 점과, 기존의 GIS 데이터베이스를 활용할 수 있으므로 자원의 낭비를 막을 수 있다는 것이다.

앞으로의 연구 과제로는 공간 객체를 단순히 시각화 하기 위한 목적의 객체 형식과 저장 구조 보다는 공간 객체에 대한 질의와 검색이 가능하도록 하는 객체 형식과 저장 구조에 대한 연구가 요구되며, 변환된 3차원 공간 객체를 효율적으로 시각화할 수 있는 시스템의 개발에 대한 연구가 수행되어야 할 것이다.

참 고 문 헌

- 한국항공, “광주시 도시종합정보시스템 구축,” 한국항공, pp.188, 1993.
- 유복모, “지형공간정보론,” 동명사, pp.777, 1994.9
- 김 원, “객체지향 데이터베이스,” 하이테크정보, pp.403, 1994.
- 이석호, 엄기현, 용환승, 송병호, “객체지향 데이터베이스 시스템”, 흥릉과
학출판사, pp.366, 1995.
- 과학기술처, “지리정보시스템 활용기법”, 과학기술처, pp.421, 1992.
- (주)캐드랜드, “ARC/INFO DATABASE DESIGN”, (주)캐드랜드, 1994.
- G. Booch, “Object-Oriented Analysis and Design with Applications”, The Benjamin/Cummings Publishing Company, Inc. , pp.589. 1994.
- I. White, “Using the Booch Method”, The Benjamin/Cummings Publishing Company, Inc. , pp.208. 1994.
- R. Laurini, “Fundamentals of Spatial Information Systems”, Academic Press Inc., pp.680, 1992.
- Won Kim and Frederick H. Lochovsky, “Object-Oriented Concepts, Databases, and

Applications”, ACM Press, pp. 602, 1989.

Henry F. Korth and Abraham Silberschatz, “Database System Concepts”, McGraw-Hill, Inc. , pp.694, 1991.

I.T. Hawryskiewicz, “Database Analysis and Design”, Maxwell Macmillan International Editions, pp. 574, 1991.

M.F. Worboys, H.M. Hearnshaw, and D.J. Maguire, "Object-Oriented Data Modeling for Spatial Databases," Intl. Journal of Geographical Information Systems, Vol.4, No.4, pp.369-383, Oct. - Dec. 1990.

M.F. Worboys, "Object-Oriented Approaches to Geo-referenced Information," Intl. Journal of Geographical Information Systems, Vol.8, No.4, pp.385-399, Jul. - Aug. 1994.

Max J. Egenhofer and Andrew U. Frank, “Object-Oriented Modeling : A Powerful Tool for GIS”, NCGIA Seminar Workbook, pp. 1-12, 1988.

ESRI, “Enterprise GIS : Using GIS in the Corporate Environment”, ESRI White Paper Series, pp. 1-24, 1993.