

SDTS와 Informap간의 데이터 변환 시스템의 설계 및 구현

Design and Implementation of a Data Conversion System between SDTS and Informap

오 병 우* 이 강 준* 한 기 준**
Oh, Byoung-Woo Lee, Kang-Jun Han, Ki-Joon

要 旨

독립적인 저장 구조로 공간 데이터를 저장하는 지리 정보 시스템간의 데이터 교환은 매우 힘들다. 더욱이, 공간 데이터의 저장을 위해서는 많은 기억 공간이 필요하고 입력을 위해서는 많은 비용이 필요하므로 중복 저장하면 낭비가 크다. 그러므로, 다른 지리 정보 시스템과 데이터를 서로 교환하여 공유할 수 있어야 하며, 공간 데이터의 교환을 위해서는 공통데이터교환 포맷이 필요하다. 본 논문에서는 기존의 매핑 시스템인 Informap의 데이터와 국가 공통데이터교환 포맷으로 지정된 SDTS 데이터의 상호 교환을 위한 데이터 변환 시스템을 설계 및 구현한다. 이를 위하여 먼저 Informap과 SDTS의 저장 구조를 분석한다. 그리고, 이를 근거로 효율적인 변환을 위한 게이트웨이 함수를 구현하고, 이러한 게이트웨이 함수를 사용하여 데이터 변환 시스템을 설계 및 구현한다.

ABSTRACT

It is very difficult to exchange geographical data among geographical information systems which store their spatial data with independent storage structures. Since, moreover, large amount of storage space is necessary to store spatial data and expensive cost is required to input them, waste will grow as they are stored redundantly. Therefore, it is essential to share them with other geographical information systems by exchanging spatial data among them. In order to exchange spatial data efficiently, there exist several international standards for data exchange format. In this paper, we design and implement a data conversion system that converts geographical data between SDTS (Spatial Data Transfer Standard) which is adopted as the national standard for common data exchange format and Informap which is the existing mapping system. We first analyze the storage structures of SDTS and Informap, respectively and develop gateway functions according to these analyses for efficient conversion. Finally, we design and implement the overall data conversion system between SDTS and Informap using the gateway functions.

* 건국대학교 컴퓨터공학과 대학원생

** 건국대학교 컴퓨터공학과 교수

1. 서론

공간 데이터의 입력에는 많은 비용과 시간이 필요하다. 그러나, 지리 정보 시스템들은 각자 독자적인 저장 구조로 공간 및 비공간 데이터를 저장하고 있다. 특정 지리 정보 시스템에서 사용하던 데이터를 다른 지리 정보 시스템에서는 그대로 사용할 수 없기 때문에 데이터의 효용성이 감소하고 중복성이 증가하게 된다. 일반적으로 공간 데이터는 대용량인 경우가 많아 데이터를 중복 저장할 경우에 커다란 기억 공간의 낭비를 초래하게 된다. 그러므로, 각 응용별로 서로 다른 공간 데이터를 제작 및 유지하기 보다는 교환을 위한 하나의 공통 데이터로 변환한 후에 이를 공유하는 방법을 채택하여야 한다. 기존에 입력된 공간 데이터의 중복 입력 및 중복 저장을 줄이기 위한 데이터 저장 구조의 표준이 필요하며 이 표준을 준수하는 데이터로의 상호 변환이 필요하다^{3,11,14,17}.

최근 국내에서는 지리 정보 시스템이 국가 경쟁력 강화 및 행정 생산성 제고 등에 기반이 되는 사회 간접 자본이라는 전제하에 국가 차원에서 지리 정보 시스템의 국가 표준을 설정하고 기본 공간 정보 데이터 베이스를 구축하며, 또한 지리 정보 시스템 관련 기술 개발을 지원하여 지리 정보 시스템의 활용 기반과 여건을 성숙시켜 가고 있다. 또한, 국가 지리 정보 시스템 사업의 일환으로 정보통신부(한국전산원)에서는 표준화분과 위원회를 구성하여 한국 표준을 제정하고 있다. 특히, 국가 기본도 포맷과 공동데이터교환 포맷의 표준 제정을 위한 소위원회를 구성하여 표준화 작업을 진행한 결과 공동데이터교환 포맷으로 SDTS(Spatial Data Transfer Standard)를 채택하였다^{12,17}.

본 논문에서는 공동데이터교환 포맷으로 채택된 SDTS와 많은 응용에 사용되고 있을 뿐만 아니라 특히 한국통신에서 전국을 수치지도화하여 선로 관리를 위해 사용하는 Informap 매핑 시스템¹⁸의 데이터 저장 구조를 분석하고, 이를 근거로 SDTS와 Informap간의 데이터 변환 시스템을 설계 및 구현한다. SDTS와 Informap간의 데이터 변환 시스템은 에러 처리 모듈, 로그 화일 처리 모듈, 화일 처리 모듈, Informap 데이터 처리 모듈, SDTS 데이터 처리 모듈, 변환 처리

모듈, FIPS 123 라이브러리¹⁰로 구성된다. 특히, 변환 처리 모듈은 변환 방향에 따라 Informap 변환 처리 모듈과 SDTS 변환 처리 모듈로 구분된다. 이러한 모듈별 프로그래밍을 통해 다양한 환경의 변화에 쉽게 대처할 수 있도록 한다. 예를 들어, 에러 처리를 다른 방식으로 처리하기 위해서 시스템 전체를 변경하지 않고 에러 처리 모듈만을 변경하여 모든 에러의 처리를 원하는대로 변경할 수 있어 재활용성을 높일 수 있다.

본 시스템에서 모든 데이터의 변환은 cm_Initialize(), cm_Close(), cm_InsertClass(), cm_EndClass(), cm_SetAttr(), cm_InsertAttr(), cm_InsertSpatial()의 7개 게이트웨이 함수(gateway functions)를 통해 이루어진다. 본 논문에서 이러한 게이트웨이 함수를 사용함으로써 변환 과정을 두 과정(Front-end와 Back-end)으로 나눌 수 있고 각 과정을 개별적으로 개발할 수 있어 독립성과 유연성을 제공할 수 있게 된다. 또한, 특정 과정을 대체하여 다양한 데이터 형식간의 변환이 가능하다. 즉, 타 시스템의 공간 데이터 처리를 Front-end로 대체하고 Back-end로서 SDTS 처리 모듈을 사용하면 타 시스템의 데이터를 SDTS로 변환할 수 있게 된다.

데이터의 변환시에 모든 데이터를 메인 메모리에 적재하기에는 공간 데이터가 일반적으로 매우 크므로 운영체제에게 무리를 주게된다. 본 논문에서는 이러한 문제점을 해결하기 위하여 한번에 한 객체만을 읽어 게이트웨이 함수를 통해 전달하는 방식을 채택하였다. 그러므로, 메인 메모리의 저장 공간이나 운영체제의 메모리 처리 방식과 무관하게 모든 데이터를 처리할 수 있다.

본 논문의 구성은 다음과 같다. 제 2 장의 관련 연구에서는 SDTS와 Informap을 분석하고, 제 3 장에서는 전체적인 구조, 작업 흐름, 게이트웨이 함수 등에 대해 설명한다. 제 4 장과 제 5 장에서는 데이터의 상호 변환에 대해 언급한다. 마지막으로 제 6 장에서는 결론 및 이후의 연구 과제에 대해 언급한다.

2. 관련 연구

2.1 SDTS 분석¹²

SDTS는 모든 종류의 공간 데이터(지리 정보, 지도)

들을 서로 전환 가능하게 해주는 표준으로 상이한 하드웨어, 소프트웨어, 운영체제간에서 데이터 교환을 가능하게 한다. 미연방에 의해 추진된 SDTS는 1992년 미연방 표준(FIPS Publication 173)으로 채택되었다. SDTS는 많은 다른 형식, 모델, 구조들로 서로 연관된 속성 데이터들을 FIPS 173을 사용하여 다른 시스템간에 교환할 수 있도록 정의되어 있으며, 이는 크게 3개의 부분인 Part 1, Part 2, Part 3으로 구성된다. 또한 Part 1, 2, 3에 의해 특정 데이터 유형에 관한 구체적인 전환 형식으로서 Part 4 벡터 데이터를 위한 TVP(Topological Vector Profile)가 존재한다.

2.1.1 Part 1

SDTS Part 1은 공간 데이터 표준의 일반적인 정의 및 규정을 포함하고 있다. Part 1은 5개의 Section으로 분류된다. Section 1은 개론, Section 2는 공간 데이터의 개념적 모델, Section 3은 공간 데이터 질, 그리고 Section 4, 5는 전환과 관련된 레코드와 필드 수준의 내용을 포함한다.

Section 2의 공간 데이터 개념적 모델은 속성을 가진 공간 개체로서 실제 세계를 표현하는 공간 현상(Spatial Phenomena), 공간 현상과 공간 객체간의 관계를 규정하는 공간 형상(Spatial Features), 그리고 다양한 공간 실체들을 수치적으로 순서적 및 추상적으로 표현하는 공간 객체(Spatial Objects)등의 모델로 구분된다. 래스터와 벡터 객체들은 Geometry Only(G)와 Geometry and Topology(GT) 같은 단순 객체와 분류 개체의 집합으로 표현된다. Geometry-Only(G) 공간 객체들에는 포인트, 라인 세그먼트, 스트링, 호, G-링, 내부 영역, G-다각형, 점, 그리드 셀 등이 있으며, Geometry and Topology(GT) 공간 객체에는 노드, 링크, 체인, GT-링, GT-다각형 등이 있다.

Section 3은 사용에 적합한 데이터 특성이나 데이터의 질에 대한 정보들로서 Lineage, 위치 정확성(Positional Accuracy), 속성 정확성(Attribute Accuracy), 논리 일관성(Logical Consistency), 그리고 완전무결성(Completeness) 등을 정의하고 있다.

Lineage는 원본 데이터 생성 시간, 데이터의 갱신 및 갱신 시간 등과 같은 시간 정보와 공간 데이터 좌표의 변환 방법 등을 기술하며, 위치 정확성은 위치 정보가 실제 위치와 얼마나 근접한지에 대한 정보를 갖는다. 속성 정확성은 비위치성 데이터에 대해 기술하며, 논리 일관성은 인코딩시 공간 데이터 구조와 관계가 충실하게 조작성되어 있는지를 검증하며, 마지막으로 완전무결성은 사용된 정의, 척도(기준), 다른 관련 규정에 대한 정보를 기술한다.

Section 4, 5는 공간 데이터 변환을 위한 논리적 변환 방식을 정의하고 있는데, 이 단계는 전환의 중간 단계로 개념적 수준에서 물리적 수준으로의 연결 역할을 수행한다. SDTS 전환은 모듈들로 구성되며, 각 모듈은 하나 이상의 모듈 레코드들로 구성된다. 그리고, 각 모듈 레코드는 하나 이상의 모듈 필드들로 구성되고, 각 필드는 하위 필드들로 구성된다.

SDTS는 34개 유형의 모듈들로 구성되며, 레코드, 필드, 하위 필드가 구성되는 방식을 상세하게 정의하고 있다. 34개의 모듈은 글로벌 모듈, 데이터 질 모듈, 공간 객체 모듈, 속성 모듈, 그래픽 표현 모듈 등 5개의 범주로 나뉘어 진다.

(1) 글로벌 모듈은 데이터 전환시 기본 내용을 갖는 식별 모듈, 각각의 모듈들의 위치, 모듈들의 관계, 범주화를 정의하는 카탈로그 모듈, 좌표 체계에 관한 내용을 갖는 공간 참조 모듈, 공간 개체와 속성의 의미, 구조, 연결을 정의하는 데이터 사전 모듈, 보안에 대한 정보를 갖는 보안 모듈, 그리고 간단한 통계적 정보를 갖는 변환 통계 모듈 등이 있다.

(2) 데이터 질 모듈은 5개의 필수 모듈 유형을 갖는다. Lineage 모듈, 위치 정확 모듈, 속성 정확 모듈, 논리 일치 모듈, 그리고 완전무결성 모듈 등이다. 하나의 질 레코드는 작게는 한 문장에서 백과 사전 전체 정도의 가변 길이의 텍스트 보고서를 가질 수 있으며, 임의 개수의 속성 레코드를 참조하거나, 개개의 모듈 레코드들의 임의 개수의 질이 외부 식별자를 통하여 설명될 수도 있다. 또한, 모든 변환에서 뿐만이 아니라 단지 카탈로그/상호-참조

레코드를 통하여 연결된 선택 모듈에 대해서도 설명될 수 있다.

- (3) 공간 객체 모듈은 다양한 형상 수준의 공간 데이터를 공간 객체 모듈로 변환한다. 공간 객체 모듈은 포인트, 라인, 다각형 등의 벡터 모듈과 래스터 형태의 데이터를 표현하는 래스터 모듈, 그리고 하나 이상의 벡터 공간 객체들로 구성된 합성 모듈로 구성된다.
- (4) 속성 모듈은 내용과 구조의 측면에서 속성 데이터와 공간 데이터를 연결해 준다. 공간 객체와 직접 연결되는 속성 주모듈과 위계를 갖는 속성 구조를 허용하고, 다른 속성 모듈과 연결될 수 있는 속성 부모듈을 갖는다.
- (5) 그래픽 표현 모듈은 공간 데이터의 표현에 대한 정보를 포함하고 있으며 텍스트 표현 모듈, 심볼 표현 모듈, 라인 표현 모듈, 영역 채움 표현 모듈, 색 인덱스 모듈, 그리고 폰트 인덱스 모듈 등이 있다.

2.1.2 Part 2

데이터 변환 과정 중에서 공간 형상의 공통적인 정의가 필요하게 되며 특히 공간 데이터 사회(Spatial Data Community)에서의 데이터 공유의 필요성은 더욱 강조되는데 SDTS의 Part 2 는 표준 개체 리스트의 개발을 위한 시도이다. Part 2 의 리스트들은 서로 다른 기관에서 저마다 다른 요구에 의해 약 2600개의 지리 형상 정의를 모두 고려할 수 있도록 수년간에 걸친 노력으로 이루어졌다. 현재 리스트들은 200개의 개체 형태 정의, 244개의 속성 정의, 1200개의 포함 용어(Included Term)들로 구성된다.

2.1.3 Part 3

ISO 8211 화일은 Data Description File(DDF)로 불리며 두 가지 형태의 레코드인 DDR(Data Descriptive Record)과 DR(Data Record)로 구성된다. DDR은 데이터 구조와 데이터 구조 설명, DR은 실질적인 데이터를 갖는다. 한 화일에는 하나의 DDR과 하나 이상의

DR이 존재한다. DDF의 레코드는 한개 이상의 필드로 구성되고, 필드는 하나 이상의 서브 필드들로 이루어지며, 서브 필드는 데이터의 기본적인 요소(element)이다. DDF의 레코드 구조와 필드 구조를 살펴보면 다음과 같다.

- (1) 레코드 구조 : DDR과 DR은 각각 3 부분으로 나뉘어진다. DDR과 DR에서 첫번째 부분인 리더(leader)는 레코드 길이, 그리고 DDR과 DR에서의 두번째 부분인 디렉토리내 개체들의 길이 등을 갖으며, 두번째 부분인 디렉토리는 각 필드의 태그, 길이, 위치에 대한 정보들을 갖는다. 세번째 부분은 DDR에서는 DDA(Data Descriptive Area)라 불리고 DR에서는 UDA(User Data Area)라 불린다. DDA는 필드들의 구조와 필드들의 설명을 갖고 있으며, UDA는 실제 데이터 필드들을 갖는다. DDA는 필드의 구조와 설명을 하는 4부분으로 구성되는데, 필드 구조와 유형을 나타내는 필드 제어 부분, 필드의 이름을 기술하는 필드명 부분, 필드를 구성하고 있는 하위 필드들을 기술하는 라벨 부분, 그리고 하위 필드들의 포맷을 설명하는 포맷 부분으로 구성된다.

- (2) 필드 구조 : 필드 구조는 하나의 값을 갖는 요소 구조, 일차원 배열을 갖는 벡터 구조, 그리고 이차원 삼차원 배열을 갖는 배열 구조를 갖는다. 요소 구조는 하위 필드가 없이 필드만으로 존재하며, 벡터 구조는 각 하위 필드들이 나열되는데 "LABEL1!LABEL2!LABEL3"처럼 '!'로 구분된다. 배열 구조는 벡터 구조들이 '*'에 의해 연결되어 다차원 배열 구조를 갖는다. 즉, A!B*X!Y는 AX AY BX BY의 구조로 표현된다.

2.1.4 Part 4

SDTS Part 1, 2, 3의 일반적인 정의에 의해 규정된 내용이 구체적으로 전환 가능하도록 SDTS의 제한된 하위집합으로 profile을 정의한 SDTS Part 4 즉, TVP(Topological Vector Profile)는 벡터 데이터의 실

제적인 전환을 위해 벡터 데이터의 규정, 전환 등을 포함한다. TVP는 지리 벡터 데이터를 플래너 그래프 위상과 함께 갖고 있으며, 전환의 적합성, 인코더 및 디코더의 적합성 등의 규정을 포함한다. 또한, 공간 객체로 노드, 플래너, 완전무결 체인, GT-다각형, 합성들이 필수적으로 필요하다. TVP 모듈명은 표준화된 4문자로 구성되어 실제 전환시 모듈의 역할, 포함되는 정보 등에 대한 설명, 모듈의 한계와 요구되는 내용, 구체적으로 ISO 8211방식으로 전환될 때 Part 1에서 규정된 내용이 어떻게 전환되는 지에 대한 내용 등을 포함한다. 공간 모듈의 모듈명은 공간 객체 코드(예를 들면, LS, NE, NO 등) 두문자로서 시작하고, 마지막의 두문자는 상이한 모듈/화일을 구분하기 위해 자유롭게 허용된다. 속성 모듈의 모듈명은 A(Attribute Primary) 또는 B(Attribute Secondary)로 시작하는 4문자로 구성된다.

2.2 Informap 분석(20)

Informap은 미국 Synercom사에서 개발한 매핑 시스템으로 공간 및 비공간 데이터를 저장한다. Informap은 관계 구조의 특징과 공간 데이터의 검색 능력이 결합된 지리 관계형 데이터베이스 구조를 채택하고 있다.

Informap 데이터베이스 시스템은 지도 데이터를 여러 유형의 요소들로 분류하며, 분류된 요소들을 해당 화일로 나누어 저장한다. 이러한 저장 구조는 모든 데이터를 한 화일에 저장하는 구조보다 데이터를 빠르고 쉽게 관리할 수 있도록 한다. Informap에는 공간 데이터와 속성 데이터가 있다. Informap에서의 공간 데이터는 라인, 주석, 심볼로 구분되며 지도나 도면의 요소들을 정의하는 모든 점들의 좌표만을 저장한다. 그리고 속성 데이터는 공간 데이터로부터 포인터(ID)로 연결되며 공간형 데이터 화일과 별도의 화일로 저장된다. 속성 데이터는 지도상의 공간 검색 및 통계 관리에 유용하게 사용된다.

Informap에서 사용되는 데이터베이스는 공간 데이터 화일과 비공간 데이터 화일, 그리고 공간 데이터 화일에 관련된 비공간 데이터를 참조하기 위한 수단

인 포인터로 구성된다. 지리 데이터 화일은 최대 128개의 레이어로 저장된다. 각 레이어는 SDT(Schema Description Table)에 정의되어 있고 실제 데이터들은 Kind 1~5 등의 형식으로 저장된다. 그리고, Informap의 모든 지도나 도면은 내부 직각 좌표계를 가지고 있다. 이 좌표계는 X, Y 값이 한정되어 있으며 이러한 제한을 극복하기 위해 원점을 이동하여 좌표를 나타낸다.

Informap에는 데이터의 유형을 정의하여 다음과 같이 분류해 놓았다.

- Kind 1 : 속성 데이터가 없는 공간 데이터를 저장한다.
- Kind 2 : 지도상의 주석을 저장한다.
- Kind 3 : 속성 데이터가 없는 공간 심볼 데이터를 저장한다.
- Kind 4 : 속성 데이터를 포함하는 공간 심볼 데이터를 저장한다.
- Kind 5 : 속성 데이터를 포함하는 공간 데이터를 저장한다.

2.2.1 SDT (Schema Description System)

Informap을 SDTS로 변환하기 위해 사용하는 SDT 화일은 Informap에서 사용되는 레이어(layer)들에 대한 정보를 기술한다. SDT 화일은 Informap 시스템과 각 테이블을 연결하고, 또한 사용되는 레이어와 레이어에 해당하는 종류(Kind 1~5)를 지정한다. 그리고, 속성 데이터의 명칭과 저장 형태를 지정하며 공간과 속성 화일의 확장자 지정한다. SDT 화일에서 레이어는 0~8 컬럼에 레이어 이름, 12~14 컬럼에 레이어 번호, 17~19 컬럼에 공간 화일 확장자, 22~24 컬럼에 비공간 화일 확장자, 26~28 컬럼에 Kind 번호, 72 컬럼에 변수 배정도(variable precision)를 저장한다. 그리고, 속성은 4~12 컬럼에 애트리뷰트 이름, 14~16 컬럼에 길이, 18 컬럼에 형식(type)을 저장한다.

2.2.2 Kind 1

Informap의 Kind 1은 비공간 데이터가 없는 라인

데이터를 저장하는 역할을 담당한다. Kind 1 공간 화일의 레코드 길이는 6바이트로 다음과 같은 구조로 저장된다.

C														X	Y
0	1	2	3	4	5	6	7	0	1	2	3	4	5		
line type - 1														cntl	

여기서 “cntl”은 레코드의 제어를 저장하며 0이면 “Curve”, 1이면 “Pen Down”, 3이면 “Pen Up”이다.

2.2.3 Kind 2

Informap의 Kind 2은 지도상의 주석을 저장하는 역할을 담당한다. Kind 2 공간 화일의 레코드 길이는 12바이트로 다음과 같은 구조로 저장된다.

C														X	Y	NGP	X	Y
0	1	2	3	4	5	6	7	0	1	2	3	4	5					
anno. type - 1														cntl				

여기서 “cntl”이 2이면 “Annotation”, 3이면 “Deleted”이다.

Kind 2의 비공간 화일은 다음과 같은 구조로 저장된다.

NUM	CHR	NUM	CHR	(NUM)	(CHR)	NUM	CHR
-----	-----	-----	-----	-------	-------	-----	-----

2.2.4 Kind 3

Informap의 Kind 3은 비공간 데이터가 없는 공간 심볼 데이터를 저장하는 역할을 담당한다. Kind 3 공간 화일의 레코드 길이는 12바이트로 다음과 같은 구조로 저장된다.

C														X	Y	S	X	Y
0	1	2	3	4	5	6	7	0	1	2	3	4	5					
symbol type - 1														cntl				
														X	Y		X	Y

여기서 “S”는 X와 Y의 확대 비율이며, “cntl”이 0이면 “Symbol”, 3이면 “Deleted”이다.

2.2.5 Kind 4

Informap의 Kind 4는 비공간 데이터를 포함하는 공간 심볼 데이터를 저장하는 역할을 담당한다. Kind 4 공간 화일의 레코드 길이는 12바이트로 다음과 같은 구조로 저장된다.

C														X	Y	NGP	X	Y
0	1	2	3	4	5	6	7	0	1	2	3	4	5					
sec. - 1														symbol type - 1		cntl		

여기서 “NGP”는 비공간 화일의 레코드를 가리키며, “cntl”이 0이면 “Symbol”, 2이면 “Secondary annotation”, 3이면 “Deleted”이다.

Kind 4의 비공간 화일은 앞서 설명한 SDT에 정의된 속성 데이터를 저장한다.

2.2.6 Kind 5

Informap의 Kind 5는 비공간 데이터를 포함하는 라인 데이터를 저장하는 역할을 담당한다. Kind 5 공간 화일의 레코드 길이는 8바이트로 다음과 같은 구조로 저장된다.

C														X	Y	NGP	
0	1	2	3	4	5	6	7	0	1	2	3	4	5				6
sec. - 1														line type - 1		cntl	

여기서 “NGP”는 비공간 화일의 레코드를 가리키며, “cntl”이 1이면 “Pen Down”, 2이면 “Secondary annotation”, 3이면 “Pen Up”이다.

Kind 5의 비공간 화일은 앞서 설명한 SDT에 정의된 속성 데이터를 저장한다.

3. 전체적인 구조

본 장에서는 SDTS와 Informap간의 데이터 변환 시스템의 전체적인 구성과 작업 흐름, 변환을 위한 게이트웨이 함수, 그리고 개발 환경 및 사용한 도구들에 대해 설명한다.

3.1 시스템 구성

본 논문에서 제시하는 SDTS와 Informap간의 데이터 변환 시스템의 전체적인 구성은 그림 3.1과 같다. 즉, 에러 처리 모듈, 로그 화일 처리 모듈, 화일 처리 모듈, Informap 데이터 처리 모듈, SDTS 데이터 처리 모듈, 변환 처리 모듈, 그리고 FIPS 123 라이브러리로 구성된다.

각각의 구성 모듈에 대한 설명은 다음과 같다.

- (1) 변환 처리 모듈 : 게이트웨이 함수들로 구

- 성되며 변환 방향에 따라 Informap 변환 처리 모듈과 SDTS 변환 처리 모듈을 호출한다.
- (2) Informap 데이터 처리 모듈 : 화일 처리 모듈을 통해 화일 시스템에 접근하며 Informap에 대한 모든 처리를 담당한다.
- (3) SDTS 데이터 처리 모듈 : FIPS 123 라이브러리를 통해 SDTS 데이터에 접근하며 SDTS에 대한 모든 처리를 담당한다.
- (4) 화일 처리 모듈 : UNIX 화일 시스템으로부터 Informap 데이터를 읽어 Informap 데이터 처리 모듈로 전달하는 역할과 Informap 데이터 처리 모듈로부터 받은 데이터를 화일 시스템에 저장하는 역할을 담당한다.
- (5) 로그 화일 처리 모듈 : 변환 과정에서 발생하는 손실을 화일 처리 모듈을 통해 저장하는 역할을 담당한다. 로그의 내용에는 좌표의 손실도 포함한다. 즉, SDTS에서 Informap으로의 변환시에 배정도(precision)의 차이에 따라 손실되는 정보를 출력하며, 손실을 최소화시킬 수 있는 옵션(확대 비율과 시작점 좌표)을 제시한다. 그리하여, 사용자가 손실을 감수하면

- 서라도 데이터의 변환을 원할 경우는 제시된 옵션을 사용하여 손실이 최소화된 변환이 가능하도록 한다.
- (6) 에러 처리 모듈 : 변환시에 발생하는 모든 에러에 대한 처리를 담당한다.
- (7) FIPS123 라이브러리¹⁰⁾ : SDTS 데이터의 물리적인 접근에 대한 처리를 담당한다.

3.2 작업 흐름

SDTS로부터 Informap으로의 변환은 우선 SDTS의 글로벌 모듈의 분석을 통해 변환될 모듈을 선정한 후 공간 데이터와 관련된 속성 모듈을 분석하여 임시 테이블을 생성한다. 임시 테이블은 순차적으로 데이터를 저장하는 SDTS의 구조에서 Informap의 레이어 구조로 매핑하는데 사용된다. 분석이 끝난 공간 데이터 모듈은 표 3.1과 같이 Informap으로 변환 가능한 Kind를 결정한 후 Informap 클래스를 생성한다. 공간 데이터의 변환시에 SDTS 공간 데이터 모듈은 Informap에서 공간 화일로 변환되며 연결된 속성 데이터는 비공간 화일로 변환된다.

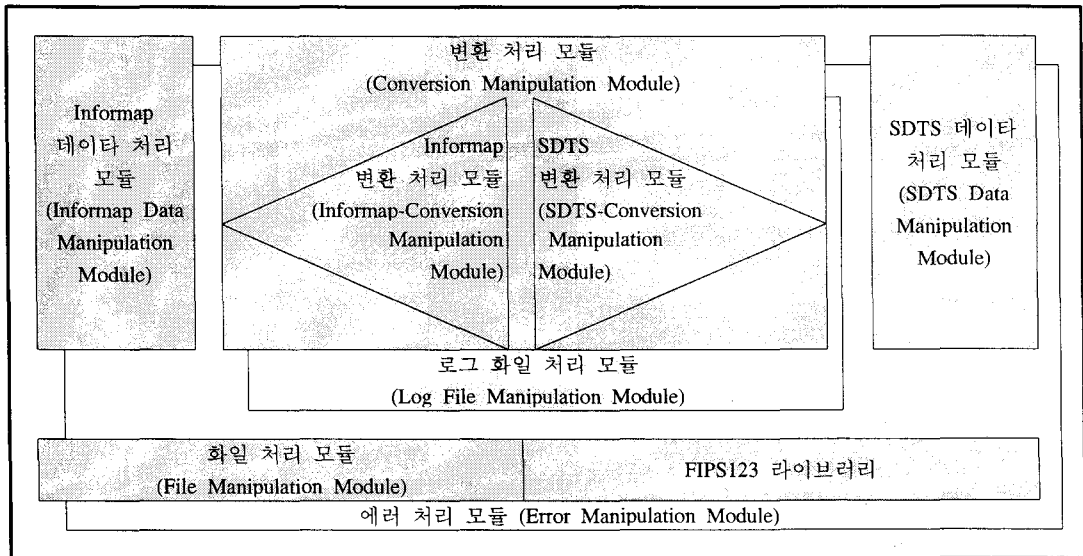


그림 3.1 전체적인 구성

표 3.1 SDTS와 Informap의 변환 매핑

	SDTS 공간 모듈	Informap의 Kind
0 차원 공간 객체	NE-개체 포인트	kind 4 (또는 kind 3)
	NL-라벨 포인트	kind 2
1 차원 공간 객체	LS-라인 스트링	kind 5 (또는 kind 1)

Informap으로부터 SDTS로의 변환은 우선 Informap의 SDT 화일의 분석을 통해 처리할 클래스(즉, Informap의 레이어)를 찾고 Kind에 의해 어떤 모듈로 변환할 지를 표 3.1을 통해 결정한다. 변환 모듈을 결정 한 후 Informap의 비공간 화일은 SDTS의 속성 모듈로 변환되며, Informap의 공간 화일은 SDTS에서 공간 데이터 모듈로 변환된다. 공간 데이터의 변환 과정에서 변환된 모듈들에 대한 객체 수, 통계적 자료들을 계산하여 SDTS의 글로벌 모듈을 생성한다. 그리고, 모듈별 물리적 화일명들로 구성된 카탈로그 모듈을 생성하며, 모듈별 공간 및 비공간 레코드 개수는 변환 통계 모듈에 저장된다.

3.3 게이트웨이 함수

공간 데이터는 대부분 대용량이므로 변환시에 모든 데이터를 메인 메모리에 적재한 후 변환하는 것은 자원 활용 측면에서 매우 좋지 못한 방법이다. 이러한 단점을 해결하기 위하여 본 논문에서는 게이트웨이 함수를 사용하여 메인 메모리 자원을 최소한만 사용하면서 모든 데이터를 변환할 수 있도록 하였다. 게이트웨이 함수는 다음과 같이 모두 7개의 함수로 구성된다.

- `cm_Initialize()` : SDTS에서 Informap으로 혹은 Informap에서 SDTS로 변환을 초기화하는 함수를 호출한다.
- `cm_Close()` : 변환을 종료하는 함수를 호출한다.
- `cm_InsertClass()` : 새로운 클래스를 삽입하는 함수를 호출한다.
- `cm_EndClass()` : 현재의 클래스에 대한 작업을 종료하는 함수를 호출한다.

- `cm_SetAttr()` : 주어진 클래스에 애트리뷰트를 지정하는 함수를 호출한다.
- `cm_InsertAttr()` : 비공간 데이터를 삽입하는 함수를 호출한다.
- `cm_InsertSpatial()` : 공간 데이터를 삽입하는 함수를 호출한다.

본 논문에서 구현한 데이터 변환 시스템은 이러한 게이트웨이 함수를 사용하므로 해당 Front-end나 Back-end를 교체하여 다른 용도의 시스템으로 손쉽게 전환할 수 있다. 즉, 게이트웨이 함수의 규약에 맞도록 다른 시스템의 저장 구조를 전달하면 다른 시스템의 데이터도 모두 SDTS 데이터로 저장할 수 있고, 역변환도 가능하다. 예를 들어, DXF(Data eXchange File)를 읽어 앞서 설명한 7개의 함수를 호출하는 Front-end를 개발하면 DXF를 SDTS로 변환하는 시스템이 되고, DXF에 맞도록 게이트웨이 함수, 즉 Back-end를 개발하면 SDTS를 DXF로 변환하는 시스템이 된다.

3.4 개발 환경 및 도구

데이터 변환 시스템을 위한 운영체제로는 Solaris 2.4 (SunOS 5.4 generic sparc)를 사용하였고, HP-UX B.10.01에서의 검증도 수행하였다. 사용 언어는 C를 사용하였으며 컴파일러로는 GNU C 버전 2.7.2를 사용하였다. 또한, 디버깅을 위한 도구로는 gdb 버전 4.14와 버전 4.16을 사용하였다. SDTS 데이터 처리를 위해서는 FIPS 123 라이브러리(UNIX 용)¹⁰⁾를 사용하였다.

4. SDTS로부터 Informap 변환

SDTS에서 Informap으로의 변환 시스템은 모듈화 프로그래밍 방법과 게이트웨이 함수를 사용하는 방법을 채택하였으며, SDTS 포맷의 데이터들을 분석하여 Informap의 대응되는 Kind의 공간 및 비공간 화일로 변환하도록 설계 및 구현되었다. SDTS 규약중에서 일반적인 Part 1, 2, 3를 충실하게 준수하면서, 구체적인

인 매핑 관계에서는 Part 4의 규약을 중심으로 따랐다.

4.1 전체적인 변환

SDTS에서 Informap의 변환에서 SDTS의 공간 모듈들은 Informap의 공간 화일로 변환되며, SDTS의 속성 모듈은 Informap의 비공간 화일로 변환된다. SDTS에서의 공간 객체와 속성 데이터의 관계 정보가 Informap으로 그대로 넘겨져 Informap의 공간 화일과 비공간 화일의 관계 정보를 구성한다. SDTS의 모듈들은 Informap에서의 Kind 1 ~ 5 화일들로 변환 가능하며, 그림 4.1에서 보는 것처럼 다양한 모듈들의 변환을 제공한다.

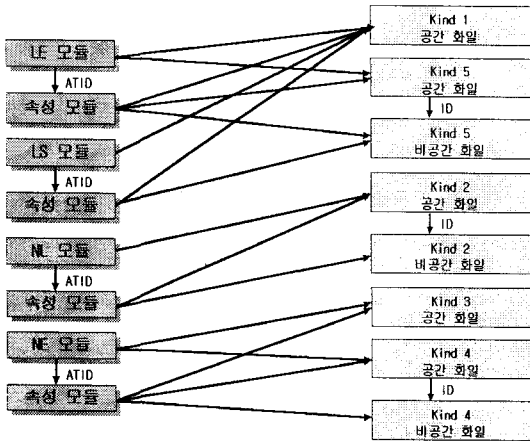


그림 4.1 SDTS로부터 Informap으로의 변환

4.2 라인 모듈 변환

SDTS에서 라인의 정보는 위상 정보를 가진 경우는 주로 LE(complete chain)으로 저장되고, 위상 정보가 없는 라인의 경우는 LS(Line String)로 저장된다. 이 두 모듈은 모두 속성 모듈과의 연결 정보를 ATID 필드의 MODN, RCID 하위 필드에 저장하고 있다. 그러므로, SDTS로부터의 Informap으로의 변환 과정에서 LE와 LS 모듈은 그림 4.2와 같이 Kind 5의 공간 화일로 저장된다.

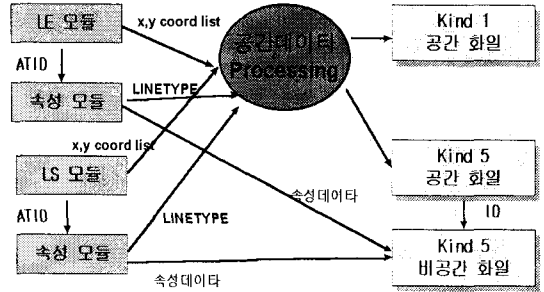


그림 4.2 라인 모듈 변환

LE 모듈 변환시에 위상 정보는 Informap에서 사용할 수 없으므로 무시된다. 그리고, LE와 LS모듈의 ATID 필드를 통해 관계를 갖는 속성 모듈은 생성된 공간 화일과 대응하는 비공간 화일로 저장된다. 만약, 이때 LE와 LS 모듈이 Informap의 Kind 1의 공간 화일(비공간 화일을 갖지 않음)로 자동적으로 변환된다. 또한, 속성 모듈이 포함하는 주석 형식을 공간 화일에 저장한다.

4.3 심볼 모듈 변환

SDTS에서 심볼의 정보는 NE(Entity Point) 모듈로 저장된다. 이 모듈은 속성 모듈과의 연결 정보를 ATID 필드의 MODN, RCID 하위 필드에 저장하고 있다. 그러므로, SDTS의 NE 모듈은 그림 4.3과 같이 Informap의 Kind 4의 공간 화일로 저장된다.

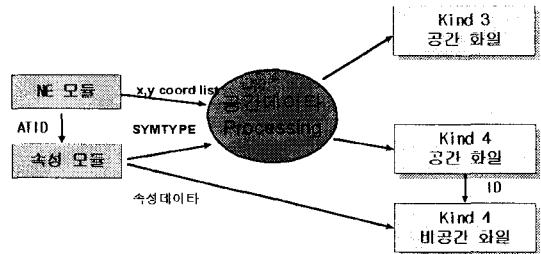


그림 4.3 심볼 모듈 변환

그리고, NE 모듈의 ATID 필드를 통해 관계를 갖는 속성 모듈은 생성된 공간 화일과 대응하는 비공간 화일로 저장된다. 만약, 이때 NE 모듈이 Informap의 Kind 3으로 변환된 모듈이라면 역변환 과정에서 Kind 4가 아닌 Kind 3의 공간 화일(비공간 화일을 갖지

않음)로 자동적으로 변환된다. 또한, 속성 모듈이 포함하는 심볼 형식을 공간 화일에 저장한다.

4.4 레이블 모듈 변환

SDTS에서 주석의 정보는 NL (Label Point) 모듈로 저장된다. 이 모듈은 속성 모듈과 연결 정보를 ATID 필드의 MODN, RCID 하위필드에 저장하고 있다. 그러므로, SDTS의 NL 모듈은 그림 4.4와 같이 Informap의 Kind 2의 공간 화일로 저장된다.

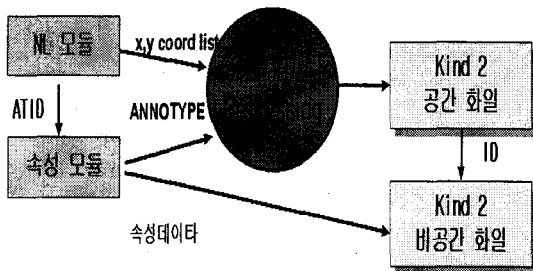


그림 4.4 레이블 모듈 변환

그리고, NL 모듈의 ATID 필드를 통해 관계를 갖는 속성 모듈은 생성된 공간 화일과 대응하는 비공간 화일로 저장된다. 만약, 이때 NL 모듈이 Informap의 Kind 2로 변환된 모듈이라면 역변환 과정에서 속성 모듈이 포함하는 주석 형식을 공간 화일에 저장한다.

5. Informap으로부터 SDTS 변환

Informap에서 SDTS로의 변환도 모듈화 프로그래밍 방법과 게이트웨이 함수를 사용하며, Informap의 SDT 화일을 분석하여 SDTS의 대응되는 모듈로 변환하도록 설계 및 구현되었다. 이때 Informap에서는 위상 정보를 제공하지 않으므로 SDTS로의 변환시에 위상 정보를 생성하지 않는다.

5.1 전체적인 변환

Informap에서 SDTS로의 변환에서 Informap의 공간 화일은 SDTS의 공간 모듈로 변환되며, Informap의 비공간 화일은 SDTS의 속성 모듈로 변환된다. 그

리고, Informap에서의 공간 객체와 속성 데이터의 관계 정보(NGP: Non-Graphic Pointer)가 SDTS 공간 모듈의 ATID 필드를 통해 그대로 넘겨져 SDTS의 공간 모듈과 비공간 모듈의 관계 정보를 구성한다. Informap의 화일들은 그림 5.1과 같이 SDTS에서의 LS, NL, NE 모듈로 변환가능하다.

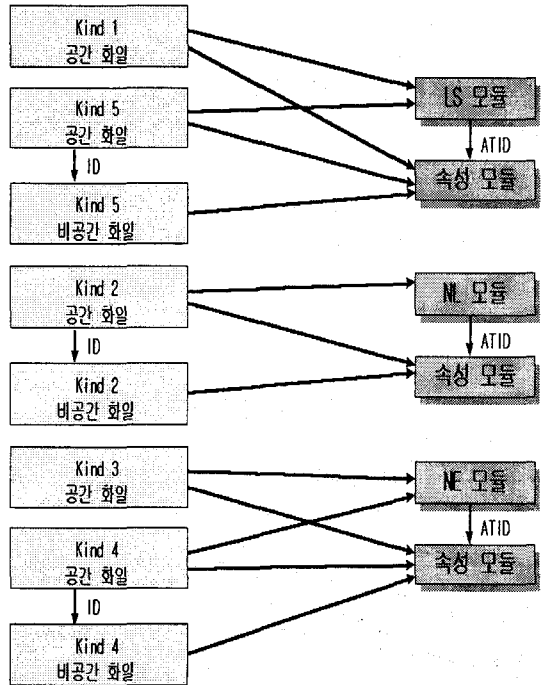


그림 5.1 Informap에서 SDTS로의 변환

5.2 Kind 1 변환

Informap의 속성 정보를 갖지 않고 라인만을 저장하는 Kind 1은 그림 5.2와 같이 SDTS의 LS 모듈로 변환된다.

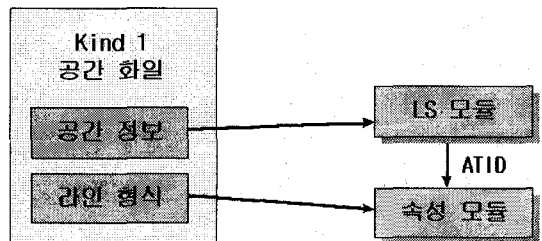


그림 5.2 Kind 1 변환

Kind 1은 라인과 라인의 형식을 저장하는 공간 화일

만으로 구성된다. 그러므로, 라인, 즉 공간 정보는 LS 모듈로 변환되고, 라인 형식은 속성 모듈로 변환된다.

공간 정보는 NE 모듈로 변환되고, 심볼 형식은 속성 모듈로 변환된다.

5.3 Kind 2 변환

Informat의 주석을 저장하는 Kind 2는 그림 5.3과 같이 SDTS의 NL 모듈로 변환된다.

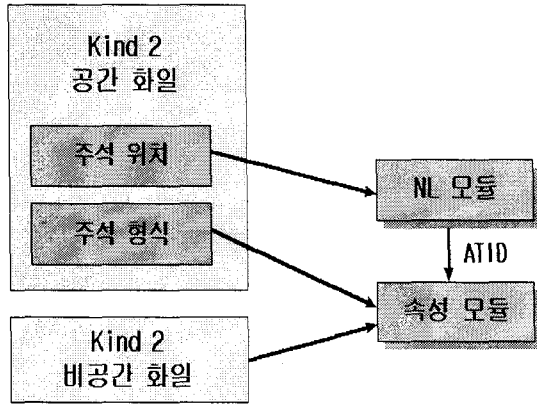


그림 5.3 Kind 2 변환

Kind 2는 주석 위치와 주석 형식을 저장하는 공간 화일과 출력될 주석을 저장하는 비공간 화일로 구성된다. 그러므로, 주석 위치, 즉 공간 정보는 NL 모듈로 변환되고, 주석 형식과 주석 데이터는 속성 모듈로 변환된다.

5.4 Kind 3 변환

Informat의 속성 정보를 갖지 않고 심볼만을 저장하는 Kind 3은 그림 5.4와 같이 SDTS의 NE 모듈로 변환된다.

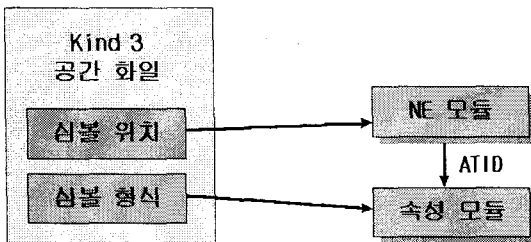


그림 5.4 Kind 3 변환

Kind 3은 심볼의 위치와 심볼의 형식을 저장하는 공간 화일만으로 구성된다. 그러므로, 심볼의 위치, 즉

5.5 Kind 4 변환

Informat의 속성을 갖는 심볼을 저장하는 Kind 4는 그림 5.5와 같이 SDTS의 NE 모듈로 변환된다.

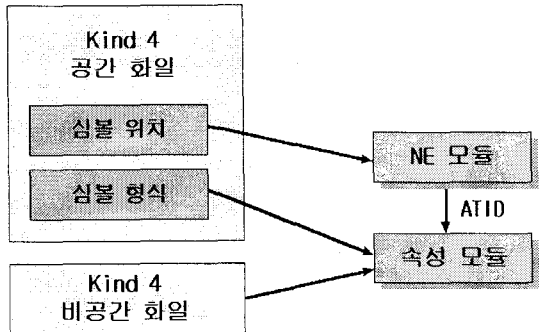


그림 5.5 Kind 4 변환

Kind 4는 심볼 위치와 심볼 형식을 저장하는 공간 화일과 속성 데이터를 저장하는 비공간 화일로 구성된다. 그러므로, 심볼 위치, 즉 공간 정보는 NE 모듈로 변환되고, 심볼 형식과 속성 데이터는 속성 모듈로 변환된다.

5.6 Kind 5 변환

Informat의 속성을 갖는 라인을 저장하는 Kind 5는 그림 5.6과 같이 SDTS의 LS 모듈로 변환된다.

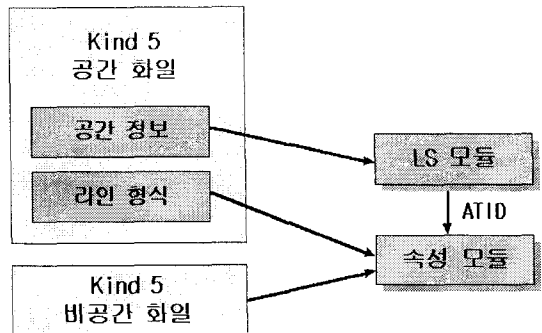


그림 5.6 Kind 5 변환

Kind 5는 라인과 라인 형식을 저장하는 공간 화일과 속성 데이터를 저장하는 비공간 화일로 구성된다. 그

러므로, 공간 정보는 LS 모듈로 변환되고, 라인 형식과 속성 데이터는 속성 모듈로 변환된다.

6. 결론

본 논문에서는 국가 공통데이터교환 포맷으로 채택된 SDTS를 분석하고, 기존의 매핑 시스템인 Informap의 데이터를 SDTS 데이터 구조와 상호 변환할 수 있는 데이터 변환 시스템을 설계 및 구현하였다. 본 논문에서 제시하는 데이터 변환 시스템은 여러 처리 모듈, 로그 화일 처리 모듈, 화일 처리 모듈, Informap 데이터 처리 모듈, SDTS 데이터 처리 모듈, 변환 처리 모듈과 FIPS123 라이브러리로 구성된다. 이처럼 모듈별 프로그래밍 방법을 사용하여 특정 환경의 변화에 쉽게 대응할 수 있도록 하였다.

본 논문에서 구현한 데이터 변환 시스템은 7개의 게이트웨이 함수를 사용하여 모든 변환이 이루어지도록 설계 및 구현되어 있기 때문에 해당 Front-end나 Back-end를 교체하면 다른 용도를 위한 변환 시스템으로 쉽게 전환할 수 있다. 또한, 공간 데이터가 대용량이라는 점을 감안하여 한번에 한 객체씩만을 메인 메모리에 적재하고 게이트웨이 함수를 통해 처리하는 방법을 채택하였다. 그러므로, 운영체제나 메인 메모리의 용량에 독립적으로 대용량의 모든 공간 데이터를 처리할 수 있다는 장점이 있다.

이후 연구과제로는 데이터의 교환시에 발생하는 손실을 더욱 줄이는 연구와 Informap에서 제공하지 않는 위상 데이터를 변환시에 자동으로 생성하여 SDTS 데이터로 변환하는 연구가 필요하겠다.

참 고 문 헌

1. Altheide, P., "An Implementation Strategy for SDTS Encoding," American Congress on Surveying and Mapping, Vol.19, No.5, Dec. 1992.
2. British Standard Institution, Electronic Transfer of Geographic Information (NTF), U.K., 1992.
3. Davis, B.A., George, J.R., and Marx, R.W.,

"TIGER/SDTS: Standardizing an Innovation," American Congress on Surveying and Mapping, Vol.19, No.5, Dec. 1992.

4. Digital Geographic Information Working Group, DIGEST : The Digital Geographic Information Exchange Standard, National Defence Headquarters, Ottawa, Canada, 1994.

5. Federal Geographic Data Committee, Cadastral Standards for the National Spatial Data Infrastructure, Reston, VA, 1994.

6. Fegeas, R.G., Cascio, J.L., and Lazar, R.A., "An Overview of FIPS 173, The Spatial Data Transfer Standard," American Congress on Surveying and Mapping, Vol.19, No.5, Dec. 1992.

7. Gunther, G., and Buchemann, A., "Research Issues in Spatial Databases," ACM SIGMOD Record, Vol.19, No.4, Dec. 1988, pp.61-68.

8. Herring, J., What is an Open GIS Application?, ANSI Working Paper, X3L 1.2/94-044, June 1994.

9. ISO/IEC JTC1/SC21 WG3, SQL Multimedia and Application Packages (SQL/MM) - Part3 : Spatial, Feb. 1995.

10. Lazar, R.A., "SDTS Support Software: The FIPS 123 Function Library," Cartography and Geographic Information Systems, Vol.19, No.5, 1992.

11. Medeiros, C.B., and Pires, F., "Databases for GIS," Proc. of ACM SIGMOD Int. Conf., Mar. 1994, pp. 107-115.

12. National Institute of Standards and Technology, The Spatial Data Transfer Standard, Federal Information Processing Standard Publication 173, U.S. Department of Commerce, 1992.

13. Object Management Group, The Common Object Request Broker: Architecture and Specification, Rev. 1.2, Framingham, Mass., Nov. 1993.

14. Open GIS Consortium, The Open Geodata Interoperability Specification Version 1, Draft 1, April 1994.

15. Tom, H., "The Geographic Information Systems

- Standards Infrastructure," StandardView, Vol.2, No.3, pp. 133-142, Sep. 1994.
16. Wiederhold, "The Role of Government in Standards," StandardView, Vol.1, No.2, pp. 11-16, Dec. 1993.
17. 오병우, 한기준, "지리 정보 시스템을 위한 표준화," 정보과학회지, 제 13권 10호, pp. 46-55, 1995.
18. 캐드랜드, "GIS 공통데이터 교환포맷 표준초안 참고자료," Sep. 1995.
19. 한국통신 선로기술연구소, "GIS DB용 데이터 포맷 변환 도구 개발," 최종보고서, Dec. 1995.
20. 한기준, "선로 요원 작업 경로 최적화를 위한 GIS에 관한 연구," 한국통신 연구개발원 '95장기기초 연구과제 최종보고서, Dec. 1995.