

웹기반 공간 OODBMS를 위한 형상 관리자의 개발†

박인하, 신명철, 이강준, 한기준
건국대학교 컴퓨터공학과

Development of a Configuration Manager for a Web-based Spatial OODBMS

In-Ha Park, Myoung-Chul Shin, Kang-Jun Lee and Ki-Joon Han
Department of Computer Science & Engineering, Kon-Kuk University

요 약

최근 웹의 활성화로 인하여 인터넷이 대중화되면서 지리 정보 시스템 분야에서도 웹 기반의 공간 데이터의 검색 및 관리의 필요성이 대두되었다. 또한, 지리 정보 시스템에서 다루는 공간 데이터는 가변적이고 대용량이기 때문에 이러한 데이터를 효율적으로 처리하기 위해서는 공간 OODBMS의 사용이 적합하다. 이러한 웹 기반 공간 OODBMS를 사용하는 실제 응용프로그램에서 다중 사용자 환경에서의 협동 작업이 효율적으로 수행되기 위해서는 개별적인 객체 중심의 버전 관리에 그치지 않고 일정 영역내의 연관된 버전화된 객체들의 집합인 복합 객체를 효율적으로 관리하는 것이 절실히 필요하다. 이에 본 논문에서는 다중 사용자 환경에서 연관된 버전화된 객체들의 집합인 복합 객체를 효율적으로 관리하는 웹기반 공간 OODBMS를 위한 형상 관리자를 설계 및 구현한다. 이를 위해 형상 관리 기능을 지원하는 기존의 시스템인 Open OODB와 RCS, 그리고 SCM에서의 형상 관리 모델에 대해서 알아보고, 형상 관리자의 관리 모델과 자료 구조를 설명한 후 형상 관리자 설계시의 고려사항에 대해서 언급한다. 마지막으로, 형상 관리자의 구성 모듈과 형상 관리자 API에 대하여 기술한다.

1. 서론

최근 지리 정보 시스템의 필요성과 중요성이 부각되고 있는 가운데 이에 관한 연구가 활발히 진행되고 있다. 더우기 국내에서는 국가적인 차원에서 중요한 연구 개발 과제로 많은 연구가 이루어지고 있다 [정98]. 예전까지의 지리 정보 시스템은 숙련된 전문가에 의해서만 제한적으로 사용되어 왔으나, 지리 정보 시스템과 웹을 연동함으로써 전문가 뿐만이 아닌 일반 사용자들도 필요한 지형 공간 데이터를 쉽게 처리할 수 있는 환경이 구축되었다 [Ple96, Tho95].

지리 정보 시스템의 개발을 위해서는 벡터, 래스터 이미지 등과 같은 기본적인 대용량인 공간 데이터를 효율적으로 처리할 수 있는 공간 OODBMS가 필요하다. 공간 OODBMS에서 버전 관리는 여러 작업자에 의해 데이터의 갱신이 빈번하게 일어나는 공간 및 비공간 데이터를 효율적으로 처리하기 위해 반드시 필요한 기능이다 [김97, Ahn91, Bur98, Kla86]. 그러나, 실제 다중 사용자 환경의 응용 프로그램에서는 협동 작업을 위해 개별적인 객체 중심의 개별적인 버전 관리보다는 일정 영역내의 버전화된 객체들의 집합인 복합 객체를 효율적으로 처리하는 것이 절실히 필요하다 [Con98, Sci94].

일정 영역내 연관된 버전화된 객체들의 집합인 복합 객체를 관리하는 것을 형상 관리라 하며 이러한 형상 관리 기능은 객체들의 버전 진화 과정에 상관없이 항상 객체들간에 일관성있는 관계를 유지하도록 한다. 또한, 형상 관리를 통하여 다중 사용자 환경에서 협동 작업을 가능하게 하신 협동 작업으로 인한 생산성 향상을 가져올 수 있다 [Ful93]. 본 논문에서는 일정 영역내의 연관된 버전화된 객체들의 집합인 복합 객체를 효율적으로 관리할 수 있는 웹기반 공간 OODBMS를 위한 형상 관리자를 설계 및 구현한다.

본 논문의 구성은 다음과 같다. 제 1 장 서론에 이어 제 2 장에서는 관련 연구로서 기존의 형상 관리 기능을 지원하는 시스템인 Open OODB에서의 형상 추상화 머신과 RCS(Revision Control System)에서의 형상 관리 기법을 설명하고, SCM(Software Configuration Management)에서의 형상 관리 모델에 관하여 소개한다. 그리고, 제 3 장에서는 형상 관리 모델과 자료 구조를 설명하고, 설계시의 고려사항

을 언급한 다음, 제 4 장에서는 형상 관리자의 모듈 구성과 형상 관리자 API에 관하여 기술한다. 마지막으로 제 5 장에서는 결론 및 향후 연구 과제에 대하여 언급한다.

2. 관련 연구

본 장에서는 형상 관리 기능을 지원하는 기존의 시스템인 Open OODB와 RCS에 관하여 알아보고, SCM에서의 형상 관리 모델에 관하여 설명한다.

2.1 Open OODB

Open OODB에서는 형상 관리를 위하여 형상 추상화 머신(CAM Configuration Abstract Machine) 명세를 제시하고 있다. 형상 추상화 머신에서는 클라이언트가 복합 객체, 추상화 형상, 버전화된 객체의 능력 형상들을 생성하거나 관리할 수 있고, 특징 객체 인스턴스의 형상들을 구체화할 수 있다 [Tho91]. 이를 위해 형상 추상화 머신은 형상 노드들로 구성된 루트로부터 가지는 비순환 방향성 그래프인 형상 그래프를 관리한다. 이 때, 각각의 형상 노드들은 형상을 관리하기 위한 필요한 형상에 관한 정보와 객체에 대한 포인터를 갖는데, 이 객체에 대한 포인터는 버전 추상화 머신이 관리하는 버전 그래프에 대한 포인터일 수도 있다. 본 논문에서는 Open OODB의 형상 추상화 머신 명세에서 제시한 것보다 길어 형상을 관리하기 위해 형상 정보를 기록하고 있는 형상 노드들로 구성된 형상 그래프를 사용하며, 버전화된 객체를 위해 객체의 버전 진화 그래프에 대한 포인터를 유지하는 방법 [김97, 박99]을 사용한다.

2.2 RCS

RCS(Revision Control System)는 소스인 버전과 버전들로 이루어진 형상을 관리하기 위한 UNIX 명령어의 집합이다 [Tic85]. RCS에서의 형상은 다른 요소 그룹들로부터 어떤 특정 길이에 의해 선택된 요소들로 이루어진 집합으로 정의된다. RCS에서는 각 요소들이 버전 진화 과정을 통하여 변경되는데, 요소들이 변경될 때에는 변경할 요소본

† 본 연구는 과학기술부의 핵심 S/W 기술개발사업에서 지원받았음.

분리하여 변경 작업을 수행하고 변경한 내용을 별도로 저장하여 변경 과정을 확인할 수 있도록 하고 있다 그리고, MAKE 파일을 사용하여 변경된 요소들간의 일관성있는 관계가 자동적으로 유지되도록 한다. 본 논문에서는 RCS와 같이 요소를 변경할 때 변경할 요소를 별도로 분리하여 변경 작업을 수행하는 방법을 사용하며, 또한 변경된 요소들 별도로 저장하여 요소의 변경 과정을 확인할 수 있도록 한다.

2.3 SCM

SCM(Software Configuration Management)은 크고 복잡한 소프트웨어 진화 과정을 관리하여 주는 방법으로서 형상 관리를 지원하기 위해 관리 지원 측면과 개발 지원 측면을 제공한다 [Con98]. 관리 지원 측면에서는 소프트웨어 구성 요소들의 검증과 구성 요소들의 버전, 변경 조절, 상태 평가, 그리고 검사와 검토 등에 관한 제반 사항을 관리한다. 개발 지원 측면에서는 개발자가 소프트웨어의 내용을 변경할 수 있도록 하면서 버전화한 소프트웨어 구성 요소들의 변경 사항을 기록하고 상호 의존적인 요소들간의 일관성을 유지·관리하며, 변경된 내용을 바탕으로 컴파일하여 실행 코드를 생성한 후 각 요소들의 특성을 갖는 새로운 형상을 제작하는 방식을 사용한다. 본 논문에서는 SCM에서와 같이 구성 요소들의 버전에 관한 사항을 이용하여 형상의 변화 과정을 확인할 수 있도록 한다.

3. 형상 관리자의 설계

본 장에서는 형상 관리자의 형상 관리 모델과 형상 관리 자료 구조에 대해서 설명하고, 형상 관리자를 설계할 때 고려되었던 사항에 관해서 언급한다.

3.1 형상 관리 모델

본 논문에서 형상은 복합 객체를 처리하기 위한 일관된 버전화된 객체들의 집합이며, 형상 관리를 위해 형상 노드들로 구성되는 루트 노드를 기저는 비순환 방향성 그래프 형태의 형상 그래프를 사용한다. 본 논문에서 제시한 형상 관리 모델에서는 사용자가 임의로 형상 그래프를 형성할 수 있다. 이를 위해 형상 관리자는 효율적인 형상 관리를 위해 사용자에게 형상 그래프를 자유로이 형성할 수 있는 인터페이스를 제공한다. 그림 1은 형상 관리 모델과 자료 구조의 사용 예를 보여준다.

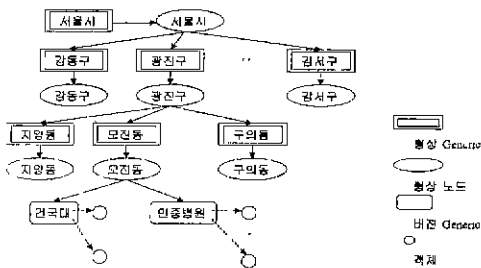


그림 1. 형상 관리 모델 및 자료 구조 사용 예

그림 1에서 사용자는 서술시라는 형상을 처음으로 생성한다. 이 때, 서술시 형상 Generic 객체와 서술시 형상 노드 객체가 생성된다. 서술시 형상을 생성한 후 강동구, 광진구, 강서구 형상 등을 생성하여 서술시의 하위 형상으로 등록한다. 이와 같은 방법으로 지양동, 모진동, 구의동 형상 등을 생성하여 광진구의 하위 형상으로 등록한다. 건국대와 민중병원은 버전 관리 기능을 제공하기 위하여 작성한 공간 객체의 버전 진화 그래프에 대한 전반적인 관리를 하는 버전 Generic 객체 [진97]이다. 이러한 건국대와 민중병원의 버전 Generic 객체를 모진동에 등록함으로써 건국대와 민중병원에 관한 다양한 버전들이 모진동에서 관리될 수 있게 된다.

3.2 형상 관리 자료 구조

본 논문에서는 형상을 관리하기 위해 형상 Generic 객체와 형상 노드 객체를 사용한다. 형상 Generic 객체는 형상 노드를 관리하는 객체로서, 형상 노드의 변경 사항을 관리하기 위해 형상 노드에 대한 포인터를 갖는 config_nodes, 상위의 형상 노드에 대한 포인터를 갖는 parent_node, 그리고 다중 사용자 환경에서 협동 작업을 하기 위해 현재 자신을 포함하여 하위 형상에서 몇 개의 형상 노드들이 check_out 되었는지를 저장하는 check_count를 포함한다. 표 1은 형상 Generic 클래스에 대한 상세한 명세를 보여준다.

형상 노드 객체는 각각의 형상에 대한 정보를 유지·관리하는 객체로서 하위 형상 Generic들에 대한 포인터를 갖는 child_nodes, 현재

형상 노드가 단일 형상 노드일 때 현재 형상 노드에서 관리하는 버전 Generic 객체들에 대한 포인터를 갖는 member_objects, 현재 형상 노드의 형상 Generic에 대한 포인터를 갖는 generic_node, 그리고 현재의 형상이 어떤 형상인가에 대한 설명을 갖는 Desc로 구성된다. 표 2는 형상 노드 클래스의 상세한 명세를 보여준다.

Name	Type	Description
config_nodes	set of references	Set of References to Configuration Node
parent_node	reference	Reference to Parent Configuration Node
check_count	long	Count of Check_out Node

표 1 형상 Generic 클래스

Name	Type	Description
child_nodes	set of refernces	Set of References to Configuration Generic
member_objects	set of references	Set of References to Version Generic Objects
generic_node	reference	Reference to its Configuration Generic
Desc	string	Description

표 2 형상 노드 클래스

3.3 설계시의 고려사항

형상 관리자를 설계할 때 특별히 고려한 형상 관리자의 인터페이스를 제공하는 방법과 버전 관리 기능 및 협동 작업 관리 기능의 수행 방법에 대하여 설명하면 다음과 같다. 우선 형상 관리자의 인터페이스에 대해서는 형상 관리에 필요한 작업들을 수행할 수 있는 인터페이스가 정의된 별도의 형상 관리 클래스를 정의하여 사용자가 형상 관리 기능을 사용하고자 할 때 형상 관리 클래스 객체를 생성하여 형상 관리 기능을 수행하도록 한다. 형상 관리자에 등록되는 공간 객체가 버전화된 객체인 경우에는 버전 관리 [진97, Kim98]의 외부 인터페이스를 통하여 공간 객체에 대한 버전 관리 기능을 사용한다. 그리하여, 사용자에게 형상 관리 기능뿐만 아니라 버전 관리 기능을 사용할 수 있도록 한다. 다중 사용자 환경에서 협동 작업을 지원하기 위해 형상 관리자는 형상 Generic 객체에 check_count를 두어 형상 노드의 체크아웃 여부를 관리할 수 있도록 한다. 형상 노드가 체크아웃될 때 현재의 형상 노드의 모든 상위 형상 Generic 객체의 check_count를 증가시키고 다시 체크인될 때는 check_count를 감소시킨다. check_count가 0보다 클 때는 현재의 형상 노드의 하위 형상 노드가 체크아웃되어 다른 사용자에 의해 변경되고 있다는 것을 의미하기 때문에 현재의 형상 노드를 체크아웃할 수 없게 된다.

4. 형상 관리자의 구현

본 논문에서 형상 관리자의 개발을 위해 사용한 시스템 환경은 다음과 같다. 운영체제는 Solaris Release 2.5를 사용하였고, 컴파일러는 GNU gcc 2.72 버전을 사용하였다. 또한, 프로그램 디버깅을 위해서는 GNU gdb 4.3을 이용하였다. 또한, ODMG 93의 [Cat97] C++ OML을 확장하여 형상 관리자 API를 제공함으로써 응용 프로그램에서 형상 관리자를 쉽게 사용할 수 있도록 하였다. 본 장에서는 3장에서 언급한 설계 사항을 바탕으로 하여 형상 관리자의 모듈 구성과 형상 관리자 API에 대하여 기술한다.

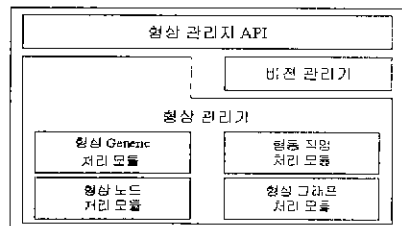


그림 2 형상 관리자 모듈 구성도

4.1 형상 관리자 모듈 구성

형상 관리자는 형상 Generic 처리 모듈, 형상 노드 처리 모듈, 형상 그래프 처리 모듈, 그리고 협동 작업 처리 모듈로 구성되어 있으며

1 구성도는 그림 2과 같다.

형성 Generic 처리 모듈은 형성의 전반적인 사항을 처리하는 모듈로서 새로운 형성 그래프를 생성하는 기능, 전체 형상을 삭제하는 기능, 그리고 형성 그래프의 한 가지(branch)를 삭제하는 기능을 제공한다

형성 노드 처리 모듈은 하위 형성 노드를 생성하는 기능, 형성 노드를 삭제하는 기능, 생성된 형성 노드를 상위 형성 노드에 등록하는 기능을 제공한다 또한, 형성 노드의 설명을 변경하거나 형성 노드의 설명을 반환하는 기능, 형상에 의해 관리되는 버전 Generic 객체를 등록하는 기능, 형성에서 관리되고 있는 버전 Generic 객체를 반환하는 기능도 제공한다.

형성 그래프 처리 모듈은 형성 Generic 처리 모듈과 형성 노드 처리 모듈에서 생성한 형성 그래프를 검색하여 주로 형성 그래프의 위상 정보를 반환하는 모듈이다 또한, 원하는 형성 노드를 반환하여 주는 기능, 현재 형성의 상위 형성 노드를 반환하는 기능, 하위 형성 노드를 반환하는 기능, 그리고 전체 형성 그래프를 반환하는 기능도 제공한다

마지막으로, 협동 작업 처리 모듈은 웹 기반의 다중 사용자 환경에서 사용자가 공간 객체들을 효율적으로 관리할 수 있도록 지원하며 데이터가 중복 변경되지 않도록 관리하는 모듈이다 형성 관리자에서는 협동 작업을 지원하기 위해 체크아웃/체크인 모델 [Kim95]을 사용하여 현재 형성에서 변경하고자 하는 형성 일부분을 체크아웃하는 기능과 변경된 형성의 일부분을 원래의 전체 형상에 반영되도록 체크인하는 기능을 제공한다

4.2 형성 관리자 API

본 절에서는 외부 모듈에서 형성 관리 기능을 사용할 수 있는 형성 관리자의 여러 함수들에 대해서 설명한다. 각각의 함수들은 형성 관리 API 클래스의 멤버 함수로 구현된다.

4.2.1 형성 Generic 처리 관련 함수

형성 생성 함수는 형성 관리를 위하여 새로운 형성 그래프를 맨 처음 생성할 때 사용되는 함수로서 형성 Generic 객체와 형성 노드 객체를 생성하고 입력받은 설명을 형성 노드에 기록한다 형성 삭제 함수는 현재의 형성 그래프를 삭제하는 함수로서 입력받은 형성 노드가 참여하는 형성 그래프를 삭제한다 형성 트리 삭제 함수는 입력받은 형성 노드의 하위 형성들을 모두 삭제한다 즉, 형성 그래프의 한 가지(branch)를 삭제한다

4.2.2 형성 노드 처리 관련 함수

형성 노드 생성 함수는 형성 그래프에 등록할 새로운 형상을 생성하는 함수로서 형성 노드를 생성하여 입력받은 형성 설명을 기록하고, 형성 노드의 변화 과정을 관리하기 위한 형성 Generic 객체도 생성한다 형성 노드 등록 함수는 생성한 형성 노드를 상위 형성 노드의 하위 형성으로 등록하는 함수로서 상위 형성 노드를 인자로 받아 하위 형성 노드 관리에 현재 생성한 형성 노드를 등록한다 형성 노드 삭제 함수는 현재의 형성 노드를 삭제하는 함수로서 하위 형성 노드가 존재하지 않을 때는 등록되어 있는 버전 Generic 객체들을 삭제하고 형성 노드도 삭제한다 형성 설명 변경 함수는 형성 노드 생성시에 설정해 놓은 형성 노드의 설명을 바꾸는 함수로서 변경하려는 형성 노드의 설명을 입력받아 형성 노드의 설명을 변경한다 형성 설명 반환 함수는 형성 노드의 설명을 반환하는 함수로서 형성 노드를 입력받아 형성 노드의 설명을 반환한다 버전 객체 등록 함수는 단일 형성 노드에 대한 Generic 객체를 등록하는 함수로서 단일 형성 노드인지를 체크하여 단일 형성 노드일 때 입력받은 버전 Generic 객체를 형성 노드에 등록한다 버전 객체 반환 함수는 형성 노드에 등록되어 있는 버전 Generic 객체들을 반환하는 함수로서 입력받은 형성 노드의 하위에 있는 모든 형성 노드에 등록되어 있는 버전 Generic 객체들을 리스트 형식으로 반환한다

4.2.3 형성 그래프 처리 관련 함수

형성 그래프 반환 함수는 형성 그래프에 속한 모든 형성 노드들의 설명을 반환한다 형성 노드 반환 함수는 형성 Generic 객체 또는 형성 설명을 이용하여 형성 그래프에 속해 있는 형성 노드를 반환한다 형성 Generic 객체를 이용하여 형성 설명을 반환하는 형상 Generic 객체에 등록되어 있는 형상 노드를 반환하고, 형성 설명을 이용할 때에는 입력받은 설명과 일치하는 형성 그래프의 형성 노드를 반환한다. 상위 형성 노드 반환 함수는 현재 형성 노드의 형성 Generic 객체에 기록되어 있는 상위 형성 노드를 반환한다 하위 형성 노드 반환 함수는 현재 형성 노드에 등록되어 있는 모든 하위 형성 노드들을 리스트 형식으로 반환한다

4.2.4 협동 작업 처리 관련 함수

체크아웃 함수는 사용자가 형성 그래프에서 일정한 영역을 변경하고자 할 때 사용하는 함수로서 현재의 형성 노드가 이미 다른 사용

자에 의해 체크아웃되었는지를 검사한다 만약 이미 체크아웃되었나면 현재의 형성 노드 체크아웃 작업은 취소된다. 만약에 체크아웃되지 않았다면 현재의 형성 노드의 상위 형성 Generic 객체들의 check_count를 증가시키고, 현재 형성 노드를 포함한 하위 형성 노드를 새로운 저장공간으로 복사하여 체크아웃을 수행한 사용자만이 변경 작업을 수행할 수 있도록 한다. 체크인 함수는 이전에 체크아웃하여 변경 작업을 완료한 후에 변경된 형성 노드들의 내용을 다시 원래의 형상 그래프에 반영하는 함수이다 또한, 체크아웃한 최상위 형성 노드의 형성 Generic에 다른 형성 노드로 등록하여 변경된 내용과 변경되기 전의 내용이 모두 관리될 수 있도록 하고, 모든 상위 형성 Generic 객체의 check_count 값을 감소시킨다

5. 결론 및 향후 연구 과제

현재 지리 정보 시스템이 웹과 연동됨으로 해서 다중 사용자의 협력 작업이 점차 중요시되고 있는 시점에서 일정 영역내의 복합 객체에 대한 효율적인 관리를 위하여 형성 관리 기능은 매우 중요한 기능이다 본 논문에서는 웹기반 공간 OODBMS를 위한 형성 관리자의 개발에 필요한 형성 관리를 지원하는 기존의 시스템인 Open OODB와 RCS 및 SCM의 형성 관리 모델에 편하여 분석하였다 또한, 형성 관리자의 형성 관리 모델, 자료 구조를 설계하고 설계시 고려사항을 기술하였고 이러한 설계에 따라 형성 관리자를 구현하였다

본 논문에서 설계 및 구현한 웹기반 공간 OODBMS를 위한 형성 관리자는 연관된 버전화된 객체들의 집합인 복합 객체를 관리하는 형성 관리를 효율적으로 지원함으로써, 다중 사용자 환경에서 사용자간에서 서로 영향을 주지 않고 동시에 공간 객체의 갱신 작업을 수행할 수 있는 협동 작업을 가능하게 한다 그리하여, 지리 정보 시스템에서 작업의 효율성과 생산성을 높일 수 있다 향후 연구 과제로는 현재 구현된 형성 관리자가 웹에서 사용될 수 있도록 자바를 이용하여 클라이언트를 구현하는 것이다

참고 문헌

[Ahn91] Ahmed, R, and Navathe, S.B., "Version Management of Composite Objects in CAD Databases," Proc of ACM-SIGMOD Conf, 1991, pp 218-227

[Bur98] Burns, R.C., and Narang, I, "Version Management and Recoverability for Large Object Data," IW-MMDBMS, 1998, pp 12-19

[Cal97] Cattell, R.G.G., et al, *The Object Database Standard ODMG 2.0*, Morgan Kaufmann, 1997.

[Con98] Conrad, R, and Westfichtel, B, "Version Models for Software Configuration Management," ACM Computing Surveys, Vol.30, No.2, Jun 1998, pp 232-282

[Fu93] Fuller, D.A, Mujica, S.T, and Pino, J.A., "The Design of an Object-Oriented Collaborative Spreadsheet with Version Control and History Management," SAC, 1993, pp 416-423

[Kim95] Kim, W, *Modern Database System*, Addison-Wesley Publishing, 1995

[Kim98] Kim, H.J, Lee, K.J, Oh, B.W., and Han, K.J, "Development of a Version Manager for a Spatial Object Management System," Proc of Geoinformatics '98 Conference, China, Jun 1998, pp 63-72.

[Kla86] Klahold, P., Schlageter, G, and Wilkes, W, "A General Model for Version Management in Databases," Proc of VLDB Conf., 1986, pp 319-327

[Ple96] Plewe, B, "Mapping on the Web: A Primer on Creating Geographic Services," GIS World, Jan 1996, pp 56-62

[Sci94] Score, E, "Versioning and Configuration Management in an Object-Oriented Data Model," VLDB Journal, Vol.3, No.1, Jan. 1994, pp 77-106

[Tho91] Thompson, C., *Change Management Module*, DARPA Open Object-Oriented Database Preliminary Module Specification, Nov. 1991

[Tho95] Thoen, B, "Interactive Mapping and GIS Thrive on the Web," GIS World, Oct 1995, pp. 28-32

[Tic85] Tichy, W., "RCS-A System for Version Control," Software Practices and Experience, Vol.15, Jul 1985, pp 637-654

[김97] 김형진, 이경준, 오병우, 한기춘, "공간 객체 관리 시스템을 위한 버전 관리자의 개발," 한국정보과학회 학술발표논문집, 24년2호, 1997, pp. 237-240

[박99] 박인하, 윤재관, 오병우, 한기춘, "버전 관리 기능을 갖는 공간 객체 집합 관리자의 설계 및 구현," 한국정보과학회 학술발표논문집, 26년1호, 1999, pp 140-142.

[정98] 정진원, 장흥근, 김철우, 박주홍, 한기춘, 허진, "국기 지리 정보 시스템을 위한 공간 객체관리 시스템의 구조," 한국정보과학회 정보과학회지, 16년3호, 1998, pp 5-9