

중복 지리정보 객체 관리를 위한 분산 지리정보 시스템의 위치 서비스 모델링

정 창 원[†] · 이 원 중^{**} · 이 재 완^{***} · 주 수 중^{****}

요 약

인터넷 기술의 발전에 따라 분산 지리정보 시스템 환경이 웹 기반 서비스로 변화되고 있는 추세이다. 기존의 웹 지리정보 서비스의 지리정보는 다양한 지도 포맷을 지원하여 상호운용하지 못하도록 독립적으로 개발되어 왔다. 동일한 지리정보임에도 불구하고 이러한 정보는 서로 다른 목적에 따라 개별적인 지리정보 시스템에 의해 중복된다. 이는 중복된 지리정보 객체를 식별하는 문제와 중복된 지리정보 객체들 중에 클라이언트에게 가장 적합한 복사본을 선정하기 위한 지능적인 전략이 요구되고 있다. 그리고 중복 객체를 관리하기 위해 OMG와 GLOBE 그리고 GRID 컴퓨팅에서 관련 프레임워크를 제안하고 있다. 그러나 지리정보 객체에 대한 연구는 미흡하다.

따라서, 본 논문에서 이름 또는 속성으로 중복된 서비스 객체들의 효율적인 관리와 중복 서비스 객체들 중 가장 적합한 객체를 선정하기 위한 위치 서비스 모델을 제시하였다. 위치 서비스 모델은 세 개의 주요 서비스로 구성되어 있다. 첫 번째로 바인딩 서비스는 사용자가 객체의 이름과 속성으로 정의하여 서비스 오퍼 단위로 저장하고 클라이언트가 서비스 오퍼를 검색 할 수 있다. 두 번째는 위치 서비스로 컨택 레코드를 갖는 위치 정보를 관리한다. 그리고 컨택 주소를 갖는 독립적인 시스템상의 LSF에 의해 성능 정보를 획득한다. 세 번째는 지능형 선정 서비스로 바인딩/위치 서비스로부터 기본/성능 정보를 획득하고, 러프집합 기반의 지능형 모델에 의한 규칙에 의해 보다 빠르게 접근하고 성능이 우수한 특징을 제공한다. 본 논문에서 제시한 위치 서비스 모델에 대한 검증을 위해 그래픽 사용자 인터페이스를 통해 위치 서비스의 처리 과정을 보였다.

키워드 : 분산 지리정보 시스템, 지리공간 정보, 위치 서비스, 중복 객체

Location Service Modeling of Distributed GIS for Replication Geospatial Information Object Management

Jeong Chang Won[†] · Lee Won Jung^{**} · Lee Jae Wan^{***} · Joo Su Chong^{****}

ABSTRACT

As the internet technologies develop, the geographic information system environment is changing to the web-based service. Since geospatial information of the existing Web-GIS services were developed independently, there is no interoperability to support diverse map formats. In spite of the same geospatial information object, it can be used for various purposes that is duplicated in GIS separately. It needs intelligent strategies for optimal replica selection, which is identification of replication geospatial information objects. And, for management of replication objects, OMG, GLOBE and GRID computing suggested related frameworks. But these researches are not thorough going enough in case of geospatial information object. This paper presents a model of location service, which is supported for optimal selection among replication and management of replication objects. It is consist of tree main services. The first is binding service which can save names and properties of object defined by users according to service offers and enable clients to search them on the service offers. The second is location service which can manage location information with contact records. And obtains performance information by the Load Sharing Facility on system independently with contact address. The third is intelligent selection service which can obtain basic/performance information from the binding service/location service and provide both faster access and better performance characteristics by rules as intelligent model based on rough sets.

For the validity of location service model, this research presents the processes of location service execution with Graphic User Interface.

Key Words : Distributed Geographic Information System, Geospatial Information Location Service, Replication Object

1. 서 론

인터넷 기술과 네트워크 기술은 지리정보시스템의 패러다임으로 변화 시켰다. 특히 컴퓨터 기술의 발전은 분산 지리

정보시스템으로 보다 실용적이고, 다양한 분야에 활용되어 가고 있다[1, 2]. 그러나 분산 지리정보시스템의 실현은 기존의 구축된 지리정보의 이질성 이유로 어려움을 겪고 있다. 이러한 이유는 GIS(Geographic Information System) 벤더가 제공하는 고가의 공간 데이터를 처리하는 툴에 의해 지리정보를 구축하고, GIS 애플리케이션을 개발하여 왔기 때문이다. 분산 지리정보 시스템에서 지리적 상호운용성은 정보 시스템이 자유롭게 지구상의 모든 종류의 지리공간 정보를

※ 본 연구는 한국과학기술부의 특정 기초연구사업(R01-2006-000-10147-0)의 지원으로 수행되었음.

† 준 회 원 : 원광대학교 전기전자 및 정보공학부 Post-Doc

** 정 회 원 : 군장대학 사회실무학부 교수

*** 정 회 원 : 군장대학교 전자정보공학부 교수

**** 정 회 원 : 원광대학교 전기전자 및 정보공학부 교수

논문접수: 2006년 8월 3일, 심사완료: 2006년 10월 9일

교환하고, 소프트웨어를 사용하여 네트워크 상에서 이러한 정보들을 조작할 수 있는 능력을 의미한다. 이러한 상호 운용을 위해서는 네트워크상에 특정 영역에 위치한 시스템간의 정보공유를 위한 메커니즘이 필요로 하며, 서로 다른 이질적인 플랫폼을 극복해야만 한다. 이러한 기존 GIS의 이질성 해결과 상호 운용이 가능한 분산 환경을 제공하기 위해 점차 컴포넌트 기반 GIS로 변화하고 있다[3, 4]. 컴포넌트 기술은 클라이언트와 서버를 이루는 애플리케이션과 운영체제에 대한 독립적인 플랫폼을 제공하는 하기 위해 OMG CORBA, 마이크로소프트의 DCOM 그리고 썬 마이크로 시스템즈의 JAVA RMI가 이에 해당하는 기술로 컴포넌트 기반 GIS는 기존 GIS 기술의 급속한 발전을 이끌어 보다 개방적인 환경을 제공하고 있다. 이에 관련된 대표적인 연구는 OGC(Open GIS Consortium)에서 주도하고 있다[5, 6]. OGC에서는 상호운용성을 지원하기 위해 GML(Geography Markup Language)명세서를 제시하였다[7]. 이를 기반으로 표준화 작업을 하는 기타 연구기관 그리고 벤더들에 의해 GIS의 지리정보의 상호운용을 위한 표준화 연구가 활발하게 진행 중이나, 기존 시스템이 갖는 플랫폼 상의 이질성을 포함하여, 구성요소들간의 상호운용성을 지원하기 위해서는 더 많은 연구를 필요로 하고 있다. 앞으로, 분산 컴퓨팅 환경의 GIS 시스템은 점차 유비쿼터스 컴퓨팅 환경으로 점차 확장될 전망이다. 이러한 환경은 클라이언트들이 원하는 지리정보 객체나 클라이언트 환경에 적합한 구성요소들을 찾아 제조를 통하여 서비스를 제공하는 형태로 변화됨을 예측하게 한다. 이를 위해, 사용자에게 보다 지리공간 객체의 위치에 상관없이 접근 가능하며, 기존의 구축된 지리정보 객체들의 이름이나 속성을 기반으로 중복된 지리정보 객체들의 관리가 요구된다. 특히, 클라이언트가 중복된 지리정보 객체들 중에 임의의 특정 지리정보 객체에 바인딩 하는 경우, 지리정보 객체가 위치한 성능 측면을 고려하여 최적의 지리정보 객체를 선정하는 지능적인 모델이 요구된다.

따라서, 본 논문에서는 중복된 지리정보 객체의 관리와 클라이언트 요청에 적합한 지리정보 객체가 중복되어 있을 경우, 이들 중 최적의 지리정보 객체를 선정 할 수 있는 위치 서비스 모델을 제시한다. 이를 위해 서비스 제공자가 정의한 지리정보 객체의 이름과 속성을 서비스 오피 단위로 저장하고, 이를 근거로 클라이언트가 검색할 수 있는 바인딩 서비스와 객체들의 위치 정보를 컨택(contact) 레코드로 관리하는 위치 서비스 그리고 이러한 지리정보 객체의 기본 속성정보와 객체가 상주하는 시스템의 부하정보를 포함한 성능에 관련된 정보를 기반으로 최적의 지리정보 객체를 선정하기 위한 지능형 선정 서비스로 구성하였다. 분산 컴퓨팅 환경에서 지리정보를 관리하는 서버간의 지리정보 객체들의 이동을 지원하기 위해 바인딩 서비스와 위치 서비스를 각각 독립적으로 운영하며, 이들 서비스가 관리하는 정보 매핑은 객체의 식별자인 객체 핸들(object handle)단위로 관리된다. 이로 인하여 지리정보 객체의 위치가 변경되더라도 위치 서비스가 관리하는 저장소에서 컨택 레코드의 위치 정보만 변경한다. 또한, 중복된 지리정보 객체들 중에 하나의

객체를 선정하기 위해 지리정보 객체가 갖는 속성정보와 성능에 관련된 정보를 이용하여, 러프집합(Rough Sets) 이론 [8, 9, 10]을 적용한 지능형 지리정보 객체를 선정 전략을 통해 가장 최적의 지리정보 객체를 선정하도록 하였다. 이러한 위치 서비스 모델은 중복된 지리정보 객체에 대한 효과적인 관리뿐만 아니라 위치 이동성 및 사용자에게 투명성을 제공한다.

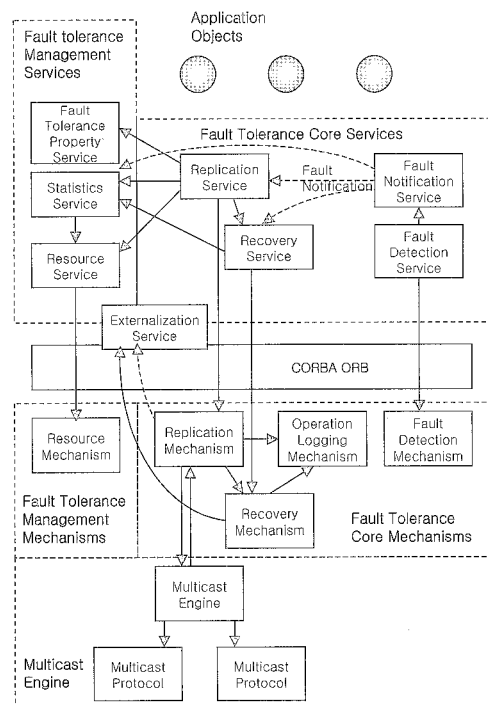
2. 관련 연구

본 장에서는 기존에 구축된 GIS 시스템들이 운영 목적에 따라 상이한 레이어를 갖는 지리공간 객체를 구축함으로써 인하여 동일한 지리공간 객체임에도 불구하고 중복되어 있으므로 이를 해결하기 위한 대표적인 중복 객체 관리 모델과 중복 객체 선정 전략에 대해서 살펴본다.

2.1 중복 객체 관리 모델

2.1.1 OMG의 Replication 모델

OMG CORBA의 Object Replication Service[11]는 분산 환경에서 객체들의 명확한 복제와 복사된 복사본의 일관성을 관리를 제공한다. 여기에서 복사본들의 상호 일관성은 매우 중요한 요구사항이다. 따라서 하나의 복사본이 갱신이 필요하다면 다른 복사본에게 변화된 부분을 전달해야 한다. 복사본 객체들은 생성되거나 사용되는 주소 공간의 밖에서 인식할 수 있는 그들만의 이름을 사용한다. Replication Service는 복사본들의 갱신을 위한 직접적인 로깅 메커니즘을 사용한다. 로깅 메커니즘은 클래스 계층에서 관련된 서브 클래스 모두의 갱신을 지원한다. 그러므로 Replication Service 동일한 객



(그림 1) 고장 허용 프레임워크의 구조

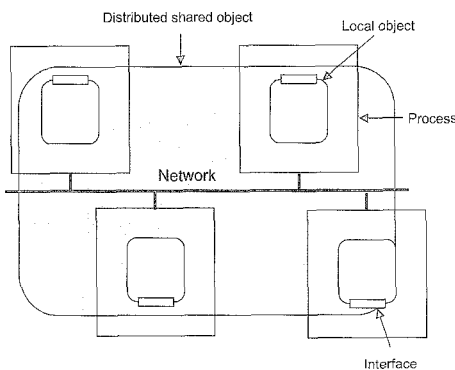
체가 작업하는 멀티 프로세스를 찾아낸다. 즉, 클라이언트에 게 필요한 공통 객체를 검사하고 찾아내는 역할을 담당한다.

최근 Replication Service는 고장 허용 프레임워크에 대한 구조에서 포함한 연구와 객체 그룹에서 복사본들을 관리하기 위한 연구로 진행하고 있다[12]. 다음 (그림 1)은 CORBA에서 고장 허용 프레임워크에 대한 구조를 보이고 있다.

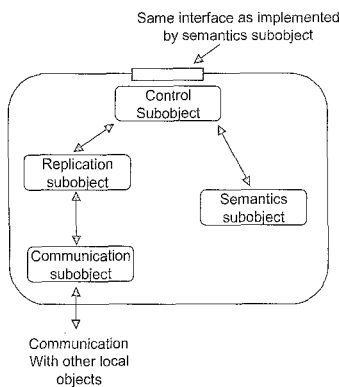
(그림 1)에서 나타난 바와 같이 고장 허용 프레임워크에서는 복사의 기본 단위인 개별적인 객체를 지원한다. 특정 객체의 복사본들은 동일한 IDL로 구현되며 동일한 구현 소스 코드를 가지므로, 동일한 오퍼레이션을 갖는다. 여기에서 복사본들의 집합을 객체들의 그룹으로 간주하며, 애플리케이션은 객체의 복사본에 직접적인 접근을 하지 못한다. 이들은 고장 허용 핵심 서비스의 제어 하에 논리적인 Replication 도메인을 구성하여 관리 된다.

2.1.2 GLOBE Replication 모델

GLOBE는 규모가 큰 광역 분산시스템 개발을 위한 매우 유용한 통일된 패러다임을 제공하는데 목적을 두고 있다. 이를 위해 통일된 객체 기반의 분산 공유 객체모델을 제시하고, 분산 투명성을 제공하기 위한 위치 서비스를 제시하고 있다[13, 14]. GLOBE는 원격 객체모델을 채택하는 대신에 객체의 상태를 분산시키고, 여러 프로세서에 복사하는 물리적인 분산을 제공한다. (그림 2)는 서로 다른 시스템 상에 운영되는 4개의 프로세스에 분산된 객체를 보이고 있다.



(그림 2) GLOBE 분산 공유 객체의 구성



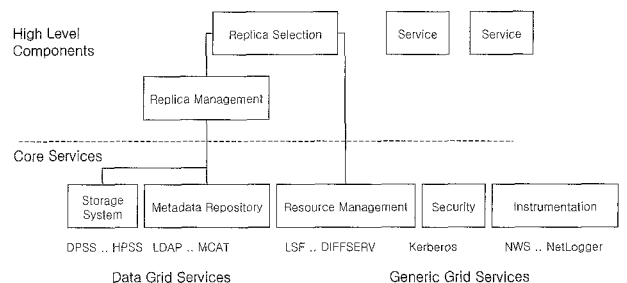
(그림 3) 분산 공유 객체를 위한 지역 객체의 구성

GLOBE에서 객체는 분산 공유 객체로 여러 프로세서간에 공유된다. 분산 공유 객체에 바인딩 되는 프로세스는 객체가 제공하는 인터페이스의 지역 구현(local implementation)을 제공한다. 이와 같은 지역 구현은 지역 표현(local representative) 또는 지역 객체라 불린다. 지역 객체는 연결될 프로세스에 투명성을 제공한다. 객체의 세부적인 구현은 프로세스가 제공하는 인터페이스에 의해 숨겨진다. 각 지역 객체는 DCOM에서 사용하는 IUnknown 인터페이스와 유사한 SOInf라 불리는 표준화된 객체 인터페이스로 구현된다. (그림 3)은 GLOBE의 객체 모델로 4개의 서브 객체들로 구성된 지역 객체를 보이고 있다.

시멘틱 서브객체는 분산 객체의 실제적인 서비스를 제공하는 객체로 개발자는 분산 객체의 클래스 객체를 구현해야 한다. 제어 서브객체는 클라이언트 프로세스로부터 호출을 책임지고, 복사 객체와 시멘틱 객체간의 상호작용을 제어한다. 또한 마샬링과 분산 객체의 상태를 관리한다. 분산 객체는 고장 허용 또는 성능의 향상을 위해 복사될 수 있다. 복사 서브객체는 이러한 복사 전략에 따라 복사된 복사본의 일관성을 유지하는데 책임을 진다. 통신 서브객체는 일반적으로 시스템에서 제공하는 지역 객체로서 서로 다른 주소 공간에 위치한 분산 객체간의 통신을 주관한다.

2.1.3 Globus Data Grid architecture

Globus data grid[15, 16]는 두개의 레이어로 코어 서비스와 이러한 코어 서비스들을 사용하여 구현하기 위한 높은 레벨 서비스로 구성된다. 이는 서비스들과 다양한 애플리케이션과 툴들을 재사용할 목적으로 제시하였다. 특정 애플리케이션 복사 관리 솔루션과 저장 관리 시스템인 저장소 요청 브로커(storage request broker)가 공통적인 하위 레벨 메커니즘으로 상호 공유할 수 있다. 다음 (그림 4)는 데이터 그리드 구조를 보이고 있다. 코어 데이터 그리드 서비스들은 그리드 환경에 존재하는 다수의 저장 시스템에 대한 메타 정보를 관리함으로써 높은 레벨 서비스들과 애플리케이션들이 이러한 저장 시스템에 접근할 수 있도록 한다. 코어 서비스들은 저장 시스템, 데이터 접근 서비스를 제공한다. 즉, 저장 시스템에 위치한 데이터 관리와 접근하기 위한 기본적인 메커니즘을 제공한다. 이러한 메커니즘은 비록 분산 된 물리적인 위치에 상관없이 저장 시스템 상에 파일의 인스턴스를 일관성 있게 생성하거나 삭제, 접근 그리고 수정하기 위한 추상화를 제공한다. 이는 기존의 유닉스 파일 시스템과 고



(그림 4) 데이터 그리드 구조

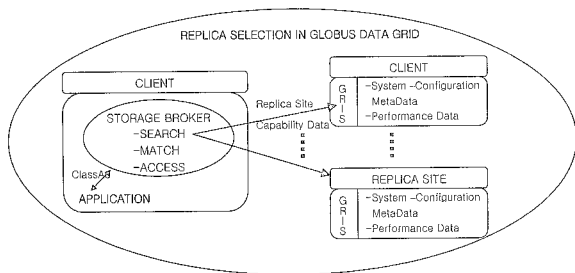
성능 저장 시스템(High Performance Storage System : HPSS)와 Unitree SRB와 같은 저장 시스템에 중점을 둔다.

메타데이터 접근 서비스는 저장 시스템에 저장된 데이터에 관한 관리와 접근 메커니즘을 제공한다. 다양한 메타데이터는 다음과 같은 서비스에 의해 제공된다.

- 애플리케이션 메타데이터는 애플리케이션이 사용하는 다양한 세부사항과 수집된 데이터에 대한 파일의 컨테츠에 대해 기술한다.
- 복사 메타데이터는 특별한 복사와 파일의 인스턴스간의 매핑을 기술한다.
- 시스템 구성 메타데이터는 저장 시스템의 가용성에 대해 기술한다. 따라서 데이터 그리드의 구조상의 정보를 제공한다.

메타데이터서비스는 다양한 메타데이터를 접근하거나 공개할 수 있는 일관성 있는 방법을 제공할 예정이다. 이러한 서비스들은 Globus 툴킷에 의해 제공되는 기본적인 보안과 정보서비스에 의해 만들어진다. 다음 (그림 5)는 Globus 데이터 그리드에서 복사본을 선정하는데 핵심 부분인 저장 브로커 구조를 나타낸다.

코어 데이터 그리드 서비스들은 여러 가지의 높은 레벨 서비스를 구축하는데 사용된다. 예를 들면 복사본 관리는 저장 사이트에 복사본들을 생성하거나 삭제를 처리한다. 이러한 복사본은 원본 파일의 복사로 추출된다. 이는 성능향상에 이점을 갖는다. 복사본 관리자는 전형적으로 복사 사이트주소와 파일 인스턴스를 포함한 복사본 카탈로그를 유지 관리한다. 복사본 선택은 애플리케이션의 성능을 향상시키기 위해 그리드 상에 분산되어 있는 복사본들 중에 하나의 복사본을 선택하는 과정을 말한다. 이를 위해 시스템의 성능에 관련된 메타데이터를 활용하여 가장 적합한 복사본을 선택한다. 이에 대한 책임은 저장 브로커의 선정 전략에 따른다.



(그림 5) 저장 브로커 구조

2.2 기존 중복 객체 선정 전략

일반적으로 분산 시스템을 설계하는데 강인성(robustness), 확장성(scalability) 그리고 성능(performance)을 고려한다. 특히 시스템의 성능을 고려하는 방법으로 중복(replication) 개념을 채택하고 있다[17, 18]. 중복 서버 구조는 각 클라이언트 트랜잭션을 위한 목표 복사본을 선택하는 기술들이 요구

된다. 중복 객체를 선정하는데 대표적인 방법으로 무작위 선정 방법은 클라이언트의 요청에 따라 가용한 중복 복사본들의 집합으로부터 하나의 복사본을 선정하는데 무작위 선정하는 방법이다. 이에 대한 예로는 FTP에서 이용하고 있다. 또한, 라운드 로빈 방법은 사용자 요구를 차례대로 각 서버에 균등하게 분배하는 방식으로 새로운 연결이 요청되는 각 시간에 동일선상의 다음 서버에게 연결을 시킨다. 서버와의 연결 수나 응답시간에 상관없이 그룹 내의 모든 서버를 동일하게 선정 처리하여 일반적인 구성에 있어서 다른 알고리즘에 비해서 가장 빠르다는 장점이 있다. 그리고 중복된 서비스 객체를 선정하는 전략에서 널리 사용되는 방법 중의 하나는 클라이언트의 요청에 따라 가용한 서비스 객체를 주기적인 스케줄링 방법으로 서비스 객체를 선정한다. 이에 대한 예로는 Orbix 제품인 OTM(Object Transaction Middleware)과 웹 서비스를 위한 DNS에서 사용한다. 이와 함께 분산 시스템에서 많이 쓰이는 중복 객체 선정 알고리즘은 다음과 같다.

- ▶ Least Connection(최소접속방식) : 오픈 커넥션이 가장 적은 서버로 사용자 요구를 연결하는 방식으로 모든 서버가 균등한 트래픽을 유지하기 위해서 처리 속도가 빠른 서버를 선정함으로써 보다 많은 네트워크의 접속처리를 담당하게 한다. 최소접속 알고리즘은 서버들의 성능이 비슷하게 구성되었을 경우에 가장 효과적인 트래픽 분산이 가능하나, 실제로는 TCP의 TIME_WAIT 상태 때문에 아주 좋은 성능을 낼 수 없다
- ▶ Weighted Least Connections(가중치 최소접속방식) : 가중치 기반 최소 접속 방식은 최소 접속 방식의 한 부분으로서 각각의 실제 서버에 성능 가중치를 부여하여, 언제나라도 가중치가 높은 서버를 선정하여 더 많은 요청을 받을 수 있다. 가상 서버의 관리자는 각각의 실제 서버에 가중치를 부여할 수 있다. 가중치의 비율인 실제 접속 자수에 따라 네트워크 접속이 할당된다. 기본 가중치는 1이다.
- ▶ Fastest Response Time(응답시간방식) : 가장 빨리 응답하는 서버를 선정하여 사용자 요구를 연결하는 방법으로 이 알고리즘은 서버 배열을 다양하게 하였을 때에는 다른 서버로 트래픽을 연결하기 전에 특정서버로 접속부하가 걸리므로 비효율적이며 웹서버는 일반적으로 커넥션의 수에 따라서 선행의 응답이 이루어지지 않기 때문에 보통 웹서버는 평범하게 응답하다가 트래픽이 증가 하는 어느 수준에 이르러서 극적으로 응답시간이 증가한다. 이 알고리즘은 커넥션 수를 고려하지 않고 서버 속도에 의지함으로써 이러한 문제를 해결할 수 있다.
- ▶ Adaptive(최소대기방식) : 이 알고리즘은 Open 또는 Pending 커넥션을 적게 가지고 있는 서버를 선정하여 네트워크 커넥션 방향을 지정한다. Pending 커넥션은 Full TCP Handshake를 완성하지 않은 것으로, 이것은 초당 클라이언트 Thread의 수가 증가할 때 더욱 잘 수행된다.

이러한 중복 객체 선정을 위한 전략과 알고리즘에 관한 연구들이 성능에 무관하게 선정하는 방법에서 동적인 측면을 고려한 방법으로 진행되고 있으며, 특히 다양한 지능형 알고리즘을 적용하고 있는 추세이다. 즉, 중복 객체의 조건을 고려한 방법으로 동적인 할당 방법에 대한 대표적인 연구로 greedy 선정 방법으로 중복된 서비스 객체를 선정하고 있다[19, 20]. 앞서 언급한 기존의 알고리즘들은 점차 서버의 상황에 따른 조건을 기준으로 선정하는 알고리즘으로 진화되고 있다. 그러나 이러한 고려 사항에는 클라이언트 측면과 서버 측면을 모두 고려해야만 하며, 보다 많은 지능적인 선정 방법이 요구되고 있다.

본 논문에서 제안하는 위치 서비스 모델은 서로 다른 도메인에서 독립적으로 운용되는 Web GIS가 갖고 있는 지리정보 즉 공간 정보와 속성정보가 동일한 지리정보 객체를 중복 지리정보 객체로 정의하고 이러한 지리정보 객체의 다양한 기본적인 속성 정보뿐만 아니라 성능에 관련된 정보를 고려하여 최적의 객체를 선정하는데 리프 집합 이론을 적용하고자 한다.

3. 위치 서비스 모델

3.1 요구사항

본 위치 서비스 모델은 분산 컴퓨팅 환경에서 각 도메인별 독립적으로 운영되는 분산 객체를 위치 서비스 모델에 의해 클라이언트에게 투명하게 분산 투명성을 제공하고자 한다. 이는 자신이 속한 도메인 내의 지리정보 객체들뿐만 아니라 다른 도메인상의 지리정보 객체에 대한 위치 정보를 필수적으로 제공해야한다. 이러한 정보 제공을 위해 고려되는 기능별 요구사항들은 다음과 같다.

- ▶ 위치 서비스는 이름 또는 속성으로 분산 객체를 찾기 위한 메커니즘을 제공한다. 또한 지리정보 객체의 위치 정보는 위치 서비스에서 관리하는 2 계층 구조를 갖는다.
- ▶ 지리정보 객체에 대한 표현은 OMG의 트레이딩 객체 서비스 사양을 따르는 서비스 오퍼 구조를 갖으며, 위치 정보는 컨택 주소로 각각 관리한다.
- ▶ 서비스 오퍼와 위치 정보의 매핑은 지리정보 객체를 등록시킬 때 생성되는 식별자인 객체 핸들(128bit)에 의해 이루어진다.
- ▶ 동일한 도메인 내에서 중복 지리정보 객체의 표현은 위치 서비스가 관리하는 저장소에 한 개 이상의 컨택 주소 값을 갖는다.

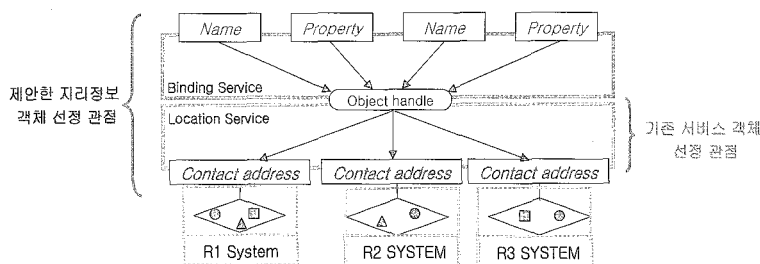
- ▶ 중복 지리정보 객체의 이동은 지리정보 객체를 운영하는 서버간의 이동이며, 클라이언트의 요청에 따라 서버상의 지리정보 객체를 다운로드한 지리정보 객체와는 구분된다. 동일 도메인 내에서 다른 시스템으로 이동한 경우에는 해당 도메인에 위치한 위치 서비스의 컨택 주소만을 변경한다. 다른 도메인으로 이동할 경우에는 해당 위치 서비스에서 서비스오퍼 정보와 컨택 주소 값을 삭제한 후, 이동한 지리정보 객체의 위치 변경을 표현한다(단, 최초 생성된 객체 핸들의 값을 변경시키거나 옮겨진 도메인 내에서 재 생성된 값을 사용하지 않는다.).
- ▶ 동일한 속성 정보를 갖는 중복 지리정보 객체 중에 최적 지리정보 객체를 선정하기 위해 지리정보 객체가 갖는 기본 속성정보 또는 기타 성능에 관련된 정보만을 이용하지 않고 모두 포함하여 지능형 선정 전략을 통해 최적의 지리정보 객체를 선정한다.
- ▶ 지리정보 객체는 관리 목적 또는 서비스 제공 목적에 따라 분산되어 위치한다.

본 연구는 이와 같은 고려 사항을 토대로 위치 서비스 모델에 제안한다. 그리고 위치 서비스에 대한 기본 구성과 선정 전략을 기술하고, 이에 대한 세부적인 설명은 다음절에서 기술한다.

3.2 제안한 위치 서비스 모델

본 논문에서는 분산 지리정보 시스템의 지리 정보를 활용하기 위해 해당 레이어 또는 특정 찾기 위해서는 기존의 통합 바인딩 서비스에서 사용했던 2단계 구조의 바인딩 서비스와 위치 서비스를 사용하며, 객체를 선정하는 관점에서 기존의 연구[26, 27, 28]에서는 컨택 주소에 해당하는 서비스 객체가 위치한 시스템의 부하만을 고려하였으나, 본 연구에서는 지리정보 객체가 갖는 기본적인 속성 정보와 컨택 주소에 해당하는 성능에 관련된 정보를 포함한 의사 결정 테이블을 통해 지리정보 객체를 선정하도록 하였다. 다음(그림 6)은 본 논문에서 중복된 지리정보 객체 선정을 위한 2 단계 구조와 고려할 관점을 보인다.

(그림 6)에서 나타난 바와 같이 중복된 지리정보 객체를 찾아 바인딩 하기 위해서는 바인딩 기능과 위치 서비스의 키가 되는 객체 핸들의 특징은 다음과 같다. 분산 지리정보 시스템 환경에서 수많은 지리정보 객체들을 각각 식별할 수 있

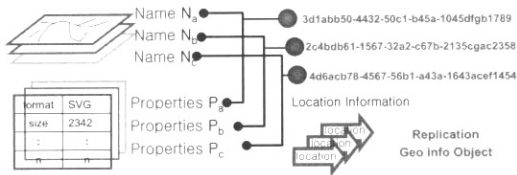


(그림 6) 본 연구에서 제안하는 지리정보 객체 선정을 위한 관점

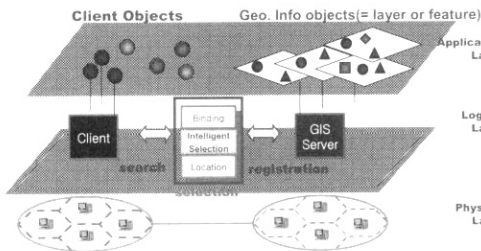
도록 해야 하며, 생성된 식별자(객체 핸들)는 객체의 생명주기 동안에 다른 객체가 재사용 할 수 없어야 한다. 또한 초기에 할당된 후 변경되거나 한 개 이상의 다른 객체 핸들을 얻을 수 없다. 객체 핸들은 지리정보 객체들의 위치를 지원하는 정보와 매핑된다. 이때 하나의 객체 핸들당 위치 정보가 한 개 이상이라면 중복되었음을 의미한다. 다음 (그림 7)은 중복 지리정보 객체와 객체 핸들 그리고 객체 핸들에 해당하는 예를 보인다.

본 논문에서 제시하는 위치 서비스 모델은 (그림 8)에서 나타난 바와 같이 3계층으로 물리적 계층은 네트워크 구조 즉 네트워크로 연결된 시스템의 모임이다.

애플리케이션 계층은 분산된 지리정보 시스템에서 사용되는 레이어 또는 특징인 지리정보 객체를 말한다. 논리적인 계층은 이러한 객체들을 관리하며, 중복된 지리정보 객체중에 최적의 객체를 선정하여 바인딩 서비스를 제공하는



(그림 7) 중복 지리정보 객체, 객체 핸들, 위치 정보



(그림 8) 위치 서비스 모델

관점을 의미한다. 이러한 위치 서비스는 클라이언트에게 분산되어 있는 지리정보 객체들을 관리하며 최적의 객체를 선정하여 제공한다.

3.3 단계별 위치 서비스 과정

▶ 이름/속성 검색

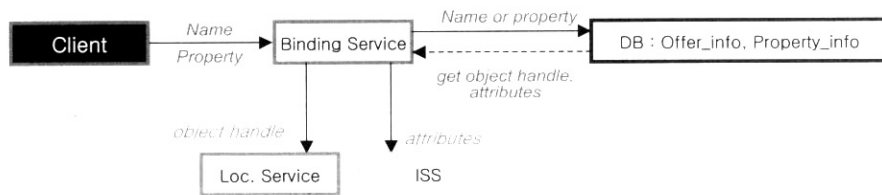
바인딩 서비스에 클라이언트가 전달하는 파라미터가 이름 또는 속성 값에 따라, 이에 가장 적합한 지리정보 객체에 대한 객체 핸들과 매핑 시킨다. 여기에서 이름 일 경우에는 오픈 정보의 객체 이름과 매핑된 객체 핸들을 반환하고, 속성일 경우에는 속성 정보의 속성 값들과 매핑한다. 매핑된 객체 핸들을 위치 서비스에 전달하고, 관련 속성 정보를 지능형 선정 서비스에 전달한다. 다음 (그림 9)는 이에 대한 처리 과정을 나타낸다.

▶ 위치 검색

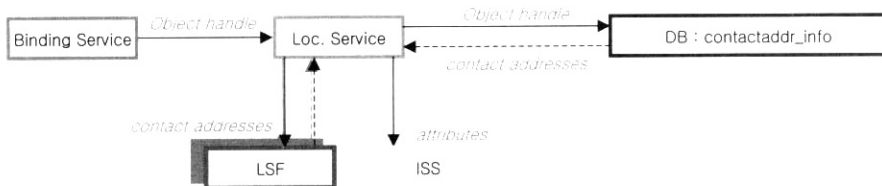
위치 서비스에서는 이름/속성 검색에서 매핑된 객체 핸들을 입력 파라미터로 해당 객체의 컨택 주소와 매핑 시킨다. 위치 서비스에서 관리하는 contactaddr_info 테이블의 컨택 레코드의 한 개 이상의 컨택 주소 값을 반환하게 된다. 여기에서 한 개 이상의 컨택 주소는 중복 지리정보 객체를 의미한다. 얻어진 컨택 주소에 해당하는 호스트 시스템 상의 LIM에게 시스템의 부하정보를 요청한다. 각 시스템의 동적인 부하 정보를 얻고, 이를 지능형 선정 서비스에 반환한다.

▶ 의사 결정

의사 결정 단계는 지능형 선정 서비스에 의해 바인딩 서비스로부터 이름/속성 검색으로 얻어진 지리정보 객체의 기본적인 속성 정보와 위치 서비스로부터 얻어진 시스템의 부하정보를 통합하여 의사 결정 데이터 테이블을 생성하고, 이러한 데이터 값의 일정한 패턴을 분석하여 얻어진 규칙을

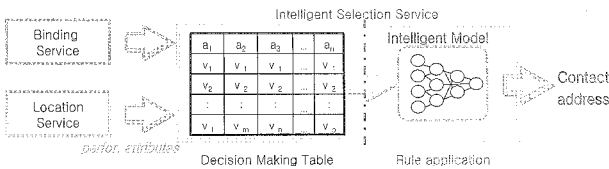


(그림 9) 이름/속성 검색 과정



(그림 10) 위치 검색 과정 및 동적 속성 정보 획득과정

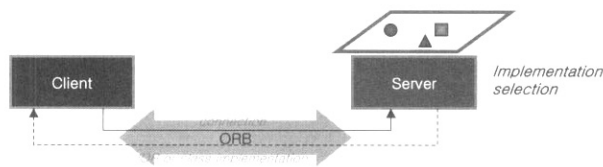
적용하여 최적의 지리정보 객체를 선정하는 과정이다. 다음 (그림 11)은 이에 대한 선정 과정을 보이고 있다.



(그림 11) 최적 지리정보 객체 선정 과정

▷ 구현 객체 선정

구현 객체 선정(Implementation selection) 과정은 클라이언트가 지리정보 객체와 접속하기 위해 클라이언트측 ORB의 스텝과 객체가 위치한 서버측 ORB 스킴레톤간의 통신을 통해 구현 객체를 선택하게 된다. 이를 위해 지리정보 객체가 위치한 시스템의 해당 구현 저장소 또는 웹 서버의 디렉토리 상에 위치한 객체를 선정하는 과정이다. 프로토콜 또는 포트 번호와 같은 하위 레벨의 통신 메커니즘에 대해선 무관하게 클라이언트는 지리정보 객체에 바인딩한다.



(그림 12) 서버 상의 지리정보 객체인 구현 객체 선정

▷ 바인딩

이 과정은 클라이언트 측에 서비스를 제공하는 객체에 접근하기 위한 스텝 설치가 완료 후, 서비스 요청하기 위해 매소드 호출을 하고, 구현 객체로부터 또는 웹 서버로부터 서비스 수행 후 결과에 해당하는 Map을 반환하는 과정이다. 이에 대한 과정은 (그림 13)에서 나타낸다.



(그림 13) 클라이언트/서버 시스템 바인딩

4. 최적 객체 선정 전략

일반적으로 지리정보 객체가 갖는 다양한 속성 정보에 따르는 불확실한 데이터를 분류하는데 가장 적합한 러프 집합 이론을 지리정보 객체를 선정하기 위한 전략으로 채택하여 최적의 지리정보 객체를 선정하기 위해 규칙을 적용하고자 한다.

지능형 선정 서버에서 사용되는 규칙을 생성하는 과정은 다음과 같다. 먼저 규칙을 생성하기 위해서 가상의 의사 결정 데이터 테이블을 구성하기 위한 조건 속성 정의가 필요하다.

지리정보의 공간 정보로는 레이어(layer)와 POI(Point Of Interest) 그리고 지도의 포맷으로 정의한다. 여기에서 레이어는 운영 시스템의 목적에 따라 구축되는 수가 다르다. 교

통정보 관리를 위한 레이어의 경우 32개의 레이어를 구축하여 운영하고 있다. 위치 안내를 제공할 경우에는 밑에 구축된 무인 단속기와 같은 레이어는 불필요하나 일반적으로 GIS 시스템 구축 목적에 따라 도로, 차선과 같은 레이어는 필수적인 레이어로 활용하고 있다.

POI의 경우도 마찬가지로 운영 시스템의 서비스의 특성에 따라 건물을 위주로 구축하고 있다. 차량의 CNS(Car Navigation System) 내비게이션으로 활용되는 가장 중요한 POI는 주유소를 비롯하여 건물에 관련된 POI를 구축한다. 그러나 PNS(Personal Navigation System)의 경우에는 차량과는 달리 다양한 POI를 요구하게 된다.

또한, 중요한 공간정보로는 구축된 시스템의 공간정보에 대한 파일 포맷이 서로 다르며 이는 성능과 밀접한 관계를 갖고 있다. 다음 <표 1>은 동일한 건물 레이어의 서로 다른 포맷의 사이즈를 나타낸다.

<표 1> 건물에 대한 포맷별 사이즈

종류	DWG	SHAPE	SVG	GML
크기(KB)	29,178	4,205	14,908	44,593

해당 지리정보 객체가 위치한 시스템의 성능에 관련된 속성 정보에 필수적인 정보는 시스템의 CPU 이용률과 응답시간으로 구분할 수 있다. CPU 이용률은 기존 PostGIS에서 성능을 개선하기 위해 활용한 기준을 활용한다[23].

- CPU <5 % 적은 CPU 이용률
- CPU >75% 높은 CPU 이용률

응답시간의 경우는 웹 GIS 서비스[24]를 위한 Web QoS에서 정의한 사용자가 요구하는 응답시간 기준을 따른다[25].

- <1000 msec : 높은 응답 시간
- >2000 msec : 낮은 응답 시간

앞서 언급한 조건 정보는 <표 2>와 같이 바인딩 서비스로부터 서비스 오피에 해당하는 속성 정보와 위치 서비스로부터 획득한 지리정보 객체가 상주하는 시스템 부하 정보를 포함한 의사 결정 데이터 테이블을 생성한다.

<표 2> 의사 결정 데이터 테이블

속성 \ 값	1	2	3	비고
a	shape	svg	gml	지도 포맷 종류
b	high response time	low response time	-	응답 시간
c	high performance	low performance	-	CPU 성능
d	40 이상	40 이하	-	POI 개수

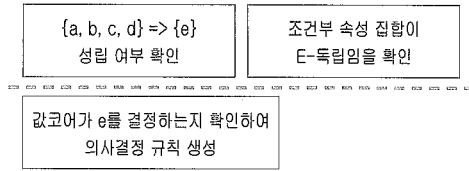
규칙 적용 과정은 중복된 지리정보 객체 중에서 최적의 객체를 선정하기 위한 알고리즘을 적용하여 분류하고 우선순위를 선정한다. 여기에서 의사결정 속성인 e는 이러한 조

건 속성정보를 기반으로 결정된 값을 의미한다. H는 high를 M은 middle 그리고 L은 low로 클라이언트의 측면에서 값을 결정하였다. 먼저 규칙을 생성하기 위해서 가상의 의사 결정 데이터 테이블을 구성하고 속성을 정의한다.

(그림 14)에 나타난 테이블을 근거로 하여 러프 집합 이론을 적용하여 규칙을 생성하기 위해서는 불필요한 조건을 찾아내 제거나 조건부 속성 중 불필요한 값을 제거하는 방법을 사용한다. 다음 (그림 15)는 러프집합 이론에서 의사결정을 하기 위한 방법을 나타낸다.

R	a	b	c	d	e
1	1	1	2	2	H
2	1	2	2	2	H
3	2	1	2	2	H
4	3	1	2	2	H
5	1	1	1	2	M
6	1	2	1	2	M
7	2	1	1	2	M
8	2	2	1	2	M
9	3	2	1	2	M
10	1	1	1	1	L
11	1	1	2	1	L
12	1	2	1	1	L
13	1	2	2	1	L
14	2	1	1	1	L
15	2	1	1	1	L
16	2	2	1	1	L
17	2	2	2	1	L
18	2	2	2	2	L
19	3	1	1	1	L
20	3	1	1	2	L
21	3	1	2	1	L
22	3	2	1	1	L
23	3	2	2	1	L
24	3	2	2	2	L

(그림 14) 의사 결정 데이터 테이블



(그림 15) 러프집합 이론에서 의사결정을 위한 방법

이러한 문제를 해결하기 위한 접근법에는 의사결정표의 간략화나 의사결정 알고리즘을 찾는 것으로 나타낼 수 있다. 전자의 방법은 식별 불가능 관계에 따른 연산에 기초하고 대수학적인 반면 후자는 논리학을 이용한다. 본 논문에서는 의사결정표의 간략화에 중점을 두어 규칙을 생성하고자 한다. 먼저 의사 결정부 속성으로 이루어진 원소범주가 모든 조건부 속성의 집합으로 정의된 기본범주로 정의 되는가와 의사결정 범주의 정의에 필요한 조건부 범주 집합의 감축에 대해 살펴본다.

(그림 16)에 나타난 바와 같이 의사결정을 나타내는 속성 e가 조건을 나타내는 속성 a, b, c, d에 종속되는가를 확인한다. 그리고 특정 속성을 제거하여 의사 결정표가 비일관적이 되는 지 확인한다. 그 결과 조건부 속성집합은 e-독립임을 알 수 있다.

두 번째 방법으로 조건부 원소범주 중에 제거할 수 있는 것이 있는지 확인을 하기 위해 조건부 속성 중 불필요한 값을 제거할 수 있는지 확인한다. 이는 특정한 값이 불필요한지 아닌지를 확인하여 동일한 의사결정부 속성값을 갖는지 확인할 수 있도록 해당 값이 코어에 속하는지 확인한다. 이를 통해 다음 (그림 17)과 같이 모든 코어값들과 이에 해당하는 리덕트값을 구하게 된다.

앞서 리덕트 값을 모두 표현한 테이블은 다음 (그림 18)과 같다.

(그림 19)에서 나타난 바와 같이 의사결정부 동치류인 것은 하나의 대표적인 규칙을 도출해내고 불필요한 의사결정 규칙들을 제거한다. 이를 통해 의사 결정 규칙은 다음 (그림 20)과 같이 정의할 수 있다.

R	a	b	c	d	e
1	1	1	2	2	H
2	1	2	2	2	H
3	2	1	2	2	H
4	3	1	2	2	H
5	1	1	1	2	M
6	1	2	1	2	M
7	2	1	1	2	M
8	2	2	1	2	M
9	3	2	1	2	M
10	1	1	1	1	L
11	1	1	2	1	L
12	1	2	1	1	L
13	1	2	2	1	L
14	2	1	1	1	L
15	2	1	1	1	L
16	2	2	1	1	L
17	2	2	2	1	L
18	2	2	2	2	L
19	3	1	1	1	L
20	3	1	1	2	L
21	3	1	2	1	L
22	3	2	1	1	L
23	3	2	2	1	L
24	3	2	2	2	L

속성 a 를 삭제

$b_2c_2d_2 \rightarrow e_H$ (규칙2)

$b_2c_2d_2 \rightarrow e_L$ (규칙 18, 24)

⇒ 모순된 의사결정 규칙

$b_1c_1d_2 \rightarrow e_M$ (규칙5)

$b_1c_1d_2 \rightarrow e_L$ (규칙 20)

⇒ 비 일관성

속성 b, c 를 삭제/ d를 삭제할 경우에도

위와 같은 비일관성 발생

따라서, 조건부 속성 집합은 e-독립임을 확인

(그림 16) e-독립임을 확인하기 위한 비일관성 발생 확인

R	a	b	c	d	e
1	-	-	2	2	H
2	1	-	2	2	H
3	-	1	2	2	H
4	-	1	2	2	H
5	-	-	1	2	M
6	-	-	1	2	M
7	2	-	1	2	M
8	-	-	1	2	M
9	-	2	1	2	M
10	-	-	-	1	L
11	-	-	-	1	L
12	-	-	-	1	L
13	-	-	-	1	L
14	-	-	-	1	L
15	-	-	-	1	L
16	-	-	-	1	L
17	-	-	-	1	L
18	2	2	2	-	L
19	-	-	-	-	L
20	3	1	1	-	L
21	-	-	-	1	L
22	-	-	-	1	L
23	-	-	-	-	L
24	3	2	2	-	L

의사결정규칙에 대한 코어 계산 => 값코어가 무엇인지

$a_1b_1c_2d_2 \rightarrow e_H$ 에서 c_2 와 d_2 가 코어

$b_1c_2d_2 \rightarrow e_H$

$a_1c_2d_2 \rightarrow e_H$

참인 반면 다음 규칙

$a_1b_1d_2 \rightarrow e_H$

$a_1b_1c_2 \rightarrow e_H$

은 거짓

각 의사결정규칙의 리덕트

전체 규칙이 참이 되도록 하는 조건부 속성값을 추가

$a_1b_1c_2d_2 \rightarrow e_H$ 의 경우

$a_1c_2d_2 \rightarrow e_H$ 과 $b_1c_2d_2 \rightarrow e_H$ 2개의 리덕트를 갖는다.

의사결정규칙에 대한 모든 코어값을 확인하여 e-리덕트값을 구한다.

(그림 17) 리덕트를 구하기 위한 코어 확인 과정

R	a	b	c	d	e
1	x	1	2	2	H
1'	1	x	2	2	H
2	1	x	2	2	H
3	x	1	2	2	H
4	x	1	2	2	H
5	1	x	1	2	M
6	1	x	1	2	M
6'	x	2	1	2	M
7	2	-	1	2	M
8	-	-	1	2	M
8'	x	2	1	2	M
9	x	2	1	2	M
10	x	x	x	1	L
11	x	x	x	1	L
12	x	x	x	1	L
13	x	x	x	1	L
14	x	x	x	1	L
15	x	x	x	1	L
16	x	x	x	1	L

R	a	b	c	d	e
17	x	x	x	1	L
17'	2	2	2	x	L
18	2	2	2	x	L
19	3	1	1	x	L
19'	x	x	x	1	L
20	3	1	1	x	L
21	x	x	x	1	L
22	x	x	x	1	L
23	x	x	x	1	L
23'	3	2	2	x	L
24	3	2	2	x	L

의사결정규칙 1, 6, 8, 17, 19, 23은 두개의 리덕트를 갖으며 나머지는 1개의 리덕트를 갖는다.

최소 의사결정 알고리즘을 찾기 위해 불필요한 의사결정규칙을 제거

=> 동치류에 해당하는 규칙을 제거

: 1과 1' 규칙 2, 3, 4 동일 제거

: 규칙 5와 6일 동일 규칙 6', 8', 9와 7, 8도 동일 제거

(그림 18) 의사 결정 규칙에 대한 리덕트

R	a	b	c	d	e
1', 2	1	x	2	2	H
1, 3, 4	x	1	2	2	H
5, 6	1	x	1	2	M
7, 8	2	x	1	2	M
6', 8', 9	x	2	1	2	M
10-17, 19', 21-23	x	x	x	1	L
17', 18	2	2	2	x	L
19, 20	3	1	1	x	L
23', 24	3	2	2	x	L



R	a	b	c	d	e
1	1	x	2	2	H
2	x	1	2	2	H
3	1	x	1	2	M
4	2	x	1	2	M
5	x	2	1	2	M
6	x	x	x	1	L
7	2	2	2	x	L
8	3	1	1	x	L
9	3	2	2	x	L

x: 속성의 don't care 값을 의미

최소 의사결정 알고리즘

$a_1c_2d_2 \rightarrow e_H$

$b_1c_2d_2 \rightarrow e_H$

$a_1c_1d_2 \rightarrow e_M$

$a_2c_1d_2 \rightarrow e_M$

$b_2c_1d_2 \rightarrow e_M$

$d_1 \rightarrow e_L$

$a_2b_2c_2 \rightarrow e_L$

$a_3b_1c_1 \rightarrow e_L$

$a_3b_2c_2 \rightarrow e_L$

(그림 19) 최소 의사결정 알고리즘 도출 과정

다음 (그림 19)는 최소 의사 결정 알고리즘을 생성해내는 과정을 보이고 있다.

```

Decision rule 1. IF(A =1) AND (C = 2) AND (D = 2) THEN (E = H);
Decision rule 2. IF(B =1) AND (C = 2) AND (D = 2) THEN (E = H);
Decision rule 3. IF(A =1) AND (C = 1) AND (D = 2) THEN (E = M);
Decision rule 4. IF(A =2) AND (C = 1) AND (D = 2) THEN (E = M);
Decision rule 5. IF(B =2) AND (C = 1) AND (D = 2) THEN (E = M);
Decision rule 6. IF(D = 1) THEN (E = L);
Decision rule 7. IF(A =2) AND (B = 2) AND (C = 2) THEN (E = L);
Decision rule 8. IF(A =3) AND (B = 1) AND (C = 1) THEN (E = L);
Decision rule 9. IF(A =3) AND (B = 2) AND (C = 2) THEN (E = L);
    
```

(그림 20) 의사 결정 규칙

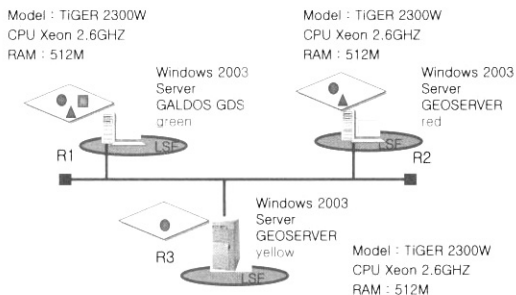
▶ 최적 객체 선정 과정

앞서 의사 결정 데이터 테이블을 분석하여 생성된 규칙을 통해 분류된 지리정보 객체들 중 우선순위가 높은 객체를 선정하여 해당 지리정보 객체가 위치한 컨택 주소를 클라이언트에게 반환한다.

5. 위치 서비스 모델 구현 및 검증

구현 환경은 TiGER 2300W로 CPU 2.6Mhz인 시스템 3대로 운영체제는 Windows 2003 서버로 구성하였으며, 지리정보 객체를 운영하기 위해서 R1 시스템에는 GALDOS의 GDS(Geography Data Server)[21]를 이용하였으며, R2 와 R3 시스템에는 GeoServer[22]를 운영한다.

분산 컴퓨팅 환경에는 CORBA 사양을 따르는 상용화된 미들웨어로 Borland의 VisiBroker 4.1을 사용하였으며, 위치 서비스 모델의 구성요소인 바인딩 서비스와 위치 서비스 그리고 지능형 위치 선정 서비스는 자바(Java2 SDK 1.3.1)로 개발하였다. 서비스 오피와 컨택 레코드 정보를 관리하기 위해 Oracle 9i를 이용하였으며, 각 시스템의 부하정보를 획득하기 위해 LSF 4.1을 이용한다. 위치 서비스 모델의 구성요소인 바인딩 서비스, 위치 서비스 그리고 지능형 선정 서비스는 동일한 시스템에 위치시키고 지리정보 객체는 각 R1, R2, R3 시스템에 배치한다. 또한 시스템의 부하정보를 얻기 위한 LSF 4.1은 각 시스템에 위치하여 부하정보를 모니터링 한다. 각 구성요소의 객체들에 대한 인터페이스는 미들웨어인 VisiBroker4.1을 사용하였다. 또한, 검증을 위한



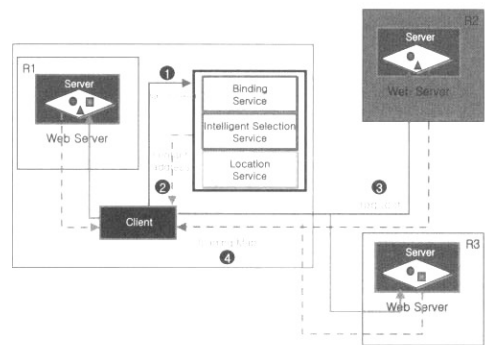
(그림 21) 시스템 구현 환경

그래픽 사용자 인터페이스는 자바의 경량 컴포넌트인 JFC/Swing(Java Foundation Class/Swing)을 이용하여 구현하였다. 위치 서비스 모델의 검증을 위한 수행 시나리오는 (그림 22)와 같다.

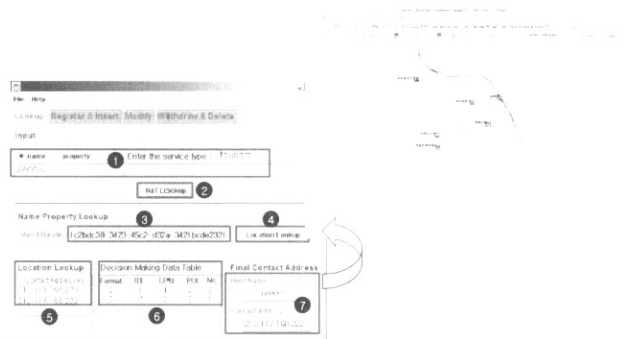
(그림 22)과 같이 수행 시나리오는 클라이언트가 위치 서비스에 지리정보 객체를 요청하고, 최적의 지리정보 객체의 컨택 주소를 받아 R2 시스템 상의 지리정보 객체를 요청하여 이를 클라이언트의 웹 브라우저에 디스플레이 한다.

다음 (그림 23)은 사용자가 Jeonju에서 여행을 할 목적으로 지리정보 객체를 검색하였다.

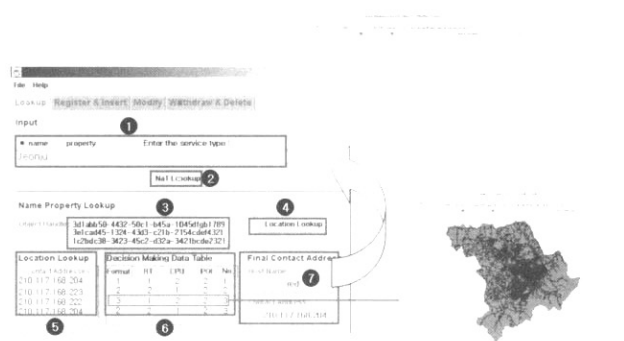
검색할 지리정보 객체의 이름을 'Jeonju'로, 서비스 타입은 'tourism'을 기입하고 [NaT Lookup] 버튼을 클릭한다. 검색한 결과 객체 핸들은 '1c2bdc38-3423- 45c2-d32a-3421bcde2321'이



(그림 22) 수행 시나리오



(그림 23) 이름으로 서비스 타입을 명시하여 검색한 결과 화면 및 서비스 수행 결과 화면



(그림 24) 이름으로 서비스 타입을 명시하지 않고 검색한 결과 화면 및 서비스 수행 결과 화면

며 [Location Lookup] 버튼을 클릭 한 결과, 이에 위치 정보는 객체 핸들에 매핑되는 컨택 주소가 2개를 보이고 있다. 이들로부터 얻은 속성을 통합하여 의사 결정 데이터 테이블을 보이고, 규칙에 적용하여 우선순위가 가장 높은 green의 컨택 주소를 얻어, 해당 서버에게 요청하여, 클라이언트의 웹 브라우저로 결과를 보였다. (그림 24)는 클라이언트가 특정 목적이 없이 전주라는 지리정보를 검색할 경우의 수행과정과 결과화면을 보이고 있다.

6. 결 론

인터넷 기술의 발전에 따라 지리정보 시스템 환경이 중앙 집중형 시스템에서 분산시스템 형태로 변화되고 있으며 특히 웹 기반 서비스로 변화되고 있다.

현재 인터넷 기반의 웹 서비스를 제공하는 지리정보는 사용 목적에 따라 서로 다른 레이어로 구축하고 있으며 특히 벤더들의 고유 포맷으로 공간정보를 구축함으로써 상호 운용하는데 문제점으로 이를 해결하기 위한 표준화 연구를 활발하게 진행 중에 있다. 이와 함께 웹 서비스로 다양한 지리정보를 클라이언트에게 제공되고 있지만, 지리정보 시스템을 구축하는데 가장 고려해야 할 분산 투명성에 관한 연구가 요구되고 있다. 특히, 앞서 언급한 바와 같이 서로 다른 포맷으로 구축한 지리정보이기 때문에 동일한 공간 정보에 해당하는 지리 정보를 중복해서 구축하고 있는 실정이다.

본 논문에서는 이러한 지리정보 시스템 환경에서 네트워크 상에 수많은 지리정보 객체에 대한 관리적인 측면과 클라이언트에게 분산 투명성을 제공하는데 중점을 두었다. 특히, 분산 투명성을 제공하기 위한 위치 투명성과 접근 투명성에 관한 연구로 전체 시스템 성능에 중점을 둔 중복(replication)에 대한 연구가 활발하게 진행되고 있다. 중복 객체를 관리하기 위해 OMG와 GLOBE 그리고 GRID 컴퓨팅에서 관련 프레임워크를 제안하고 있다.

그러나 중복된 서비스 객체를 식별하는 문제와 중복된 서비스 객체들 중에 클라이언트에게 가장 적합한 복사본을 선정하기 위한 능동적인 전략이 필요하다. 특히 분산 컴퓨팅 환경에서는 수많은 서비스 객체들이 이름이나 속성에 의해 중복되어 있다.

따라서, 본 연구는 이름 또는 속성으로 중복된 서비스 객체들의 효율적인 관리와 중복 서비스 객체들 중 가장 적합한 지리정보 객체를 선정하기 위한 위치 서비스 모델을 제시하였다. 이를 위해 이름과 속성 모두를 지원하는 바인딩 서비스와 실제 서비스를 제공하는 객체의 컨택 주소를 관리하는 위치 서비스 그리고 이들 서비스로부터 얻어진 정적인 속성정보와 동적인 속성정보를 포함하여 최적의 지리정보 객체를 선정하기 위해 패턴을 분석하여 규칙을 생성하고 이를 적용하였다.

본 논문에서 제시한 위치 서비스 모델에 대한 검증은 위해 서버 시스템 상에 중복된 지리정보 객체를 위치시키고, 이러한 지리정보 객체를 위치 서비스의 검색 오퍼레이션에 대한 수행 결과 화면을 보였다. 이를 위해 구현 환경은 서버 시스템 3대와 시스템의 동적 속성 정보를 추출하기 위해

LSF 도구를 사용하였다. 개발 언어는 자바(Java SDK 1.3.1)를 이용하였으며, 미들웨어는 VisiBroker 4.1을 사용 하였다. 수행 검증 윈도우는 JFC/Swing을 이용하여 구현하였다.

향후 연구로는 논문에서 제시한 위치 서비스를 OGC에서 제안하고 있는 OWS(OGC Web Service) 표준화에 적용시키기 위한 연구가 필요하다. 특히, 지리정보 객체의 관리적인 측면에서 메타데이터 또는 카탈로그 서비스에 적용 가능하다. 그리고 본 모델의 검증을 위해 헬스케어 정보 시스템 분야에 적용시키고자 한다.

참 고 문 헌

- [1] Preston, M., Clayton, P., and Wells, G. "Dynamic Run-time Application Development Using CORBA Objects and XML in The Field of Distributed GIS," International Journal of Geographical Information Science, Vol.17, No.4, pp.321-341. 2003.
- [2] Tsou, M.H. and Buttenfield, B.P., "A Dynamic Architecture for Distributed Geographic Information Services," Transactions in GIS. 6(4), pp.335-381, 2002.
- [3] Coddington, P.D., Hawick, K.A., Kerry, K.E., Mathew, J.A., Silis, D.L., Webb, P.J., Whitbread, C.G., Irving, M.W., Grigg, R., and Jana, K.T. "Implementation of a Geospatial Imagery Digital Library using Java and CORBA," Proc of Technologies of Object-Oriented Languages and Systems (TOOLS) Asia'98, Beijing, China, pp.280-290, 1998.
- [4] Goddard, S., Harms, S., Reichenbach, S., Tadesse T., and Waltman, W., "Geospatial Decision Support for Drought Risk Management," Communication of the ACM, Vol. 46, No.1, pp.35-37, 2003.
- [5] Xueming Wu, Shifeng Zhang and Steve Goddard, "Development of a Component-based GIS using GRASS," Proceedings of the FOSS/GRASS Users Conference - Bangkok, Thailand, 12-14 September, 2004.
- [6] OGC(OpenGIS Consortium), <http://www.opengeospatial.org/initiatives/?iid=65>.
- [7] GML 3.1.1, http://portal.opengeospatial.org/files/?artifact_id=4700.
- [8] Lawrence A. West Jr., Traci J. Hess. "Metadata as a knowledge management tool: supporting intelligent agent and end user access to spatial data. Decision Support Systems," 32, pp.247-264, 2002.
- [9] Massimo Dragan, Enrico Feoli, Michele Ferneti, etc.. "Application of a spatial decision support system (SDSS) to reduce soil erosion in northern Ethiopia," Environmental Modelling & Software 18, pp.861-868, 2003.
- [10] Z. Pawlak, "Rough sets," Int. J. Inform. Computer Science., vol.11, no.5, pp.341-356, 1982.
- [11] M. O. Killijian, J. C. Fabre, J. C. Ruiz-Garcia and S. Chiba, "A meta object protocol for fault-tolerant CORBA applications," Proceedings of the IEEE 17th Symposium on Reliable Distributed Systems, West Lafayette, IN (October 1998), pp.127-134.
- [12] Louise E. Moser, P. M. Melliar-Smith, Priya Narasimhan: A Fault Tolerance Framework for CORBA. FTCS pp. 150-157, 1999.

[13] G. Ballintijn, M. van Steen, A.S. Tanenbaum. "Scalable Naming in Global Middleware," Proc. 13th Int'l Conf. on Parallel and Distributed Computing Systems (PDCS-2000), Las Vegas, pp.624-631, August 8-10, 2000.

[14] I. Kuz, M. van Steen, H.J. Sips, "The Globe Infrastructure Directory Service," Proc. 7th Annual ASCI Conference, pp.115-122, May, 2001.

[15] K. Ranganathan and I. Foster. "Design and Evaluation of Dynamic Replication Strategies for High Performance Data Grids." Proceedings of International Conference on Computing in High Energy and Nuclear Physics, Beijing, China, September, 2001.

[16] S. Vazhkudai, S. Tuecke, I. Foster. "Replica Selection in the Globus Data Grid," Proceedings of the First IEEE/ACM International Conference on Cluster Computing and the Grid (CCGRID 2001), pp.106-113, IEEE Computer Society Press, May, 2001.

[17] Server Load Balancing. TechBrief from Extreme Networks.

[18] Rabinovich, M., Xiao, Z., and Aggarwal, A. "Computing on the Edge: A Platform for Replicating Internet Applications." In Proceedings of the 8th International Workshop on Web Content Caching and Distribution, (Hawthorne, New York, September 29 through October 1, 2003).

[19] Fu, Z. and Venkatasubramanian, N. "Combined Path and Server Selection in Dynamic Multimedia Environments," In Proceedings of the 7th ACM International Conference on Multimedia(Part 1). (Orlando, Florida, 1999). ACM pp.469-472.

[20] Guo, M. Ammar, M. Zegura, E., "Selecting among Replicated Batching Video-on-Demand Servers," In Proceedings of the 12th International Workshop on Network and Operating System Support for Digital Audio and Video. (Miami, Florida, May 12-14, 2002).

[21] GALDOS In., <http://www.galdosinc.com/>.

[22] Open Gateway for Geospatial Data, <http://docs.codehaus.org/display/GEOS/Home>.

[23] PostGIS : Improving Performance, <http://postgis.refrains.net/piermail/postgis-user/2004-February/004032.html>.

[24] UWE MEYER, Dortmund, "Managing Large Imagery Databases via the Web" <http://ifp.uni-stuttgart.de/publications/phowo01/Meyer.pdf>.

[25] Mike Tsykin, "On Web Quality of Service : Approaches to Measurement of End-TO-END Response Time," 2001.

[26] 정창원, 오성권, 주수중, "광역 객체 컴퓨팅 환경에서 이름/속성 기반의 통합 바인딩 서비스 방안", 한국정보처리학회 제 9-A권 2호, pp.241-248, 2002. 6.

[27] 정창원, 오성권, 주수중, "광역 객체 컴퓨팅 환경에서 부하를 고려한 통합 바인딩 서비스의 설계 및 구현", 한국정보과학회 논문지 제30권 3호, pp.293-306, 2003. 6.

[28] Chang-Won Jeong, Su-Chong Joo, and Sung-Kook Han, "Integrated Binding Service Model for Supporting Both Naming/Trading and Location Services in Inter/Intra-Net Environments", Lecture Notes in Computer Science, Vol. 3032, pp.754-761, 2003. 12.

정 창 원



e-mail : medibblue@wonkwang.ac.kr
 1993년 원광대학교 컴퓨터공학과(학사)
 1998년 원광대학교 컴퓨터공학과 졸업
 (석사)
 2003년 원광대학교 컴퓨터공학과 졸업
 (공학박사)

2004년~2006년 전북대학교 차세대 LBS 응용 연구센터
 연구교수

2006년~현재 원광대학교 전기전자 및 정보공학부 Post-Doc.
 관심분야: 분산객체 컴퓨팅, 멀티미디어 데이터베이스, LBS,
 텔레메틱스, 헬스케어

이 원 중



e-mail : wjlee@kunjang.ac.kr
 1991년 2월 전북대학교 전산통계학과(학사)
 1998년 2월 원광대학교 교육대학원
 컴퓨터공학과(석사)
 2005년 2월 원광대학교 일반대학원
 컴퓨터공학과(박사)

2001~현재 군장대학 사회실무학부 교수

관심분야: 분산객체 컴퓨팅, 지리정보시스템, 멀티데이터베이스

이 재 완



e-mail : mrlee@chonbuk.ac.kr
 1894년 중앙대학교 전자계산학과
 (이학사)
 1987년 중앙대학교 대학원
 전자계산학과(이학석사)
 1992년 중앙대학교 대학원
 컴퓨터공학과(공학박사)

1996년~1997년 한국학술진흥재단 전문위원

2002년 Florida Atlantic University 교환교수

1992년~현재 : 군산대학교 전자정보공학부 교수

관심분야: 분산시스템, 운영체제, 데이터마이닝 등

주 수 중



e-mail : scjoo@wonkwang.ac.kr
 1986년 원광대학교 전자계산공학과(학사)
 1988년 중앙대학교 컴퓨터공학과(공학석사)
 1992년 중앙대학교 컴퓨터공학과(공학박사)
 1993년 미국 University of Massachusetts
 at Amherst, Post-Doc.

2003년 미국 University of California at Irvine, Visiting Professor.

1990년~현재 원광대학교 전기전자 및 정보 공학부 교수

관심분야: 분산 실시간 컴퓨팅, 분산객체모델, 시스템 최적화,
 멀티미디어 데이터베이스