

그리드 컴퓨팅을 위한 온톨로지 기반의 시맨틱 정보 시스템[☆]

Ontology Based Semantic Information System for Grid Computing

한 병 전* 김 형 래** 정 창 성***
Han Byong-John Kim Hyung-Lae Jeong Chang-Sung

요 약

그리드 컴퓨팅 환경은 기존의 분산 컴퓨팅 기술의 확장된 개념으로, 다양한 분야에서 저비용의 고성능 컴퓨팅 퍼포먼스를 얻기 위하여 그리드를 이용하고 있다. 그러나 공통적으로 이용할 수 있는 정보 시스템의 부재로, 현재의 그리드는 대규모의 단일한 환경으로서의 그리드가 아닌, 기존의 클러스터 컴퓨터와 같은 의미로 사용되고 있다. 따라서 자신이 구성한 분산 컴퓨팅 환경을 그리드 환경의 한 부분으로 포함시키기 위해서는 컴퓨팅 환경에 대한 정보를 그리드의 이용자가 공유할 수 있어야 하며, 정보 서비스를 공개하여 공유할 수 있도록 해야 한다. 따라서 본 논문에서는 정보의 공유와 확장이 용이하며, 플랫폼에 독립적인 서비스를 제공할 수 있는 정보 시스템을 구축하기 위한 웹 서비스, 온톨로지 기반의 Semantic Information System Framework(WebSIS)을 제안한다. WebSIS를 통하여 온톨로지를 이용하는 정보 시스템의 제작을 용이하게 하며, 온톨로지의 공유와 재사용, 확장의 특징을 활용할 수 있는 구조를 제안한다. 또, 그리드 컴퓨팅 환경이 기존의 분산 컴퓨팅 환경에 비해 복잡해짐에 따라 작업을 수행하기 위한 적절한 자원의 할당에도 좀 더 많은 고려사항이 존재하게 되었고, 이를 효과적으로 처리하기 위해서는 좀 더 고수준의 정보처리를 필요로 하게 되었다. 이에 우리는 WebSIS와 함께 WebSIS를 이용하여 그리드 환경을 이용하는 작업의 자동화를 위한, 온톨로지 기반의 고수준의 정보처리가 가능한 Resource Information Service를 구현하였고, 이를 통하여, 효율적인 공유와 확장을 위한 온톨로지 모델을 제안한다. 또한, 기존의 용어 매칭이 아닌 의미 정보를 이용한 검색과 정보 처리를 제안한다.

Abstract

Grid computing is an expanded technology of distributed computing technology to use low-cost and high-performance computing power in various fields. Although the purpose of Grid computing focuses on large-scale resource sharing, innovative applications, and in some case, high-performance orientation, it has been used as conventional distributed computing environment like clustered computer until now because Grid middleware does not have common sharable information system. In order to use Grid computing environment efficiently which consists of various Grid middlewares, it is necessary to have application-independent information system which can share information description and services, and expand them easily. Thus, in this paper, we propose a semantic information system framework based on web services and ontology for Grid computing environment, called WebSIS. It makes application and middleware developer easy to build sharable and extensible information system which is easy to share information description and can provide ontology based platform-independent information services. We present efficient ontology based information system architecture through WebSIS. Discovering appropriate resource for task execution on Grid needs more high-level information processing because Grid computing environment is more complex than other traditional distributed computing environments and has various considerations which are needed for Grid task execution. Thus, we design and implement resource information system and services by using WebSIS which enables high-level information processing by ontology reasoning and semantic-matching, for automation of task execution on Grid.

☞ keyword: Grid middleware, Web service, Ontology, Semantic, 그리드 미들웨어, 웹 서비스, 온톨로지, 시맨틱

-
- * 정 회 원 : 고려대학교 전자전기공학과 연구원(박사)
guru1013@korea.ac.kr
** 정 회 원 : 고려대학교 전자전기공학과 연구원(박사)
nolight@korea.ac.kr
*** 정 회 원 : 고려대학교 전자공학과 정교수
csjeong@korea.ac.kr

[2009/01/07 투고 - 2009/01/24 심사 - 2009/02/09 심사완료]
☆ 이 논문은 서울시 연구 비즈니스 개발 프로그램, 정보통신 연구진흥원 ITRC 지원 프로그램, 고려대학교 BK21사업단 지원에 의한 연구이었습니다(This work was supported by the Seoul Research and Business Development Program of Seoul, Korea, the ITRC support program of IITA, a Brain Korea 21 project and a grant of Korea University)

1. 서론

그리드 컴퓨팅 환경[1]은 방대한 분산 컴퓨팅 리소스를 조직, 운용하여 사용자에게 저비용으로 고성능의 컴퓨팅 파워를 제공하는 환경이다[2, 29]. 인터넷과 마찬가지로 그리드 컴퓨팅은 상용이 아닌 과학 프로젝트들을 지원하기 위한 목적으로 개발되어 진화하고 있으며, 현재 e-Science, bioinformatics, astrophysics, Earthquake engineering, weather forecast 등의 다양한 분야에서 이러한 그리드 컴퓨팅 환경의 고성능 컴퓨팅 파워를 이용하기 위한 연구들이 진행 중이다. 기존의 분산 컴퓨팅 환경과는 달리 그리드 컴퓨팅 환경은 보다 큰 규모의 컴퓨팅 환경을 구축하고, 다양한 분야에서 단일한 대규모의 환경을 공유하여 작업을 수행하는 환경이다. 그러나 각 분야의 어플리케이션은 필요로 하는 정보의 종류도 다르고, 이용하는 시스템의 종류도 다르기 때문에, 기존의 글로벌버스[3]와 같은 그리드 미들웨어는 각 분야에서 공통적으로 필요로 하는 정보, 예를 들어, 리소스의 현재 상태나 명세에 관한 정보만을 제공하였고, 그리드 환경을 이용하는 응용프로그램 개발자는 추가적으로 필요한 정보를 어플리케이션 레벨에서 직접 구현하여 미들웨어의 정보 서비스와 함께 이용하였다. 따라서 응용프로그램 개발자는 미들웨어에서 제공하는 정보 외에 필요한 정보를 독자적으로 정의하여 사용하였고, 이로 인하여 어플리케이션들 사이에서는 구축된 정보를 공유하거나 재사용하기 힘든 구조로 되어 있었다. 따라서 새로운 응용프로그램을 개발하거나, 기존의 응용프로그램의 기능을 확장하고자 할 때, 필요한 정보 시스템을 구축하기 위하여 많은 시간과 비용이 발생하게 되었다.

Global Grid Forum(GGF)[4]에서는 분산된 이질적이고 동적 특성을 갖는 가상조직(Virtual Organizations)에서 서비스들의 통합과 관리가 요구되는 시스템과 응용프로그램을 위한 기반으로 그리드 환경의 표준 규약이 되는 Open Grid 서비

스 구조(OGSA)[5]를 제안하였고, 변화 및 발전되는 컴퓨팅 환경에 맞추어 개선되고 있다. 또한, 웹 서비스(Web Services)[6]와 그리드 기술을 토대로, OGSA는 핵심 그리드 서비스를 정의하고 그리드 환경에 필수적인 여러 가지 기본적인 그리드 서비스들을 정의한 Open Grid 서비스 Infrastructure(OGSI)[27]를 제안하고 있다. 이러한 표준 규약을 만족하는 어떠한 미들웨어도 그리드 환경을 구성할 수 있으며, 궁극적으로는 다양한 미들웨어로 단일한 형태의 그리드 환경을 구성하는 것이 그리드의 목적이다.

현재의 그리드는 가장 널리 이용되고 있는 Globus Toolkit을 포함하여, Legion[7]의 상업용 버전인 Avaki, Condor[8, 30]등의 다양한 미들웨어로 구성되고 있으며, myGrid[9], GridLab[10], Unicore[11]등의 새로운 그리드 Infrastructure가 연구 진행 중에 있다. 그리드 환경을 이용하는 응용프로그램이 이처럼 다양한 미들웨어로 구성된 그리드 환경을 이용하기 위해서는 각 미들웨어에서 제공되는 정보 서비스를 공통적으로 이용할 수 있어야 하며, 효율적인 작업 자동화를 위해서는 다양한 형태의 정보 서비스를 필요로 한다. 하지만, 현실적으로 각 미들웨어의 서비스를 이용하기 위해서는 각 미들웨어의 정보 서비스를 이용할 수 있는 또 다른 정보시스템을 응용프로그램 레벨에 요구하며, 이 역시도 해당 미들웨어에서 제공하는 정보 서비스만 이용할 수 있다는 한계를 가지고 있기 때문에, 다양한 미들웨어의 정보를 공유하기에는 많은 어려움이 따른다.

따라서 본 논문에서는 사용자가 필요로 하는 다양한 정보서비스를 쉽게 구축할 수 있고, 다양한 분야의 어플리케이션에서 정보와 서비스를 공유할 수 있는 웹 서비스에 기반한 시맨틱 정보 시스템 프레임워크 그리고 그리드 컴퓨팅을 위한 온톨로지 (이하 WebSIS)을 제안한다. WebSIS는 크게 Knowledge Base, 시맨틱 정보 시스템 프레임워크, 시맨틱 정보 서비스의 3단계로 구성되어, 그리드 상의 정보와 서비스의 공유 및 확장에 유

연한 구조를 목표로 디자인 되었다. 다양한 응용 프로그램과 미들웨어로 구성된 단일한 환경인 그리드를 효율적으로 이용하기 위해서 모든 응용 프로그램과 미들웨어에서 정보와 서비스를 공유할 수 있는 독립적인 정보 시스템이 필요하다. 이를 위해서 각 시스템이 자신의 정보 서비스를 위하여 자체적인 명세 언어를 가지고 작성하던 정보 명세를 World Wide Web Consortium(W3C) 표준의 Ontology Web Language(OWL)[13]를 이용하여 정의 하였다. 또한, 작성된 정보 서비스를 모든 응용 프로그램이나 미들웨어에서 공통적으로 사용하기 위하여 웹 서비스 인터페이스로 제공된다. WebSIS는 이러한 온톨로지 기반의 정보 시스템을 쉽게 구축할 수 있도록 도와주는 프레임워크이며, WebSIS를 이용하여 개발자는 자신이 원하는 서비스를 쉽게 제작하고 공유할 수 있다. 또한, 기존에 정의된 정보 명세를 이용할 수 있고, 자신이 원하는 정보를 추가하여 새로운 정보 서비스로의 확장도 용이하다.

본 논문은 다음과 같이 기술된다. 먼저 2절에서는 본 연구와 관련된 기술 및 연구에 대하여 설명하고, 3절에서는 WebSIS의 구조에 대하여 설명한다. 4절에서는 WebSIS를 이용한 그리드 환경에서의 리소스할당을 위한 리소스 정보 서비스의 구현에 관하여 설명하고, 5절에서는 WebSIS의 구현과 평가에 대하여 서술한다. 마지막으로 6절의 결론으로 본 논문을 끝맺는다.

2. 연구 동향

2.1 그리드 컴퓨팅

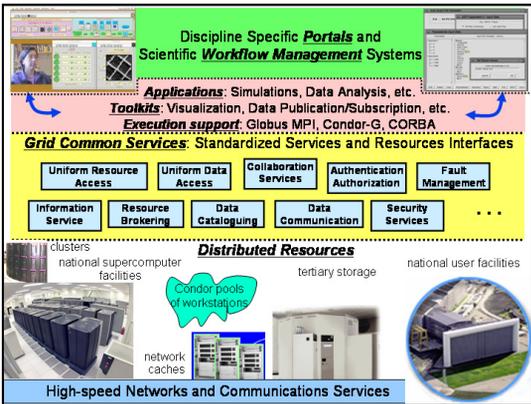
그리드 컴퓨팅은 최근 활발히 연구가 진행되고 있는 분산 병렬 컴퓨팅의 한 분야로서, 네트워크로 연결된 다수의 이 기종의 컴퓨팅 리소스들을 묶어, 가상의 대용량 고성능 컴퓨팅 환경을 구성하여, 고도의 연산 (computation intensive jobs) 혹은 대용량 연산 (data intensive jobs)을 수행하는 것이다. 기존 분산 병렬 컴퓨팅 기술인 클러스터

컴퓨터와는 달리 네트워크 상에서 서로 다른 기종의 머신들을 연결한다는 점으로 인하여, 클러스터 컴퓨터에서는 고려되지 않았던 여러가지 표준 규약들이 필요해 졌고, 현재 글로벌스 프로젝트를 중심으로 표준들이 정립되고 있다.

현재 Globus[1]를 포함하여 Legion[7]의 commercial version인 Avaki, condor [8, 30]등의 다양한 미들웨어가 그리드 컴퓨팅 환경의 구축을 위하여 이용되고 있으며, myGrid[9], GridLab[10], Unicore[11]등의 새로운 Grid infrastructure가 연구 진행 중이다. 이러한 다양한 그리드 미들웨어 중, Globus가 가장 널리 사용되고 있으며, 현재 진보된 Web Service Functionality와 WSRF(Web Service Resource Framework)[14]를 추가하여, 보다 향상된 Service-oriented 구조[15]의 Globus Toolkit version 4 (GT4)[3]가 공개 되었다. Globus는 그림 2-1에서 보는 것과 같이 OGSA에 제시된 그리드 환경에 필요한 서비스의 표준안을 충족하기 위하여 보안 서비스에 Grid Security Infrastructure(GSI), 리소스의 관리와 원격 작업수행을 위한 GRAM(Grid Resource Allocation Manager), 그리드 내에서 데이터의 관리와 전송을 위한 GridFTP(Grid File Transfer Protocol), 정보 서비스를 위한 MDS(Meta Directory Service), remote secondary storage의 연결을 위한 GASS(Grid Access to Secondary Storage)등 그리드 환경의 이용에 필요로 하는 다양한 기능을 제공한다[3, 4, 5, 27, 28].

그리드 컴퓨팅 환경은 지역적으로 넓게 분포된 다수의 이 기종의 컴퓨팅 환경으로 구성되어 있고, 각각의 컴퓨팅 리소스는 각기 다른 플랫폼과 응용프로그램을 가지고 있기 때문에, 사용자가 자신이 수행하고자 하는 작업을 수행할 수 있는 리소스를 찾기 위해서는 그리드 환경에 존재하는 다양한 정보를 이용할 필요가 있다. 그러나 사용자는 동적으로 변화하는 그리드 환경의 변화를 모두 관리할 수 없기 때문에, 이러한 정보들을 그리드 환경을 구성하는 미들웨어로부터 얻어야 한다.

현재 릴리즈 되고있는 GT4에서는 MDS4[16]를 통하여 사용자가 필요로 하는 정보 서비스를 제공한다. 그러나 Grid환경은 기존의 분산 컴퓨팅 환경과는 달리 다양한 분야의 다양한 목적을 위하여 사용되는 작업 수행 환경이기 때문에 MDS4에서는 모든 시스템에서 공통적으로 필요로 하는 정보만을 제공하며, 특정 도메인에 한정적으로 필요한 정보의 제공은 할 수 없다. 이 때문에, 글로벌 버스를 이용하는 많은 응용프로그램에서는 추가적으로 필요한 정보를 얻기 위하여, 자체적으로 새로운 정보시스템을 개발하고, MDS와 함께 이용하는 형태로 개발되었다. 이는 그리드 환경의 이용을 어렵게 만들고, 동일한 환경에서 같은 역할을 하는 서비스가 중복되는 등의 비용의 증가를 야기한다. 따라서, 우리는 이 기존의 컴퓨팅 리소스 사이에서 좀 더 편리하게 필요한 정보를 위한 정보 시스템을 구축하고, 작성된 서비스를 여러 시스템에서 공유하여 사용할 수 있는 정보 시스템 프레임워크가 필요하다. 그림 2-1은 어플리케이션과 그리드 서비스(미들웨어), 분산된 자원간의 레이어 구조를 나타낸다.



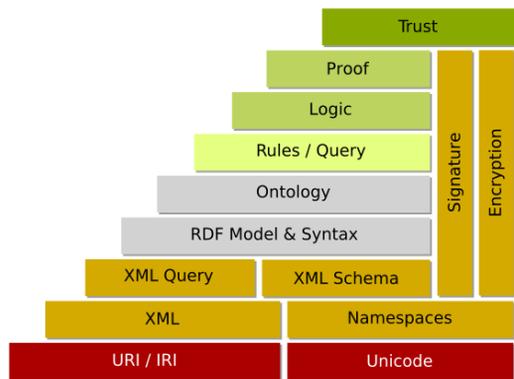
(그림 2-1) 그리드의 구조

2.2 시맨틱 웹 기술

시맨틱 웹 기술[12]은 현재의 인터넷과 같은 분산환경에서 리소스(웹 문서, 각종 파일, 서비스

등)에 대한 정보와 리소스 사이의 관계와 의미 정보(시맨틱s)를 컴퓨터가 처리할 수 있는 온톨로지의 형태로 표현하고, 이를 자동적으로 처리하도록 하는 기술이다. 웹의 창시자인 팀 버너스 리에 의하여 제안되었고, W3C에 의하여 표준화 작업이 진행 중이다[17].

기존의 HTML이 단순히 사람이 시각적으로 볼 수 있는 것을 표현하는 것에 목적을 두고 설계된 것과는 달리, 시맨틱 마크업 언어는 XML을 기반으로 트리플 구조를 이용하여 대상의 의미 정보를 함께 표현하여, 컴퓨터도 정보의 의미를 파악하고 이용할 수 있도록 설계된 언어이다. 따라서, 컴퓨터는 기존의 검색이 특정한 구문의 일치를 바탕으로 하는 방식이 아닌, 대상의 의미를 분석하고, 그 의미를 바탕으로 검색하는 시맨틱 매칭이 가능하게 된다. 그림 2-2는 이러한 시맨틱 웹의 구성을 나타낸다.



(그림 2-2) W3C 시맨틱 웹 레이어

2.2.1 온톨로지

현재 시맨틱 웹 온톨로지를 기술하는 표준 언어로 W3C에서 제안한 리소스 기술 구조 Resource Description Framework(RDF)[18], Ontology Web Language(OWL)[13] 그리고 ISO에서 제안한 TopicMaps[19] 등이 있으나 국지적으로는 N3[20], N-Triple[21] 또는 인공지능 분야에서 예전부터 쓰이던 KIF[22] 형태로도 온톨로지가 작성된다. 본

논문에서는 W3C의 RDF, OWL을 이용하여 온톨로지를 작성하였다.

RDF, OWL은 XML에 기반한 시맨틱 마크업 언어이다. 시맨틱 마크업 언어는 모두 <Subject, Property, Object>의 트리플(Triple) 형태로 개념을 표현한다. 온톨로지는 이러한 트리플 구조를 기반으로 하는 그래프 형태로 이루어진다[23].

OWL은 Class와 Property로 구성되며, 추가적으로 XML에 정의된 primitive data type의 사용도 가능하다. Class와 property, primitive data type은 고유성을 위하여 모두 네임스페이스를 포함하는 URI의 형태로 표현되며, 이 들을 트리플 형태로 표현하여, class 사이의 관계를 표현하여 의미 정보를 담는다. 또, property에 부여되는 axiom을 통하여 property의 특성이나 다른 property와의 관계도 정의할 수 있다.

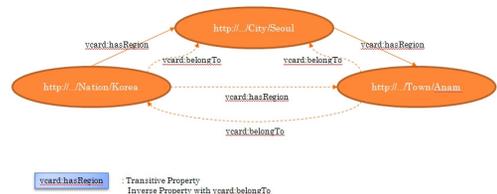
이렇게 구성된 온톨로지의 각 트리플과 property axiom으로 컴퓨터는 추론기(Reasoner)를 이용하여 온톨로지에 정의되지 않은 새로운 트리플을 추가적으로 알아낼 수 있다. 이 추론 기능은 온톨로지에 정의된 정보를 보다 가치 있고, 풍부하게 사용할 수 있게 해준다. 이렇게 OWL의 Class와 property로 정의된 정보 명세를 온톨로지 스키마라 한다. 온톨로지 스키마는 각 도메인에 대하여, 정의된 클래스에 대한 정의나 다른 클래스와의 관계 등 그 도메인의 지식을 가지고 있는 부분이며, 각 클래스로부터 생성된 Individual이 실질적인 데이터를 나타낸다. 클래스의 인스턴스인 Individual은 온톨로지의 각 클래스에 대한 실질적인 데이터 값만을 가지고 있고, 다른 클래스와의 관계나 의미정보 등은 가지고 있지 않다. 따라서, 온톨로지를 이용하는 소프트웨어 에이전트는 Individual에 해당하는 온톨로지 스키마를 분석하여, Individual의 데이터가 갖는 의미를 분석할 수 있으며, 이때, 의미를 분석하는 도구가 추론기이다. 따라서, 온톨로지 스키마는 해당 도메인에 대하여 컴퓨터가 이해할 수 있는 지식이라고 할 수 있다.

작성된 온톨로지는 웹 상에 공개 되고, 다른 사람들과 공유될 수 있다. 온톨로지는 특정 분야의 개념을 정의해 놓은 컴퓨터가 이해할 수 있는 지식이다. 같은 지식을 가지고 있는 컴퓨터 사이에는 효과적인 정보의 교류가 가능하다. 기존의 연구에서 XML기반의 독자적인 명세 언어를 제작하여 사용하던 때와는 달리, 여러 분야에서 널리 사용되고 있는 OWL로 정의된 온톨로지는 서로 공유와 재사용이 가능하며, 다른 사람이 정의해 놓은 온톨로지를 확장하여 이용할 수 도 있다.

온톨로지는 잘 정의된 표준의 메타데이터 언어이다. 정의된 온톨로지는 그 자체로 메타데이터로 이용이 가능하다. 이에 추가적으로 온톨로지의 property axiom과 클래스 사이의 관계를 통하여, 추론이 가능하며, 추론을 통해 정의한 정보 이상의 것을 얻을 수 있는 강력한 언어이다.

2.2.2 추론

온톨로지의 추론 기능은 다양한 목적으로 사용될 수 있다. 기본적으로 온톨로지서 추론은 그림 2-2의 시맨틱 Web 레이어에서 Rules과 Logic으로 표현된다. Rules은 온톨로지 property axiom과 해당 axiom에 대한 폴리시를 가지고 표현되며, Rules을 이용하여 추론 논리를 정의한다. 정의된 Logic으로 추론하는 것은 해당 Logic의 정보 처리를 수행할 수 있는 추론기에 의하여 이루어진다.



(그림 2-3) 추론 예제

그림 2-3은 온톨로지의 추론을 통한 정보의 확장에 대한 간단한 예제를 나타낸다. 온톨로지 제작자는 Nation Class의 Korea, City Class의 Seoul,

Town Class의 Anam이라는 individual 사이에 vcard:hasRegion property로 정의하였고(실선), vcard:hasRegion property에 Transitive Property axiom을 부여하고 vcard:belongsTo property와의 Inverse Property axiom을 부여하였다. Transitive Property와 Inverse Property는 OWL에서 제공되는 Logic이며, OWL 추론기를 이용하여, 그림 2-3의 점선에 해당하는 정보를 추가적으로 얻을 수 있다. 따라서 온톨로지의 추론은 기존의 시스템에서 단순한 논리규칙을 정해서 예상된 결과를 매칭하는 것이 아닌, 트리플의 Property에 의미를 부여하고, 해당 Property를 통하여 관계를 정의함으로써, 트리플의 정보를 의미를 바탕으로 확장할 수 있는 기술이다.

OWL은 property axiom, 즉 표현의 정도를 가지고 OWL-Lite, OWL-DL(Description Logic) 그리고 OWL-Full의 3종류의 sub-language로 나뉘어진다 [13]. 각 sub-language는 미리 정의된 axiom을 제공하며, 해당 axiom에 대한 Logic을 수행할 수 있는 추론기를 이용하여 추론한다. 현재 OWL-DL이 가장 널리 쓰이는 온톨로지 명세 언어이며, OWL-DL의 추론이 가능한 추론기로는 Racer[24], Pellet[25] 등이 있다.

Property에 각 sub-language에서 제공하는 다양한 axiom을 부여함으로써, 정보 명세에 의미 부여가 가능하다. 그림 2-3에서 설명한 것과 같이 단순히 트리플을 정의해 놓은 것만으로도 컴퓨터는 그 트리플의 의미를 분석하여 새로운 정보를 추론해 낼 수 있다.

2.3 웹 서비스

웹 서비스(Web services)[6]는 XML기반의 플랫폼 독립적이고 어플리케이션 독립적인 정보 통신을 가능하게 하는 기술이다. 웹 서비스는 Web Service Description Language(WSDL), Universal Description Discovery Integration (UDDI), Simple Object Access Protocol(SOAP)로 구성된다. WSDL은 웹 서비스에 대한 정의를 담고 있는 문서이며,

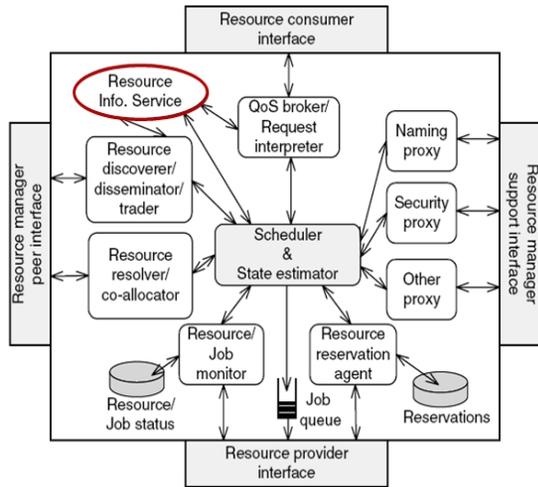
SOAP는 HTTP기반의 통신 프로토콜이다. UDDI는 웹 서비스의 위치에 대한 정보를 가지고 있는 저장소로, 사용자는 UDDI에 SOAP을 통하여 서비스 요청 메시지를 보내고, 필요한 서비스의 위치를 받아 그 서비스의 위치로부터 WSDL을 받고, WSDL을 이용하여 서비스를 호출 할 수 있다. 현재 인터넷과 같은 분산 컴퓨팅 환경이 확산되면서, 많은 시스템들이 웹 서비스를 바탕으로 하는 서비스-oriented 구조를 도입하고 있다. 다수의 이기종의 컴퓨팅 리소스로 구성된 그리드 환경에서 모든 응용프로그램과 미들웨어에서 플랫폼에 관계없이 이용할 수 있게 하기 위하여, 이러한 공통의 인터페이스가 필수적이며, 따라서 웹 서비스를 이용하는 Service-oriented 구조를 도입하였다.

2.4 그리드를 위한 정보 시스템

복잡한 그리드 환경에서 사용자에게 사용 편의성을 제공하기 위하여, 그리드 미들웨어가 개발되었고, 그리드 미들웨어는 대규모의 환경에서도 사용자의 작업에 적합한 리소스를 제공하기 위하여 리소스에 대한 다양한 정보를 제공한다. 다양한 플랫폼과 응용프로그램을 가진 이기종의 리소스들 사이에서 자신의 작업에 적합한 리소스를 선별하는 것은 그리드 환경의 이용에 있어서 무엇보다 중요한 문제이기 때문에, 이를 위한 다양한 형태의 리소스 정보 시스템이 개발되고 있다. 또한, 작업의 자동화를 위해서 응용프로그램은 미들웨어에서 제공하는 정보를 자동적으로 수집하고, 자체적으로 정의하거나 외부의 또 다른 정보 서비스를 이용하는 정보 시스템을 포함하고, 이용한다. 따라서 기존의 정보 시스템이나 정보 서비스는 리소스에 편중된 경향을 보였으며, 그림 2-4는 Abstract한 기존의 리소스 관리 시스템(RMS) 내에서의 정보 서비스의 위치를 나타낸다[29].

지금도 Globus나 Legion과 같은 그리드 미들웨어와 그리드 워크플로우 시스템인 ASKALON의 RMS인 GridARM[30], 온톨로지 기반의 리소스 검색 시스템인 Matchmaker[31]등의 그리드를 이용

하는 응용프로그램들은 자체적으로 정보 서비스를 구현하여 사용하고 있으며, 서비스의 기능도 기존의 리소스 중심적 정보에서 작업 정보를 고려한 리소스 검색 등, 점차적으로 개선 및 발전되어가고 있다.

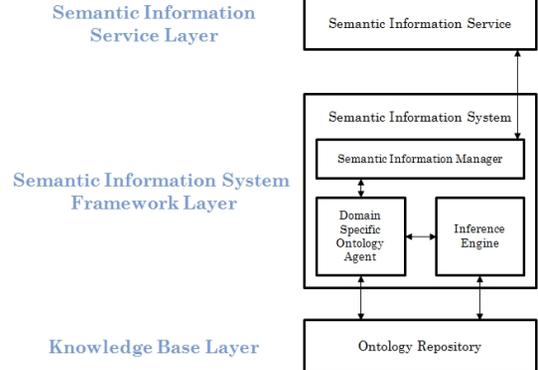


(그림 2-4) 추상적 리소스관리 시스템 구조

3. WEBSIS 구조

3.1 개요

WebSIS는 그리드 환경의 미들웨어나 어플리케이션에서 작업의 자동화를 위해 필요한 정보를 제공하는 온톨로지를 활용한 정보 서비스 구축을 위한 시맨틱 정보 시스템 프레임워크다. 그림 3-1은 WebSIS의 아키텍처를 나타낸다. WebSIS는 크게 시맨틱 정보 서비스 레이어, 시맨틱 정보 시스템 프레임워크 레이어 그리고 Knowledge base 레이어로 나눌 수 있으며, 시맨틱 정보 시스템 레이어는 다시 시맨틱 정보 관리자(SIM), 도메인 특정 온톨로지 에이전트(DSOA), 그리고 추론 엔진(IE)으로 구성된다.



(그림 3-1) 웹시스의 구조

Knowledge base는 그리드 컴퓨팅 환경의 변화에 동적으로 대응할 수 있도록 확장 및 변경에 유연하게 하기 위하여 다수의 독립적인 도메인 온톨로지의 계층적인 구조로 이루어져 있다. 또한, 시스템적으로 Knowledge Base의 변경이나 확장이 기존 서비스에 영향을 미치지 않도록 하기 위하여, 시맨틱 정보 시스템 프레임워크 레이어의 정보 처리를 정보 처리와 추론 과정의 두 단계로 나누어 처리한다. 추론 과정의 경우 온톨로지의 처리를 위한 복잡한 프레임워크를 이용해야 하기 때문에, 개발자에게 온톨로지의 이용에 필요한 기능을 제공하여, 시맨틱 웹 기술에 익숙하지 않은 개발자도 쉽게 정보 시스템을 구축할 수 있도록 하기 위하여 정보처리와 분리 하였다. 분리된 두 단계의 정보처리는 SIM과 IE에 의하여 처리되며, DSOA는 SIM과 IE 간의 상호 작용을 담당한다. DSOA는 각 도메인 온톨로지마다 생성되어, 텍스트 형태로 이루어진 온톨로지 정보를 SIM이 필요로 하는 형태로 변환하여 제공한다. SIM은 자신이 필요한 온톨로지의 DSOA만을 이용하면 되기 때문에, 차후에 새로운 온톨로지와 DSOA가 추가되더라도, 기존 서비스에는 영향을 받지 않는다. 또한, 추가된 온톨로지와 DSOA를 SIM에서 추가하면 기존의 서비스를 보다 확장하거나 성능의 향상을 얻을 수도 있다. 게다가 IE를 이용한 추론을 통하여, 미리 정의해 놓은 Knowledge Base 이

상의 정보를 활용 할 수 있다. 이렇게 SIM에서 제작된 각 정보 처리 함수를 웹 서비스화 하여 외부에 제공하게 된다.

다음 섹션에서는 아키텍처의 각 부분에 대하여 좀 더 세부적으로 설명한다.

3.2 Knowledge Base 레이어

Knowledge Base는 Ontology Repository안에 저장되는 온톨로지 스키마, Individual과 rule로 구성된다. 기본적으로 온톨로지는 XML기반의 텍스트 파일이다. 따라서 온톨로지를 사용하는 소프트웨어 에이전트는 해당하는 텍스트 파일을 파싱하여 이용하게 된다. 온톨로지 스키마는 OWL 언어를 사용하여 정의되고, 웹 상에 공개되어, 사용자들이 공유할 수 있도록 한다. Individual data들은 RDF로 정의되고, 파일의 형태로 저장되거나 Relational Database에 저장될 수 있다. 현재 WebSIS는 파일 형태로 Individual data를 관리한다. WebSIS에서 사용하고 있는 rule은 Jena(Jena 2 Semantic Web Framework)[26]에서 제안하는 rule syntax로 정의하여 파일로 저장하고 관리된다. 온톨로지 스키마가 저장된 웹 상의 저장소와 Individual과 rule을 저장하는 저장소를 포함하여 Ontology Repository라 하고, WebSIS의 Knowledge Base가 된다.

3.3 시맨틱 정보 시스템 프레임워크 레이어

시맨틱 정보 시스템 레이어는 시맨틱 정보 관리자(SIM), 도메인 특정 온톨로지 에이전트(DSOA), Inference Engine(IE)의 3개의 컴포넌트로 구성된다. 시맨틱 정보 시스템 프레임워크 레이어에서는 온톨로지에 정의되어 있는 정보를 이용하여 추론 작업과 정보 처리를 통하여 사용자의 요청에 맞는 정보 서비스를 제공하는 역할을 담당한다.

WebSIS는 온톨로지의 공유, 재사용 성을 극대화하고, 서비스의 확장에 유연하도록 하기 위하여, 시맨틱 정보 시스템 프레임워크 레이어에서 처리하는 역할을 3개의 컴포넌트로 나누어 처리

하도록 디자인 하였다. 정보 처리는 SIM에 의하여 처리되고, 추론 과정은 IE에서 담당한다. DSOA는 각 도메인 온톨로지와 1:1로 매칭되어, SIM과 IE사이에서 메시지를 전달하는 역할을 담당하며, 해당 온톨로지를 관리하고, individual의 생성과 삭제를 수행하는 등의 역할을 담당한다. 이처럼 시맨틱 정보 시스템 프레임워크 레이어는 정보 서비스 개발자에게 온톨로지를 활용한 정보 시스템을 보다 쉽게 구현할 수 있는 기능들을 제공한다. 시맨틱 정보 시스템 프레임워크 레이어의 각 컴포넌트가 제공하는 기능은 다음과 같다.

먼저, 시맨틱 정보 관리자는 시맨틱 정보 서비스에 실질적인 정보 서비스를 제공하는 역할을 한다. 온톨로지 기반의 정보시스템을 적용하고자 하는 개발자는 기존에 자신이 직접 정의한 정보 명세를 사용할 때와 비교하여, 온톨로지를 새로 공부해야 하는 부담을 가지고 있다. 또한, 온톨로지의 추론과 같은 강력한 기능을 사용하기 위해서는 온톨로지의 처리 과정에 대한 이해도 필수적이다. 우리는 이러한 추가적인 시간과 비용을 절약하기 위하여, 온톨로지를 사용에 필요한 기능을 DSOA와 IE으로 제한하였고, 온톨로지의 로딩, 정보 검색, 추론 등의 온톨로지와 밀접한 관계가 있는 기능 등을 제공하도록 하였다. 시스템 개발자는 프레임워크에서 제공하는 기능을 이용하여 정보 관리자를 제작하여, 자신이 원하는 서비스를 제공하는 시맨틱 정보 시스템을 제작할 수 있으며, 정보 관리자를 구현하는 과정에서 외부 시스템의 서비스를 이용할 수도 있다.

DSOA는 Knowledge Base에 정의된 각 도메인 온톨로지에 1:1로 매칭된다. 온톨로지는 정보를 컴퓨터가 이해할 수 있는 언어로 개념을 정의해 놓은 구성 요소의 모음이다. 따라서 온톨로지 내에 있는 구성 요소는 온톨로지 마다 고유한 형태를 가지고 있기 때문에 이를 처리할 수 있는 소프트웨어 에이전트가 필수적으로 필요하다. 이처럼 DSOA는 온톨로지 마다 생성되고, 해당 온톨로지의 individual data를 관리하거나, SIM으로부터 요

칭된 정보를 제공하는 기능을 담당한다. DSOA는 자신의 온톨로지와 Rule을 이용하여, IE에게 추론을 요청할 수 있으며, 이정보를 SIM에게 제공할 수 있다.

IE는 순수하게 온톨로지의 추론기능을 담당한다. 추론을 하기 위해서는 온톨로지 스키마와 온톨로지를 정의한 언어의 rule을 필요로 한다. 본 논문에서, IE는 디스크립션 로직 규칙을 이용하는 OWL-DL 추론기와 보다 강력한 추론을 위하여 사용자가 직접 추론 규칙을 정의할 수 있는 사용자 커스텀 룰을 사용하는 Jena[26]의 유전적 룰 추론기를 함께 사용할 수 있도록 하였다. 따라서 IE는 DSOA를 통하여, 추론을 위한 온톨로지의 정보와 함께, 추론 규칙에 대한 서식을 받아 추론을 수행하고 결과를 DSOA에게 돌려주는 역할을 수행한다.

이처럼 정보를 처리하는 프로세스를 3개의 컴포넌트로 나누어 설계함으로써 우리는 기존의 서비스에 영향을 주지 않고, Knowledge base와 서비스의 확장성에 유연성을 얻었다.

3.4 시맨틱 정보 서비스 레이어

시맨틱 정보 서비스 레이어는 WebSIS가 제공하는 모든 서비스를 포함한다. WebSIS는 SIM에서 제공되는 정보를 웹 서비스의 형태로 제공한다. 웹 서비스 기술은 플랫폼에 독립적인 통신 프로토콜이자 서비스 인터페이스로, 누구나 서비스를 자유롭게 이용할 수 있도록 하기 위하여 개발된 기술이다. WebSIS의 서비스를 웹 서비스화하여 제공함으로써, 그리드 환경의 정보 서비스를 필요로 하는 모든 사용자에게 제약 없이 제공할 수 있다.

4. WebSIS를 사용한 리소스 정보 서비스

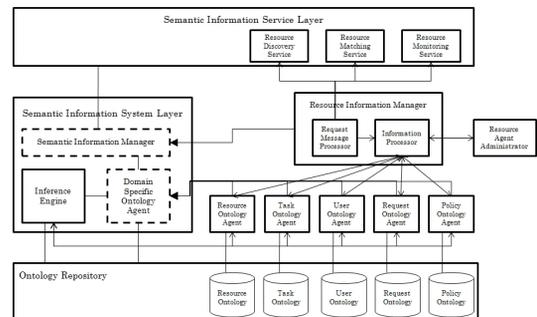
그리드 환경에서 사용자가 자신이 사용할 리소스를 결정할 때, 다양한 조건을 고려하여 결정한다. 작업수행에 필수적인 요구, 리소스에 대한 사용자의 권한 등과 같은 필수 요구사항을 고려하

여, 이를 만족하는 리소스를 찾고, 이 가운데에서 자신의 작업의 성격에 따라 리소스를 정보를 보고 최종적으로 결정하게 된다. 따라서 단순히 리소스의 작업 수행 성능의 우열로 선택하는 것이 아닌, 작업의 환경에 존재하는 다양한 정보들을 고려하여, 서비스를 제공하기 위하여, 지능적인 정보 시스템이 필요하다.

따라서 본 논문에서는 WebSIS 프레임워크와 함께, WebSIS를 이용한 그리드 환경에서 효과적인 리소스할당을 위한 리소스 정보 서비스를 제안한다. 리소스 정보 서비스 위하여 리소스 정보 관리자와 온톨로지 model을 설계 및 구현 하였고, 리소스 정보 관리자에서 동적 리소스 정보를 얻고, 리소스의 등록과 제거를 자동화 하기 위하여 외부 컴포넌트인 리소스 에이전트를 제작하여 이용하였다. 현재 리소스 정보 관리자가 제공하는 서비스에는 리소스 디스커버리 서비스, 리소스 매칭 서비스, 리소스 모니터링 서비스며, 차후에 필요한 정보 서비스의 추가는 리소스 정보 관리자의 기능 추가를 통하여 가능하다.

4.1 리소스 정보 관리자

그림 4-1은 WebSIS를 이용한 리소스 정보 시스템의 구조다. 리소스 정보 관리자는 WebSIS의 시맨틱 정보 관리자 class를 상속하여 SIM에서 제공하는 기능을 활용하여 실질적인 서비스를 제공하는 역할을 한다.

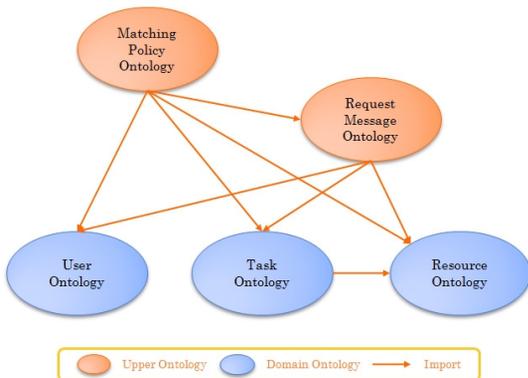


(그림 4-1) WebSIS를 사용한 리소스 정보 시스템 구조

본 연구의 리소스 정보 관리자는 서비스 호출 시에 전달 된 사용자의 리퀘스트 메시지를 분석하고, 분류하는 리퀘스트 메시지 프로세서와 리퀘스트 메시지를 이용하여 사용자가 원하는 정보를 DSOA와 추가적인 컴포넌트인 리소스 에이전트를 이용하여 처리하고, 제공하는 정보 프로세서로 구성된다. 리퀘스트 메시지 프로세서로부터 분류된 리퀘스트 메시지를 받은 정보 프로세서는 리퀘스트 온톨로지 에이전트에게 해당 리퀘스트 메시지에 대한 Individual의 생성을 요청하고, 폴리시 온톨로지 에이전트에게 필요한 리소스의 정보를 추론하여 제공할 것을 요청한다. 폴리시 온톨로지 에이전트로부터 제공된 리소스 리스트를 가지고 리소스 에이전트 관리자를 통하여 리소스 에이전트로부터 리소스 목록의 모든 리소스에 대한 동적 정보를 얻을 수 있다.

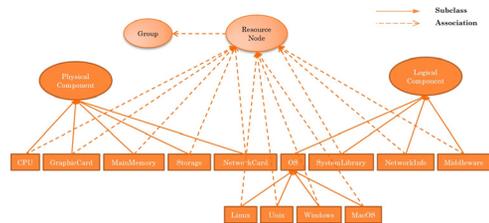
4.2 온톨로지 모델

Knowledge base를 구성하는 온톨로지의 확장과 재사용성의 극대화를 위하여, 우리는 온톨로지를 각각의 도메인별로 따로 정의하고, 중복되거나 다른 도메인의 개념을 사용할 필요가 있을 경우에는 다른 온톨로지를 임포트하여 사용하는 계층적 구조로 모델링 하였다.



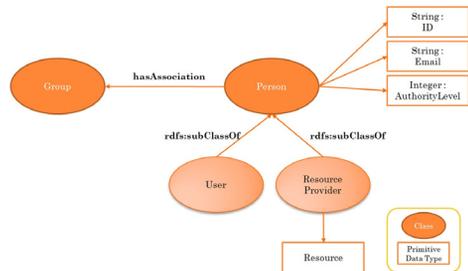
(그림 4-2) 온톨로지 모델 구조

그리드 환경의 리소스 할당을 위한 리소스 정보 서비스가 이용하는 Knowledge base는 리소스, 사용자, 태스크, 리퀘스트 메시지 그리고 매칭 폴리시 온톨로지의 총 5개의 도메인 온톨로지로 구성되며, 추론을 위하여 디스크립션 로직 외의 로직을 정의한 사용자 임의의 규칙을 이용한다. WebSIS의 리소스 정보 서비스는 사용자의 최소한의 입력으로 신뢰할 만한 정보 서비스를 제공하기 위하여, 다양한 온톨로지의 정보를 이용한다. 각 온톨로지의 역할은 다음과 같다.



(그림 4-3) 리소스 온톨로지 모델

먼저 리소스 온톨로지는 그리드를 구성하는 개개의 컴퓨팅 리소스를 정의한다. 컴퓨팅 리소스의 하드웨어와 소프트웨어 컴포넌트들을 조합하여 하나의 리소스를 정의 할 수 있으며, 이렇게 생성된 리소스 인스턴스 (리소스 온톨로지 individual)를 통하여 Knowledge Base 안에 가상의 리소스 풀을 구축할 수 있다. 따라서 실제의 리소스 정보와 리소스 인스턴스의 정보를 주기적으로 동기화 함으로써 실제 작업 수행 시 모든 리소스와 통신하지 않아도 되기 때문에, 작업 수행시간을 단축시킬 수 있다.

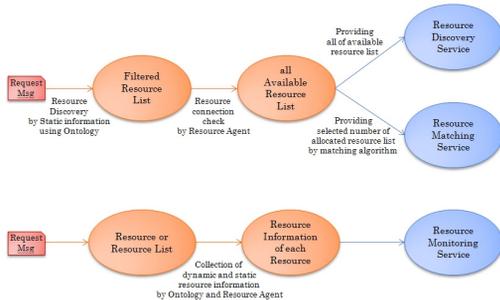


(그림 4-4) 사용자 온톨로지 모델

소스를 선택하기 위한 기준이 되는 선택 요구사항을 추론한다. 즉, 사용자가 수행하고자 하는 작업을 정의한 태스크 온톨로지의 **Individual**의 정보와 사용자의 데이터가 포함된 사용자 온톨로지의 **Individual** 정보를 통하여 필수 요구사항과 선택요구사항을 결정하고, 이 요구사항을 바탕으로 리소스 검색을 수행한다.

4.3 리소스 정보 서비스

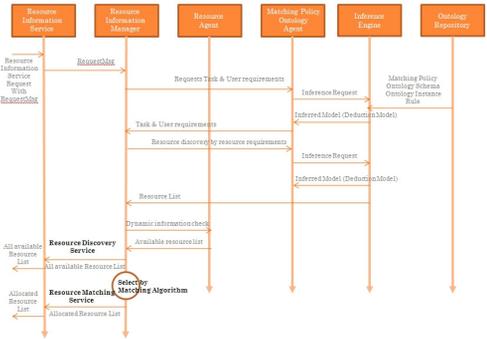
그림 4-8은 리소스 정보 서비스의 흐름도를 나타낸다. 그림 4-8에서 보는 것과 같이 리소스 정보 서비스는 리소스 디스커버리 서비스, 리소스 매칭 서비스, 리소스 모니터링 서비스의 3가지 서비스로 이루어진다.



(그림 4-8) 리소스 정보 서비스의 흐름도

리소스 디스커버리 서비스는 사용자의 작업에 적합한 모든 리소스를 제공하여 사용자가 직접 리소스를 선택할 수 있도록 하는 서비스이다. 반면, 리소스 매칭 서비스는 사용자가 필요한 리소스의 수를 입력하면, 리소스 디스커버리 서비스에 의하여 나온 리소스 리스트에서, 태스크 온톨로지의 작업의 성격에 맞는 가장 최적의 리소스를 필요한 수만큼 제공하는 서비스이다. 즉, 리소스 디스커버리 서비스와 리소스 매칭 서비스는 리소스 할당의 최종 결정 단계를 자동화 여부로 구분된다. 리소스 모니터링 서비스는 리소스 온톨로지 **individual**의 정보와 리소스 에이전트로부터 얻은 동적 리소스 정보를 제공하는 서비스이다. 따라서

리소스 디스커버리 서비스와 리소스 매칭 서비스는 온톨로지의 추론기능을 이용한 서비스이며, 리소스 모니터링 서비스는 특정 리소스의 정적 정보와 현재 상태정보를 제공하기 위한 서비스이다.



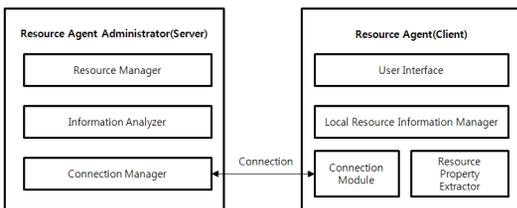
(그림 4-9) 리소스 디스커버리와 매칭 서비스의 시퀀스 다이어그램

리소스 디스커버리는 2단계의 정보처리를 통하여 이루어진다. 먼저, 리퀘스트 메시지 프로세서로부터 받은 분류된 리퀘스트 메시지의 태스크 디스크립션과 사용자의 id를 이용한 사용자 정보를 이용하여, 매칭 폴리시 온톨로지에 필요한 리소스의 요구를 추론하고, 추론된 요구를 이용하여, 최종적으로 리소스 온톨로지에 해당하는 요구를 만족하는 리소스를 검색한다. 검색된 모든 리소스는 1차적으로 작업을 수행할 수 있는 리소스 이나, 온톨로지만을 이용한 정보는 정적인 정보이므로 현재의 상황을 알 수 없다. 따라서 1차적으로 제공받은 리소스 목록의 리소스 에이전트를 이용하여, 현재의 연결상태를 확인하고, 사용 가능한 상태의 리소스만을 사용자에게 제공한다. 이것이 리소스 디스커버리 서비스의 흐름이며, 리소스 매칭 서비스는 이 단계에서 리소스 정보 관리자에 의하여 사용자가 요청한 리소스의 수만큼만 제공한다. 시스템 어플리케이션은 리소스 디스커버리 서비스를 통해 얻은 리소스 리스트에 대하여 리소스 모니터링 서비스를 호출함으로써 사용자에게 해당 리소스의 정보를 제공할 수 있으며, 어플리케이션 사용자는 리소스의 정보를 보고 최

중 선택을 할 수 있다.

4.4 리소스 에이전트

그림 4-10은 리소스 정보 관리자에서 리소스의 동적 정보를 위하여 이용하는 리소스 에이전트(RA)와 각각의 RA의 관리를 담당하는 서버의 역할을 하는 리소스 에이전트 관리자(RAA)의 구조를 나타낸다.



(그림 4-10) 리소스 에이전트의 구조

설치된 리소스의 물리적인 정보와 동적으로 변화하는 상태 정보를 추출하기 위한 리소스 특성 추출기와 추출한 정보를 관리하는 지역 리소스 정보 관리자, 리소스 제공자를 위한 간단한 사용자 인터페이스와 RAA와의 연결을 담당하는 연결 모듈로 구성된 간단한 소프트웨어 클라이언트인 RA는 각 리소스의 정보를 리소스 정보 관리자에게 제공하기 위한 외부 컴포넌트이다. 따라서 WebSIS의 리소스 정보 서비스를 이용하고자 하는 모든 리소스는 RA가 설치되어 있어야 한다. RA는 RAA를 통한 리소스 정보 관리자의 요청에 해당 리소스의 정보를 제공하는 역할을 한다. 또, RA가 처음 실행될 때, 리소스의 정보를 제공함으로써 리소스 온톨로지에 해당 리소스의 Individual을 자동으로 생성 및 제거할 수 있게 한다.

5. 구현과 평가

5.1 구현

표 5-1은 WebSIS의 개발 환경을 보여준다.

(표 5-1) 개발 도구

	Tools
Web services	Apache Axis Apache Tomcat server
Program Language	Java SDK
Semantic Web Technology	Jena 2 Semantic Web Framework
Reasoner	Pellet OWL-DL Reasoner
Ontology Editor	Protégé 3.2

WebSIS는 Java언어로 작성되었으며, 온톨로지의 이용을 위하여 Jena 2 시맨틱 Web 프레임워크(Jena)를 이용하였다. Jena는 XML기반의 OWL로 정의된 온톨로지 문서를 파싱하여 자바 오브젝트의 형태로 변환하고 이를 이용하는 다양한 기능을 제공하여 개발을 편리하게 하는 프레임워크이다. 또한, 온톨로지의 추론을 위하여 사용자 정의 룰 기반의 유전 규칙 추론기를 포함하여 다양한 추론기를 제공한다. 하지만, 현재까지 릴리즈 된 버전에서는 OWL-DL의 추론을 충분히 지원하지 못하기 때문에, OWL-DL 추론을 위하여, 추가적으로 외부의 추론기를 이용하였다. Pellet은 OWL-DL 언어로 작성된 온톨로지를 추론할 수 있는 추론기이며, Jena에서의 사용을 위한 라이브러리를 제공한다. 따라서 WebSIS에서는 Jena의 유전 규칙추론기와 함께 OWL-DL추론을 위한 Pellet 추론기를 이용한다.

(표 5-2) WebSIS의 메인 메소드

Class	Main Method
Semantic Information Manager	RequestMsgAnalyze() RequestIndividualSet(Class name, rule file name)
DomainSpecific OntologyAgent	RequestIndividualSet(ontology Class name, rule file name) ontModelCreation() reasonerCreation(ontology model, rule file name) inferenceModelCreation(reasoner, ontology model) reasoning(inference model) getIndividualSet(ontology model, ontology class name)
InferenceEngine	ontModelCreation() reasonerCreation(ontology model, rule file name) inferenceModelCreation(reasoner, ontology model) reasoning(inference model)

표 5-2는 WebSIS에서 제공하는 주요 기능을 나타낸다. WebSIS에서는 정보 서비스의 개발자가 Jena 프레임워크에 익숙하지 않더라도 온톨로지 기반의 정보 시스템 개발을 할 수 있도록 하기 위하여, 온톨로지와 관련된 정보처리 및 추론은 Inference Engine과 DSOA에서 처리하도록 구현하였다. 따라서 시스템 개발자는 SIM을 구현함으로써, 정보 시스템을 개발할 수 있으며, 온톨로지를 이용하기 위하여 SIM에서 제공하는 기능을 이용하고, 필요한 정보를 DSOA의 함수의 호출을 통하여 이용할 수 있다. 각각의 함수의 역할은 다음과 같다.

RequestMsgAnalyze()은 하나의 요청 메시지에 여러 도메인의 정보를 담고 있을 경우 도메인 별로 분류하는 코드가 들어가게 된다. 각각의 도메인은 자신이 가진 정보만을 이용하므로, 도메인에 맞는 정보만 처리가 가능하다. RequestIndividualSet(Class name, rule file name)는 SIM에서 DSOA에게 온톨로지를 이용한 정보처리를 요청하는 함수이다. 온톨로지 내의 필요한 클래스이름과 사용하고자 하는 룰을 인자로 필요한 온톨로지의 DSOA에게 정보를 요청하면, DSOA는 자신의 온톨로지를 이용하여 IE에게 온톨로지 생성(ontModelCreation()), 추론기 생성(reasonerCreation(ontology model, rule file name)), 추론모델 생성(inferenceModelCreation(reasoner, ontology model)), 추론(reasoning(inference model))의 과정을 통하여 추론 작업을 진행하고, 추론 결과로 생성된 모델에 getIndividualSet

(ontology model, ontology class name)함수를 이용하여 SIM이 요청한 class의 정보를 제공한다. DSOA의 함수들은 IE에게 해당 작업을 요청하는 함수이며, 실질적인 모델의 생성과 추론 과정은 대응되는 IE의 함수를 통하여 이루어진다.

본 논문에서는 WebSIS를 이용하여 그리드 환경에서 작업에 적합한 리소스를 찾고 그에 대한 정보를 제공하는 리소스 정보 관리자를 구현하였다. 표 5-3은 리퀘스트 메시지 구성 요소를, 표 5-4은 리소스 정보 관리자의 주요 기능을 나타낸다.

(표 5-3) RequestMsg 클래스

```
public class RequestMsg implements java.io.Serializable {
    private float cpuspeed;
    private int jobnumber;
    private java.lang.String jobtype;
    private float memorycapacity;
    private float memoryspeed;
    private java.lang.String networkgroup;
    private int numberofresource;
    private int specnumber;
    private float storagecapacity;
    private int storagespeed;
    private java.lang.String useremail;
    private int usernumber;
    private java.lang.String vgachipset;
    private float vgamemorycapacity;
}
```

(표 5-4) 리소스 정보 관리자의 주요 기능

Class	Main Method
Resource Information Manager	requestMsgAnalyze() getIndividualSet(Class name, rule file name) resourceRanking(ResourceList) connectionCheck(ResourceList) getResourceListwithInfo(ResourceList)

리소스 정보 관리자에서는 SIM의 RequestMsg Analyze() 와 getIndividualSet(Class name, rule file name) 함수를 구현하고, 추가적인 정보 처리 함수를 구현하였다.

RequestMsgAnalyze()는 사용자 정보, 작업 정보, 리소스 정보로 구성된 요청메시지를 도메인에 맞게 분류하는 역할을 하며, getIndividualSet(Class name, rule file name) 를 통하여 DSOA에게 필요한 리소스의 List를 얻을 수 있다. DSOA로부터 받은 리소스 List는 온톨로지 내에 저장된 정적인 정보만을 이용하여 검색된 정보이므로, 이것만으로는 현재의 이용 가능성을 확신할 수 없기 때문에 connectionCheck(ResourceList) 함수를 이용하여 리소스 에이전트에 현재의 연결상태를 확인하는 작업을 거치게 된다. Connection Check까지 마친

후 결과로 나온 리소스 List는 현재 사용자가 요청한 작업을 수행할 수 있는 모든 리소스를 포함하고 있다. 이 리소스 목록을 ResourceRanking (ResourceList) 함수를 통하여 작업의 성격에 따라 정렬한다. 리소스 디스커버리 서비스의 경우엔 현재의 리소스 List를 그대로 전달하며, 사용자가 자신이 사용할 리소스를 최종 결정할 수 있도록 한다. 리소스 매칭 서비스의 경우에는 사용자가 입력한 필요 리소스의 개수만큼의 리소스를 제공한다. 따라서 작업 수행의 모든 과정을 자동화 하기 위해서는 리소스 매칭 서비스를 이용하면 된다. getResourceListwithInfo(ResourceList) 함수는 리소스의 목록을 제공하고, 온톨로지와 리소스 에이전트를 통하여 그 리소스 목록 안의 모든 리소스에 대한 정적, 그리고 동적인 정보를 받아오는 함수로 리소스 모니터링 서비스를 위한 함수이다.

5.2 평가

표 5-5는 WebSIS와 다른 정보 시스템간의 특성 비교표이다. WebSIS는 정보의 공통적인 이용과 재사용 등의 특성을 이용하여, 그리드 컴퓨팅 환경의 변화에 유연하게 대응할 수 있고, 서비스를 제공하는 프로그램에 독립적인 종합 정보 시스템을 목표로 개발되었다.

현재 그리드 관련 연구에서는 WebSIS와 같이 그리드 환경의 종합적인 정보 제공 시스템의 연구가 이루어지지 않고 있다. 또한, 대부분의 정보 시스템에 관한 연구는 특정 작업의 자동화를 위한 정보 서비스의 역할만을 수행하였고, 정보 시스템의 연구는 이루어지지 않고 있었다. 따라서 현재 제공되는 시스템 독립적인 정보시스템은 Globus의 MDS나 Ganglia, ASKALON의 GridARM 등 주로 리소스의 정보에 관한 시스템이 대부분이었다.

(표 5-5) 다른 정보 시스템과의 비교

	WebSIS	MDS4	Matchmaker	GridARM
Service Interface	Web services	Web services	Message Passing	Message Passing
Required component	Resource Agent	Globus (GT4)	Non	Globus
Extensibility	Service & Information description	Bad	Information description	Bad
Reusability	Good	Bad	Good	Bad
Information	Static & Dynamic	Static & Dynamic	Static	Static & Dynamic
Description Language	OWL-DL Ontology language	LDAP	RDFS Ontology language	LDAP & Ontology language

따라서 그리드 환경을 이용하는 응용프로그램을 개발하는 개발자는 제공되지 않는 정보에 대해서는 스스로 서비스를 만들고 이용해야하는 불편함이 존재하였고, 이에 우리는 서비스의 제작을 쉽게 하여, 같은 환경을 이용하는 이용자가 서비스를 쉽게 공유할 수 있는 구조를 제안하였다. 또한 그리드 컴퓨팅과 같이 대규모의 컴퓨팅환경에서, 모든 리소스와 통신하여 정보를 얻어오고 이를 통한 리소스 할당은 작업의 성능을 많이 떨어뜨리게 된다. 본 논문에서 제안한 WebSIS를 이용한 리소스 정보 서비스에서는 리소스에 대한 정보를 온톨로지의 Individual로 저장하고, 그 정보를 이용하여 작업을 수행할 최소한의 요건을 갖춘 리소스만 찾아 통신을 하기 때문에, 성능에 가장 영향을 많이 미치는 통신 시간을 대폭 줄일 수 있게 하였다.

6. 결론

그리드 컴퓨팅 환경은 기존의 분산 컴퓨팅 환경에 비하여 매우 큰 규모로 무한한 잠재력을 가지고 있다. 그러나 규모가 커짐에 따라, 기존의 분산 컴퓨팅과는 달리 다양한 이종의 컴퓨팅 리소스들로 구성이 되었고, 이러한 환경을 쉽게 이용하기 위해서는 기존의 클러스터 컴퓨터와 같은 소규모의 환경에서는 불필요했던 요소들 까지도 고려해야 하게 되었다. 따라서 사용자가 쉽게 이

용할 수 있도록, 작업 환경의 자동화가 필요하며, 효율적인 작업 자동화는 고수준의 정보를 필요로 한다. 이에, 본 논문에서는 대규모의 분산 컴퓨팅 환경인 그리드 상에서 공유와 재사용이 용이한 정보 서비스를 제공하는 온톨로지 기반의 정보 시스템을 쉽게 제작할 수 있게 하기 위한 시맨틱 정보 시스템 프레임워크와 이를 활용하여 그리드 환경의 리소스 얼로케이션을 위한 플랫폼 독립적인 리소스 정보 서비스를 제안하였다. WebSIS는 온톨로지를 이용하여, 정의된 정보를 공유할 수 있고, 재사용이 가능하며, 추론을 통한 고수준의 정보 처리를 가능하게 하였다. 또한, 작성된 서비스를 함께 공유함으로써, 그리드 환경을 거대한 단일 컴퓨팅환경으로의 이용을 가능하게 한다. 따라서 WebSIS는 시간에 따라 새로운 리소스의 추가와 삭제가 이루어지고, 그리드 환경을 이용하는 작업이 늘어나도, WebSIS가 제공하는 기존의 서비스는 그대로 유지하며 새로운 개념을 온톨로지 로 새롭게 정의하고, 이를 활용하여 새로운 서비스를 추가할 수 있는 변화하는 환경에 유연하게 대응할 수 있는 시스템이다.

참 고 문 헌

- [1] Grid Computing, http://en.wikipedia.org/wiki/Grid_computing
- [2] Foster, I., Kesselman, C. and Tuecke, S. The Anatomy of the Grid: Enabling Scalable Virtual Organizations. International Journal of Supercomputer Applications, 15 (3). 200-222. 2001.
- [3] Globus Toolkit, <http://www.globus.org/toolkit/>.
- [4] Global Grid Forum, <http://www.ggf.org>
- [5] Foster, I., Kesselman, C., Nick, J. and Tuecke, S. The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration. Globus Project, 2002. www.globus.org/research/papers/ogsa.pdf.
- [6] W3C. Web Services, <http://www.w3.org/2002/ws/>.
- [7] S.J. Chapin, D. Katramatos, J.F. Karpovich, and A.S. Grimshaw, "Resource Management in Legion," Journal of Future Generation Computing Systems, vol. 15, pp. 583-594, 1999.
- [8] The Condor Project, <http://www.cs.wisc.edu/condor/>
- [9] The myGrid project, <http://www.mygrid.org.uk/>
- [10] The GridLab project, <http://www.gridlab.org/>
- [11] The unicore project, <http://www.unicore.org>
- [12] W3C Semantic Web, <http://www.w3.org/2001/sw/>
- [13] OWL Web Ontology Language Reference, <http://www.w3.org/TR/owl-ref/>, 2004
- [14] OASIS Web Services Resource Framework http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsrf
- [15] SOA and Web Services, <http://java.sun.com/developer/>.
- [16] GT Information Services : Monitoring & Discovery System (MDS), <http://www.globus.org/toolkit/mds/>
- [17] W3C, World Wide Web Consortium, <http://www.w3.org/>
- [18] RDF, Resource Description Framework, <http://www.w3.org/RDF/>
- [19] XML Topic Maps (XTM) 1.0, <http://www.topicmaps.org/xtm/index.html>
- [20] Notation 3, Tim Berners-Lee, 2006, <http://www.w3.org/DesignIssues/Notation3.html>
- [21] Knowledge Interchange Format (KIF), <http://logic.stanford.edu/kif/dpans.html>
- [22] Semantic Web, http://en.wikipedia.org/wiki/Semantic_web
- [23] RacerPro OWL reasoner and inference server for the semantic web, <http://www.racer-systems.com/index.phtml>
- [24] Pellet OWL-DL reasoner, <http://pellet.owldl.com/>
- [25] Jena 2 Semantic Web Framework, <http://jena.sourceforge.net/>
- [26] I. Foster, C. Kesselman, J. Nick, S. Tuecke, Open

Grid Service Infrastructure WG, Global Grid Forum. "The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration." June 22, 2002.

- [27] S. Tuecke, K. Czajkowski, I. Foster, J. Frey, S. Graham, C. Kesselman, T. Maguire, T. Sandholm, D. Snelling, "Open Grid Services Infrastructure", <http://www.ggf.org/documents/GWD-R/GFD-R.015.pdf>

[28] Foster I, Kesselman C (eds.). The Grid: Blueprint for a New Computing Infrastructure. Morgan Kaufmann: San Fransisco, CA, 1999.

[29] Tannenbaum, T., Wright, D., Miller, K., Livny, M., "Condor - A Distributed Job Scheduler" in Sterning, T. (ed) Beowulf Cluster Computing with Linux, The MIT Press, 2002..

● 저 자 소개 ●



한 병 진

2007년 고려대학교 전자전기전과공학과 졸업(학사)
2009년 고려대학교 대학원 전자전기공학과 졸업(석사)
2009년 고려대학교 대학원 전자전기공학과 입학(박사)
2009~현재 고려대학교 전자전기공학과 연구원(박사)
관심분야 : 그리드, 미들웨어, 분산처리
E-mail : guru1013@korea.ac.kr



김 형 래

2004년 고려대학교 전자전기공학과 졸업(학사)
2004년 고려대학교 대학원 전자전기공학과 입학(석박 통합)
2004~현재 고려대학교 전자전기공학과 연구원(박사)
관심분야 : 그리드, 미들웨어, 웹서비스, 분산처리
E-mail : nolight@korea.ac.kr



정 창 성

1981년 서울대학교 전기공학과 졸업(학사)
1984년 Northwestern대학교 대학원 전자계산학과 졸업(석사)
1987년 Northwestern대학교 대학원 전자계산학과 졸업(박사)
1987년 ~ 1992년 포항공학대학교 전자계산학과 조교수
1992년 ~ 1998년 포항공학대학교 전자공학과 부교수
1998년 ~ 현재 고려대학교 전자공학과 정교수
관심분야 : 그리드, 미들웨어, 분산처리, 분산시뮬레이션, 이미지 프로세싱
E-mail : csjeong@korea.ac.kr