그리드 기반의 고성능 과학기술지식처리 프레임워크 개발

Development of a Grid-based Framework for High-Performance Scientific Knowledge Discovery

정창후, 최성필, 윤화묵, 최윤수
한국과학기술정보연구원 정보기술연구실

Chang-Hoo Jeong(chjeong@kisti.re.kr), Sung-Pil Choi(spchoi@kisti.re.kr),
Hwa-Mook Yoon(hmyoon@kisti.re.kr), Yun-Soo Choi(armian@kisti.re.kr)

요약

본 논문은 그리드 컴퓨팅을 이용한 고성능 과학기술지식처리 프레임워크인 SINDI-Grid의 개발에 관련된 연구이다. SINDI-Grid 프레임워크는 대용량의 데이터 저장소 및 고속의 컴퓨팅 파워를 제공하는 그리드 컴퓨팅의 장점을 이용하여 분산 데이터 분석과 과학기술지식처리를 위한 다양한 그리드 서비스들을 제공한다. 그리고 SINDI-Workflow 도구는 이러한 서비스들을 이용하여 다양한 지식처리 알고리즘을 통합하는 복잡한 과학기술지식처리 애플리케이션을 설계하고 실행하는 역할을 수행한다.

■ 중심어 : | 지식처리 | WSRF | 워크플로우 | 프레임워크 |

Abstract

In this paper, we propose the SINDI-Grid which is a high-performance framework for scientific and technological knowledge discovery using the grid computing. By using the advantages of the grid computing providing data repository of large-volume and high-speed computing power, the SINDI-Grid framework provides a variety of grid services for distributed data analysis and scientific knowledge processing. And the SINDI-Workflow tool exploits these services so that performs the design and execution for scientific and technological knowledge discovery applications which integrate various information processing algorithms.

■ keyword : | Knowledge Processing | WSRF | Workflow | Framework |

I. 서론

정보화 시대로 인해 처리해야 하는 데이터의 양이 기하급수적으로 증가하면서 데이터 안에 숨겨진 유용한 정보를 자동으로 추출하는 데이터 마이닝 애플리케이션의 컴퓨팅 리소스에 대한 요구 사항도 커지게 되었다. 이러한 요구 사항은 비단 데이터를 저장하는 하드디스크나 메인 메모리에 국한되지 않고 복잡한 알고리즘을 빠르게 처리하기 위한 성능 좋은 컴퓨팅 파워 및 전체적인 실행 관리 기법도 함께 요구하고 있다. 기존의 방법으로는 대용량의 데이터를 고속으로 처리함에 있어 많은 한계가 있기 때문에 새로운 데이터 마이닝 방법론이 요구되고 있는데, 그리드는 분산된 데이터 저장소 및 컴퓨팅 파워의 집합체로서 계산 복잡도가 매우 높거나 많은 자원을 활용해야 하는 데이터 마이닝 애플리케이션의 문제를 비교적 쉽게 해결해준다. 따라서 현

접수번호 : #091014-001
접수일자 : 2009년 10월 14일
심사완료일 : 2009년 11월 19일
교신저자 : 최윤수, e-mail : armian@kisti.re.kr

재 급속하게 축적되고 있는 과학기술정보를 신속하게 가공하고 지식화하기 위해서 이러한 응용을 논문이나 연구보고서와 같은 과학기술 전문 콘텐츠의 지식 추출을 위한 시스템에 적용해보는 연구가 그 어느 때보다도 필요한 시점이다.

본 논문의 구성은 2장에서 그리드 컴퓨팅을 이용한 기존의 데이터 마이닝 시스템에 대해서 기술하고, 3장에서는 데이터 마이닝을 과학기술 전문 콘텐츠의 지식처리 분야에 특화시켜서 발전시킨 고성능 과학기술지식처리 프레임워크에 대해서 설명한다. 과학기술지식처리 프레임워크에 대한 설명을 위해서 과학기술지식처리란 용어의 기본적인 개념을 우선적으로 기술하고, 이러한 개념을 그리드 환경에서 어떻게 구현하였는지에 대해서 설명한다. 다음으로 4장에서 프레임워크와 상호작용하는 워크플로우 도구에 대해서 설명하고, 마지막으로 5장에서 결론 및 향후 연구에 대해서 기술한다.

II. 관련 연구

대용량 데이터를 처리하기 위해서 다양한 형태의 분산 데이터 마이닝 프레임워크에 대한 연구가 이루어지고 있다. 이 중에서 그리드 컴퓨팅 기반의 데이터 마이닝 프레임워크인 GridMiner[1], MyGrid[2], DiscoveryNet[3], ADMIRE[4]와 같은 시스템들이 대용량의 데이터 처리 및 복잡한 알고리즘을 실행하기 위한 효과적인 플랫폼이라는 것이 입증되면서 그리드 컴퓨팅 기술의 활용 가치가 점점 높아지고 있다. 더욱이, OGSA(Open Grid Services Architecture)[5]와 WSRF(Web Services Resource Framework)[6] 표준이 발표되고 GT4(Globus Toolkit 4)[7]에서 이러한 표준들이 구현되면서, 그리드 연구·개발 커뮤니티들은 웹 서비스의 확장된 개념으로서 그리드 서비스를 정의하고 최종 사용자가 직접적으로 활용할 수 있는 보다 고차원적인 분산 그리드 서비스를 개발하기 시작했다[8-10]. 결과적으로 그리드 기반 데이터 마이닝 프레임워크는 기능의 유연한 통합 및 확장을 위해서 SOA(Service Oriented Architecture) 구조로 변화하고 있으며, 사용자가 직접 자신이 원하는 지식을 추출하고 이를 서비스할 수 있는 개방형 정보 분석 체제로 발전하고 있다. OGSA와 WSRF, 그리고 SOA와 같은 표준 기술을 구현에 반영한 대표적인 시스템으로는 Knowledge Grid[11]와 Data Mining Grid[12]가 있다.

본 논문에서는 기존에 수행되었던 이러한 연구들의 장점을 취합 및 보완하여 논문이나 연구보고서와 같은 과학기술 전문 콘텐츠의 지식처리 분야에 특화시켜서 발전시킨 고성능 과학기술지식처리 프레임워크에 대해서 설명한다. 과학기술지식처리 알고리즘 고유의 복잡성 극복과 대용량 자원에 대한 효율적인 지식가공을 위하여 과학기술지식처리 작업에 필수적인 핵심 기술을 직접 개발하고[13][14], 이것들을 프레임워크에 내장하여 사용자에게 보다 고차원적인 서비스를 제공하도록 하였다. 본 논문에서는 지식처리 과정의 세부적인 알고리즘 및 성능에 관련된 내용은 다루지 않고 주로 과학기술문서를 효과적으로 처리하기 위한 프레임워크 구축에 중점을 두어 설명한다. 대용량의 과학기술문서를 효과적으로 처리하기 위해서 SINDI(Scientific INtelligence DIscovery)-Grid 프레임워크를 제안하는데, SINDI-Grid 프레임워크는 과학기술문서에서 중요한 의미를 갖는 핵심개체를 인식하고 인식된 개체들 간의 상호작용을 기술하는 연관관계를 추출 및 가공하여 보다 고차원적인 전문지식을 생성하는 과학기술지식처리 애플리케이션의 개발을 지원하는 그리드 컴퓨팅 기반의 프레임워크이다.

III. 고성능 과학기술지식처리 프레임워크 개발

본 장에서는 그리드 환경에서 다양한 종류의 지식처리 알고리즘의 통합을 지원하는 고성능 과학기술지식처리 프레임워크에 대해서 설명한다. 그리드 컴퓨팅 환경을 구성하기 위해서 OGSA를 따르는 GT4 미들웨어를 사용하였고, 과학기술지식처리를 위한 그리드 서비스 개발을 위하여 상태 정보를 유지하는 WSRF의 특징을 효과적으로 활용하였다.

1. 과학기술지식처리 개요

SINDI는 대용량의 과학기술문서에서 중요한 의미를 갖는 핵심개체를 인식하고 인식된 개체들 간의 상호작용을 기술하는 연관관계를 추출 및 가공하여 보다 고차원적인 전문지식을 생성하는 과학기술지식처리 아키텍처이다. SINDI는 크게 CORE, REX, 그리고 SET으로 구성되어 있다. 우선 첫 번째로 SINDI-CORE는 데이터베이스를 구성하는 개별 서지의 제목 및 초록, 그리고 본문 정보에 내재된 과학기술분야 인명, 기관명, 기술용어, 유전자, 단백질, 질병명, 신체조직명, 약품명 등 다양한 종류의 개체를 자동으로 식별하고 분류할 수 있는 과학기술핵심개체 인식 엔진을 의미한다. 다음으로 SINDI-REX는 데이터베이스 내에 존재하는 복수 핵심개체들 간의 연관성을 파악하고 의미적 연관관계를 자동으로 추정할 수 있는 과학기술핵심개체 간 관계추출 엔진을 의미한다. 마지막으로 SINDI-SET은 개발된 두 가지 핵심 엔진들에 대한 객관적인 성능 평가 및 기능 개선을 위한 테크 마이닝 기반기술 평가·검증 컬렉션을 의미한다.

SINDI의 개념적 시스템 아키텍처는 [그림 1]과 같다.

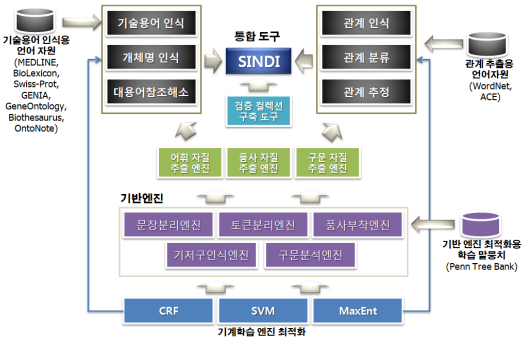


그림 1. SINDI의 개념적 시스템 아키텍처

우선 기반 엔진으로 문장분리엔진, 토큰분리엔진, 품사부착엔진, 기저구인식엔진, 구문분석엔진이 있다. 기반 엔진 최적화를 위하여 학습 말뭉치를 사용하고 있으며, CRF, SVM, MaxEnt와 같은 최신의 기계학습 알고리즘을 사용하고 있다. 이러한 기반 엔진을 이용하여 보다 상위 레벨에서 기술용어 인식, 개체명 인식, 대용어 참조 해소와 같은 과학기술핵심개체를 식별하는 엔진이 완성되고, 관계 인식, 관계 분류, 관계 추정과 같은 과학기술핵심개체 간 관계를 추출하는 엔진이 완성된다. 이때 핵심개체 식별 엔진과 연관관계 추출 엔진의 개발을 위하여 부가적으로 다양한 언어 자원과 자질 정보를 활용한다.

2. 그리드 기반 과학기술지식처리 프레임워크

SINDI의 개념적 시스템 아키텍처에서 설명한 다양한 내부 엔진들은 엔진이 추구하는 목적에 따라서 실행에 필요한 컴퓨팅 리소스 및 처리 속도에 대한 요구 사항이 아주 다양하다. 이러한 엔진들을 하나의 시스템에서 일괄적으로 개발 및 배치하는 방법은 기존의 소규모 데이터 처리 시스템에서는 크게 문제가 되지 않겠지만 대용량의 데이터베이스를 대상으로 하였을 경우에는 심각한 성능 저하를 가져올 수 있다. 시스템을 구성하는 하부 엔진의 일부분에서 과도한 리소스 및 실행 시간이 필요한 경우에도 전체 시스템의 성능에 악영향을 끼치게 된다. 이와 같이 기존의 시스템이 처리할 수 없는 대용량의 문헌을 고속으로 처리할 수 있는 새로운 방법론으로 그리드가 소개되면서 다양한 지식처리 시스템에 적용되기 시작했다. 더 나아가 웹 서비스의 호출에 있어서 상태 정보를 유지할 수 있는 기능이 제공되면서 WSRF 기반의 분산 지식 마이닝 시스템의 개발이 활성화되고 있다. 본 연구에서는 이러한 특징을 잘 활용하여 그리드 컴퓨팅 환경에서 SINDI와 같은 과학기술지식처리 애플리케이션의 생성을 쉽게 도와주는 SINDI-Grid 프레임워크를 정의하고 개발하였다. SINDI-Grid의 실제적인 시스템 아키텍처는 [그림 2]와 같다.

SINDI-Grid는 Globus Toolkit, SINDI-Grid Core, SINDI-Grid Service, SINDI-Workflow 도구로 구성되어 있다. 첫 번째로 Globus Toolkit은 실행 관리, 데이터 관리, 정보 서비스, 그리드 보안, 그리고 실행 환경을 제공하는 그리드 미들웨어를 의미한다. 이러한 그리드 미들웨어는 그리드 환경을 구성하는 컴퓨팅 리소스를

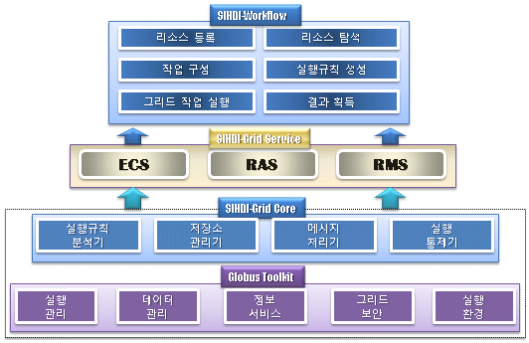


그림 2. SINDI-Grid 아키텍처

효율적으로 통합 및 관리하는 역할을 수행한다. 두 번째로 SINDI-Grid Core는 상위의 SINDI-Grid Service에서 제공하는 다양한 서비스들을 효과적으로 지원하기 위한 핵심 기능들을 구현하고 있다. 작업에 대한 실행규칙을 해석하고 실제적인 그리드 실행 명세서를 생성하는 실행규칙 분석기, 리소스에 관련된 정보의 등록 및 검색을 위한 저장소 관리기, 웹 서비스와 SINDI-Workflow 도구 사이에 전달되는 다양한 XML 메시지를 처리하기 위한 메시지 처리기, 그리고 그리드를 구성하는 여러 컴퓨팅 노드들에게 실행을 명령하고 감시하는 실행 통제기로 구성된다. 세 번째로 SINDI-Grid Service는 과학기술지식처리 프레임워크를 구축하기 위해서 개발된 다양한 핵심 기능들을 그리드 서비스 개념으로 외부에 개방하고 워크플로우 도구와 효과적인 상호 작용을 수행한다. 그리드 상의 리소스(프로그램, 데이터, 언어자원 등) 등록, 삭제, 검색을 위한 RMS(Resource Management Service), 워크플로우 작업 구성을 통하여 생성된 실행규칙을 그리드 상의 실제적인 작업 명령으로 변환하고 실행을 수행하는 ECS(Execution Control Service), 그리고 실행규칙의 최종 수행 결과를 워크플로우 도구에게 제공하는 RAS(Result Access Service)로 구성된다. 그리고 마지막으로 과학기술지식처리에 대한 다양한 작업 흐름도를 쉽고 빠르게 생성하도록 도와주는 SINDI-Workflow 도구가 있다.

3. 서비스 개발을 위한 WSRF의 효과적인 활용

WSRF를 이용하여 SINDI-Grid의 웹 서비스를 개발하였는데, 본 장에서는 다중 리소스를 처리할 수 있는 Factory/Instance 디자인 패턴[15]이 과학기술지식처리 서비스 개발을 위하여 어떻게 효과적으로 활용되었는지에 대해서 설명한다.

Factory/Instance 패턴은 소프트웨어 디자인에서 잘 알려진 디자인 패턴으로, 객체의 Instance를 new 연산을 이용하여 직접적으로 생성하지 않고 create 연산을 제공하는 Factory를 통하여 생성한다. GT4의 WSRF 명세서는 다중 리소스를 처리할 때 이 패턴을 따르도록 권고한다. 따라서 상태 정보를 유지하는 웹 서비스를 제공하기 위해서는 리소스를 생성하는 의무를 지닌 Factory Service와 리소스 안에 포함된 정보를 실제로 접근하여 사용하는 Instance Service를 구현해야 한다. [그림 3]은 SINDI-Grid 서비스 중의 하나인 Execution Control Service를 수행할 때, 워크플로우 도구와 웹 서비스 그리고 리소스 사이에서 Factory/Instance 디자인 패턴이 어떻게 적용되고 있는지를 보여준다.

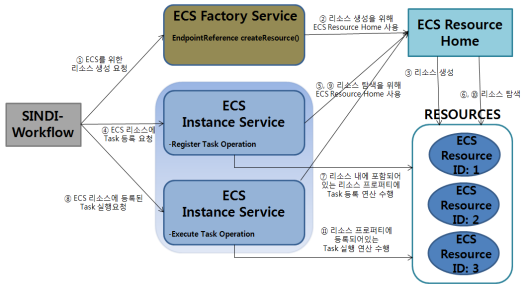


그림 3. Factory/Instance 디자인 패턴을 사용한 Execution Control Service 수행 과정

Execution Control Service의 웹 서비스 부분은 ECS Factory Service와 ECS Instance Service로 나누어지고 리소스 부분은 리소스 관리를 담당하는 ECS Resource Home과 실제 ECS Resource로 구성된다. Factory Service는 새로운 WS-Resource에 대한 EPR(Endpoint Reference)을 반환하는 createResource 연산을 제공하고 Instance Service는 SINDI-Grid 프레임워크에서 분산 과학기술지식처리를 위해서 실제적으로 수행해야 하는 registerTask 혹은 executeTask와

같은 연산을 제공한다. 워크플로우 도구에서 Execution Control Service에 대한 새로운 리소스 생성 요청이 발생하면 ECS Factory Service는 ECS Resource Home을 사용하여 새로운 ECS Resource를 생성하고 초기화한다. 그리고 ECS Resource Home은 생성된 ECS Resource를 식별할 수 있는 키와 함께 해당 리소스를 저장 및 관리한다. 그리고 나서 ECS Factory Service는 ECS Instance Service와 최근에 생성된 ECS Resource로 구성되어 있는 WS-Resource에 대한 EPR을 반환하는데, 이 EPR을 이용하여 이후 모든 작업에서 특정 WS-Resource를 유일하게 참조할 수 있다. WS-Resource는 특정 리소스와 서비스의 쌍으로서 Instance Service의 URI와 새롭게 생성된 리소스의 키를 가지고 있다. ECS Factory Service에 의해서 반환된 EPR을 이용하여 워크플로우 도구는 ECS Instance Service에서 제공하는 과학기술지식처리를 위한 그리드 서비스를 호출할 수 있다.

IV. 과학기술지식처리 워크플로우 도구 개발

최종 사용자는 워크플로우 도구를 이용하여 애플리케이션을 설계하고 실행할 수 있는데, 이러한 도구의 예로는 Triana[16], Tarverna[17], 그리고 Kepler[18]와 같은 것들이 있다. 이와 같은 워크플로우 도구들은 생물학, 지질학, 천체 물리학, 화학 등 여러 분야의 데이터베이스와 웹 서비스에 접근하여 다양한 분석을 수행하면서 도구의 유용성을 입증 받아왔다. 하지만, 다양한 분야의 적용을 위한 기능 확대에 주력한 나머지 시스템이 복잡해지고 사용하기가 어려워지는 문제점이 발생하였다. 본 장에서는 관련 지식이 없는 사용자도 단시간에 습득하여 쉽게 활용할 수 있는, 과학기술 콘텐츠의 지식처리 분야에 특화된 SINDI-Workflow 도구에 대해서 설명한다. SINDI-Workflow 도구는 시스템의 복잡한 기능으로 인해 사용성이 떨어지는 기존 워크플로우 도구의 문제점을 보완하기 위해서 언어처리, 개체 인식, 관계추출, 지식생성 등과 같은 과학기술지식처리에 자주 사용되는 여러 가지 작업들을 템플릿 기반으로 제공하고 사용자는 단지 초기 입력 데이터 및 실행 옵션을 설정함으로써 원하는 결과를 쉽게 얻을 수 있도록 하였다. 이와 같이 과학기술지식처리에 특화된 서비스를 기반으로 워크플로우 도구를 개발하고 자주 사용되는 서비스 시나리오에 대한 기본 템플릿을 제공함으로써 일반 사용자도 과학기술지식처리를 위한 다양한 애플리케이션을 쉽게 개발할 수 있다. 또한 SINDI-Grid 프레임워크에서 제공하는 컴퓨팅 노드와 데이터 노드, 그리고 언어분석 및 지식처리를 위해서 사용되는 다양한 하부 엔진 노드들을 이용하여 복잡한 지식처리 과정이 요구되는 애플리케이션을 세부적으로 설계할 수 있다. [그림 4]는 이러한 애플리케이션을 설계하기 위해서 사용되는 워크플로우 도구를 보여준다.

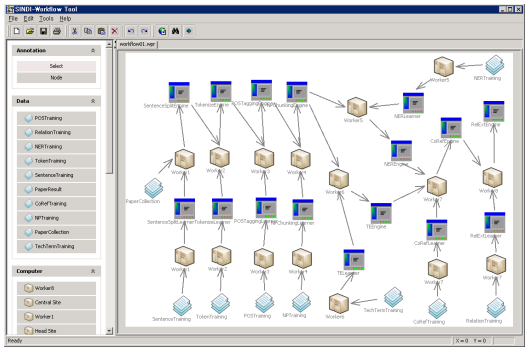


그림 4. SINDI-Workflow 도구

SINDI-Workflow 도구의 기능은 리소스 등록, 리소스 탐색, 작업 구성, 실행규칙 생성, 그리드 작업 실행, 결과 획득의 6개의 모듈로 구성된다. 리소스 등록 모듈은 지식 처리를 위해서 사용되는 다양한 데이터 마이닝 알고리즘 및 언어 처리 엔진 등의 프로그램적인 리소스와 논문, 특허, 연구 보고서와 같은 데이터적인 리소스를 등록하고 삭제하는 기능을 수행한다. 리소스 탐색 모듈은 메타데이터 저장소에 등록되어 있는 리소스에 관련된 정보를 검색해서 사용자에게 제공하는 기능을 수행한다. 작업 구성 모듈은 워크플로우 도구의 좌측 화면에 존재하는 다양한 리소스 관련 노드를 이용하여 과학기술지식처리에 적합한 작업 흐름도를 중앙 화면에 생성하고 실행 옵션을 설정하는 기능을 수행한다.

실행규칙 생성 모듈은 작업 구성을 통하여 생성된 최종적인 작업 흐름도를 XML 문서로 기술하는 역할을 수행한다. 이렇게 기술된 XML 문서는 그리드 작업 실행 모듈을 통하여 그리드 서비스로 전달되어 분산 작업 실행을 위한 실제 명령으로 변환된다. 마지막으로 결과 획득 모듈은 그리드 상에서 수행된 작업 실행 결과를 가져와서 사용자에게 보여주는 역할을 수행한다.

SINDI-Workflow 도구를 이용하면 과학기술지식처리에 관련된 작업 흐름도를 매우 세부적으로 정밀하게 설계할 수 있다. 이것은 SINDI-Grid 프레임워크가 각 하부 엔진들을 저수준의 언어처리 엔진부터 고수준의 개체 및 관계 추출 엔진까지 서비스 지향적인 모듈로 개발함으로써 독립적으로 사용될 수 있는 환경을 제공하고, 각 엔진들을 적용하는 도메인의 특성에 맞게 최적화시킬 수 있는 학습기를 제공함으로써 최상의 실행 환경을 구성할 수 있는 기반을 제공하고 있기 때문이다. [그림 4]의 작업 흐름도에서 보이는 바와 같이 학습 모델을 생성하기 위한 노드로부터 시작해서 데이터 및 언어 자원을 제공하기 위한 노드, 실제적으로 실행을 수행하기 위한 노드, 그리고 최종적으로 결과를 저장하기 위한 노드까지 다양한 방식의 프로세스 통합을 실현시켜 준다. 따라서 각각의 개별 엔진은 단일 프로세스로 존재하지만, 이러한 프로세스들을 결합하여 새로운 서비스를 구성하였을 때 통합된 프로세스 그룹은 또 다른 하부 엔진의 기능으로 사용될 수 있다. 지식 생성을 위한 핵심 요소 모듈들을 서비스 지향적인 구조로 설계함으로써 얻는 또 다른 이점은 각 단위 엔진에 가장 적합한 리소스를 할당함으로써 전체 성능을 개선시킬 수 있다는 것이다. 단위 엔진 별로 요구되는 컴퓨팅 파워나 메모리 공간을 최적화된 상태로 지원함으로써 실행 과정 중의 어느 한 부분이 비효율적으로 지연되는 것을 방지할 수 있다. 따라서 과학기술 지식 생성에 필요한 대용량의 기반 자원을 신속하고 정확하게 처리할 수 있는 분산·병렬 환경을 구성할 수 있다.

워크플로우 도구를 이용하여 생성되는 작업 흐름도는 내부적으로는 실행규칙으로 변환되어 Excution Control Service로 전달된다. 실행규칙을 기술하기 위해서 BPEL(Business Process Execution Language)[19]과 WSCI(Web Service Choreography Interface)[20]와 같은 표준규약을 사용하기도 하지만, 본 논문에서는 간결성을 유지하기 위해서 XML 기반의 실행규칙을 새롭게 정의하였다. [그림 5]는 XML을 기반으로 실행규칙을 새롭게 정의한 것을 보여준다.

```
<?xml version="1.0" encoding="EUC-KR"?>
<!-- 워크플로우 도구를 이용하여 생성되는 작업 흐름도에 대한 실행규칙을 정의 -->
<!ELEMENT ExecutionRule (BasicInformation, TaskSequence)>
<!ATTLIST ExecutionRule
        id          CDATA       #REQUIRED>
<!-- 실행규칙에 관련된 일반적인 정보를 정의 -->
<!ELEMENT BasicInformation (Name, Explanation, Date, Creator)>
<!ELEMENT Name (#PCDATA)>
<!ELEMENT Explanation (#PCDATA)>
<!ELEMENT Date (#PCDATA)>
<!ELEMENT Creator (#PCDATA)>
<!-- 실행규칙에 명시된 작업명령의 흐름을 정의 -->
<!ELEMENT TaskSequence (Task+)>
<!ELEMENT Task (FlowList)>
<!ATTLIST Task
        no          CDATA       #REQUIRED>
<!ELEMENT FlowList (Flow+)>
<!ELEMENT Flow (SourceList?, Destination)>
<!-- 작업명령 수행을 위한 Source 정의 -->
<!ELEMENT SourceList (Source+)>
<!ELEMENT Source (Computer, DataList)>
<!ELEMENT Computer (#PCDATA)>
<!ELEMENT DataList (Data+)>
<!ELEMENT Data (#PCDATA)>
<!-- 작업명령 수행을 위한 Destination 정의 -->
<!ELEMENT Destination (Computer, Program)>
<!ELEMENT Program (Command, OptionList?, ResultList)>
<!ELEMENT Command (#PCDATA)>
<!ELEMENT OptionList (Option+)>
<!ELEMENT Option (Name, Value)>
<!ELEMENT Name (#PCDATA)>
<!ELEMENT Value (#PCDATA)>
<!ELEMENT ResultList (Result+)>
<!ELEMENT Result (#PCDATA)>
```

그림 5. XML 기반 실행규칙 DTD

과학기술지식처리 프레임워크와 상호작용하는 워크플로우 도구를 이용하여 작업 흐름도를 생성하는 과정을 간략하게 기술하면 다음과 같다.

① 사용자는 워크플로우 도구를 이용하여 원하는 리소스를 검색한다. 리소스 검색 기능은 프레임워크와 상호작용하여 검색 결과들을 반환받는다. 원하는 리소스가 없는 경우에 자신이 보유하고 있는 프로그램이나 데이터를 등록하여 사용할 수 있다.

② 사용자는 과학기술지식처리를 위한 기반 엔진으

로 사용될 프로그램 노드들을 선택하여 작업 구성 창에 올려놓는다. 그리고 프로그램의 입력으로 사용될 데이터들을 선택하여 작업 구성 창에 올려놓고 화살표를 이용하여 해당 프로그램의 입력으로 지정한다. 이때 이전 프로세스의 결과물을 다음 프로세스의 입력으로 재지정할 수 있다. 그리고 노드의 속성 창을 이용하여 프로그램의 실행 옵션 정보를 설정한다. 마지막으로 각 프로그램이 실행되기에 적합한 그리드 상의 컴퓨팅 노드를 할당한다. 이와 같은 과정을 통하여 최종적인 작업 흐름도를 생성할 수 있다.

③ 사용자는 워크플로우 도구의 작업 흐름도 실행 버튼을 누른다. 그러면 워크플로우 도구는 작업 구성 창에 생성된 최종적인 작업 흐름도를 분석하여 XML 포맷의 작업 실행규칙으로 변환하고 이것을 프레임워크로 전달한다. 이때 프레임워크는 작업 관리 아이디를 반환해준다.

④ 프레임워크는 전달된 실행규칙을 해석하여 실제적인 그리드 상의 분산 작업 명령으로 변환하고 이것을 각 컴퓨팅 노드에 전달하여 실행한다. 프레임워크는 작업 실행 과정에 변화가 있을 때마다 이벤트 통지 매커니즘을 이용하여 워크플로우 도구에게 알려준다.

⑤ 워크플로우 도구는 프레임워크로부터 전달되는 이벤트 통지를 인식하여 작업 구성 창의 작업 흐름도에 작업 실행 과정을 표시해준다. 그리고 최종적으로 작업이 끝나면 관리 아이디를 이용하여 프레임워크로부터 실행 결과를 가져와서 사용자에게 제공한다.

V. 결론 및 향후 연구

지금까지 그리드 컴퓨팅을 이용한 고성능 과학기술지식처리 프레임워크인 SINDI-Grid에 대해서 설명하였다. SINDI-Grid는 하부에 존재하는 여러 용도의 지식처리 엔진을 세분화된 독립 모듈로 개발하여 WSRF 기반의 그리드 서비스와 통합 및 연동하는 서비스 지향적인 분산 지식 마이닝 프레임워크이다. 더 나아가 이러한 서비스들을 SINDI-Workflow 도구를 이용하여 다양한 방식으로 조합함으로써 보다 향상된 지식처리 애플리케이션을 설계하고 실행할 수 있다. 뿐만 아니라 워크플로우 도구를 이용하여 생성된 애플리케이션은 그 자체로서 독립적으로 수행될 수 있는 컴포넌트이기 때문에 새롭게 구성하는 애플리케이션의 하부 모듈로 재사용할 수 있어서 애플리케이션의 개발 속도를 줄이고 확장을 쉽게 할 수 있는 장점이 있다. 향후 연구로는 대규모의 그리드 망을 이용하여 개발된 시스템의 성능을 다양한 시나리오로 평가하는 작업이 필요하다. 그리고 평가된 결과를 바탕으로 미흡한 부분을 보완하고 유지보수를 할 필요가 있다. 또한 지식처리에 필요한 여러 가지 기술들의 최신 연구 동향을 빠르게 파악하여 시스템의 하부 컴포넌트로 사용되는 기반 엔진 및 언어 자원들을 지속적으로 개선하고 보강할 필요가 있다. 이와 같이 성능 평가를 통하여 시스템의 결점을 보완하고 최신의 지식처리 기술들을 지속적으로 반영함으로써 분산 과학기술지식처리 프레임워크는 과학기술핵심개체 인식 및 인식된 개체들 간의 연관관계 추출 영역을 넘어서 과학기술분야의 새로운 기술 지식을 생성할 수 있는 유용한 프레임워크로 발전할 수 있다.

참 고 문 헌

[1] P. Brezany, I. Janciak, and A. M. Tjoa, "GridMiner: A Fundamental Infrastructure for Building Intelligent Grid Systems," The 2005 IEEE/WIC/ACM International Conference on Web Intelligence, pp.150-156, 2005.

[2] C. Goble, C. Wroe, and R. Stevens, "The myGrid project: services, architecture and demonstrator," in All Hands Meeting, pp.595-603, 2003.

[3] S. Alsairafi, F. S. Emmanouil, M. Ghanem, N. Giannadakis, Y. Guo, D. Kalaitzopoulos, M. Osmond, A. Rowe, J. Syed, and P. Wendel,

"The Design of Discovery Net: Towards Open Grid Services for Knowledge Discovery," International Journal of High Performance Computing Applications, Vol.17, No.3, pp.297-315, 2003.

[4] Nhien-An Le-Khac, Tahar Kechadi, and Joe Carthy, "ADMIRE Framework: Distributed Data Mining on Data Grid Platforms," Proceedings of 1st International Conference on Software and Data Technologies, pp.67-72, 2006.

[5] http://www.globus.org/ogsa

[6] http://www.globus.org/wsrf

[7] http://www.globus.org/toolkit

[8] D. Talia, P. Trunfio, and O. Verta, "The Weka4WS framework for distributed data mining in service-oriented Grids," Concurrency and computation : practice & experience, Vol.20, No.16, pp.1933-1951, 2008.

[9] A. Congiusta, D. Talia, and P. Trunfio, "Service-oriented middleware for distributed data mining on the grid," Journal of Parallel and Distributed Computing, Vol.68, No.1, pp.3-15, 2007.

[10] D. Talia and P. Trunfio, "How Distributed Data Mining Tasks can Thrive as Services on Grids," In Proc. of National Science Foundation Symposium on Next Generation of Data Mining and Cyber-Enabled Discovery for Innovation, 2007.

[11] A. Congiusta, D. Talia, and P. Trunfio, "Distributed data mining services leveraging WSRF," Future Generation Computer Systems, Vol.23, pp.34-41, 2007.

[12] V. Stankovski, J. Trnkoczy, M. Swain, W. Dubitzky, V. Kravtsov, A. Schuster, T. Niessen, D. Wegener, M. May, M. Rohm, and J. Franke, "Digging Deep into the Data Mine with DataMiningGrid," IEEE Internet Computing, pp.69-76, 2008.

[13] S. P. Choi, S. H. Myaeng, and H. Y. Cho, "Guiding Practical Text Classification Framework to Optimal State in Multiple Domains," Transactions on Internet and Information Systems, Vol.3, No.3, pp.285-307, 2009.

[14] S. P. Choi, C. H. Jeong, Y. S. Choi, and S. H. Myaeng, "Relation Extraction based on Extended Composite Kernel using Flat Lexical Features," Journal of KIISE : Software and Applications, Vol.36, No.8, pp.642-652, 2009.

[15] http://gdp.globus.org/gt4-tutorial

[16] A. Harrison, I. Wang, I. Taylor, and M. Shields, "WS-RF Workflow in Triana," International Journal of High Performance Computing Applications Special Issue on Workflow Systems in Grid Environments, 2007.

[17] D. Hull, K. Wolstencroft, R. Stevens, C. Goble, M. Pocock, P. Li, and T. Oinn, "Taverna: a tool for building and running workflows of services," Nucleic Acids Research, Vol.34, Web Server issue, pp.729-732, 2006.

[18] I. Altintas, C. Berkley, E. Jaeger, M. Jones, B. Ludascher, and S. Mock, "Kepler: An extensible system for design and execution of scientific workflows," 16th International Conference on Scientific and Statistical Database Management, pp.423-424, 2004.

[19] http://www.oasis-open.org/committees/wsbpel

[20] http://www.w3.org/TR/wsci

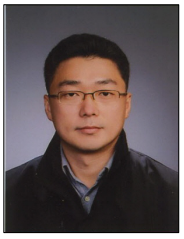
저 자 소 개

정 창 후(Chang-Hoo Jeong) 정회원

- 1999년 : 충남대학교 컴퓨터과학과 졸업(학사)
- 2002년 : 충남대학교 대학원 컴퓨터과학과 졸업(석사)
- 2003년 ~ 현재 : 한국과학기술정보연구원 정보기술연구실

<관심분야> : 정보검색 및 추출, 분산 데이터마이닝

최 성 필(Sung-Pil Choi) 정회원



- 1996년 : 부산대학교 전자계산학과 졸업(학사)
- 1998년 : 부산대학교 대학원 전자계산학과 졸업(석사)
- 2009년 : 한국과학기술원 대학원 정보통신공학과(박사 수료)
- 1998년 ~ 현재 : 한국과학기술정보연구원 정보기술연구실

<관심분야> : 기계학습, 정보검색, 자연어처리, 정보추출, 텍스트마이닝

윤 화 묵(Hwa-Mook Yoon) 정회원

- 1992년 : 서울산업대학교 전자계산학과 졸업(학사)
- 1997년 : 공주대학교 대학원 전자계산학과 졸업(석사)
- 2008년 : 배재대학교 컴퓨터공학과 졸업(박사)
- 현재 : 한국과학기술정보연구원 정보기술연구실 책임연구원

<관심분야> : 데이터베이스, 정보검색, 온톨로지

최 윤 수(Yun-Soo Choi) 정회원

- 1993년 : 충남대학교 컴퓨터공학과 졸업(학사)
- 1995년 : 충남대학교 대학원 컴퓨터공학과 졸업(석사)
- 1995년 ~ 현재 : 한국과학기술정보연구원 선임연구원

<관심분야> : 데이터베이스, 정보검색