

그리드에서 작업 흐름을 효과적으로 제어하기 위한 JCML과 GUI 기반의 편집기 (A JCML and a GUI-based Editor for Specifying Job Control Flow on Grid)

황 석 찬[†] 최 재 영^{**} 이 상 산^{***}
(Seogchan Hwang) (Jaeyoung Choi) (Sangsan Lee)

요 약 그리드는 기존의 컴퓨팅 환경을 대체하는 새로운 환경으로 각광받고 있다. 그러나 그리드가 원천적으로 가지고 있는 이질적인 특성으로 인한 사용상의 복잡함 때문에 일반 사용자가 사용하기 어렵다. 본 논문에서는 그리드 환경에서 사용자에게 편리하고 향상된 작업 환경을 제공하고 효율적인 작업의 수행을 돕는 JCML(Job Control Markup Language)과 편집기에 대해 소개한다.

JCML은 기존의 글로버스에서 사용하는 RSL을 향상시킨 작업 제어 언어이다. JCML은 표준 XML 문서 형식으로 설계되어 그리드 서비스와의 데이터 호환성을 제공하며, 작업의 표현과 의존성 정보를 표시하기 위해 GXL(Graph eXchange Language)을 이용한 그래프 방식으로 노드와 에지를 사용하여 두 노드간의 의존성을 표현한다. 그리고 JCML 편집기는 사용자가 직관적으로 이해할 수 있도록 GUI 기반의 비주얼 환경을 제공하며, 끌어다 놓기와 같은 간단한 조작만으로도 복잡한 작업 지시를 수행할 수 있도록 구성되었다.

키워드 : 그리드 미들웨어, RSL, Job Control Markup Language, GUI 편집기

Abstract The Grid system is an emerging computing infrastructure that will substitute for existing distributed systems. However end users have a difficulty in using the Grid because of its complicated usage, which is an inherent characteristic from the heterogeneous mechanism of the Grid. In this paper, we present the JCML(Job Control Markup Language) and its GUI-based editor, which not only provide users with ease of use, improved working environment, but assist users to execute their jobs efficiently.

The JCML is a job control language that improves the RSL of Globus, which defines global services in Grid. The JCML is designed to support flexibility among various Grid services using standard XML. And it makes use of a graph representation method, GXL(Graph eXchange Language), to specify detailed job properties and dependencies among jobs using nodes and edges. The JCML editor provides users with GUI-based interface. With the JCML editor, a complicated job order can be easily completed using very simple manipulations with a mouse, such as a drag-and-drop.

Key words : Grid Middleware, RSL, Job Control Markup Language, GUI Editor

1. 서 론

그리드는 기존의 병렬, 분산 컴퓨팅 환경이 가지고 있는 한계를 극복하고자 1990년대부터 연구되고 있는 컴

퓨팅 환경이다. 그리드는 지리적 또는 물리적으로 분산되어 있는 이기종의 자원을 공유하여 단일 작업 환경에 응용될 수 있는 시스템을 제공한다. 그리드를 이용하면 여러 기관이 가지고 있는 컴퓨팅 자원을 동시에 이용할 수 있게 되므로, 기존의 컴퓨팅 환경이 갖고 있는 능력 이상의 성능을 제공받을 수 있고, 각 컴퓨팅 자원의 효율성을 크게 향상시킬 수 있다[1,2].

그러나 그리드 환경을 구축하기 위해서는 해결해야 할 문제들이 많이 있다. 기존의 컴퓨팅 환경은 서로 다른 시스템 구조와 운영 환경 등 물리적 또는 논리적인

[†] 비 회 원 : 송실대학교 컴퓨터학부
seogchan@ss.ssu.ac.kr

^{**} 종신회원 : 송실대학교 컴퓨터학부 교수
choi@comp.ssu.ac.kr

^{***} 비 회 원 : (주)다산네트워크 연구본부장
sslee@da-san.com

논문접수 : 2003년 2월 21일

심사완료 : 2003년 11월 21일

로 다른 구조를 갖고 있으며 자원들의 체계가 일정하지 않아서 접근이 쉽지 않다. 이러한 문제들을 해결하기 위해서는 표준화된 기반의 하부 구조가 필요하다. 대표적으로 미국의 아르곤 국립 연구소(ANL)를 추축으로 한 연구팀은 글로버스 툴킷(Globus Toolkit)[3]을 제안하였다. 글로버스는 이기종 자원의 단일한 접근 방법을 제공하고 필요한 자원의 검색을 위해 디렉토리 서비스를 제공한다. 이 밖에도 원격지 데이터의 전송 방법, 사용자의 인증과 보안 방법 등을 제공한다.

글로버스는 하위 자원들에 대한 단일화된 접근방법을 제공하지만 상위 계층의 그리드 어플리케이션이 요구하는 다양한 응용 서비스를 모두 충족시킬 수 없다. 이를 해결하기 위해서는 추가적인 서비스에 대한 연구가 필요하다. 글로버스 기반의 그리드에서 작업을 수행시키기 위한 명령을 기술하는 방법으로 RSL(Resource Specification Language)[4]를 이용한다. 텍스트 기반의 명령 형식인 RSL은 일반 연구자가 이용하기 어렵고, 조금 복잡한 작업 정의를 하기 위해서는 매우 복잡한 다중 문항의 RSL 문법을 이해해야 하며, 또한 작업들 간의 의존 관계를 표현할 수 없다.

이러한 문제점을 개선하고자 본 논문에서는 그리드의 작업 명령을 기술할 수 있는 언어인 JCML(Job Control Markup Language)과 사용자 중심의 편리한 그래픽 방식의 편집기를 제안한다. JCML은 그리드를 이용하는 작업의 흐름을 제어할 수 있는 마크업 언어이며, 기존의 RSL 형식의 부족한 부분을 개선하여 작업들 간의 의존성 및 제약 사항을 자세히 기술할 수 있다. 그리고 편집기는 사용자가 직관적으로 이해할 수 있도록 GUI 기반의 비주얼 환경을 제공하여 간단한 조작만으로도 복잡한 작업 지시를 수행할 수 있도록 구성하였다.

JCML은 그리드를 필요로 하는 어플리케이션 중에서도 여러 곳에 분산된 데이터를 이동하여 처리하거나 다단계의 복잡한 처리 절차를 갖는 작업을 배치 작업처럼 흐름을 정의하여 작업을 단계별로 수행하는 어플리케이션을 잘 표현할 수 있다.

본 논문의 구성은 다음과 같다. 2장에서는 본 연구와 관련된 연구를 살펴보고, 3장에서는 JCML을 위한 요구사항 분석과 설계, 편집기의 설계와 구현을 다룬다. 4장에서는 향후 연구 방향에 대한 토의와 결론을 맺는다.

2. 관련 연구 비교

그리드 환경에서는 분산시스템과 같이 단일 자원이 아닌 여러개가 물리적으로 분산되어 있고 특히 규격과 형식이 서로 다른 자원을 공유하여 사용하기 위해서 각 자원의 정보를 표현하는 방법이 필요하다. 그리고 분산된 자원을 이용하는 작업의 경우에도 마찬가지로 작

업을 분리시켜 처리하기 위해서는 작업의 특징과 필요한 자원, 처리 프로세스를 충분히 기술할 수 있는 방법이 필요하다.

Condor[5]에서 사용되는 ClassAd[6]는 유휴 컴퓨팅 자원정보를 기술하고 이를 필요로 하는 작업에서 필요로 하는 자원에 대한 요구조건을 기술할 수 있는 방법을 제공한다. ClassAd는 분산 배치 작업에 대한 명세를 바탕으로 자원 풀(pool)에 등록된 자원 명세를 비교하여 적합한 자원과 작업을 연결시켜 주는 역할을 수행한다.

RSD(Resource and Service Description)[7]는 CCS(Computing Center Software)[8] 프로젝트를 구성하는 자원에 대한 내용과 관리방법을 기술하는 언어이다. RSD는 센터내에 존재하는 메타 컴퓨터에 대한 자원의 명세를 기술하여 사용되며 주로 자원의 관리 측면이 강조되어 자원 관리면에서는 다양하고 매우 강력한 성능을 나타내고 있지만 작업 프로세스를 기술하고 관리하는 부분은 매우 부족하다.

GSFL(Grid Service Flow Language)[9]은 그리드의 OGSA(Open Grid service Architecture)[10]를 바탕으로 하는 웹 서비스를 이용한 작업의 흐름을 기술할 수 있는 방법을 지원하기 위해 진행중인 프로토타입 모델이다. GSFL은 WSDL[11]에 기반하여 웹 서비스간의 상호작용에 대한 작업 정의를 할 수 있다. 그러나 웹 기반의 서비스만을 이용할 수 있는 작업에만 적용할 수 있는 방법으로서 기존의 어플리케이션이 웹 서비스 형태의 작업을 지원해야만 가능하다.

ClassAd와 RSD는 기존의 분산 시스템에 대한 연구의 일부분으로서 전용의 시스템에 대한 의존적인 목표를 가지고 설계되었기 때문에 그리드 환경에서 요구되는 작업을 수행하는데 필요한 처리 절차와 특히 복잡한 작업이 갖을 수 있는 프로세스간의 의존관계(consumer-provider 관계)에 대한 처리 방법을 지원하지 못한다. ClassAd는 작업과 자원의 연결 방법만을 지원하며 작업에서 요구될 수 있는 처리 절차에 대한 프로세스의 정의를 수행할 수 없으며, RSD는 자원에 대한 정확한 명세를 구성할 수 있는 장점이 있으나 ClassAd와 마찬가지로 작업처리에 대한 프로세스를 정의할 수 없다. GSFL은 작업 처리에 대한 의존관계 부분을 명세할 수 있고 전체적인 작업의 흐름을 기술할 수 있지만, 웹 기반의 서비스만을 기준으로 작성되어서 웹 기반 서비스가 아닌 기존의 일반 분산 환경에서 수행되는 어플리케이션에 적용할 수 없는 점과 함께 현재 OGSA를 만족시키는 그리드 환경을 위한 글로버스 미들웨어(GT3)에 대한 연구가 진행중에 있어 적용할 수 있는 플랫폼이 아직 없다는 문제점이 있다. 현재 그리드 포럼(GGF)[12]에서는 이질적인 배치 시스템을 갖는 그리드

에 작업을 제출할 수 있는 표준적인 작업 제출 방법에 대한 정의를 위해 워킹그룹(JSDL, Job Submission Description Language)[13]을 운영하여 연구하고 있다.

JCML은 기존의 구축 가능한 그리드 환경(GT2)에서 분산 환경의 어플리케이션을 수정하지 않고 그대로 사용할 수 있으며 어플리케이션의 복잡한 처리와 의존관계를 갖는 작업에 대한 순차적인 프로세스를 정의할 수 있는 방법을 제공한다.

3. JCML 시스템

그리드 시스템은 그 자체가 다양한 환경을 연결하는 이질적인 성격을 갖고 있다. 그리드를 구성하는 각 시스템은 시스템 구조부터 운영 방법까지 모두 다를 수 있기 때문에, 일반 사용자가 각 시스템의 특성을 모두 이해하고 그 특성에 맞도록 작업을 실행시킨다는 것은 매우 어렵다. 일반적으로 그리드에서는 글로벌스에서 사용되는 RSL과 같은 작업 제어 언어를 이용하여 규격화된 작업 정의 리스트(Job Definition List)를 작성하고, 이를 각 시스템에서 분석하여 그에 대응하는 작업을 한다.

작업 제어 언어로 작성된 작업 정의 리스트는 그리드의 상위 레벨에서 하위 시스템으로의 스케줄링과 관련된 정보를 제공한다. 작성된 작업 정의 리스트는 하위의 각 단계를 거치며 작업과 관련한 정보를 해당 서비스에 제공한다. 예를 들면, 자원 스케줄링 단계에서는 자원 브로커 서비스에 의해 분석되어 최적의 자원을 선택할 수 있는 정보를 주며, 실행 단계에서는 각 프로그램이 실행될 때 다른 프로그램의 의존성에 대한 제어 정보와 입출력 데이터의 정보를 준다. 마지막으로 각 프로그램의 최종 결과 값의 처리에 관한 정보를 작업 정의 리스트를 통해 얻을 수 있다.

그림 1은 JCML을 그리드에 적용한 예를 나타내었다. 사용자가 JCML 편집기를 이용하여 작업 정의 리스트를 작성하면, 자원 브로커(Resource Broker), 작업 스케줄러(Job Scheduler), 데이터 관리자(Data Manager) 등의 그리드 서비스에서 이를 이용할 수 있다. 자원 브로커에서는 작업 수행에 필요한 자원을 선택하고 예약하는 자원 스케줄링에 필요한 정보를 얻을 수 있으며, 작업 스케줄러에서는 어떠한 작업이 어떠한 작업 노드에서 수행되어야 하는지 또는 어떤 작업이 우선으로 수행되어야 하는지 등의 정보를 얻을 수 있다. 그리고 데이터 관리자에서는 사용자 작업에 필요한 데이터의 위치와 이동에 관련된 정보를 파악하여 빠른 수행을 도울 수 있도록 데이터 프리페치(pre-fetch) 등의 관련 서비스를 수행할 수 있다.

그리드 환경에서 일반 사용자는 그리드를 구성하는 시스템에 대해서 자세히 알지 못하기 때문에, 사용하는 방법도 이해하기 쉽고 사용하기 편리해야 하고 그리드는 다양한 구조를 갖는 시스템으로 구성되기 때문에 정보의 교환에 있어서 호환성이 보장되어야 하며 자세한 정보 표현이 가능해야 한다. 그리고 편집기는 일반 사용자가 직접 조작하는 도구이기 때문에 사용상의 편리함을 우선으로 하였다.

3.1 Job Control Markup Language

JCML은 글로벌스 툴킷으로 구축되어 있는 그리드 환경에서 사용자 작업 흐름을 제어하기 위해 설계한 마크업 언어이다. JCML은 앞 절에서 논의한 작업 제어 언어가 가져야 할 기본적인 조건들을 충족시키기 위해 다음의 특징들을 갖는다.

JCML은 다양한 시스템에서 자료의 호환성을 유지하기 위해 XML[14] 기반의 문서 형식으로 구성한다. XML은 자료의 호환성을 유지할 수 있으며 또한 요소(Element)내에 속성과 다른 요소를 포함할 수 있는 구조적인 특징을 가지고 있다. 특히 XML은 거의 모든 웹 서비스에서 호환성을 보장하는 표준화된 문서 형식을 지원하기 때문에 그리드 포럼의 OGSA에서 지원하려고 하는 그리드와 웹 서비스의 결합된 서비스에 유연한 적용을 보일 수 있다.

JCML은 각 구성 요소들의 의존 관계를 기술하기 위해 그래프 방식으로 XML을 표기할 수 있는 GXL(Graph eXchange Language)[15]을 이용한다. GXL은 소프트웨어에서 표현되는 그래프 방식의 정보를 다른 형식으로 변경할 수 있는 형식을 제공하기 위해 개발된 XML 기반의 언어이며, 노드(Node)와 에지(Edge)로 표현되는 그래프를 XML 스키마를 이용하여 다양한 형식으로 변형할 수 있다.

JCML은 사용자에게 내부의 복잡함을 감추고 작업을

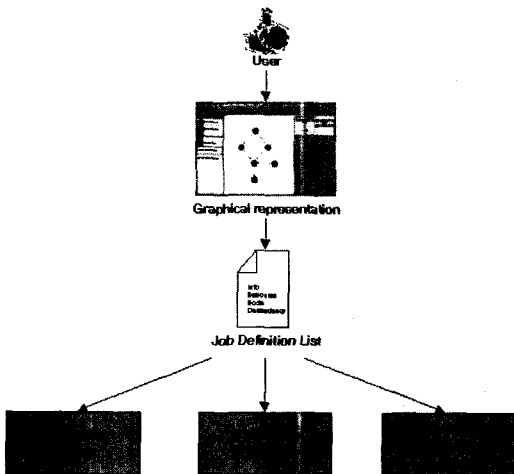


그림 1 JCML의 적용 예

기술할 수 있는 간단한 조작 방법을 제공하기 위해 그래픽 인터페이스를 같이 제공한다. JCML은 GXL 기반의 그래프 방식으로 설계되었기 때문에 그래픽 화면에서 보여주는 그래프를 이용한 노드와 에지 아이콘을 사용하여 사용자의 이해를 높일 수 있다

JCML은 XML 스키마로 정의되며 다음과 같은 구조를 갖는다. 그림 2는 XML spy[16]에서 제공하는 스키마 트리형식으로 표현된 것이다.

가. 문서 정보

JCML 문서의 이름과 문서의 범위, 요소에서 사용되는 타겟 네임스페이스를 갖는다. 그리고 작성한 사용자와 작성날짜와 함께 간단한 설명을 포함한다.

나. 자원 정보

Resource 요소는 JCML에서 기술되는 작업이 수행되기 위해 필요로 하는 하드웨어와 소프트웨어, 종료 시간 요소로 구성된다. 하드웨어 요소는 컴퓨터 구조, 중앙 처리 장치, 메모리, 디스크 용량, 네트워크의 대역폭 요소로 구성되고, 소프트웨어 요소는 운영체제, 설치된 어플리케이션,

작업 스케줄러 요소로 구성된다. 시간 요소는 사용자 작업이 종료되기를 희망하는 시간을 나타낸다.

다. 작업 구성 정보

Component 요소는 응용 프로그램이 정상적으로 수행되기 위한 여러 단계의 프로세스들을 표현하는 단위이며 노드(Node)와 그룹(Group)으로 구성된다. 노드는 태스크 노드와 자원 노드로 구분된다. 태스크 노드는 실제 작업이 실행되는 주체를 나타내며 일반적으로 컴퓨터 시스템에서 수행되는 단일 프로세스라고 할 수 있다. 자원 노드는 태스크 노드가 수행되기 위해 필요한 보조 노드로서 프로세스가 실행되기 위해 필요한 자원(저장 장치, 컴퓨터 등의 물리적 자원 장치)을 나타낸다. 태스크 노드는 실행 파일, 입출력 데이터, 환경 변수를 포함한다. 자원 노드는 해당 자원에 대한 태스크 노드가 접근하여 사용할 수 있는 정보를 포함하고 있다. 그룹은 그리드 작업을 프로세스에 따라 분리할 필요가 있을 경우 한개 또는 여러 개의 노드를 묶어 지정할 수 있는 논리적인 노드이다. 지정된 그룹은 외부에서는 하나의

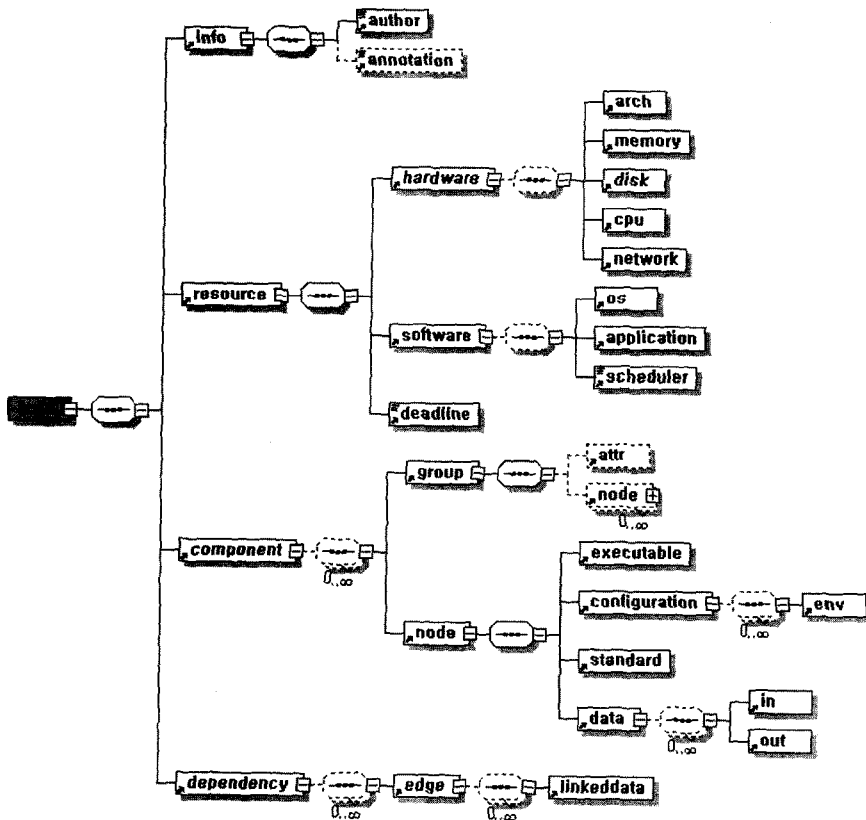


그림 2 JCML 스키마

노드로 인식되어 다른 노드와의 상호 작용을 수행한다. 그리고 그룹은 다른 그룹을 포함할 수 있다.

그림 3은 JCML을 이용하여 작업을 모델링한 예제이다. 그림에서 실선의 원형은 노드를 표시하고 있다. 전체 작업은 저장장치 S1, S2와 태스크 A, B, 그룹 A로 구성된다. 그룹 A는 다시 4개의 태스크 노드 C, D, E, F로 구성된다. 그룹 A에 속해있는 태스크들은 사용자가 다른 작업 프로세스와 별도의 구분이 필요하다고 결정하는 경우, 다른 노드와 분리되어 그룹으로 지정될 수 있다. 분리된 그룹은 다른 노드의 수행과는 상관없이 별도의 수행 과정을 거칠 수 있고 다른 노드와의 연관 관계를 가질 수 있다. 그림 3의 예제에서는 태스크 B에 의해 그룹 A의 노드들이 작업을 실행하도록 표기되고 있다. 노드들 간의 의존 관계에 대한 내용은 다음 소절에서 자세히 설명한다.

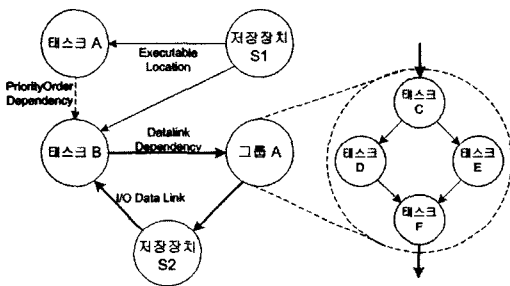


그림 3 JCML 모델

라. 의존 관계 정보

Dependency 요소는 component 요소를 구성하는 그룹 또는 노드들 간의 관계를 기술하는 에지(Edge)로 구성된다. 에지는 그룹 또는 노드를 연결하여 두 노드(또는 그룹)의 실행 순서, 입출력 데이터의 의존 관계 등을 표시한다.

에지는 우선 순위(PriorityOrder)와 데이터 링크(Data-link)로 구성된다. 그리고 에지는 방향(direction) 속성을 가지고 있어 시작점(from)과 끝점(to)으로 연결된 두 노드의 순서를 나타낸다. 우선 순위는 에지의 방향 속성에서 표시되는데 두 노드나 그룹에 대한 실행 우선 순위를 표시하지만 데이터링크는 두 노드간의 데이터의 이동도 함께 나타낸다. 데이터 링크는 태스크 노드에 포함된 입출력 데이터 요소를 참조하여 태스크 노드가 실행되기 위한 데이터의 관계를 표시하여 실행시에 관리 시스템이 이를 처리할 수 있도록 알려준다.

그림 3에서 각 노드간의 연결된 화살표는 에지를 나타내고 있다. 모든 에지는 방향을 가지고 있는 화살표로 이루어져 있으며 시작점과 끝점을 나타내고 있다. 점선

으로 된 에지는 노드간의 우선 순위를 표시한다. 우선 순위 에지는 태스크 또는 그룹 노드 간에만 사용되며 자원 노드에서는 사용되지 않는다. 태스크 A와 B는 저장 장치 S1과 실선 에지로 연결되어 있다. 이 에지는 각 태스크를 수행하기 위한 실행파일이 있는 위치를 지정하는 것이며 실제 태스크가 실행될 때에는 자원 스케줄러에서 선택된 컴퓨팅 자원으로 이동된다. 태스크 B와 그룹 A의 실선 에지는 B의 출력 데이터가 그룹 A의 입력 데이터로 사용되는 입출력 데이터의 의존 관계를 데이터 링크 요소로 표시한다. 그리고 저장 장치 S2와 연결된 에지는 태스크 B의 입력으로 S2의 데이터를 이용하며 그룹 A의 출력 데이터를 S2에 저장한다는 것을 표시하고 있다. 그룹 A의 내부 태스크 C, D, E, F는 외부(태스크 노드 B)로부터의 입력을 C가 받아 수행한 다음 출력 데이터를 태스크 D, E의 입력으로 받아 각 출력력을 F의 입력으로 넘겨준다. 태스크 F의 출력은 다시 저장 장치 S2에 저장된다.

3.2 JCML 편집기

편집기는 사용자가 직관적으로 이해할 수 있는 구성을 가져야 하며 간단한 동작만으로도 작업 정의 리스트를 작성할 수 있는 편리함이 같이 제공되어야 한다. JCML 편집기는 작업을 구성하는 요소를 하나의 객체로 나타내는 아이콘으로 표현하여 마우스로 필요한 아이콘을 선택하여 작업 화면에 끌어다 놓기만으로 간단한 작업 정의 리스트를 작성할 수 있다. 그림 4는 작업 정의 리스트를 작성중인 편집기 화면이다.

편집기는 크게 3부분으로 구성되어 있다. 좌측의 도구 모음 창에는 작업 정의 리스트에서 사용되는 노드와 에지의 종류를 표시한다. 아이콘 모양의 노드는 작업의 실행 형태와 컴퓨팅 리소스의 종류로 분류하며, 화살표 형태의 에지는 단방향과 양방향의 데이터 이동과 같은 노

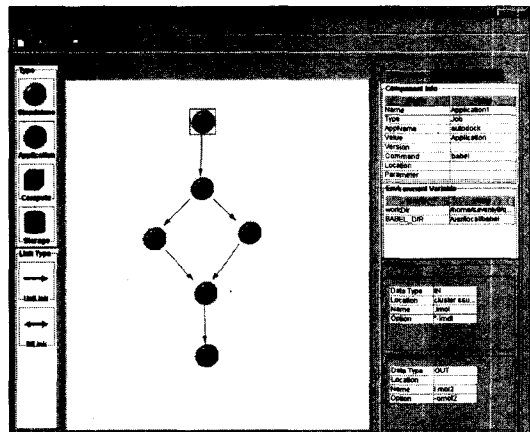


그림 4 JCML 편집 화면

드 사이의 관계를 표시한다. 가운데의 작업 창은 사용자가 원하는 형태의 작업 형태를 표현한다. 좌측의 아이콘 모음을 이용하여 작업에 필요한 프로그램들과 외부 저장 장치를 선택하여 배열한 다음, 에지를 이용하여 데이터의 이동과 같은 의존 관계를 표시한다. 우측의 속성 창은 작업 창에서 선택된 객체의 속성을 기입할 수 있는 부분으로 실행 프로그램 이름, 위치, 실행 옵션, 입·출력 데이터 위치 등과 같은 내용을 기술할 수 있다.

사용자는 좌측의 아이콘을 마우스를 이용한 드래그 앤 드롭으로 작업 창에 끌어다 놓고, 화살표를 이용하여 미리 선택한 두개의 노드 객체 사이에 에지를 연결한다. 각 객체에 속성 값을 입력하기 위해서는 먼저 마우스로 객체를 선택한 후 우측의 속성 입력창을 이용하여 입력한다.

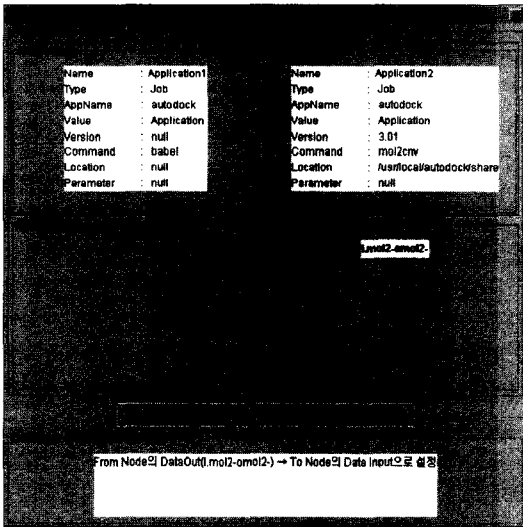


그림 5 에지 정보 편집 화면

그림 5는 두 노드 사이의 관계를 정의하는 에지 편집 창이다. 에지를 마우스의 오른쪽 버튼으로 선택하며 두 노드에 대한 정보가 나오고 각 노드의 입력과 출력 데이터를 선택하여 아래의 버튼을 누르면 데이터의 의존 관계가 성립된다.

4. 구현 및 실험

JCML과 편집 시스템은 JDK 1.4.0을 이용하여 구현하였으며, JCML 형식은 W3C의 표준 권고안인 XML 스키마 1.1 버전을 따르고 있다. 그리고 XML 파싱을 위한 도구로서는 SAX[17] 버전 2를 지원하는 아파치 그룹의 Xerces2[18] 자바 버전을 사용하였다.

본 연구에서는 JCML에 대한 실험을 위하여 생명 공

학 분야에서 신약을 설계할 때 사용되는 가상 검색 어플리케이션에 대한 테스트를 수행하였다. 가상 검색은 특정 질병에 관여하는 단백질이 활성화된 곳에 잘 반응할 것이라고 예상되는 화합물을 하나씩 결합하는 도킹(docking)이라는 과정을 통하여 신약 후보 물질을 검색하는 방법이다. 가상 검색을 하기 위해서는 화합물이 가지고 있는 고유한 형태를 도킹에 필요한 형태로 바꾸는 과정을 여러 단계 수행해야 한다. 본 실험에서 사용되는 Autodock[19]은 그림 6에서와 같은 과정을 수행한다.

그림 7은 그림 6의 Autodock 수행 단계를 JCML을 이용하여 작업을 정의하기 위한 프로세스 모델이다. 원안은 각 단계별 수행되는 실행 명령을 표시한다. babel 명령을 통해서 mol 형식의 파일을 mol2 형식의 파일로 바꾼다. mol2cnv는 화합물을 pdbq 형식으로 변환시킨다. 단백질의 pdbqs 파일과 변환된 화합물의 pdbq 파일은 동시에 mkgpf3와 mkdpf3 명령을 수행하여 각각

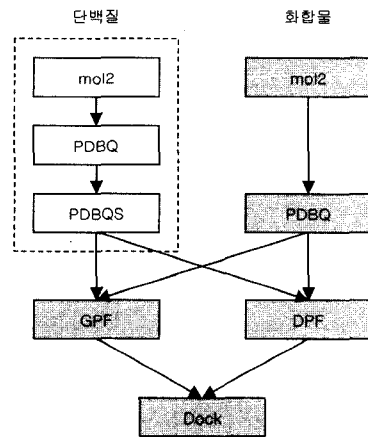


그림 6 Autodock 실행 과정

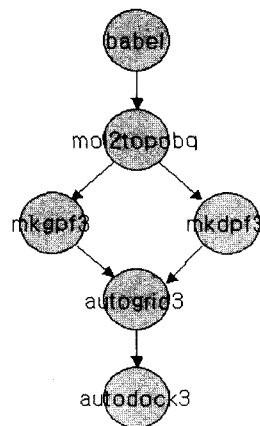


그림 7 프로세스 모델

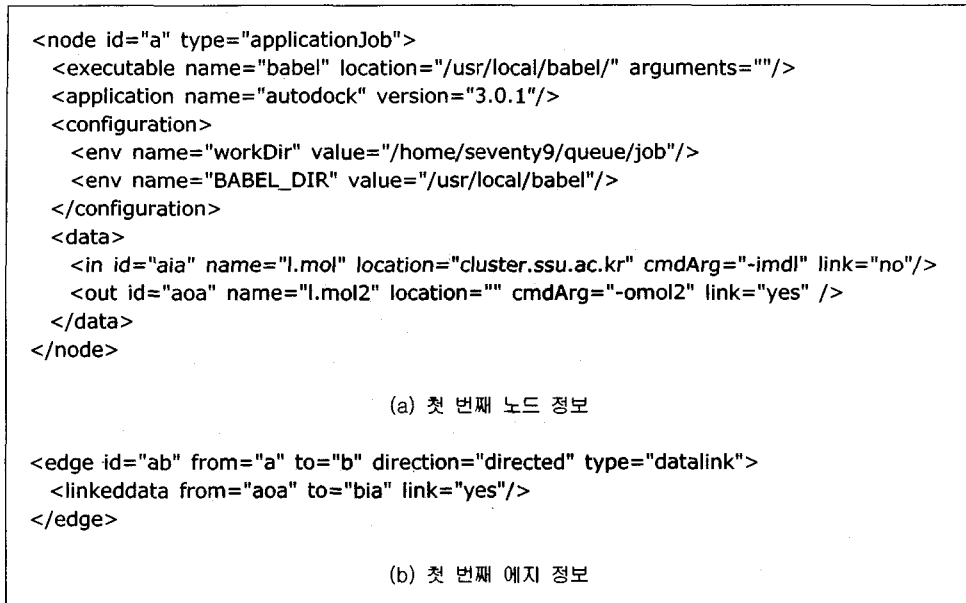


그림 8 작업 정의 리스트

gpf와 dpf 파일을 생성한다. 생성된 두 파일은 auto-grid3 명령과 autodock3 명령을 차례대로 수행하여 최종적인 에너지 값을 출력한다.

그림 7의 프로세스 모델을 이용하여 앞절의 그림 4와 그림 5처럼 JCML 편집기를 이용하여 작업을 표현한다. JCML 편집을 마치고 publish 버튼을 누르면 그림 8과 같은 작업 정의 리스트를 생성한다. 그림 8(a)에서 노드 번호는 "a"이며 타입으로 applicationJob으로 지정하고 autodock 어플리케이션을 실행 환경으로 요구한다. 환경 변수는 작업 디렉터리와 BABEL_DIR 패스를 지정한다. 입력 데이터는 cluster 컴퓨터에서 l.mol 파일을 복사하여 "-imdl"을 입력 파라미터로 지정하고, 출력 데이터는 "-omol2" 파라미터와 l.mol2 파일로 지정된다. 그림 8(b)는 노드 번호 "a"와 "b"를 연결하며 순차적인 정보를 나타낸다.

5. 결론 및 향후 연구

JCML 시스템은 사용자에게 복잡한 그리드 작업에 대한 수행 내용을 쉽고 빠르게 정의할 수 있는 환경을 제공하며 다음과 같은 특징을 갖는다. JCML은 XML 표준 스키마를 이용하여 이기종 시스템과의 데이터 호환성을 제공하며, 작업을 표현하는데 편리한 구조적인 특징과 그리드가 웹 서비스와의 연결을 위한 요구사항 [8, 20]에 빠르게 적용할 수 있도록 확장성을 제공한다. 그리고 복잡한 어플리케이션이 갖는 데이터, 또는 태스크간의 의존 관계에 대한 프로세스 모델을 표현할 수

있다. 또한 편리한 작업 환경을 위하여 그래프로 표현되는 그래픽 기반의 편집 도구를 같이 지원한다. 이와 같은 특징은 그리드를 이용하는 일반 연구자에게 매우 필요한 부분으로서 그리드 관련 연구에 매우 유용하게 쓰일 수 있다.

JCML 시스템에 대한 연구는 메타 스케줄링 프레임워크(MSF, Meta Scheduling Framework)를 개발하는 프로젝트의 한 부분으로 시작되었다. MSF는 그리드의 글로벌스 미들웨어 기반으로 그리드 서비스의 상위 레벨에서 그리드 자원과 사용자 작업에 대한 스케줄링을 수행하는 프레임워크이다. MSF는 그리드 작업에서 필요로 하는 다양한 서비스를 제공하기 위해 워크플로우 서비스를 제공한다. MSF에서 워크플로우 서비스를 제공하기 위해서는 수행되는 작업에 대한 자세한 내용을 필요로 한다. 이러한 문제를 해결하기 위해 JCML의 연구가 시작되었고 작업의 의존성 문제를 기술할 수 있는 구조를 포함하고 있다.

글로벌스에서 사용되는 서비스는 그 내용이 구현에 따라 조금씩 달라질 수 있으며 많은 서비스가 아직 구현되어 있지 않아 계속해서 그리드에 대한 새로운 연구가 진행되고 있다. 그리고 사용자의 요구 사항도 증가하므로 이에 대한 연구가 지속적으로 이루어져야 할 것이다. 또한 사용자와 그리드 시스템간의 원활한 유기적인 결함을 위한 MSF에 대한 연구도 지속되어야 할 것이다.

참고 문헌

[1] I. Foster, C. Kesselman, *The Grid: Blueprint for a New Computing Infrastructure*, Morgan Kaufmann, 1999.

[2] I. Foster, C. Kesselman, S. Tuecke, "The Anatomy of the Grid: Enabling Scalable Virtual Organizations," *International Journal supercomputer Applications*, Vol. 15, No. 3, 2001.

[3] I. Foster, C. Kesselman, "Globus: A Metacomputing Infrastructure Toolkit," *International Journal supercomputer Applications*, Vol. 11, No. 2, 1997.

[4] RSL specification v1.0, http://www-fp.globus.org/gram/rs1_spec1.html.

[5] M. Litzkow, M. Livny, M. Mutka, "Condor - A Hunter of Idle Workstations," *Proceedings of the 8th International Conference of Distributed Computing Systems*, June 1988.

[6] R. Raman, M. Livny, M. Solomon, "Matchmaking: Distributed Resource Management for High Throughput Computing," *Proceeding of the 7th IEEE International Symposium on High-Performance Distributed Computing*, July 1998.

[7] M. Brune, J. Gehring, A. Keller, A. Reinefeld, "RSD-Resource and Service Description," *Proceeding of the International Conference on High-Performance Computing Systems, HPCS98*, May 1998.

[8] A. Keller, A. Reindfeld, "Anatomy of a Resource Management System for HPC Clusters," *Annual Review of Scalable Computing*, Vol. 3, 2001.

[9] S. Krishnan, P. Wagstrom, G. Laszewski, "GSFL: A Workflow Framework for Grid Services," <http://www.globus.org/cog/papers/gsfl-paper.pdf>.

[10] I. Foster, C. Kesselman, J. Nick, S. Tuecke, "The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration," *Global Grid Forum*, June 2002.

[11] Web Service Description Language, <http://www.w3.org/TR/wsdl>.

[12] Global Grid Forum, <http://www.ggf.org/>.

[13] Job Submission Description Language Working Group, *Global Grid Forum*, 2003.

[14] Extensible Markup Language, <http://www.w3c.org/XML/>.

[15] A. Winter, B. Kullbach, V. Riediger, "An Overview of the GXL Graph Exchange Language," *Software Visualization, LNCS 2269*, pp. 324-336, 2002.

[16] XML spy, <http://www.xmlspy.com/>.

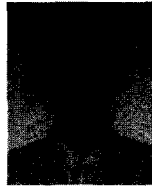
[17] Simple API for XML, <http://www.saxproject.org/>.

[18] Xerces 2, <http://xml.apache.org/xerces2-j/index.html>.

[19] AutoDock, <http://www.scripps.edu/pub/olson-web/autodock/>.

[20] S. Tuecke, et al, Grid Service Specification, Draft 4, Oct 2002, at <http://www.ggf.org/ogsi-wg/drafts/>

draft-ggf-ogsi-gridservice-04_2002-10-04.pdf.



황 석 찬

1998년 숭실대학교 컴퓨터학과 석사
2003년 숭실대학교 컴퓨터학과 박사
2003년 8월~현재 숭실대학교 컴퓨터학부 박사후과정. 관심분야는 분산/병렬 컴퓨팅, 인터넷 컴퓨팅, 시스템 소프트웨어

사진은 2003년 2월호의 정보과학회논문지: 시스템 및 이론, 제9권 제1호, 32쪽에 있습니다.



최 재 영

1984년 서울대학교 제어계측공학과 학사
1986년 미국 남가주대학교 전기공학과 석사. 1991년 미국 코넬대학교 전기공학부 박사. 1992년 1월~1994년 2월 미국 국립 오크리지연구소 연구원. 1994년 3월~1995년 2월 미국 테네시 주립대학교 연구교수. 2001년 8월~2002년 8월 미국 국립 슈퍼컴퓨팅 응용센터 (NCSA) 초빙연구원. 1995년 3월~현재 숭실대학교 정보과학대학 컴퓨터학부 부교수. 관심분야는 초고속 컴퓨팅(HPC), 병렬/분산처리, 시스템 소프트웨어



이 상 산

1984년 서울대학교 기계공학과 학사. 1986년 서울대학교 기계공학과 석사. 1992년 미국 스탠포드대학교 기계공학과 박사. 1992년~1993년 미국 NASA AMES 연구소 박사후연구원. 1993년~1994년 미국 스탠포드대학교 전임연구원. 1994년~현재 한국과학기술정보연구원(KISTI) 슈퍼컴퓨팅센터 연구원/센터장. 2003년~현재 (주)다산네트워크 부사장/연구본부장. 관심분야는 슈퍼컴퓨팅, 차세대정보통신기술, 난류 시뮬레이션