

# 그리드 기반 워크플로우 에디터 개발

조종화<sup>0</sup>, 김현대, 윤경원, 박근혜, 송하운 홍익대학교 컴퓨터공학과  
김은경, 허정민, 김윤희 숙명여자대학교 정보과학부  
박형우\* 한국과학기술정보연구원 슈퍼컴퓨팅센터

(pokoso<sup>0</sup>, hayoon)@cs.hongik.ac.kr, ifreeyou2@hanmail.net, ykw172@hotmail.com, rednugoori@dreamwiz.com  
kimek@sookmyung.ac.kr, happy98@lycos.co.kr, yulan@sookmyung.ac.kr, hwpark@hpcnet.ne.kr

## Developing a Grid-based Workflow Editor

Jong-Hwa Cho<sup>0</sup>, Hyun-Dae Kim, Kyung-Won Yoon, Kun-Hye Park, Ha-Yoon Song  
Dept. of Computer Engineering, Hongik University  
Eun-Kyung Kim, Jung-Min Huh, Yoonhee Kim  
Dept. of Computer Science, Sookmyung Women's University  
Park, Hyung Woo<sup>0</sup>, Korea Institute of Science and Technology Information

### 요약

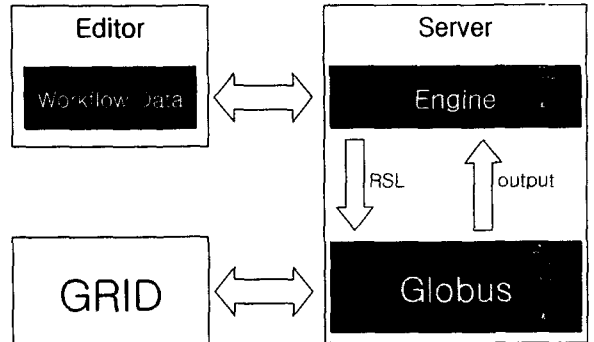
그리드 환경을 이용한 분산된 자원의 활용은 현재 많은 분야에서 활용되고 있으며 그 연구가 활발히 진행되고 있다. 그러나 자신의 분야에 관련된 PSE나 그리드용 어플리케이션이 개발되지 않으면 유저의 입장에서는 그리드 환경을 접하는 것이 어려워진다. 본 논문에서는 다양한 분야의 사용자들이 워크플로우를 직접 디자인하여 그리드의 분산된 환경을 통한 병렬처리가 가능하도록 그래픽 인터페이스를 제공하는 워크플로우 에디터의 구현에 관한 전반적인 내용을 기술하였다. 에디터의 구체적인 기능들을 살펴보면 유저에게 제공되어야 하는 기능들이 무엇인지 고찰해 본다.

### 1. GUI기반의 PSE(Problem Solving Environment)

분산된 자원을 효과적으로 사용할 수 있는 그리드 환경은 그 효율성의 뛰어난으로 인해 이미 많은 분야에서 활용되고 있으며 아직도 그에 대한 연구가 진행되고 있다. 이러한 그리드 환경의 활발한 보급을 위해서 Globus[1]가 현재 개발되고 있으며 그리드 PSE Portal 개발을 위해 GPKD(Grid Portal Development Kit)[2]나 Gridsphere[3] 등 그리드 환경을 기본으로 하는 여러 시스템들이 현재 개발 중이다.

그러나 그리드의 성능이나 그 효율성이 대단히 뛰어나다고 할지라도 그것을 사용하기 위해선 자신의 연구 분야 외에 컴퓨터 공학적인 지식을 추가로 습득하여야지만 제대로 된 그리드 환경을 사용할 수 있다. 그리드는 컴퓨터공학 분야뿐만 아니라 물리학, 생명공학, 나노테크놀로지 등 많은 분야에서 활용될 수 있지만 그리드 환경을 접하기 위해 추가로 컴퓨터공학적 지식을 습득해야 한다는 것은 대단히 비효율적이다. 따라서 유저가 그리드 환경에 쉽게 접근할 수 있도록 쉬운 조작과 GUI(Graphic User Interface)를 제공하는 그리드용 어플리케이션의 제작이 필수적이며 현재 많은 분야에서 이러한 서비스를 제공하기 위해 연구가 활발히 진행되고 있다.

또한 그리드 환경에서의 효율을 높이기 위한 워크플로우의 효과적인 디자인기능도 제공되어야 한다. 워크플로우 디자인을 통한 작업의 병렬처리는 그리드 환경에서 제공되는 분산된 자원을 효과적으로 이용하기 위해 필요하다. 유저는 워크플로우 편집을 통해 효율적으로 자신의 작업을 처리할 수 있으며 이 기능은 그리드 어플리케이션에서 지원되어야 한다. 유저는 에



[그림 1] 전체 시스템의 기본 구성

디터를 이용해 워크플로우를 직접 편집, 저장, 수행 하면서 자신의 작업해야할 전반적인 과정을 디자인할 수 있다. 이러한 과정은 복잡한 작업 과정과 많은 시간을 소비하는 작업에서 유저에게 편리한 환경을 제공하게 된다.

본 논문에서는 유저가 복잡한 그리드 배경지식을 알고 있지 않아도 그리드 환경을 사용할 수 있고 유저가 직접 편집한 워크플로우를 수행할 수 있는 에디터와 그 전반적인 구현 과정에 대해서 서술한다. 2절에서는 전체 시스템의 구성과 에디터의 기능을 제공하기 위해 서버 측에 필요한 엔진, 유저 측에서 사용되는 에디터에 대한 기본적인 기능을 설명하며 3절에서는 에디터의 각 기능들을 설명한다. 4절에서는 결론과 향후 과제 등에 관해서 생각해 본다.

### 2. 시스템의 구성

그리드 환경을 구축하기 위해서 Redhat Linux 7.2버전과

\* 본 연구는 2003년 한국과학기술정보연구원의 국가 그리드 기반 구축과제에서 지원을 받았습니다.

GT(Globus Toolkit) 2.2.4버전을 사용하였다. GT는 [1]에서 다운로드 할 수 있다. 미들웨어인 글로버스를 설치하고 전반적인 시스템의 기본 형태를 잡았으며 그리드 환경을 통한 프로그램의 실행, 전체 클러스터 간의 인증, 노드의 설정 등 그리드 환경을 구축하기 위한 여러 옵션들을 설정하였다. 전체적인 시스템의 구성은 [그림 1]과 같다. 각 구성은 그리드 관련 작업을 처리하는 미들웨어인 글로버스, 에디터와 글로버스간의 상호 데이터 처리와 부가적인 정보를 에디터에게 전송해주는 엔진, 그리고 사용자가 직접 워크플로우를 편집하고 그리드 환경에서 수행할 수 있게 해주는 GUI기반의 워크플로우 에디터로 분류된다.

### 2.1 엔진

에디터와 글로버스간의 상호 데이터 교환을 위해서 서버 측에 엔진을 개발하고 설치하였다. 엔진은 자바를 이용하여 구현하였으며 글로버스와의 연동을 위한 내부적인 기능은 CoG Kit[4]을 이용하여 구현하였다.

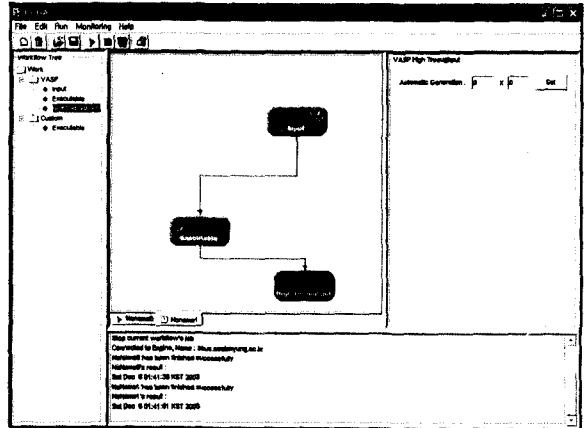
엔진은 에디터에서 전송되어진 워크플로우 데이터를 분석하고 글로버스에서 수행될 수 있도록 RSL(Resource Specification Language)[5]형식으로 변환하는 역할을 담당한다. 사용자가 디자인한 워크플로우에 따라 엔진은 각 작업들을 상황에 맞게 처리하게 되며 전체적인 작업이 특정한 순서를 따라 실행되거나 여러 작업이 병렬 처리되는 과정을 엔진은 워크플로우를 분석하여 처리하게 된다.

CoG Kit을 이용하여 특정 작업을 글로버스를 통해 수행하면 실시간으로 모니터링 정보를 얻게 된다. 따라서 엔진은 이 모니터링 정보를 XML 형식으로 변환, 에디터에게 전송한다. 작업에 따라 결과가 터미널에 출력되거나 결과 파일의 형태로 나오는 경우가 있으며 엔진은 이 두 가지의 경우를 고려하여 처리하도록 되어있다. 터미널에 출력되는 경우 출력 결과를 에디터에게 전송하며, 파일의 형태로 결과가 나오는 경우 글로버스에 의해 제공되는 Gridftp를 사용하여 결과 파일을 에디터에게 전송한다.

엔진은 작업의 수행 현황을 로그파일로 제작하여 엔진 내부에 저장한다. 이 파일은 사용자가 수행한 작업의 실행에 대한 정보나 시작·종료 시간 등을 담고 있어서 사용자가 요구할 경우 에디터를 통해 유저에게 제공하게 된다.

### 2.3 워크플로우 에디터

그리드의 분산된 환경의 효율성은 대단히 뛰어나지만 유저는 복잡한 인터페이스와 반복되는 작업 그리고 전체 워크플로우의 일괄적인 표현이 불가능하여 접근하는데 많은 어려움이 따른다. 워크플로우 에디터는 유저에게 GUI환경과 워크플로우 편집 기능을 제공하여 이와 같은 어려움을 완화시킨다. 에디터를 통해 유저는 자신이 디자인한 워크플로우를 그리드 환경에서 쉽게 수행시킬 수 있으며 그 결과를 파일이나 그래픽 형태로 제공받게 된다. 또한 워크플로우 디자인을 통한 작업의 일괄 처리와 워크플로우 저장 기능은 반복적인 작업을 줄이게 되어 작



[그림 2] 워크플로우 에디터

업의 능률을 효과적으로 증가시킨다.

워크플로우 에디터는 유저가 디자인한 워크플로우를 엔진으로 전송하여 글로버스를 통한 그리드의 분산된 자원을 활용하도록 하며 수행되는 작업의 모니터링 정보와 작업 로그 정보를 제공하여 유저가 그 진행 상황을 일괄적으로 파악할 수 있게 한다.

에디터에서 제작된 워크플로우 데이터는 XML형태로 변환되어지며 이 데이터는 엔진으로 전송되어 처리된다. 또한 유저가 디자인한 워크플로우를 파일 형태로 저장하는 경우 이 파일의 형태도 XML로 이루어진다.

## 3. 에디터의 주요 기능

### 3.1 GUI(Graphic User Interface)

GUI는 그리드의 복잡한 텍스트 인터페이스를 다양한 분야의 사용자들이 쉽게 접근할 수 있는 환경을 제공한다. 그래픽을 이용하여 워크플로우의 전체 흐름을 쉽게 파악할 수 있으며, 모니터링 정보, 결과 등을 그래픽으로 표현하여 유저가 전체 작업의 수행과 결과의 도출을 효과적으로 하도록 한다.

[그림 2]에서 볼 수 있듯이 에디터의 각 기능들을 유저가 한 눈에 쉽게 파악할 수 있다. 워크플로우를 생성하는데 필요한 오브젝트를 쉽게 추가할 수 있도록 Object List Tree를 왼쪽 창에 제공하여 유저는 마우스의 Drag&Drop만으로 원하는 오브젝트를 쉽게 추가할 수 있다. 생성된 오브젝트를 마우스로 클릭하면 오른쪽에 설정을 할 수 있는 창이 제공되며 유저는 오브젝트의 설정에 필요한 항목을 설정하게 된다.

자주 사용되는 기능을 에디터의 상단에 아이콘으로 배치하여 유저는 마우스의 클릭만으로 필요한 기능을 사용할 수 있다. 즉 새로운 워크플로우 편집 창을 생성하거나 유저가 생성한 워크플로우를 저장할 때 등 에디터에서 제공되는 기능을 사용하는 경우 유저는 마우스로 원하는 기능을 사용할 수 있다.

에디터의 하단부는 작업의 수행 결과, 워크플로우 디자인의 오류, 모니터링 정보 등 필요한 Text 정보를 표시한다. 실행 결과가 터미널을 통해 출력되는 작업의 경우 엔진에서부터 전송되어지는 결과가 이 창을 통해 출력되어진다. 따라서 유저는 자신이 디자인한 워크플로우를 확인하면서 모니터링 정보와 작

업의 결과 등을 한눈에 파악하게 된다.

### 3.2 XML기반의 데이터

에디터 내부의 모든 데이터와 엔진으로 전송되는 워크플로우 정보는 모두 XML로 처리된다. 사용자가 디자인한 워크플로우는 작업에 따라 입력 오브젝트, 실행 오브젝트, 결과 오브젝트 등으로 세분화되며 각각에 필요한 내부 데이터를 XML로 정의하였다. 입력 오브젝트의 경우 실행 오브젝트의 입력 데이터로 쓰이며 실행 오브젝트에 필요한 여러 입력 데이터를 포함한다. 실행 오브젝트는 작업이 실행되는데 필요한 클러스터의 정보와 필요한 노드의 수, 인자 값 그리고 전체 워크플로우의 스케줄 정보 등을 포함한다. 우선순위에 관한 데이터가 이 부분에 존재하며 엔진은 이 데이터를 분석하여 모든 작업의 우선순위를 설정하게 된다. 그리고 출력 오브젝트의 경우 다른 실행 오브젝트의 입력으로 다시 쓰이는 경우에 대한 정보를 포함하여 전체 워크플로우를 일괄적으로 처리할 수 있는 데이터를 포함하게 된다.

워크플로우를 파일로 저장하는 경우 그 파일의 형태는 XML으로 이루어지며 에디터 내부에서 필요한 각 오브젝트의 위치 정보, 오브젝트간의 연결정보, 내부 설정 정보 등을 포함하여 사용자가 텍스트 에디터를 통해 저장된 파일을 확인하여도 대략적인 워크플로우 정보를 파악하는 것이 가능하다.

XML Parser는 JDOM<sup>[6]</sup>을 사용하였다. JDOM은 XML 데이터를 트리 형태로 제공해주는 기능을 갖고 있어서 XML 태그 정보를 읽거나 XML Document를 생성할 때 등 필요한 기능을 제공해 준다.

### 3.3 워크플로우 편집기능

에디터는 사용자가 원하는 워크플로우를 쉽게 제작·편집할 수 있도록 제작되었다. 따라서 에디터를 사용하는 유저는 원하는 워크플로우를 직접 제작하거나 내부 데이터를 수정할 수 있으며 워크플로우 내부 각 작업들의 스케줄링을 통해 작업의 우선순위를 줄 수 있다. 전체적인 스케줄링 제어는 유저가 수작업으로 처리해야 하는 복잡한 작업 과정을 워크플로우의 디자인을 통해 일괄적으로 처리하게 해준다. 또한 필요에 따라 각 작업들이 결과를 공유해야 하는 경우 먼저 실행된 작업의 결과를 다음 작업의 입력으로 제공한다. 이 경우 작업 간의 의존성으로 인해 병렬처리는 불가능해지나 반복되는 작업을 최대한 줄이고 처리의 자동화를 제공하게 된다.

병렬처리가 가능한 작업들의 경우 유저는 워크플로우 편집을 통해 필요한 작업들을 병렬처리 시킬 수 있다. 병렬 처리가 가능한 작업은 글로벌스에 의해 분산된 시스템에서 처리되며 유저는 단일 시스템에서 작업을 수행할 때 보다 더 좋은 환경에서 작업을 처리하는 것이 가능해진다.

유저는 특정 작업을 원하는 클러스터에서 몇 개의 노드를 사용하여 수행할 것인지 선택할 수 있다. 따라서 유저는 가장 효율이 좋은 클러스터를 선택할 수 있으며 한 작업을 몇 개의 노드를 사용하여 돌릴 것인지 직접 선택하게 된다. 클러스터의

상태에 따라 같은 작업을 수행하는데 차이가 있으며 제공되는 노드의 수가 다르므로 유저는 각 상황에 맞는 설정을 통해 작업을 수행하게 된다.

### 3.4 실시간 모니터링과 작업 로그 정보

유저는 실행한 작업의 모든 상황을 에디터를 통해 실시간으로 모니터링 하게 되며 작업의 실행상태 정보를 확인하게 된다. 모니터링의 경우 각 상태를 세분화 하여 제공해 주며 각 상태는 작업이 수행되기 전 준비 상태를 나타내는 Ready 상태, 클러스터에서 작업이 현재 대기 중임을 나타내는 Pending 상태, 대기 중이던 작업이 실행을 시작하고 현재 실행 중임을 나타내는 Active 상태, 성공적으로 실행을 마친 후인 Done 상태, 작업이 실행되지 못했거나 실행 중 어떠한 문제로 인해 수행이 실패한 Failed 상태 등으로 나뉘어져 있다. 모니터링 정보는 유저가 편집한 워크플로우 오브젝트의 색 변화로 표시되며 각 상태 정보의 색을 구분지어서 쉽게 파악이 가능하다.

유저는 작업의 수행 여부나 시간 정보 등을 에디터를 통해 제공받게 된다. 엔진은 내부에서 처리되어진 모든 작업들의 로그 정보를 관리·저장하며 유저는 에디터를 통해 그 정보를 받게 된다. 이와 같은 로그 정보의 제공을 통해 유저는 자신이 수행한 워크플로우의 전체 소요 시간과 워크플로우 정보 등을 확인하게 된다.

## 4. 결과 및 향후 과제

본 논문에서는 그리드 환경과 워크플로우 편집 기능을 제공하는 워크플로우 에디터의 전반적인 내용에 대해서 기술하였다. 에디터의 GUI를 통해 유저는 복잡한 그리드 환경을 쉽게 접근할 수 있으며 워크플로우의 디자인을 통해 작업의 일괄적인 처리를 할 수 있다. 또한 그리드를 이용한 작업의 병렬처리는 분산된 환경을 이용한다는 장점을 최대한 이끌어 내게 된다.

그러나 현재 워크플로우의 표현법이나 내부 디자인 등에 대한 국제 표준이 확정되지 않아서 에디터를 일반화 하여 모든 분야에서 사용 가능하게 하는 것은 불가능 하였다. 하지만 관련된 국제 표준이 확정된다면 여러 분야에서 전반적인 활용을 할 수 있는 에디터를 개발할 수 있게 되리라고 보여진다. 그렇게 되면 에디터가 특정 분야에 한정되어진 것이 아닌 다양한 분야에서 활용이 가능하도록 제작되어질 수 있을 것이라 판단된다.

### 참고문헌

- [1] The Globus, Alliance <http://www.globus.org/>
- [2] J. Novotny., "The Grid Portal Development Kit", *Concurrency: Pract. Exper.* Vol. 00, pp.1-7, 2000
- [3] The Gridsphere, <http://www.gridisphere.org/gridisphere/gridisphere>
- [4] CoG kits, <http://www-unix.globus.org/cog/>
- [5] RSL, [http://www-fp.globus.org/gram/rs1\\_spec1.html](http://www-fp.globus.org/gram/rs1_spec1.html)
- [6] jdom, <http://www.jdom.org>