# Multi-Grid and Multi-VO Job Submission based on a Unified Computational Model


G. Pierantoni, B. Coghlan, E. Kenny, O. Lyttleton, D. O'Callaghan, G. Quigley

Department of Computer Science,Trinity College Dublin, Ireland
*email:*
[pierantg,coghlan,ekenny,lyttleo,david.ocallaghan,gquigle]@cs.tcd.ie
*phone:* (+353 1)608 1536, *fax:* (+353 1) 677 2204



Keywords: multi-grid, multi-VO, WebCom, metagrid, interoperability.



**Abstract**

Interoperability in Grid Computing is the research field that aims at using heterogeneous Grids within a single metagrid environment. In order to achieve this goal, two main issues have to be addressed: there must be a bridge infrastructure able to communicate with different Grid middlewares and there must be a unified model capable of expressing the workflow of operations to be executed by the different Grid platforms. In this paper we describe an architecture for interoperability that exploits the power of a Computational Model known as Condensed Graphs to express workflows and that uses GT4.0 WSRF Web Services as a bridge technology to submit jobs that encompass WebCom, LCG2 and GT4.0 submissions.

The proposed architecture is based on a design concept of internal and border regions: An internal region is a set of machines that run either metagrid, WebCom, or a homogeneous set of Grid Services, while a border region is a set of machines where WebCom instances and Grid services can coexist and interact with metagrid services.

To design and implement this system two processes were used in parallel: A top-down approach led to Analysis and Design documentation while a bottom-up approach produced a set of prototypes to evaluate the soundness and feasibility of the solutions proposed by the analysis and design process. The interoperability infrastructure has been successfully implemented and is now undergoing a set of stress tests to evaluate its capacity to cope with computationally demanding workflows.


## 1 Introduction

The Grid concept gives rise to many different implementations. These may be data-centric, computation-centric or a hybrid with each having particular strengths and weaknesses. This opens a wide and interesting field of research which aims to harness the different capabilities of each of these Grid implementations in a metagrid. To use heterogeneous Grids within a single environment two main issues have to be addressed; there must be a bridge infrastructure able to communicate with different Grid middlewares and there must be a unified model capable of expressing the workflow of operations to be executed by the different Grid solutions.

While not the only possible solution, we contend that the Unified Computational Model [4] known as Condensed Graphs [3] is ideal for expressing such workflows, where its implementation (WebCom [1, 2, 5]) allows its execution with eager or lazy evaluation policies. At the same time stateful Web Services, such as those provided by the Globus Toolkit 4 (GT4), offer a promising technology for the implementation of the bridge infrastructure. In this paper we describe an architecture that demonstrates this combination to exploit the power of the Condensed Graphs Unified Computational Model to express complex workflows of operations that are executed in a metagrid environment encompassing WebCom, the LHC Grid (LCG2) and GT4.

To implement this solution a design concept of internal and border regions is used: An internal region is a set of machines that run either metagrid, WebCom, or a homogeneous set of Grid Services (either LCG2 or GT4.0 Services) while a border region is a set of machines where WebCom instances and Grid services can coexist and interact with metagrid services. The metagrid services provide the bridge infrastructure so that the border region effectively extends the internal sets such that they overlap with the metagrid set.

In the metagrid region, a set of services allows the communication and the interoperability among the Grid Services hosted in the internal regions. These metagrid services are implemented using one or more metagrid technologies. The services necessary for interoperability include file staging, job submission and security, while a logging service, although not strictly necessary, is implemented.

## 2 Condensed Graphs representing Multi-Grid Jobs

The Condensed Graphs Computational Model is based on directed acyclic graphs in which every node has not only operand ports, but also an operator and a destination port and where the flow of entities on arcs are used to trigger execution. Nodes can be condensations, or abstractions, of other Condensed Graphs (CGs). CGs can thus be represented (i.e. condensed) to a single node in a graph at a highler level of abstraction. Conversely, the condensation can be undone (i.e. evaporated) to reveal the internal CG. When the proper entities (operands, operators and destination, all described by CGs) are present at all the ports of a node, the node can be fired, resulting in the execution of the instructions it represents.

With Condensed Graphs, workflows of jobs to be executed on multiple grids can be represented as follows. The dependencies among the different jobs are represented with the topology of a Condensed Graph such as the one represented in Fig.1. This Condensed Graph represents a workflow that comprises 6 jobs and their dependencies. The jobs are:

- WebCom nodes (Nodes E,X,$W_1$ and $W_2$) that have to be executed sequentially in the WebCom internal region.

- E and X are particular WebCom nodes that represent the beginning and end of a CG.
  - W1 and W2 are general WebCom jobs.
- LCG2 nodes (Nodes $L_1$ and $L_2$) that have to be executed sequentially in the LCG2 internal region.
- GT4 nodes (Nodes $G_1$ and $G_2$) that have to be executed in parallel in the GT4 internal region.

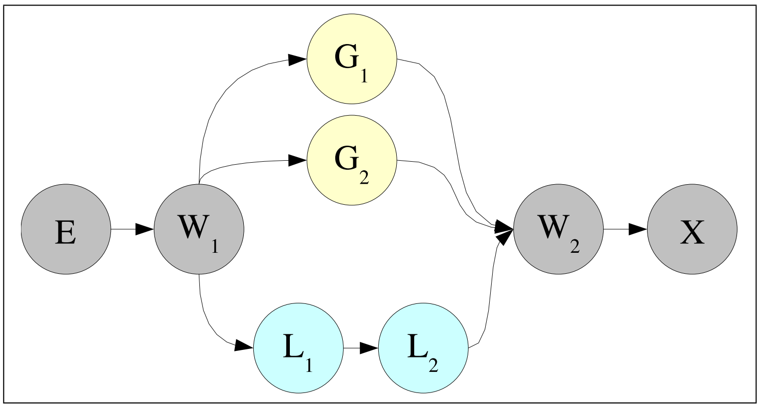


Fig. 1: A Condensed Graph representing a Multi Grid Submission.

In the topology representing the workflow the various jobs are represented by CG nodes as shown in Fig.1. Each of these nodes is, in turn, represented by a node such as the one shown in Fig.2 with the following meanings:

- Operands: The operands represent the data needed to perform the job submission:
  - Job Description: The name of the file with the job description, it can either be a JDL file [6, 7] for LCG2 submissions or a RSL file [8] for GT4 submissions
  - Main Submission Service: This is the address of the main metagrid service where the entire job was originally submitted. It is used to invoke the metagrid services for File Staging, Security and Logging.
  - User: The User Identity on behalf of whom the job will be submitted.
  - Dependencies: Dependencies on other jobs.
  - Others: Other various parameters specific to the job
- Operator: The Job submission operator.
- Output: Jobs that follow in the workflow (i.e. output dependencies).

The node in Fig.2 represents an atomic job submission to the Grid that comprises all the necessary steps to perform a job in a Grid (submission, status handling and output retrieval). Atomic nodes cannot be evaporated. However, more fine-grained operations that implement each of the possible operations on

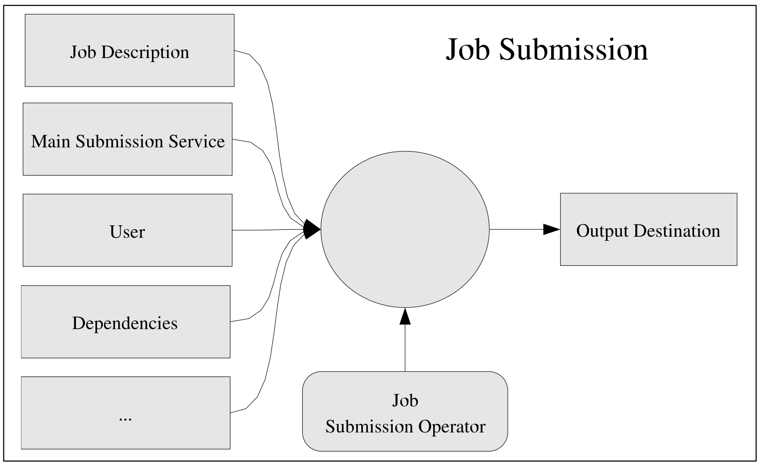


Fig. 2: A Node representing a Job Submission.

a grid are also provided as atomic operations that can be combined in different eager or lazy CGs to describe particular job submission needs such as job submissions with cancellation after a given timeout. The extra atomic operations currently implemented as CGs are:

- ListMatch: Return the compatible resources for a given job description, available only on LCG2 grids.
- Submit: Submit a job. Returns a unique identifier.
- WaitForCompletion: Polls the status of the job and waits for the completion or the failure of the job.
- GetOutput: Returns the output of the job.
- GetLogInfo: Returns the Logging information of the job.
- Cancel: Cancels a job specified by a unique identifier.

## 3 The Interoperability Architecture

The Condensed Graphs are executed by the WebCom computational engine. When a condensed graph is submitted to the WebCom engine it is evaporated (the operation opposite to condensation) until a node representing an atomic operation is reached, then the operation is executed on one of the machines hosting a WebCom instance.

In order to effectively run a composition of Grid Jobs represented in a CGs, an interoperability architecture must be provided that supports the execution of jobs on heterogeneous Grids on behalf of users belonging to different Virtual Organizations (VOs), in a resilient and scalable way. Firstly, the design of the architecture has to cope with a fundamental difference between WebCom and the conventional Grid solutions. The current release of WebCom is based on a highly dynamic topology in which different instances of WebCom are instantiated on different machines and connect to each other in a peer-to-peer structure to execute hierarchical graphs, while conventional Grids are based on rarely changing topologies where services run for most of the time on the same machines. Secondly, additional file staging mechanism are necessary. Finally, security interoperability [5] issues have to be addressed for each of the actors involved (WebCom, LCG2 and GT4).

To comply with these design constraints we have conceived an architecture based on the concept of internal and border regions.

### 3.1 Internal Regions

An internal re gion is a set of machines that run either metagrid, WebCom or a homogeneous set of Grid Services (LCG2 or GT4). We describe the regions as follows.

- $\{R_W\}$ : A WebCom internal region where only WebCom technology exists.
- $\{R_g\}$ : A Grid (either LCG2 or GT4) internal region where only that Grid technology exists.
- $R_m \backslash \{R_W \cup R_g\} = R_m \cap \overline{\{R_W \cup Rg\}}$: A metagrid region where only metagrid service technology exists.

### 3.2 Border Regions

A border region is a set of machines where at least border technologies exist, and where they can coexist with other technologies. Various types of borders can be supported within the metagrid region. We describe these border subsets as follows:

- $\{R_{mW}\} = R_m \cap R_W$ : where metagrid Service(s) and WebCom technologies coexist.
- $\{R_{mg}\} = R_m \cap R_g$ : where metagrid Service(s) and Grid Services technologies coexist.
- $\{R_{mWg}\} = R_m \cap R_W \cap R_g$ : where metagrid Service(s), WebCom and Grid Service technologies coexist.

### 3.3 Simple and Complex Borders

Borders can be of two main types: Simple and Complex. Simple Borders, called Extended Borders, such as those represented in Fig.3 are subsets of metagrid machines, each of which hosts either WebCom and metagrid technology or Grid and metagrid technology. This simple technology has two aims: It is the base upon which metagrid services such as File Staging and Security are built, and it acts as a technology decoupler between WebCom and the Grid technology.

Complex borders, called Collapsed Borders, such as those represented in Fig.4 are subsets of metagrid machines where all three technologies coexist. WebCom interacts directly with the metagrid services and the Grid technology. The metagrid technology primarily serves the metagrid services such as File

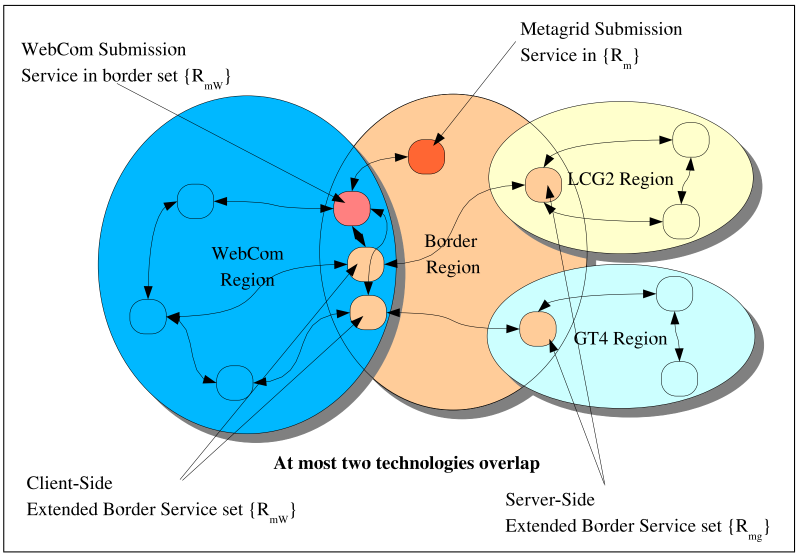


Fig. 3: A Simple Border region

Staging and Security while Job Submission is performed directly from WebCom to Grid technology.

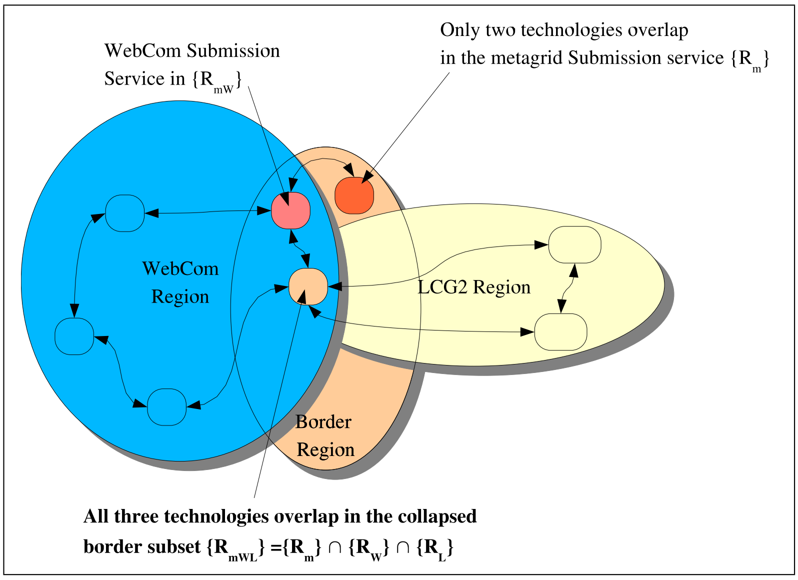


Fig. 4: A Complex Border region

The WebCom job described by a CG of Fig.1 is submitted to the metagrid machine hosting the submission services, as in Fig.3 or Fig.4; this machine is part of a simple border hosting a WebCom instance and the server-side instances of various metagrid Services that will be used throughout the entire job submission. The CG is executed by WebCom in its internal region but, when a node representing a Grid job submission is scheduled for execution it is targeted to a machine in the client side border $\{R_{mW}\}$ and there to the server-side border $\{R_{mg}\}$ (either LCG2 or GT4) and, from there, is submitted to the internal regions of the Grids. Before the job is submitted to the Grid's internal region, metagrid services such as File Staging and Security ensure that the data needed for the job submission (Files, Job Description, Certificates) are present at the border.

The services are in client-server pairs as shown in Fig.5.

- Server-Side Main Submission Service: This service represents the "Gateway" to the interoperability world. CGs representing jobs are submitted here. All necessary files and user credentials are present. This service comprises the following metagrid server-side services.
  - File Closet Service Server
  - Logging Service Server
  - Security Service Server
- Collapsed Border: This is a complex border, it hosts a WebCom instance, the Grid technology and the clients of the metagrid services.
- Extended Border: These are simple borders, one hosts a WebCom instance and the client of the Job Submission metagrid service (A metagrid technology), while the other hosts the server of the Job Submission Service and the Grid technology.

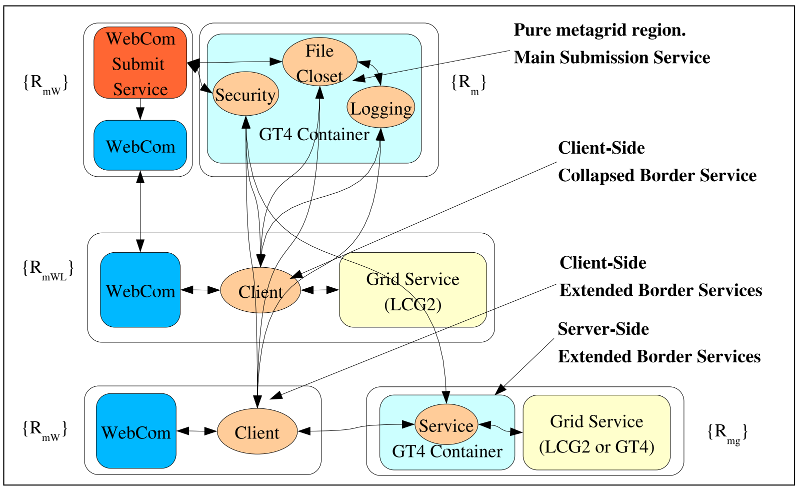


Fig. 5: Architecture of Border Services

## 4 The Metagrid Services

Metagrid Services are themselves of two main kinds: One is the Job Submission Service that enables WebCom to submit and retrieve jobs while the other, which we term "Ancillary Metagrid Services" ensure that, before the job is submitted to the internal Grid region, all the data and files needed are present where needed.

Metagrid Services, thus, interact in a choreography of Submission and Ancillary services during the execution of the entire job. A description of the choreography is as follows:

- CG Submission: The Condensed Graph representing the job is submitted to WebCom at the main submission service. All necessary files (including JDLs and/or RSLs) and certificates are present here.
- Execution of WebCom nodes: WebCom Nodes are executed directly in the WebCom internal region.
- Execution of a Grid Job Submission: When a node representing a Grid Submission is to be executed, it is sent to a specific metagrid machine where the following events take place:
  - Job Description Retrieval: The file with the description of the job is retrieved from the File Closet Service.
  - Job Description Analysis: The file with the description of the job is parsed and information regarding any other necessary file is extracted
  - Job Pre-Processing: The necessary files are downloaded from the File Closet Service, The user's credential is obtained from the Security Service
  - Job Submission: The job is submitted to the internal grid region
  - Waiting and Output: The status of the job is polled and the output of the job is retrieved
  - Job Post-Processing: The output files of the job are staged back to the File Closet Service and a log file that stores all the metagrid operations with a time-stamp is sent back to the Logging Service

## 5 Results

At the current time a first implementation of this interoperability architecture has been completed, although the Main Submission Service and the WebCom Submission Service were merged for rapid prototyping. This implementation is used both as a testbed for interoperability within the WebComG project and to perform performance testing with particular topologies of jobs to assess the overhead of the interoperability infrastructure.

## 6 Future Work

As interoperability is a very active field of research and the results of this first implementation were promising, we have further pusued interoperability in two main directions: Firstly, to implement an enhanced version of the current architecture, secondly to extend the architecture from the simplex submission of jobs from WebCom to two or more grids to a more versatile multiplex interoperability that allows any internal region to interact with any other internal region.

## References


1. John P. Morrison, James J. and David A. Power: WebCom: A Web Based Volunteer Computer, The Journal of Supercomputing 18(1), 47-61, 2001.
2. John P. Morrison, David A. Power, James J. Kennedy: An Evolution of the WebCom Metacomputer, J. Math. Model. Algorithms 2(3), 263-276, 2003.
3. John P. Morrison: Condensed Graphs: Availability-Driven, Coercion-Driven and Control-Driven Computing, Ph.D. Thesis.
4. Morrison, J.P. and Coghlan, B. and Shearer, A. and Foley, S. and Power, D. and Perrot, R., WebCom-G: A Condidate Middleware for Grid-Ireland, High Performance Computing Applications and Parallel Processing, 2005
5. David O'Callaghan and Brian Coghlan, Bridging Secure WebCom and European DataGrid Security for Multiple VOs over Multiple Grids, Proc. ISPDC 2004, IEEE Computer Society, 2004.
6. Fabrizio Pacini: Job Description Language HowTo.
7. Fabrizio Pacini: JDL Attributes.
8. http://www.globus.org/toolkit/docs/3.0/gram/rsl-schema.html.