

R-GMA: A GRID INFORMATION AND MONITORING SYSTEM

Brian Coghlan⁷, Andrew W Cooke¹, Ari Datta⁴, Abdeslem Djaoui⁵, Laurence Field⁸, Steve Fisher <s.m.fisher@rl.ac.uk>⁵, James Magowan², Werner Nutt¹, Manfred Oevers², Manish Soni³, Norbert Podhorszki⁶, John Ryan⁷, Antony J. Wilson³ and Xiaomei Zhu³

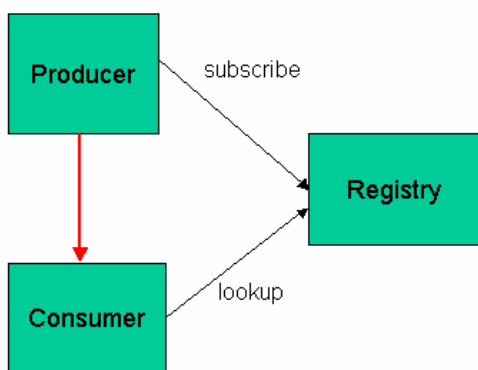
1: Heriot-Watt, Edinburgh, 2: IBM-UK, 3: PPARC, 4: Queen Mary, University of London, 5: Rutherford Appleton Laboratory, 6: SZTAKI, Hungary and 7: Trinity College Dublin

Key words to describe the work: relational, information, mediator, GMA, monitoring, EDG, DataGrid

Key Objectives: to design, implement and deploy an innovative Grid Information and Monitoring system which is based on a powerful data model, is compatible with the Grid Monitoring Architecture (GMA) of the Global Grid Forum (GGF) and, as it is being developed within the European DataGrid fully meets the requirements of that project. Note that throughout we use the term *information*, for what some may prefer to call *data*.

Motivation for the work (problems addressed): The existing LDAP based solutions do not fully meet the European DataGrid requirements because of lack of support for the *join* operation, unsuitability for application monitoring and no support for streaming of data. With the newly developed GMA from the GGF we felt that it would be possible to devise a system which was compatible with this architecture and suitable not only for monitoring (both of the Grid and applications running on the Grid) but also as a general information service. We believe we have done this.

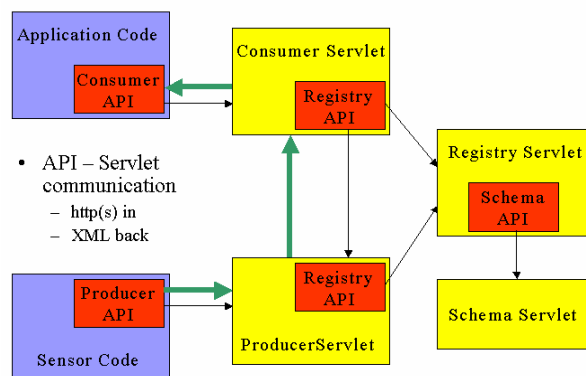
We have designed and implemented a system [1,2] with a number of components following the GMA [3] notions as highlighted in the diagram below.



Producers of information announce themselves to a registry. Consumers (wanting information) consult the registry to find suitable sources of information, and having done so connect directly to the Consumer either to transfer data for a single request or to have data streamed to them.

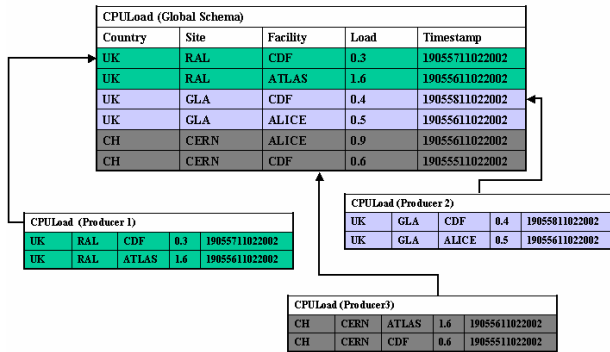
We have chosen a relational implementation that we have been able to apply both to information and to monitoring. All records carry a time stamp – by default this is the current time [4]. The system creates the impression that you have one RDBMS per Virtual Organization (VO) as far as inserting and retrieving information is concerned. Please note that we have not attempted to build a general distributed

RDBMS system, but a way to use the relational model in a distributed environment. Currently R-GMA uses servlet technology. Each distributable component is a servlet. Each servlet receives http(s) requests and makes an XML coded response. Code has been written for each servlet to deal with the http and XML and offer a simple (thin client) API. The API code has been constructed in Java, C, C++, Perl and Python to suit all users. Some of the components are shown below:



The thin arrows denote invocation and the broader (green) ones show the main information flow. The Schema (holding the description of each table type) is actually separate from the Registry, which has an entry for each table from each Producer. In fact we also register Consumers as explained later. To provide resilience we use a soft-state registration scheme.

As stated earlier we want to create the impression to a Consumer of information that she is dealing with one large database. This is indicated in the figure below, which shows a CPUload table that is published by 3 different producers but can be seen as 3 views of a single table.



The query:

```
SELECT * FROM CPUload WHERE Country = 'UK' AND Site = 'RAL'
```

Corresponds to the view of Producer 1. It is this definition of the view, which is stored in the Registry to describe a Producer. The Consumer contains a mediator component, which consults the Registry on behalf of the user and finds the Producer or Producers needed to satisfy the query and then executes the query. Currently the mediator can merge information from multiple Producers but cannot handle joins over Producers. Consumers are also registered so that when a new Producer registers itself, all the Consumers of that table are alerted and the mediator can take into account the new Producer.

We have constructed two kinds of Producer: a CircularBufferProducer and a DataBaseProducer. The CircularBufferProducer uses a circular buffer to hold the published data. It has just one *write pointer* but has a *read pointer* for each Consumer that is streaming data from it. As there is no RDBMS involved we have to handle the SQL queries in our code. Writing is never blocked but then a Consumer may miss records if it is too slow. The DataBaseProducer uses an RDBMS to hold data and so does not lose information but it is slower and it needs a clean-up strategy to be defined otherwise the database will just grow indefinitely. Another drawback of the DataBaseProducer is that we do not currently support streaming from it. Streaming will be added when we have defined the semantics of this operation for a database.

To be able to introduce it into the European DataGrid testbed easily, we have written a piece of code, which is able to invoke the MDS info provider scripts and publish the information via R-GMA. It is currently being integrated into the development testbed and should move into production in release 2.0 in autumn. Discussions with the other DataGrid work packages have brought us a number of new challenging requirements. For example we have been asked for a Producer with the performance of the CircularBufferProducer but which keeps all the data safe like the DataBaseProducer. We have also been asked to provide an overwrite mode so that a tuple with the same primary key (except for the timestamp component) is updated by the new insertion. There are also requests for more functionality in the mediator. These are being addressed [5].

Finally, we are very pleased with the interest the Grid community is showing in OGSA. R-GMA is already conceptually close to web services. We will shortly repackage it first as regular web services and then as OGSA services [6].

1. WP3, *Information and Monitoring (WP3) Architecture Report*, DataGrid-01-D1.2-0112-0-3, September 2001 (current version: <http://hepunix.rl.ac.uk/edg/wp3/documentation/doc/arch.pdf>)
2. Steve Fisher, *Relational Model for Information and Monitoring*, GGF, 2001, (<http://www-didc.lbl.gov/GGF-PERF/GMA-WG/papers/GWD-GP-7-1.pdf>)
3. Brian Tierney, Ruth Aydt, Dan Gunter, Warren Smith, Valerie Taylor, Rich Wolski and Martin Swamy, *A Grid Monitoring Architecture*, GGF, 2001 (revised 2002), (<http://www-didc.lbl.gov/GGF-PERF/GMA-WG/papers/GWD-GP-16-2.pdf>)
4. Brian Coghlan, Abdeslem Djaoui, Steve Fisher, James Magowan and Manfred Oevers, *Time, Information Services and the Grid*, BNCOD 18, Chilton, UK, July 2001, (<http://edms.cern.ch/document/349802>)
5. WP3, *R-GMA revisited*, presented at GGF5 and to be published.
6. WP3, *OGSification of R-GMA*, presented at GGF5.

