XML Schema for e-Science Projects, Grid Users, Applications and Resources.
IeSE Developers' Notes (IeSE-3).

Rob Allan, Xiao Dong Wang, Andy Richards and Dharmesh Chohan
CCLRC e-Science Centre, Daresbury Laboratory,
Daresbury, Warrington WA4 4AD, UK

Mark McKeown
UK Grid Support, Manchester Computing,
University of Manchester, Oxford Road, Manchester M13 9PL, UK

e-Mail: `r.j.allan@dl.ac.uk`

April 18, 2007

Abstract

This paper proposes XML schema for Grid users, applications and resource descriptions which can be used for authorisation and contact purposes, discovery and for mapping applications to resources. These schema are being developed in the CCLRC IeSE project `http://esc.dl.ac.uk/IeSE/about.html` and some prototype services using them are described. We will describe how these schema extend the UDDI schema for e-Science projects which has been proposed and provide a semi-static information base complementary to the Globus Monitoring and Discovery Service (MDS). We will consider how information services based upon such schema will grow in functionality when CIM is widely used, e.g. for OGSI/ GT3 service data.

Keywords: HPC Applications, HPC Resources, XML Schema, Computational Grid, Globus, Grid Architecture, OGSA, Grid Security, Resource Management, Grid Information Systems, e-Science projects, Grid users, Integrated e-Science Environment, CIM.

Contents

1 Introduction **1**

1.1 Common Information Model 1

2 Schema for Projects **2**

3 XML Schema for HPC Resources **2**

3.1 Relationship to Globus MDS 5

3.2 Relationship to CIM 6

3.3 Relationship to SNMP 7

3.4 UNICORE Incarnation Resource Database Schema 7

3.5 Relationship to DataGrid R-GMA 7

3.6 Brokering and Distributed Accounting 7

4 XML Schema for HPC Applications **7**

4.1 Globus MDS Application Schema 10

4.2 Relationship to CIM 10

4.3 Considerations of Distributed Software Management 10

4.4 Application Licenses on a Grid 10

5 Schema for Users **10**

5.1 Globus MDS Contact Person Schema 12

5.2 Relationship to CIM 13

5.3 Consideration of User Management and Authorisation Issues 13

6 Use of XML and Relational Databases **13**

7 Grid Community InfoPortal and Resource Management Services **13**

8 IeSE Application Discovery and Job Submission Services **15**

9 Acknowledgements 15

A Application Schema 18

B Resource Schema 21

C Person Schema 23

1 Introduction

In a Grid there are some things that change a lot (dynamic information) and some things that don't change very often (static information). In working with the Grid Information System we have identified a need for the LDAP-based Globus MDS system to give information about which resources are "up" and their load and other utilisation figures and a UDDI registry to provide information and links to VO projects and their services. The latter is discussed in a separate paper [17].

There is also a need to describe applications which are pre-installed on the Grid resources as discovered via MDS, and to get a more static view of the architecture of these resources. We have chosen to use XML schema in addition to MDS for this information, giving the possiblity of getting the XML documents from a Web server on the actual resource, perhaps via a Web search engine, or of having them stored in an XML database either centrally or distributed.

Quantities may or may not depend on time depending on how they are viewed. For instance if an application is installed on a resource, its version number may depend on time, but looking up that application version its characteristics are constant. In this case it may equally well be viewed as a second application, but the application set depends on time. CPU load or memory usage on a resource may vary, but many of its other capabilities will be fixed. We propose using an XML element attribute for date and time, so that a query may specify an element name and a time and any database provider should return the element value for closest time recorded.

This paper proposes XML schema for describing e-Science projects and Grid users, applications and resources which can be used in authorisation and contact, discovery and for mapping applications to resources. It will be seen that some data elements are shared between schema, for instance a resource will contain a list of authorised users (possibly recorded as a function of time), but an individual user may contain a similar list for the resources that he/ she is allowed to use. There is a clear case for relational tables here which go beyond simple XML files.

These schema are being developed in the CCLRC IeSE project and some prototype services using them, including a Community InfoPortal for the UK e-Science Grid, are described. The schema proposed here will also form the basis of "facility metadata" definitions used in the IeSE DataPortal.

1.1 Common Information Model

The Common Information Model (CIM) is an object-oriented information model defined by the Distributed Management Task Force (DMTF) that provides a conceptual framework for publishing data required for computational resource management [30]. CIM is designed for the management of systems, software, users, devices, networks and more. There are other instrumentation and management standards, including SNMP, DMI, CMIP etc. and CIM aims to unify some of these previous models and incorporates data from a variety of sources.

CIM makes strong use the structuring and abstraction of object oriented programming, as opposed to centrally managed tabular data structures used in some previous systems. It is therefore more appropriate for a Grid environment and has been adopted by the Globus development team for GT3.

Of particular interest here are the CIM models for system (compute resources, databases, storage), devices, networks, users and applications. These are referred to as the common models. We note that much of the documentation on the CIM Web site has not changed since August 2000.

2 Schema for Projects

A Schema for a Virtual Organisation (or Project) is being developed. This is considered to map onto the UDDI schema, with a one to one analogy between e-Science Project information and e-Business information. This is discussed in a separate document [17]. Contact details in the UDDI schema may point to a person schema for each contact. Resources available to the project may appear as services.

Figure 1 shows the mapping of the main elements of the schema and their relationship with UDDI which is discussed separately [17].

There may be additional information which could be useful. In UDDI v3.0 it is proposed to have extensible schema, that means that our person schema could replace the current limited contact schema. It may also be required to add information about funding sources to supplement information which could be contained in the identifierBag in v2.0.

3 XML Schema for HPC Resources

In discussion with members of the UK Grid Engineering Task Force (ETF) it was realised that there is a requirement to describe a Grid computational resource which is a heterogeneous cluster or set of clusters. This is not currently catered for in the Globus MDS schema. We proposed initially to test an XML schema for this purpose integrated with the Grid Community InfoPortal in IeSE to present information to users (see below).

We consider that a resource is logically divided into partitions and that each partition can comprise a number of head nodes and a number of worker nodes with different characteristice. Work nodes may also include special-purpose i/o nodes. Characteristics of these different nodes can be recorded including the presence of communications subsystems and resource management systems (and Globus gatekeeper and jobmanagers).

resType: name, key, description and partitions;

resDescType 1: Web site URL, contact person, description of resource;

partitionType 1..*: number of head nodes, number of worker nodes, detailed descriptions of head and worker nodes;

accountingScheme 1: how accounting information is collected and provided;

appsInstalled 1..*: name, version and location of installed applications which link back the the application schema via a unique key.

Information provided about a node include: description; DNS name; IP address; number of processors; OS version; available comms subsystems; total available disc storage in MB; total available memory in

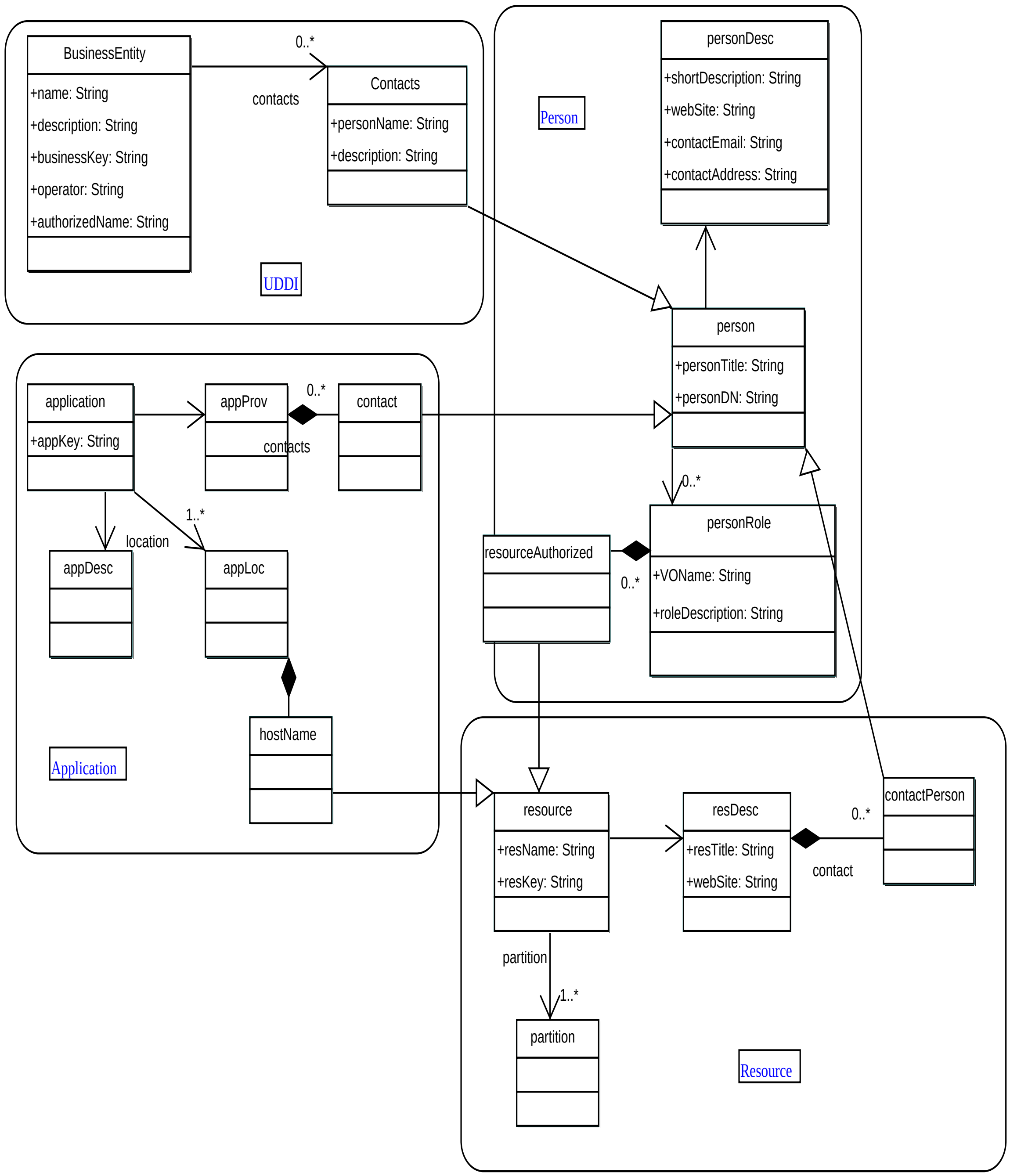

Figure 1: Connections between Schema

MB; peak performance in Gflop/s; available RMS sub systems; Globus gatekeeper if present; available Globus jobmanagers if present.

We also decided that hooks should be included in the schema to allow extensions for accounting in the future.

The following example describes the Loki alpha cluster at Daresbury.


```
<?xml version="1.0" encoding="UTF-8" ?>
<?xml-stylesheet href="resSchema.xsl" type="text/xsl" ?>
<resource>
<!-- description and identifier -->
<resName>Loki</resName>
<resKey>6c220b36-6ee0-4f7e-a7d3-c26abec07f71</resKey>
<resDesc>
<resTitle>Loki alpha cluster at Daresbury</resTitle>
<webSite>http://loki.dl.ac.uk</webSite>
<contactPerson>Steve Andrews</contactPerson>
<contactEmail>s.j.andrews@dl.ac.uk</contactEmail>
<shortDescription>
Loki is a 64-processor Alpha Beowulf cluster. 24 nodes have a pair of
Compaq Alpha EV67 667 MHz processors in a DS20 chassis with shared
memory and disk. 8 nodes have 833 MHz Alphas. All nodes are connected
by Fast Ethernet and QSNet switches which can sustain 100 Mb/s and 140
MB/s respectively. A master node is provided for login and
compilation. The application nodes run jobs via the Quadrics RMS
resource management system. Quadrics 128-way switch (QSNet).
</shortDescription>
</resDesc>
<partition>
<headNode>
<NodeNumber>1</NodeNumber>
<nodeDesc>Loki master node. UW-SCSI user disk.
</nodeDesc>
<DNSname>loki.dl.ac.uk</DNSname>
<ipAddress>193.62.112.157</ipAddress>
<numProcs>1</numProcs>
<processor>Alpha 21264A</processor>
<MHz></MHz>
<osVersion>Linux</osVersion>
<storageAvail>16 GB</storageAvail>
<memoryAvail>1 GB</memoryAvail>
<rmsAvail>PBS</rmsAvail>
<globusGateKeeper>loki.dl.ac.uk</globusGateKeeper>
<jobManagerAvail>loki.dl.ac.uk:2119</jobManagerAvail>
<jobManagerAvail>loki.dl.ac.uk:2119/jobmanager-pbs</jobManagerAvail>
<qName></qName>
</headNode>
```

```
<workerNode>
<NodeNumber>24</NodeNumber>
<nodeDesc>Pool 1.  Dual 667 MHz Alphas connected to the master via a
Fast Ethernet switch. 2nd Fast Ethernet switch dedicated to the
parallel communications traffic. High Performance Interconnect via
Quadrics Elan III NICs. ECC SDRAM and SCSI disk.
</nodeDesc>
<numProcs>2</numProcs>
<processor>Alpha 21264A</processor>
<MHz>667 MHz</MHz>
<osVersion>Linux</osVersion>
<globusGateKeeper>loki.dl.ac.uk</globusGateKeeper>
<commsAvail>MPI on Ethernet</commsAvail>
<commsAvail>MPI on QSNet</commsAvail>
<storageAvail>9 GB</storageAvail>
<memoryAvail>1 GB</memoryAvail>
<rmsAvail>PBS</rmsAvail>
</workerNode>
<workerNode>
<NodeNumber>8</NodeNumber>
<nodeDesc>Pool 2.  Dual 833 MHz Alphas connected to the master via a
Fast Ethernet switch. 2nd Fast Ethernet switch dedicated to the
parallel communications traffic. High Performance Interconnect via
Quadrics Elan III NICs. ECC SDRAM and SCSI disk.
</nodeDesc>
<numProcs>2</numProcs>
<processor>Alpha 21264A</processor>
<MHz>833 MHz</MHz>
<osVersion>Linux</osVersion>
<globusGateKeeper>loki.dl.ac.uk</globusGateKeeper>
<commsAvail>MPI on Ethernet</commsAvail>
<commsAvail>MPI on QSNet</commsAvail>
<storageAvail>9 GB</storageAvail>
<memoryAvail>1 GB</memoryAvail>
<rmsAvail>PBS</rmsAvail>
</workerNode>
</partition>
</resource>
```


3.1 Relationship to Globus MDS

MDS is being developed to provide dynamic information about a computational Grid. It currently, via the default LDAP schema which is distributed with Globus, gives information about a Globus gatekeeper including cpu type and number, real memory, virtual memory, filesystems and networks. Additional information providers can be plugged into this framework. The following comments were provided by Gabriel Mateescu of NRC, Canada.

The GRAM reporter provider gives some information about batch jobs, .e.g.,

```
> # 8.pt.sao, workq, jobmanager-pbs, pt.sao.nrc.ca, local, grid
> dn: Mds-Job-id=8.pt.sao, Mds-Job-Queue-name=workq, Mds-Software-deploym
>  ent=jobmanager-pbs, Mds-Host-hn=pt.sao.nrc.ca,Mds-Vo-name=local,o=grid
> objectClass: Mds
> objectClass: MdsJob
> objectClass: MdsGramJob
> Mds-Job-id: 8.pt.sao
> Mds-Gram-Job-localId: 8.pt.sao
> Mds-Gram-Job-globalId: NULL
> Mds-Gram-Job-localOwner: gabriel
> Mds-Gram-Job-globalOwner: NULL
> Mds-Gram-Job-status: running
> Mds-Gram-Job-schedulerSpecific: NULL
> Mds-validfrom: 200210141626.30Z
> Mds-validto: 200210141627.00Z
> Mds-keepto: 200210141627.00Z
```

So, we get the job ID, owner, and status.

For purposes such as meta-scheduling, additional information would be useful, e.g., about the resources allocated to the job (compute nodes in a cluster, number of CPUs, memory), as well as CPU time used by running jobs. Will that be provided at some time by the GRAM reporter (or another package)?

It looks like there are two options:

1. extend the GRAM reporter to provide more information about batch jobs managed by a local scheduler;
2. write add-on providers that convert local scheduler information to a scheduler independent representation.

The first choice has the advantage that it promotes consistency across different local schedulers, while the second allows sites to go beyond what the reporter supports at a given time.

3.2 Relationship to CIM

The Common Information Model (CIM) is an object-oriented information model defined by the Distributed Management Task FOrce (DMTF) that provides a conceptual framework for describing management data [30]. CIM is an approach to the management of systems, software, users, networks and more that applies the basic structuring and abstraction of object oriented programming.

3.3 Relationship to SNMP

– to follow.

3.4 UNICORE Incarnation Resource Database Schema

– to follow

3.5 Relationship to DataGrid R-GMA

– to follow

3.6 Brokering and Distributed Accounting

Brokering and accounting are linked in that one might trade time to completion with cost. However many of the aspects of a practical accounting system can be treated separately. We only intend to provide the necessary hooks in the schema to say how this can be done. A separate paper [2] considers how a distributed billing and price publication system might be deployed.

4 XML Schema for HPC Applications

The UK e-Science Grid can most effectively be used for pre-installed and validated applications [20]. Such a Grid, based on the Globus middleware, is currently being deployed by the UK Grid Engineering Task Force `http://www.grid-support.ac.uk/etf`. Indeed Mike Surridge [27] has pointed out that there are potentially overwhelming problems with security if this is not the case, e.g. in an "open grid". It is therefore necessary to know which version of each application is present on each resource, its dependencies e.g. input/ output files and its conditions of use, e.g. license required. This allows for verification that the executable is correctly installed and tested and for search facilities to find applications appropriate to a particular scientific discipline.

It is particularly useful to be able to choose an appropriate application based on a thematic keyword-based search, with keywords derived from an appropriate ontology, and to see if other applications exist which are useful in a similar area of research. These may offer different parameter space coverage or guarantee different levels of accuracy to time to solution. They may also require different operating environments and provide more or less suitable interfaces for the user.

Other work on application metadata is being carried out in the GGF Grid Computing Environments research group, e.g. at Cardiff (Rana et al.) [28], Indiana (Geoffrey Fox), ... and at Southampton, BAe Systems etc. The aim is to develop working tools which build upon previous work.

The high-level application metadata which we define has the classical set of three sections:

1. its own metadata – description of the application itself. This is likely to include such things as:
   - title, version and description, e.g. sufficient to permit a keyword search;
   - dependencies such as environment and other requirements;
   - input/ output files required together with metadata describing content and format;
   - license and other usage conditions which apply;
2. provenance – history of code version, author may be described by a link to a person schema (see below);
3. location – the install base of executables, including Grid host and full path.

Main sections in the schema file which encapsulate this information are:

appType: title, key, version;

appDescType 1: Web URL, textual description including keywords;

licenseScheme 1: information about the type of license required.

appProvType 1..*: contact e-mail, contact address;

appLocType 1..*: type (binary or source), host, full path to executable, operating system;

appFileType 1..*: information about i/o files, their content and format;

appDependencyType 1..*: information about any dependencies or operational requirements, e.g. O/S/, memory or temporary disc required, compiler version;

The appLocType section can be repeated for each host on which there is an application matching this description.

Issues yet to be fully addressed include:

- specification of input/ output files and formats;
- specification of additional requirements.

The following example describes the REALC application.

```
<?xml version="1.0" encoding="UTF-8" ?>
<?xml-stylesheet href="appSchema.xsl" type="text/xsl" ?>
<application>
<!-- description and identifier -->
<appUUID>803e385c-5edc-49c0-8ff3-f5fc9230c4aa</appUUID>
<appDesc>
<appTitle>REALC</appTitle>
```

```
<webSite>http://www.cse.dl.ac.uk/Activity=CCP6;</webSite>
<shortDescription>Parallel program for computation of rates of
chemical reactions of the form A+BC->AB+C. Uses time-dependent
propagation of a real wavepacket on a DVR coordinate grid.
</shortDescription>
<licenseScheme license="academic">
<licenseIssuer contact="person">
<contactName>Rob Allan</contactName>
<contactKey>/C=UK/O=es-grid/OU=dl.ac.uk/CN=Robert Allan</contactKey>
</licenseIssuer>
</licenseScheme>
</appDesc>
<!-- provenance -->
<appProv contact="person">
<contactName>Rob Allan</contactName>
<contactKey>/C=UK/O=es-grid/OU=dl.ac.uk/CN=Robert Allan</contactKey>
</appProv>
<!-- location of installed app on Grid resources -->
<appLoc>
<appType>binary</appType>
<appHost>esc5.dl.ac.uk</appHost>
<hostPath>/home/rja/tdrs_codes/run_realc/realc.out</hostPath>
<hostType>Linux</hostType>
</appLoc>
<appLoc>
<appType>binary</appType>
<appHost>esc4.dl.ac.uk</appHost>
<hostPath>/home/rja/tdrs_codes/run_realc/realc.out</hostPath>
<hostType>Linux</hostType>
</appLoc>
<appLoc>
<appType>binary</appType>
<appHost>beowulf1.dl.ac.uk</appHost>
<hostPath>/home/rja/tdrs_codes/run_realc/realc.out</hostPath>
<hostType>Linux</hostType>
</appLoc>
<appLoc>
<appType>binary</appType>
<appHost>loki.dl.ac.uk</appHost>
<hostKey>6c220b36-6ee0-4f7e-a7d3-c26abec07f71</hostKey>
<hostPath>/home/rja/tdrs_codes/run_realc/realc.out</hostPath>
<hostType>Linux</hostType>
</appLoc>
</application>
```

4.1 Globus MDS Application Schema

– to follow

4.2 Relationship to CIM

– to follow.

4.3 Considerations of Distributed Software Management

– to follow.

4.4 Application Licenses on a Grid

Licensing, authorisation and accounting go hand in hand. We have chosen however to deal with them separately, but to provide sufficient links between the relevant schema to allow practical tools to be developed.

Whilst it is not yet the case, we believe that the current license schemes will be extended to include payment on demand in the near future. This is one area of research to be targetted in the *Computational Markets* proposal of Darlington et al.

5 Schema for Users

In the course of discussion it was realised that an XML schema describing a user would also be useful. This could be populated with information enabling user-management tools to be developed. A single user can belong to many VOs and have many roles. These should be described in the schema. Accounting hooks should also be included.

The schema includes:

personType: person's name, DN and short description of them;

personRoleType 1..*: person's role in a VO and a list of resources to which they have access;

paymentScheme 1..*: how the person pays for Grid resources.

This schema could to some extent be populated, or updated, by automatic tools. One such is the tool to determine to what resources the user has acccess. Another might be the CA.

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<?xml-stylesheet href="personSchema.xsl" type="text/xsl" ?>
<person>
<personTitle>Dr. Robert John Allan</personTitle>
<personDN>/C=UK/O=es-grid/OU=dl.ac.uk/CN=Robert Allan</personDN>
<personDesc>
<shortDescription>A very nice guy who has written several papers on
developing the UK Grid.</shortDescription>
<webSite>http://www.e-science.clrc.ac.uk</webSite>
<contactEmail>r.j.allan@dl.ac.uk</contactEmail>
<contactAddress>CCLRC e-Science Centre, Daresbury Laboratory, Daresbury,
Warrington WA4 4AD, UK</contactAddress>
</personDesc>
<!-- my various roles in projects -->
<personRole>
<voName>CLRC</voName>
<voKey>95ee146c-f83d-4d94-8f62-0ffd3780e85a</voKey>
<roleDescription>Technical Manager</roleDescription>
<roleKey>cc2c56b9-aa44-4374-bdf2-d04b9aab09f2</roleKey>
<resourceAuthorized>esc.dl.ac.uk</resourceAuthorized>
<resourceAuthorized>esc4.dl.ac.uk</resourceAuthorized>
<resourceAuthorized>esc5.dl.ac.uk</resourceAuthorized>
<resourceAuthorized>loki.dl.ac.uk</resourceAuthorized>
<resourceAuthorized>tcs10.dl.ac.uk</resourceAuthorized>
<resourceAuthorized>splogin1.dl.ac.uk</resourceAuthorized>
<resourceAuthorized>login.hpcx.ac.uk</resourceAuthorized>
</personRole>
<personRole>
<voName>UK e-Science Grid</voName>
<voKey>f7753190-3ddc-4ad7-825f-d85a51cf07ff</voKey>
<roleDescription>Deployment Manager</roleDescription>
</personRole>
<personRole>
<voName>UK e-Science Grid</voName>
<voKey>f7753190-3ddc-4ad7-825f-d85a51cf07ff</voKey>
<roleDescription>WP2 Leader</roleDescription>
<resourceAuthorized>ginfo.grid-support.ac.uk</resourceAuthorized>
<resourceAuthorized>esc.dl.ac.uk</resourceAuthorized>
</personRole>
<personRole>
<voName>UK e-Science Grid</voName>
<voKey>f7753190-3ddc-4ad7-825f-d85a51cf07ff</voKey>
<roleDescription>HPCPortal Developer</roleDescription>
<roleKey>aca2e483-ae82-472d-aa86-38eab1b373e9</roleKey>
<resourceAuthorized>wk-pc1.dl.ac.uk</resourceAuthorized>
<resourceAuthorized>wk-ibm1.dl.ac.uk</resourceAuthorized>
<resourceAuthorized>splogin1.dl.ac.uk</resourceAuthorized>
```

```
<resourceAuthorized>loki.dl.ac.uk</resourceAuthorized>
<resourceAuthorized>hrothgar.esc.rl.ac.uk</resourceAuthorized>
<resourceAuthorized>login.hpcx.ac.uk</resourceAuthorized>
<resourceAuthorized>herschel.cam.ac.uk</resourceAuthorized>
<resourceAuthorized>metropolis.soton.ac.uk</resourceAuthorized>
<resourceAuthorized>green.cfs.ac.uk</resourceAuthorized>
<resourceAuthorized>bouscat.cs.cf.ac.uk</resourceAuthorized>
</personRole>
<personRole>
<voName>eHTPX</voName>
<voKey>06425db6-e378-40d8-8884-33c384454d17</voKey>
<roleDescription>Grid WP2 Leader</roleDescription>
<roleKey>99a7ab26-e7d2-49ac-b8a4-4bb8295f4358</roleKey>
</personRole>
<!-- funding schemes for me, map to various roles requiring payment -->
<paymentScheme>
<paymentDescription>OST funding to CCLRC</paymentDescription>
<roleKey>aca2e483-ae82-472d-aa86-38eab1b373e9</roleKey>
<roleKey>cc2c56b9-aa44-4374-bdf2-d04b9aab09f2</roleKey>
<paymentType>on demand</paymentType>
</paymentScheme>
<paymentScheme>
<paymentDescription>BBSRC grant</paymentDescription>
<roleKey>99a7ab26-e7d2-49ac-b8a4-4bb8295f4358</roleKey>
<paymentType>quarterly bill</paymentType>
</paymentScheme>
</person>
```


5.1 Globus MDS Contact Person Schema

There is an entry in the MDS2.2 schema for a contact person for each Grid resource.

Gabriel Mateescu and Darcy Quesnel of the CanadaGrid project have proposed a new information provider for the Globus Monitoring and Discovery Service (MDS) to provide information about the contact person associated with a compute resource. This currently includes the name, email address, postal code, and country of the contact person. and takes the form:

```
grid-info-search -x '(objectClass=MdsHostContact)'
  version: 2

# manager, si.sao.nrc.ca, local, grid
dn: Mds-Host-contact=manager,Mds-Host-hn=si.sao.nrc.ca,Mds-Vo-name=local,
 o=grid
objectClass: Mds
objectClass: MdsComputer
objectClass: MdsHostContact
```

```
Mds-Host-Contact-name: Gabriel Mateescu
Mds-Host-Contact-email: gabriel.mateescu@nrc.ca
Mds-Host-Contact-postalcode: K1A 0R6
Mds-Host-Contact-country: Canada
```

The source bundle as well as installation instructions are available at: http://www.sao.nrc.ca/imsb/rcsg/gc/globus.html\#install_host_contact_sensor

5.2 Relationship to CIM

– to follow.

5.3 Consideration of User Management and Authorisation Issues

Role-based authorisation is a complex issue which must be tackled if Grids are to provide a transparent means of accessing and using resources of many different types. Several research groups are investigating this area.

CAS – to follow

VOMs – to follow

VOM (Imperial College) – to follow

6 Use of XML and Relational Databases

Xindice – see separate report [18]

eXist – to follow

ORACLE – maps XML into relational structures.

??? – presents relational output as XML (ask Kevin...)

OGSA-DAI – see separate report [18]

7 Grid Community InfoPortal and Resource Management Services

The Grid Community InfoPortal http://esc.dl.ac.uk/InfoPortal can be used to obtain information as described above about projects, applications and users of the UK e-Science Grid. XML

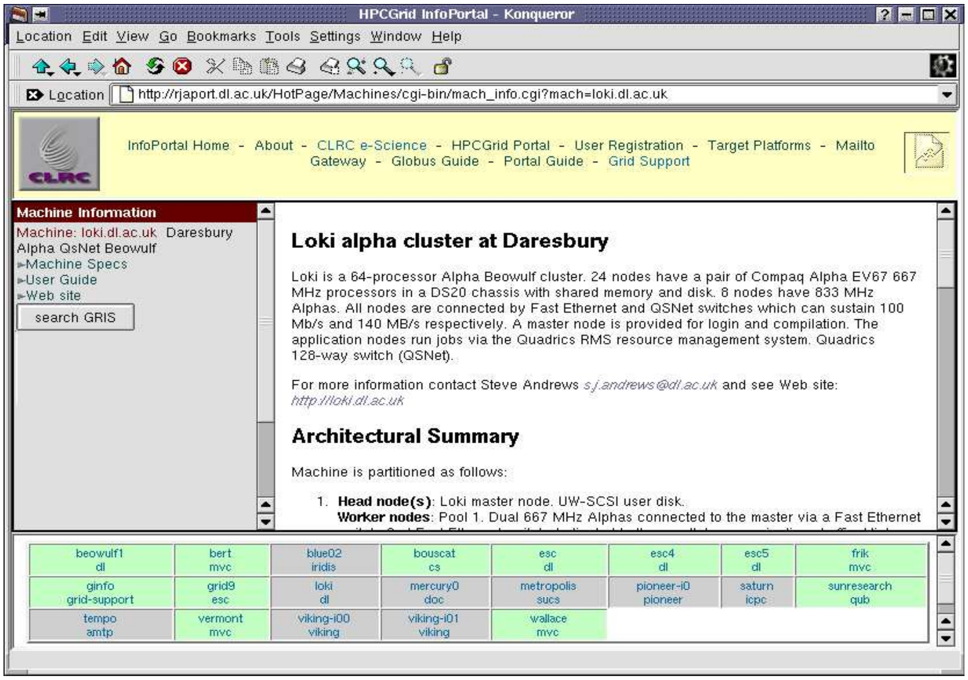

Figure 2: InfoPortal view of Resource Information

stylesheet transformations are applied to the descriptor files. In InfoPortal the Perl *xslt-parser* is invoked as a CGI task to transform the data into HTML format for the Web information pages. Other tools are being developed which provide different views of the information.

In InfoPortal the UK National MDS GIIS server is queried once per hour using a Unix `cron` application to update a local LDAP database. This is used as a dynamic information provider cache.

Additional static XML descriptions are held in an Apache Xindice database and an OGSA-DAI prototype service is used to access this data. Xpath queries can simply be applied to search the database via an instance of the DB service.

XML stylesheet transformations are applied to the resource descriptor files which are extracted from the DB. In InfoPortal the Perl *xslt-parser* is invoked as a CGI task to transform the data into HTML format for the Web information pages. This is accessible as `http://esc.dl.ac.uk/InfoPortal/Machines/mach_specs.cgi?mach=$MACH` where $MACH is the FQDN of a Grid resource, e.g. *vermont.mvc.mcc.ac.uk*.

Other similar tools are being developed which provide different views of the information. One such can provide an IDB file for use by a UNICORE job-submission system. These will be provided as Web services as well as traditional HTML GET operations.

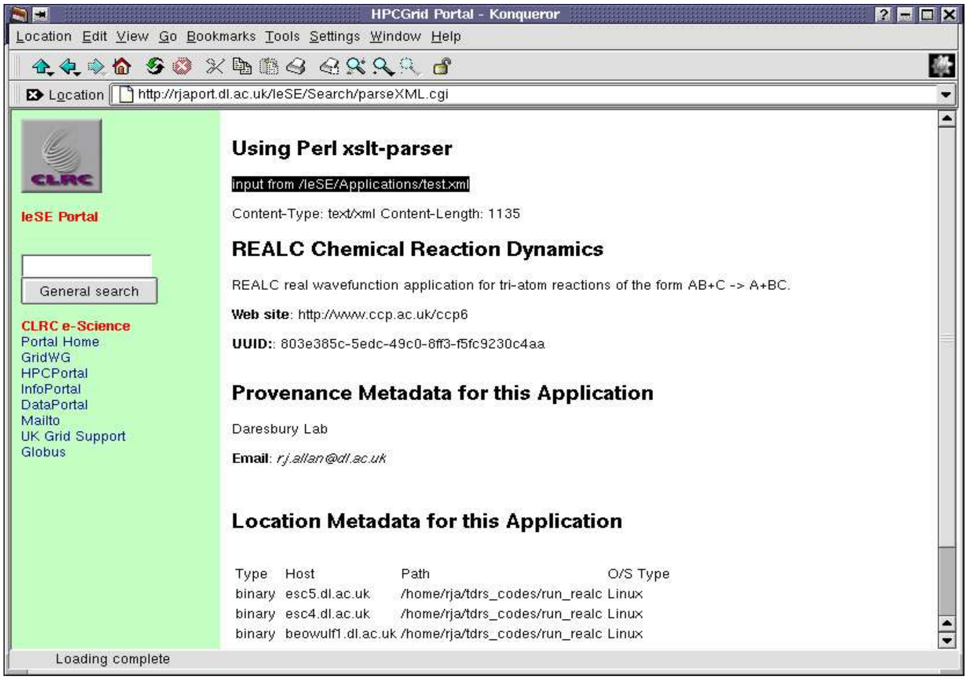

Figure 3: IeSE Application Search Web Interface

8 IeSE Application Discovery and Job Submission Services

The CCLRC Integrated e-Science Environment portal is at `http://esc.dl.ac.uk/IeSE`. In IeSE, stylesheet transformations also produce forms which are suitable for direct job submission to the resources specified in the file.

9 Acknowledgements

This paper follows the previous paper on *UDDI and WSIL for e-Science* [17]. Some of the areas discussed were identified in meetings with e-Science project teams at the regional e-Science Centres and workshops such as the NeSC Opening (April 2002), the All-Hands Meeting (September 2002) and the BAe Systems e-Engineering Workshop (October 2002). We have presented preliminary ideas to members of the Globus Team and GGF groups working on the definition of services and standards for Grid information systems and applications. The recommendations reported here are partly based on investigations funded by the OST and DTI via the CCLRC e-Science Centre and UK Grid Support Centre.

We thank the following people who have provided additional input to this paper:

Mary Thomas, Gabriel Mateescu, Omer Rana, David Baker, Mark Baker, Kerstin Kleese, Shoaib Sufi, Stephen Pickles, Kevin O'Neill.

References

[1] R.J. Allan *Survey of Computational Grid, Meta-computing and Network-based Tools* Edition 2 Technical Report DL-TR-99-02 (Daresbury Laboratory, February 2000) `http://esc.dl.ac.uk/StarterKit`

[2] R.J. Allan *Resource Selection and Billing via Agents, Brokering and a Grid Hub* (CCLRC e-Science Centre, 2000) draft

[3] R.J. Allan, D. Hanlon, R.F. Fowler and C. Greenough *A Globus Developers' Guide with Installation and Maintenance Hints* technical report (UKHEC Collaboration and CCLRC e-Science Centre, 2001) `http://esc.dl.ac.uk/GridWG`

[4] F. Costen, J.M. Brooke, R.J. Allan and M. Westhead *Grid-based High Performance Computing* (UKHEC, 2000) `http://www.ukhec.ac.uk`

[5] R.J. Allan, S. Pickles, J.M. Brooke, F. Costen and M. Westhead *UK HEC Grid: background and first steps* UKHEC Report to the Research Councils (UKHEC, 2000) `http://www.ukhec.ac.uk`

[6] R.J. Allan, S.J. Andrews, M.F. Guest, P.M. Oliver, D. Henty, L. Smith, S. Telford and S. Booth *Designing and Building Beowulf-class Cluster Computers* (UKHEC, 2000) `http://www.ukhec.ac.uk`

[7] R.J. Allan *Developing a Web Portal for the Computational Grid* technical report (UKHEC Collaboration and CCLRC e-Science Centre, 2001) `http://esc.dl.ac.uk/GridWG`

[8] R.J. Allan, D.R.S. Boyd, T. Folkes, C. Greenough, D. Hanlon, R.P. Middleton and R.A. Sansum *Evaluation of Globus and Associated Grid Middle-ware.* report to the DTI (CCLRC e-Science Centre, May 2001) `http://esc.dl.ac.uk/StarterKit`

[9] R.J. Allan and D.R.S. Boyd *Some Architectural Issues to be considered in building a UK Grid* report for the UK Grid Engineering Task Force (CCLRC, 2001)

[10] R.J. Allan, D.R.S. Boyd, T. Folkes, C. Greenough, D. Hanlon, R.P. Middleton, R.A. Sansum, Elson Mourão, Rob Baxter, David Henty, Mark Parsons, John Brooke, W.T. Hewitt, Mike Daw, Jon MacLaren, Jon Gibson, Graham Riley and Stephen Pickles *Globus and Associated Grid Middleware. Consolidated Evaluation Report from UKHEC Sites.* (UKHEC, December 2001) `http://esc.dl.ac.uk/StarterKit`

[11] R.J. Allan, A.B. Mills and D.R.S. Boyd *HPCGrid* the national magazine for HPC Grid applications, issue 26 (CCLRC, November 2002) `http://www.cse.clrc.ac.uk/Activity/HPCProfile`

[12] M. Baker, H. Ong and G. Smith *A Report on Experiences operating the Globus Toolkit through a Firewall* (University of Portsmouth, 2001) `http://esc.dl.ac.uk/GridWG`

Andrew Richards, Rob Allan and Daniel Hanlon *Globus Toolkit Firewall Port Selection* (UK Grid Support Centre, October 2002)

[13] M. Baker, H. Ong and G. Smith *A Status Report on developing a Grid-enabled Web Portal using Commodity Toolkits* (University of Portsmouth, 2001) `http://esc.dl.ac.uk/GridWG`

M. Baker, H. Ong and G. Smith *A Portal for Liquid Crystal Modelling Applications* (University of Portsmouth, 2002) `http://esc.dl.ac.uk/GridWG`

[14] M. Baker, H. Ong and G. Smith *Grid Monitoring Architecture* report (University of Portsmouth, 2002)

[15] R.J. Allan, D.J. Hanlon, K. Kleese van Dam, S. Sufi, L. Sastry, D.R.S. Boyd and B.M. Matthews *An Integrated e-Science Environment for CCLRC: Grid Portals, HPC Computation, Data Management and Advanced Visualisation* project overview (CCLRC, 2002), `http://esc.dl.ac.uk/GridWG/`

R.J. Allan, K. Kleese van Dam, L. Sastry, and B.M. Matthews *An Integrated e-Science Environment for CCLRC: Grid Portals, HPC Computation, Data Management and Advanced Visualisation* project definition, use cases and component specifications (CCLRC, 2002),

[16] O.F. Rana, V. Getov, E. Sharakan, S.J. Newhouse and R.J. Allan *Developing Grid Services with Jini and JXTA* white paper presented at GGF4, Toronto (GGF Jini Working Group, February 2002). `http://esc.dl.ac.uk/WebServices`.

[17] R.J. Allan, D. Chohan, X.D. Wang, M. McKeown, J. Colgrave and M. Dovey *UDDI and Web Services Inspection for e-Science* (UK Grid Support Centre, 2002). `http://esc.dl.ac.uk/WebServices`.

[18] Rob Allan, Dharmesh Chohan, Xiao Dong Wang and Daniel Hanlon *An Introduction to Web Services and related Technology for building an e-Science Grid* (Grid Technology Group, Daresbury Laboratory, 2002). `http://esc.dl.ac.uk/WebServices`.

[19] Rob Allan and Xiao Dong Wang *A Review of some Web Services Tools.* (Grid Technology Group, Daresbury Laboratory, 2002). `http://esc.dl.ac.uk/WebServices`.

[20] R.J. Allan and D.R.S. Boyd *The UK e-Science Grid – proposal for deployment at Level 2.* (UK Grid Support Centre, 2002)

R.J. Allan, D.R.S. Boyd and A.B. Mills *The UK e-Science Grid – Level 2 deployment plan and progress statement.* (UK Grid Support Centre, 2002-3)

[21] Rob Allan, Xiao Dong Wang, Andy Richards, Dharmesh Chohan and Mark McKeown *XML Schema for HPC Applications and Resources* (Grid Technology Group, Daresbury Laboratory, 2002). `http://esc.dl.ac.uk/WebServices`.

[22] paper with KK et al. on DataPortal for env conference...? (CCLRC e-Science Centre, 2002)

[23] R.J. Allan, D. Chohan, X.D. Wang, A.J. Richards, M. McKeown, J. Colgrave, M. Dovey, M. Baker and S.M. Fisher *Grid Information Services* Proc. AHM'2003 (Nottingham, September 2003)

D. Baker, S.J. Cox, J. Hillier, M. McKeown and R.F. Fowler *Providing a Software Toolkit to enable Operational Monitoring and Grid Integration* Proc. AHM'2003 (Nottingham, September 2003)

R.J. Allan et al. *Middleware, Applications and Tools deployed at Level 2* Proc. AHM'2003 (Nottingham, September 2003)

M.J. Leese, R. Tasker *GriMon – Grid Network Performance Monitoring* Proc. AHM'2003 (Nottingham, September 2003)

R.P. Tyer, L.J. Blanshard, K.K van Dam, R.J. Allan, A.J. Richards and M.T. Dove *Building the e-Minerals Mini-Grid* Proc. AHM'2003 (Nottingham, September 2003)

A.J. Richards, R.J. Allan, G. Drinkwater and K. Kleese *Advanced Grid Computing* Proc. AHM'2003 (Nottingham, September 2003)

[24] Rob Allan, Chris Awre, Mark Baker and Adrian Fish *Portals and Portlets 2003* UK e-Science Technical Report UKeS-2004-06 `http://www.nesc.ac.uk/technical_papers/UKeS-2004-06.pdf`

[25] *Hierarchical MDS* paper on Globus Web site

[26] David Baker *Grid Integration Test Script* ISS, University of Southampton (2002)

[27] Mike Surridge *A Rough Guide to Grid Security* IT Innovation, University of Southampton (2002)

[28] Matthew Shields *VCCE and XML Schema* University of Cardiff

[29] R. Al-Ali, O.F. Rana, D.W. Walker, S. Jha and S. Sohail *G-QoSM: Grid Service Discovery using QoS Properties* Computing and Informatics Journal, Special Issue on Grid Computing 25(5), (2003) in press

[30] Distributed Management Task Force *Common Information Model (CIM) SPecification* v2.2 (March 2000)

[31] DMTF CIM working group white paper on compute resources.

[32] DMTF CIM working group white paper on databases.

[33] DMTF CIM working group white paper on application management.

[34] DMTF CIM working group white paper on user management.

A Application Schema

The first example shows an XML description of a simple application with instances on several machines.

```
<?xml version="1.0 encoding="UTF-8" ?>
<xsd:schema xmlns="http://www.w3.org/2001/XMLSchema"
        xmlns:xsd="http://www.w3.org/2001/XMLSchema"
        xmlns:app="/IeSE/Search/appSchema.html"
        targetNamespace="/IeSE/Search/appSchema.html">

<!-- this schema is used to describe a Grid application, its
     provenance, dependencies and install base -->
<!-- ------------------------------------------------------------- -->
<element name="application" type="appType" />

<!-- separate provenance metadata from technical metadata and dynamic
     metadata (e.g. location of source, bytecode or binary) -->
<xsd:complexType name="appType">
<element name="appName" type="xsd:string" />
```

```
<element name="appKey" type="xsd:string" />
<element name="appDesc" type="appDescType" />
<element name="appProv" type="appProvType" />
<element name="appLoc" type="appLocType" />
</xsd:complexType>

<!-- technical description of code and its capabilities -->
<!-- ------------------------------------------------------------ -->
<xsd:complexType name="appDescType">
<element name="appTitle" type="xsd:string" />
<element name="webSite" type="xsd:string" minOccurs="0" />
<element name="appVersion" type="xsd:string" />
<element name="shortDescription" type="xsd:string" />
<element name="licenseScheme" type="licenseSchemeType" />
</xsd:complexType>

<!-- this schema should describe
        the type of license required for this application -->
<xsd:complexType name="licenseSchemeType">
<xsd:attribute name="license" type="licenseType" />
<xsd:simpleType name="licenseType">
        <xsd:restriction base="xsd:string">
        <xsd:enumeration type="on demand" />
        <xsd:enumeration type="once only" />
        <xsd:enumeration type="academic" />
        <xsd:enumeration type="industry" />
        <xsd:enumeration type="GPL" />
        <xsd:enumeration type="Apache" />
        </xsd:restriction>
</xsd:simpleType>
<element name="licenseIssuer" type="issuerType" />
</xsd:complexType>

<xsd:complexType name="issuerType">
<xsd:attribute name="contact" type="contactType" />
<element name="contactName" type="string" />
<!-- note key will be a UDDI key if its and organisation or a DN if
     its a person. This could facilitate automatic contract negociation. -->
<element name="contactKey" type="string" />
<!-- what about linking into a VO's tModel ? -->
<element name="tModel" type="string" minOccurs="0" />
</xsd:complexType>

<xsd:simpleType name="contactType">
        <xsd:restriction base="xsd:string">
        <xsd:enumeration type="person" />
        <xsd:enumeration type="organisation" />
        </xsd:restriction>
```

```
</xsd:simpleType>

<!-- provenance -->
<!-- ------------------------------------------------------------ -->
<xsd:complexType name="appProvType">
<xsd:attribute name="contact" type="contactType" />
<element name="contactName" type="xsd:string" />
<!-- if the contact has a DN this is all the extra information that is
     required. We can reference the person schema -->
<element name="contactKey" type="xsd:string" minOccurs="0" />
<element name="contactEMail" type="xsd:string" minOccurs="0" />
<element name="contactAddress" type="xsd:string" minOccurs="0" />
</xsd:complexType>

<!-- location of source or binaries, could be dynamically updated from MDS.
     appHost names should match res:resName names -->
<!-- ------------------------------------------------------------ -->
<xsd:complexType name="appLocType">
<element name="appType" type="xsd:string" />
<!-- need to possibly enumerate [source | binary | bytecode] ? -->
<element name="appHost" type="xsd:string" />
<!-- if the host has a key we can reference its resource schema.
     Note the dns name should also be unique -->
<element name="hostKey" type="xsd:string" minOccurs="0" />
<element name="hostPath" type="xsd:string" />
<element name="hostType" type="xsd:string" />
<element name="appFile" type="appFileType" minOccurs="0" maxOccurs="*" />
<element name="dependencies" type="appDependencyType"
         minOccurs="0" maxOccurs="*" />
</xsd:complexType>

<!-- this schema should help to describe input and
     output files, their contents and formats -->
<xsd:complexType name="appFileType">
<xsd:attribute name="open" type="readwriteType" />
<xsd:simpleType name="readwriteType">
        <xsd:restriction base="xsd:string">
        <xsd:enumeration type="readonly" />
        <xsd:enumeration type="writeonly" />
        <xsd:enumeration type="readwrite" />
        <xsd:enumeration type="append" />
        </xsd:restriction>
</xsd:simpleType>
<xsd:attribute name="dependency" type="dependencyType" />
<xsd:simpleType name="dependencyType">
        <xsd:restriction base="xsd:string">
        <xsd:enumeration type="required" />
        <xsd:enumeration type="optional" />
```

```
        </xsd:restriction>
</xsd:simpleType>
<element name="name" type="xsd:string" />
<element name="description" type="xsd:string" />
<element name="format" type="formatType" />
</xsd:complexType>

<xsd:complexType name="formatType">
<!-- to be defined -->
</xsd:complexType>

<!-- this schema should specify any other dependencies the
     program might have in order to run, e.g. memory or temporary
     disk space required, compiler version etc. -->
<xsd:complexType name="appDependencyType">
<!-- to be defined -->
</xsd:complexType>

</schema>
```

B Resource Schema

Ths first example is an XML description of a Beowulf cluster, loki.dl.ac.uk.

```
<?xml version="1.0 encoding="UTF-8" ?>
<xsd:schema xmlns="http://www.w3.org/2001/XMLSchema"
        xmlns:xsd="http://www.w3.org/2001/XMLSchema"
        xmlns:res="/IeSE/Search/resSchema.html"
        targetNamespace="/IeSE/Search/resSchema.html">

<!-- This schema is to describe a heterogeneous Grid compute
   resource -->
<element name="resource" type="resType" />

<!-- a machine has a description and may have several logical
   clusters (partitions) -->
<xsd:complexType name="resType">
<element name="resName" type="xsd:string" />
<!-- this could in fact be the fully qualified dns name -->
<element name="resKey" type="xsd:string" />
<element name="resDesc" type="resDescType" /*
<element name="accountingScheme" type="accountingSchemeType" maxOccurs="*" />
<element name="appsInstalled" type="appsInstalledType" maxOccurs="*" />
<element name="partition" type="partitionType" maxOccurs="*" />
</xsd:complexType>
```

```
</element>

<!-- description of overall machine and contacts -->
<xsd:complexType name="resDescType">
<element name="resTitle" type="xsd:string" minOccurs="0" />
<element name="webSite" type="xsd:string" minOccurs="0" />
<element name="contactName" type="xsd:string" minOccurs="0" />
<!-- if the contact has a DN this is all the extra information that is
     required. We can reference the person schema -->
<element name="contactKey" type="xsd:string" minOccurs="0" />
<element name="contactEmail" type="xsd:string" minOccurs="0" />
<element name="shortDescription" type="xsd:string" />
</xsd:complexType>

<!-- describe a partition of nodes, could be just head nodes or just
   worker nodes or various combinations -->
<xsd:complexType name="partitionType">
<element name="headNode" type="nodeType" minOccurs="0" maxOccurs="*" />
<element name="workerNode" type="nodeType" minOccurs="0" maxOccurs="*" />
</xsd:complexType>

<!-- describe an individual node of the machine, could be shared memory
    with several processors -->
<xsd:complexType name="nodeType">
<element name="nodeNumber" type="xsd:integer" />
<element name="nodeDesc" type="xsd:string" />
<element name="DNSName" type="xsd:string" minOccurs="0" />
<element name="ipAddress" type="xsd:string" minOccurs="0" />
<element name="numProcs" type="xsd:integer" />
<element name="processor" type="xsd:string" />
<element name="MHz" type="xsd:integer" />
<element name="osVersion" type="xsd:string" />
<element name="commsAvail" type="xsd:string" minOccurs="0" maxOccurs="*" />
<element name="storageAvail" type="xsd:integer" minOccurs="0" />
<element name="memoryAvail" type="xsd:integer" minOccurs="0" />
<element name="peakPerf" type="xsd:integer" minOccurs="0" />
<element name="rmsAvail" type="xsd:string" minOccurs="0" />
<element name="globusGateKeeper" type="xsd:string" minOccurs="0" />
<element name="jobManagerAvail" type="xsd:string" minOccurs="0" maxOccurs="*" />
<element name="qName" type="xsd:string" minOccurs="0" maxOccurs="*" />
</xsd:complexType>

<!-- describe the accounting done on this resource -->
<xsd:complexType name="accountingSchemeType">
<!-- to be defined -->
</xsd:complexType>

<!-- describe the applications installed on this resource -->
```

```
<xsd:complexType name="appsInstalledType">
<element name="appName" type="xsd:string" />
<element name="appKey" type="xsd:string" />
<element name="appType" type="xsd:string" />
<element name="hostPath" type="xsd:string" />
<element name="appFile" type="appFileType" minOccurs="0" maxOccurs="*" />
<element name="dependencies" type="appDependencyType"
         minOccurs="0" maxOccurs="*" />
</xsd:complexType>

</schema>
```

The second example is a UNICORE Incarnation Resource Database entry for the SGI Origin3000 green.cfs.ac.uk.

C Person Schema

This is a simple example of using XML to describe a Grid user.

```
<?xml version="1.0 encoding="UTF-8" ?>
<xsd:schema xmlns="http://www.w3.org/2001/XMLSchema"
        xmlns:xsd="http://www.w3.org/2001/XMLSchema"
        xmlns:person="/IeSE/Search/personSchema.html"
        targetNamespace="/IeSE/Search/personSchema.html">

<!-- prototype schema for a Grid user -->
<element name="person" type="personType" />

<xsd:complexType name="personType">
<element name="personName" type="xsd:string" />
<!-- person has a unique DN, so does not need a UUID.
     Other schema can refer to this. -->
<element name="personKey" type="xsd:string" />
<element name="personDesc" type="personDescType" />
<!-- list of roles the person has. Can belong to many VOs and have
     several roles in each VO -->
<element name="personRole" type="personRoleType" maxoccurs="*" />
<element name="resAuthorized" type="resAuthorizedType" maxoccurs="*" />
<element name="paymentScheme" type="paymentSchemeType" maxoccurs="*" />
</xsd:complexType>
</element>

<!-- this rescribes the person and how to contact -->
<xsd:complexType name="personDescType">
```

```
<element name="personTitle" type="xsd:string" />
<element name="shortDescription" type="xsd:string" />
<element name="webSite" type="xsd:string" minOccurs="0" />
<element name="contactEMail" type="xsd:string" minOccurs="0" />
<element name="contactAddress" type="xsd:string" minOccurs="0" />
</xsd:complexType>

<!-- this describes a person's roles in various organisations -->
<xsd:complexType name="personRoleType">
<!-- unique key to identify this role -->
<element name="roleKey" type="xsd:string" />"
<element name="voName" type="xsd:string" />"
<!-- unique business key to locate this VO in the UDDI -->
<element name="voKey" type="xsd:string" />"
<element name="roleDescription" type="xsd:string" />"
</xsd:complexType>

<!-- this describes a person's abilty to pay for Grid resource usage.
     There will be 1 or more entries for each role the user has  -->
<xsd:complexType name="paymentSchemeType">
<!-- unique key to relate this payment method to a role -->
<element name="paymentDescription" type="string" />
<element name="roleKey" type="xsd:string"  maxOccurs="*" />
<element name="paymentMethod" type="string" />
<!-- need to define main types -->
</xsd:complexType>

<!-- list of authorised resources. Could be updated by a Grid tool
     based on DN entries in grid-mapfile but supplemented with role
     information.
     There will be 1 or more entries for each role the user has  -->
<xsd:complexType name="resAuthorizedType">
<!-- unique key to relate this authorisation to a role -->
<element name="roleKey" type="xsd:string" />"
<element name="resName" type="string" minOccurs="0" maxOccurs="*" />
<!-- relate this resource to resSchema database -->
<element name="resUUID" type="uuid" /></xsd:complexType>

</schema>
```