Portalization Process for the Access Grid

Andrew Rowley, Michael Daw, Anja Le Blanc, Tobias Schiebeck, Martin Turner

Manchester Computing, The University of Manchester, UK

Abstract

The Open Middleware Infrastructure Institute has funded a project that aims to create a portal version of the Access Grid Toolkit (AGTk) client. This aims to solve problems that restrict the uptake of Access Grid technologies for a wide range of users. The main issues we intend to solve are restrictions imposed by network firewalls, bandwidth limitations, and complicated installation procedures. Our Access Grid portlet will be a fully-featured client that is compatible with both open source and proprietary versions of the technology (AGTk and inSORS). The proposed solution includes automatic multicast/unicast switching, client side bridging and stream selection for low bandwidth connections.

The original vision of the Access Grid was as an advanced collaboration environment to enable sharing of audio, video, data and applications via the Grid. Most usage has tended to focus on its videoconferencing aspects and away from Grid scenarios. With AGTk version 3 and the move to open standards such as SOAP, WSDL and XML-RPC, the development of a portal version of the client and further integration with Grid resources are now achievable.

1. Description

The Access Grid[1] is an advanced collaboration environment, in which users not only communicate using audio and video, but are also able to share data and collaborate using various add-on applications. It is designed around the idea of a group-to-group conferencing system, allowing users from different sites around the world (of which there are currently over 20,000) to join in meetings by using *"Virtual Venues"*. Various virtual venue servers have been set up internationally to allow users to connect and navigate to their chosen venues; users connect via a GUI application, namely the Access Grid Toolkit (AGTk), or a proprietary version from the company inSORS Communications[2]. The original vision was an advanced collaboration environment to enable sharing of audio, video, data and applications via the Grid although most usage has tended to focus on its videoconferencing aspects and away from Grid scenarios.

AGTk version 1 (2000) was a web-based application that allowed the user to navigate to the virtual venue and launch various applications from within that venue. AGTk version 2 (2003) was a stand-alone client so that the Access Grid was no longer accessible from a browser and AGTk version 3 (2006) replaced product-specific standards that were used to communicate with the client and were based on the Globus Toolkit, with open standards (namely SOAP, WSDL and XML-RPC) that are supported across various platforms and programming languages.

Because of this move to open standards, the opportunity has arisen to implement a new Access Grid Toolkit client integrated within a portal environment using JSR-168[3]. This portlet will have the same features as the AGTk version 3 (AGTk3) client as it currently stands, but will not require the user to install any software (other than Java) on their computer. The portlet will be able to connect to any AGTk3 server, and allow the navigation of venues, communication using audio and video, access to the jabber text chat within the venue, as well as access to shared applications and data within the venue. This will create a portlet that is fully interoperable with other AGTk3 clients and designed to be easy to incorporate future AGTk updates. This project also intends to introduce access to new forms of data storage in addition to those currently offered by AGTk. The integration with Grid technologies such as Storage Resource Broker (SRB) and GridFTP based services allows the user to access data stored within these services through the portal, and to choose data to share with other AGTk users.

The implementation of a new client will also allow us to implement some significant new features related to the operation of the Access Grid from behind restrictive firewalls and low bandwidth connections. These include the use of the AGSC Static Bridge[4] which uses a small range of ports for connection, and the use of a new bridge that uses only outgoing TCP

connections. The portlet will use a "client bridging" solution that will connect the AG services (such as audio and video tools) to a bridge running on the local machine. This will allow future networking upgrades to the Access Grid to be incorporated without the need to change the service tools themselves. This will also allow the use of low-bandwidth solutions such as the rcBridge[5] to be used within the portlet, and will allow the portlet to perform automatic multicast/unicast selection based on the current network status.

The main part of this paper describes the processes we have gone through in order to compile from relevant user feedback, the requirements and reasons for developing a portalised version of the AGTk3 and its integration within the Open Middleware Infrastructure Institute (OMII-UK) software release. The final section describes specific technology that is currently available to create this next generation vision.

2. Gathering User Feedback

The AG community has over 20,000 users across 56 countries worldwide. In the UK, the Access Grid Support Centre (AGSC) has over 100 registered room-based nodes, and over 200 additional users registered with desktop-based AG nodes. This is a broad user community, who can provide a constituency for feedback for developments arising from this project. AGSC training workshops, documentation, and the AGSC website, all provide readily accessible means for dissemination of development outcomes.

In 2006, the Access Grid Support Centre carried out a survey[6] of the Access Grid community. A similar survey has now been carried out in 2007. The 2006 survey revealed that the most common problem across sites was *"Difficulty connecting to a venue (e.g. firewall or multicast problems)"*. It was also noted that *"not being a multicast site"* was a problem as then the users had to remember to *"…switch to the unicast bridge …"*. Then in the 2007 survey, it was further noted that *"AG Firewall rules and Muilticast/Unicast are a nightmare – These need to be simplified and made more reliable"* and another user stated that *"Many partners have problems connecting because they are setting up temporary nodes. Most of the problems are due to inexperience with NAT (port forwarding) & firewall configurations"*, which shows that the situation had not improved.

In the 2006 survey, users were asked what improvements they would most like to see in the Access Grid. The first most requested improvement (requested by 35% of the respondents) was *"More reliability"* with the second being *"Greater coverage of Access Grid across institutions that do not currently have it"* (requested by 16% of the respondents). Users also requested, as extra features, that the *"Speed of the venue client"* was too slow, *"Make it more user friendly"* and *"…integration into portals …"*. In the 2007 survey, users were asked what they would like to see the AGSC provide. The users commented that *"The AGSC should work towards providing training and advice for user managed desktop access to the system"* and *"Increase in use of the personal access grid"*.

In both surveys, users were asked for additional comments. From the 2006 survey, users commented that *"Generally a lot of time is spent configuring the software …"* and that *"…Many systems exist for supporting web seminars … the potential of the AG in this respect seems to have been ignored."* From the 2007 survey one user commented that *"It needs to be as easy to use and reliable as any of the commercial video conferencing software you can buy."* Further examples of comments are *"We only use personal AG, which is a much better option for getting people interested in and using the AG …"*, *"I have to phone from China because my student didn't manage to install the AG S/W and get it working"*, *"Messenger … ConferenceXP … are much easier to setup, configure and to use"*, *"The process of configuring a personal node seems far more troublesome / complicated than it should"* and *"If I could connect to some meetings from the desktop, I would use the AG system more"*

This work addresses these issues by providing the specifications and functionality for a reliable, fast and highly usable portlet that requires minimal setup time and configuration, and that allows the Access Grid to be used more effectively from behind restrictive firewalls, and also detects and switches between multicast and unicast without user interaction. InSORS users will also benefit from this development, as the portlet allows them to access the shared applications within a venue without first installing the Access Grid Toolkit. This enables a greater degree of collaboration in meetings involving participants with inSORS and AGTk software.

2.1 Proposed User Scenarios

This portal project aims to solve problems for five user groups:

- Users located behind an institutional firewall
- Users on unreliable network
- Users on restricted/centrally managed machines
- Non-technical users
- Conference attendees

Security and firewalls are a very important issue for all networked environments. Most network and system administrators have good reasons to be paranoid about opening ports in their firewall. As understandable as this is, it always causes problems if a user wants to use network communication systems especially the Access Grid. The portal allows a user to connect to the virtual venue without opening any additional ports as it is connecting through the standard web (http) or possibly secure web (https) ports. All connections that are necessary to have a successful Access Grid session will either be done to the server running the Portal Site or have to be relayed through the web ports (http/https).

Firewalls and the reduced communication capabilities caused by them is only one problem when communicating. Another issue encountered by the second user group is the available network quality. The network bandwidth used by Access Grid is considerable, but not in itself a problem as a user on a slow connection can select which and how many streams he receives. A user on a high speed connection with multicast enabled can select all the streams available, whereas a "home" user would only select the stream of the current speaker. In the case of Access Grid the issue is not as much the speed of the network itself or the availability of multicast as it is the reliability and stability of an established communication path. If there is no multicast available for a communication a unicast bridge can be used. The problem at the moment appears when multicast is available and breaks down in the middle of a session. The portal and its background tools will aim to provide an automatic switching mechanism between multicast and unicast.

For the third user group the portal aims to cater for users on restricted/centrally managed desktops. Managed desktops are common in academia as this is the only way to allow a small number of support staff to maintain the huge number of machines in an institution. The problem with centrally managed systems is that a standard user cannot or should not install any software as any new software diverts from the "image" which makes it quickly unmaintainable. As understandable as these procedures are, they currently hinder communication through the Access Grid. This portal solution for Access Grid overcomes the issue of installing additional software, as it only uses standard applications such as a web browser with JavaScript support and a Java runtime environment configured with Java Web Start (to allow the execution of external applications), which we can safely assume to be part of any managed desktop system. Clearly these desktop systems may not have a webcam installed, and so remote participants will not be able to see the user, but the user will still be able to use audio, assuming the system has a microphone and headphone sockets (which are found and accessible on most modern systems).

Having no installation procedure also helps the fourth group of users which we call the "non technical" user. This is a group of users which will happily use the Access Grid technologies, sometimes in more innovative ways then most of the "technical" users, but don't want to go through all the hoops of setting it up. A simple solution of an "out of the box" Access Grid is very appealing to them as they are not necessarily able and most certainly not willing to go through a complicated installation procedure. The portal solution is a good alternative as it provides the "non-technical user with an "instant" Access Grid (log in and go). This again assumes the presence of a webcam on the user's computer, and access to a headset. Again, these are fairly common on modern computer systems, and where a webcam is not available, the user will be able to communicate using audio only.

The final use scenario is a conference attendee that is asked by his home institution to join an important Access Grid meeting. This case is a combination of the four previous cases as all or any of the issues discussed before may apply. Using the portlet, the user will be able to run the Access Grid without installing any software, simply by signing into the portal environment. A major barrier to using the Access Grid 'on the road' has been removed.

3. System Integration within the OMII-UK Software Release

Users within the UK e-Science community were among the early adopters of Access Grid, and this technology is still a major tool for communicating with distributed partners and collaborators. This project provides support for

using the Access Grid collaboration environment from within the OMII-UK Software release and provides a much-needed way around institutional firewalls and installation restrictions. Additionally, the software allows the sharing of files stored within SRB and/or within GridFTP based services with users of the Access Grid to broaden the integration of this technology with wider Grid technologies.

The OMII-UK software release includes a GridSphere[7] based portal which will be the base for our development. As the OMII-UK roadmap proposes to authenticate users of their respective services through Shibboleth, we will ensure that this portlet does not interfere with, but use this authentication. As the portal supports the grouping of users which have certain Shibboleth attributes it creates the basis for Virtual Organizations[8]. These Virtual Organizations allow the portlet to provide special settings such as preconfigured default venues and encryption keys.

The software will be released under a BSD style license[9] within the OMII-UK software release, to be part of the roadmap that is currently planned by OMII-UK.

3.1 Issues of Standards

The portlet will be developed to the JSR-168 standard (ensuring compatibility with JSR-268, when it becomes available). This will ensure compatibility with the OMII-UK software environment and also many other portlet environments.

The Access Grid Toolkit version 3 currently uses SOAP for communication between the client and the server, and for communication between the client and the services architecture. The event channel between the server and the client transmits SOAP-encoded event data. All communications are secured using SSL sockets. The Access Grid Toolkit client and bridge communicate using XML-RPC version 2.

H.261 video and L16 audio are used by the AGTk, and PCMU audio is used by inSORS. We will ensure that these formats are supported by the audio and video tools in our software. To this end, we will use VIC and RAT, currently being enhanced by the SUMOVER project[10], to perform the video and audio communication. Both the Access Grid and inSORS tools use RTP over UDP multicast and unicast to communicate with remote participants. We will ensure that any output traffic (from any bridges including the client-level bridges) from our portlet supports these protocols, and similarly that any traffic in this format received by the portlet is understood.

3.2 The role of a Portal within the Access Grid Environment

We now review the advantages that a portal has and how its key features relate to users' needs. Portals have developed a long way from the original Yahoo portal in the 90's. Recently Dolphin *et al.* (2002) [11] presented a simple description of a portal for their University of Hull portal project:

"A layer which aggregates, interrogates, personalises and presents information, transactions and applications to the user according to their role and preference"

They considered portals to be 'thin' if they consist of minimal linkages to relevant data and services which in implementation aids in simplified management and architecture, whereas 'thick' portals are defined as a structure that aims to provide a richer and deeper environment. JISC have considered a definition of portals as either a 'thin' or 'thick' "*…network service that provides a personalised, single point of access to a range of heterogeneous network services, local and remote, structured and unstructured. Portal functionality often includes resource discovery, email access and online discussion fora. Portals are intended for (human) end-users using common Web 'standards'...*" [12]

From our Access Grid users' needs what is required initially is a thin architecture that can evolve over time. First we should consider the features that a portal can bring to this project. Franklin (2006)[13] introduced four key features that a portal can provide. In order of preference for our Access Grid community we propose, that two of these features are essential:

1. **Desktop** – provides an abstraction that hides the operating system from the user as well as knowledge of locations of data and applications.
2. **Adaptive** – enables applications to change their behaviours depending on environment, and in this case network loading and firewall access.

and two of these features are useful but slightly less relevant at the present time:

3. **Customised** – allowing the system to learn about its users, providing for example different interfaces to serve specific needs.
4. **Personalised** – providing mechanisms for users to change parts of the portal including; the interface, behaviour and operational preferences.

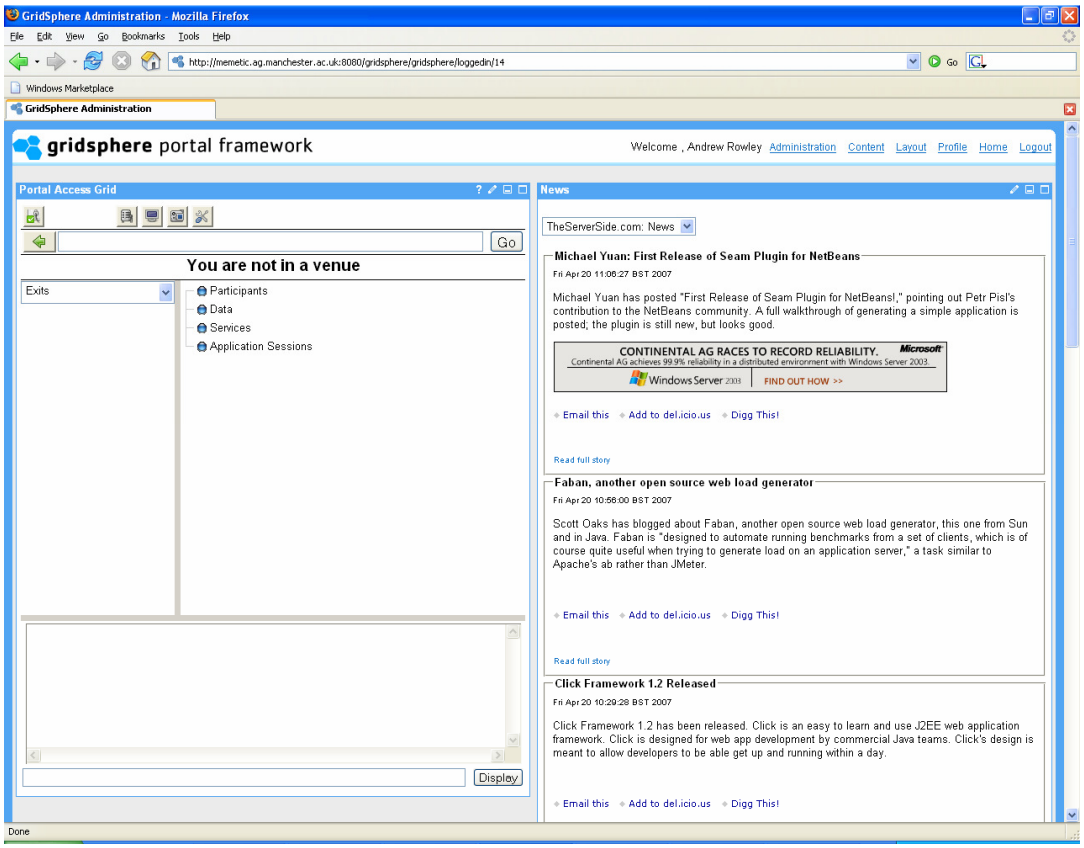

Figure 1: An example of the Access Grid portlet as it might appear in the Gridsphere portal framework. The Access Grid portlet is shown on the left, with a news feed being shown on the right.

Franklin went on to describe the key points of 'adaptive' and especially 'desktop' as being the hardest parts and often the least well integrated features within modern portals.

3.3 Proposed feature list

From these requirements, and the tools now available with current technology, we can consider a feature list for a portalised Access Grid client:

1. An Access Grid client user interface accessible entirely through a portal interface, with minimal requirements for software installation (i.e. Java only).
2. An implementation of the Access Grid node services architecture compatible with the existing AGTk node services architecture. This improves upon the current architecture by providing "client-level bridging" where node services connect to a local bridge which forwards traffic on to the Access Grid. This allows for future Access Grid network upgrades, since now only the client bridge has to be changed rather than the node service tools themselves.
3. Access to new bridging technology with minimal requirements for ports to be opened in firewalls.
4. An implementation of the current node services provided by the Access Grid Toolkit, including, but not limited to, audio and video services. These services will use the existing node service executables, and use Java to deliver the executable packages to the client computer.
5. An implementation of shared applications currently available within the AGTk, including the Shared Presentation tool, the Shared Browser, and those listed on the AGTk Contributed Software page[14].
6. Access to the AGTk shared data architecture, and the extension of this to provide access to files stored within the SRB and/or GridFTP based services.

3.4 Project timeline

The last sections itemised a series of user requests, a set of user case scenarios that are valid, and a generic technology within portal development that can be used to achieve these goals. We are currently recruiting early adopters for pilot studies to occur throughout 2007, with a final solution being integrated within the OMII-UK software release, as well as being supported by the AGSC. We aim for a pre-release in the 3rd quarter of 2007, and a first official release of the fully featured product in spring 2008.

4. Proposed technology

This section describes a more detailed set of technologies that are to be used in order to carry out the tasks described in the last section.

The intention of the project is that the portlet will provide all the features of the Access Grid Toolkit using a web interface (see Figure 1). This will be achieved through a combination of Java Server Pages (JSP), JavaScript, AJAX, Java Applets and Java Web Start.

Additionally, the portlet will provide some features not seen in the Access Grid Toolkit. These include the ability to switch between venues and between multicast and unicast without needing to restart the node services, and the implementation of a bridge client-side interface to allow the client to connect to different types of multicast-unicast bridges. This will be achieved through the use of client-level bridging. From these requirements the following lists the technical tools that will need to be implemented within the initial development phase.

4.1 The user interface

The user interface will be a JSR-168 JSP Portlet. This will look similar to the Access Grid Toolkit interface. To achieve this, the client will use HTML DIV elements to provide the overall interface, with JavaScript to allow the resizing and interaction with the interface. Where the interaction with the interface requires information from the Access Grid Toolkit server, this will be done with AJAX. This will ensure that the web page containing the portlet does not need to be reloaded in order to update the information. Where the information is to be obtained from the Access Grid Toolkit server, the JSP pages on the portal server will interact with the toolkit using SOAP XML message passing.

The Access Grid Toolkit interface contains a list of the users, data and shared applications within the venue. This list is updated dynamically whenever a user enters or leaves the venue, new data is uploaded, or a new application is started. This can be implemented in the client by having an AJAX request to the JSP on the server that only returns when a new event occurs. The same method can be used to implement the Access Grid Toolkit jabber text chat.

While this portlet will use AJAX to minimize page reloading, other portlets may not. Therefore, AJAX will also be used to store the current state of the portlet so that it can continue to operate when the page is reloaded.

4.2 Node services and shared applications

The Access Grid Toolkit currently launches external processes whenever a node service or shared application is to be used. The same executables will be used in the portlet client, but these will be distributed using signed Java Web Start applications. This has the advantage that when the service or application is updated, the client will automatically download the latest version. Additionally, when new services become available, only one user will need to install them, and they will then be available for any other user of the portlet to use.

4.3 Multicast-Unicast bridge architecture

The existing Access Grid Toolkit implementation has support for the discovery of multicast-unicast bridges. On the client side, this makes some assumptions about the bridge that is running on the server side. These assumptions make it difficult to implement bridges that do not behave like the Access Grid Toolkit default bridge implementation. The portal will implement a pluggable bridge-architecture in the client to allow other bridge types to be used without the need to update the client code.

4.4 Client-level bridging

The Access Grid Toolkit currently relies on the node services for the transmission of node service data (such as video and audio). The toolkit simply supplies the service with the multicast addresses and ports where the data is supplied from. This means that unless the service binary supports changes within this information whilst it is running, the service must be restarted to switch between venues and to switch between unicast and multicast.

Client-level bridging simply means that the client contains a network-level bridge. The client can instruct each of the services to connect to this bridge and then forward traffic to the service from the outside network and vice-versa. This means that if the user decides to change to a different virtual venue, they do not need to restart the tools. This will result in faster movement between venues since they no longer have to wait for the tools to stop and restart. Additionally, if the user experiences difficulties with multicast data, they can switch to using unicast without needing to restart the services. This means that, for example, any layout of the video streams that has been performed will be preserved.

Many of the tools used for Access Grid services are designed to work over multicast. For example, the toolkit currently has a video consumer service to receive video and one or more video producer services to send video from each of the client's cameras. All these services use the same video program which runs multiple instances, all connected to the same port number. Most operating systems allow these instances to bind to the same port number when using multicast, but when using unicast, only one of the instances will be able to receive traffic from the network. If the user is unfortunate, one of their transmitting instances may be the instance that receives the traffic, which means that the video streams will not be displayed. The client-level bridging solution will communicate with the local services using multicast with a TTL of 0, meaning that the multicast traffic will never leave the local machine. This means that the video services will receive traffic in multicast, ensuring the same experience whether multicast or unicast is being used.

Now that the client bridge is in place, additional advanced features can be implemented. Since the client bridge is passing all data to the services, it can detect when the data suddenly stops flowing from one or more of the remote participants. If the client is using multicast, this could indicate that multicast has stopped working, and therefore the client can switch to using unicast via a multicast-unicast bridge.

5. Conclusions

This portal project aims to extend the benefits of the Access Grid to a wider audience through increasing both usability and access as well as returning to its roots to integrate itself within certain Grid resources and philosophy.

We have shown firstly that there is a need for a web-based version of Access Grid Toolkit. There are many users groups that will benefit from an Access Grid client that does not have to be first installed, along with a long list of pre-requisite elements. They will also benefit from the new set of features that will be available in this client, such as automatic switching between unicast and multicast traffic, and access to low-bandwidth and firewall-friendly bridging software.

We have then demonstrated how current tools and standards can be used to create this portlet such that the user will not need to install any software other than Java.

---

[1] *Welcome to AccessGrid.org*; http://www.accessgrid.org

[2] *InSORS "Video Collaboration" company site*; http://www.insors.com

[3] *JSR-000168 Portlet Specification*; http://jcp.org/aboutJava/communityprocess/final/jsr168/index.html

[4] *Access Grid Support Centre staticBridge Service*; http://www.agsc.ja.net/services/staticbridge.php

[5] *Access Grid Support Centre rcBridge Service*; http://www.agsc.ja.net/services/rcbridge.php

[6] *Access Grid Support Centre – 2006 and 2007 Access Grid Survey Summaries*; http://www.agsc.ja.net/survey/2006/surveysummary.php http://www.agsc.ja.net/survey/2007/surveysummary.php

[7] GridSphere portal framework http://www.gridsphere.org/

[8] *Shibboleth Enabled Bridge to Access the National Grid Service (SHEBANGS)*; http://www.mc.manchester.ac.uk/research/shebangs (JISC Funded Project)

[9] Open Source Initiative OSI - The BSD License http://www.opensource.org/licenses/bsd-license.php

[10] *SUMOVER Project*; http://www.cs.ucl.ac.uk/research/sumover/

[11] Dolphin, I., Miller, P. Sharratt R. (2002) Portals, PORTALs Everywhere *Ariande* **33** http://www.ariadne.ac.uk/issue33/portals/

[12] JISC Information Environment Architecture, glossary of technical terms:

http://www.ukoln.ac.uk/distributed-systems/jisc-ie/arch/glossary/.

[13] Franklin T., (2006) Portal Architectures, in *Portals – People, Processes and Technology* ed. A. Cox, Facet Publishing.

[14] *Contributed Software*; http://www-new.mcs.anl.gov/fl/research/accessgrid/wiki/moin.cgi/ContributedSoftware