Experiences with Writing Grid Clients for Mobile devices

David E. Millard, Arouna Woukeu, Feng (Barry) Tao, Hugh C. Davis
School of Electronics and Computer Science
University of Southampton
Southampton, UK
{dem, aw1, ft, hcd}@ecs.soton.ac.uk

Abstract

This paper describes our attempts to write GRID clients for Mobile Devices, such as a PDA, which have restrictive computational and storage facilities. Our experiences are based on an implementation of a mobile GRID client for Finesse, an existing web-based e-learning system. At this stage of our work we are not looking to novel applications of mobile learning, but rather are exploring the feasibility of mobile devices as GRID platforms, with novel learning applications as our future aim. We examine Java, .NET and proxy solutions and look ahead to the next generation of WSRF-based GRID Infrastructure.

1   Introduction

There is a continuing trend for computing devices to become more heterogeneous, with rapid developments in the areas of ubiquitous and mobile systems [3]. This offers new opportunities for many application domains, for example in e-learning it opens up new possibilities for technology to enhance or ease the processes of learning and teaching; both by making existing tools more available (to remote or mobile participants) but also by offering the possibility of new applications that are not possible with traditional desk-top setups (for example, by enhancing field work).

Simultaneously there is a trend in the distributed computing and middleware areas of computing towards Service-Oriented-Architectures (SOA), and in particular the GRID is evolving as a SOA for securely orchestrating and sharing stateful services and resources across distributed organisations [4].

Mobile devices will become important resource and computing platforms in the service-oriented world, but have serious limitations, in terms of computational and storage power, compared with traditional machines.

In this paper we will report on our experiences of implementing a mobile GRID client for the Finesse e-learning system. This investigation is part of the ELeGI project, which aims at creating a common European GRID infrastructure. The ultimate goal of our work is to ensure that this infrastructure can support the next generation of novel e-learning applications, and the work on Finesse is an important first step in establishing the feasibility of various technical approaches to mobile GRID computing.

The remainder of this paper is structured as follows: Section 2 describes why we believe mobility to be an important goal for e-learning systems. Section 3 introduces Finesse and the Finesse Grid Services (FIGS). Section 4 discusses our experience in developing a Mobile Grid Client for FIGS and describes the current support in Java and .Net for mobile Grid clients, and Section 5 concludes the paper and looks ahead to the new generation of WSRF-based GRID infrastructure.

2   Why Mobility?

Arguments about how mobility could be useful for e-learning could be categorized into two areas:

- Mobility as increasing accessibility
- Mobility as an essential part of a learning application

The first of these arguments suggests that it is generally beneficial to both students and teachers to make learning resources more easily available, and mobile devices are one (of several) valid modes of access [9].

This seems to be a sound argument, and justifies our mobile Finesse prototype, but the benefits do not always match up to the costs. In particular it seems that student habits are hard to change, and that students

given mobility facilities will not always use them, even if they are seemingly obvious applications such as lecture scheduling and assessment deadline notifications [10]. Perhaps this is because of the shortcomings of mobile devices as compared to desktop systems (i.e. speed, display, interfaces, etc).

The second argument concentrates on mobility enabling computers to augment an entirely separate paradigm of learning. This paradigm is not new, it includes such things as laboratory work and field trips, but it is rarely augmented or enabled by computing devices. For example, these mobile devices can be used instead of traditional pen and paper technology to make organizing these experiences, or collecting student information, easier [11]. But they can also be used to create novel learning experiences, such as the Ambient Wood project [12], in which children were able to explore a real wood and gather data using mobile sensor technologies. They could then test their hypotheses on how the environment would change given some new parameters (such as increased rainfall), by taking new readings from the wood that were effectively from a simulation overlaid on the physical space.

We believe that constructing these rich new learning applications requires the kind of software infrastructure envisaged by ELeGI. Our goal in exploring GRID technology on mobile devices is to ensure that the infrastructure is flexible enough to support such novel learning experiences.

3 Finesse and Finesse Grid Services

Finesse (Finance Education in a Scalable Software Environment) is a web-based collaborative learning and teaching environment for the finance domain developed at the University of St. Andrews [8]. The system offers a variety of tools that allow students to manage on-line portfolios and buy and sell shares using real-time market data.

Figure 1 shows a screenshot (taken from the finesse.ac.uk website) of the interface for buying and selling shares. As well as this core functionality Finesse also allows students to compare and contrast their portfolios in order to give them direct experience of the various investment strategies used by investors.

As part of the ELeGI project's investigation into requirements for an e-learning GRID infrastructure work was undertaken at St. Andrews to create a set of GRID services that replicated the functionality of the original Finesse system.

This resulted in the Finesse Grid Services (FIGS), which is composed of five Grid Services:

- *Interest* – manages interest rate data
- *Notebook* – implements a student forum
- *Portfolio* – handles the management of virtual portfolios (including transactions)
- *Sharedata* – provides share information
- *Userdata* – provides information on FIGS users (e.g. students, tutors)

The project partners decided that to adequately test FIGS it would be necessary to work on independent client interfaces. St. Andrews developed a reference console client and also a web version that reuses the original interface (shown in Figure 1).

Figure 1: The Finesse Web Interface for Buying and Selling Shares (from finesse.ac.uk)

At Southampton, due to our interest in mobile clients, we decided to work on a mobile client interface that would make the portfolios accessible via a PDA. While this is advantageous in itself (as Finesse is all about making the simulation widely accessible, and interaction with the system is time-dependant as share prices rise and fall) it does not constitute a novel mobile learning application as we have defined them in Section 2.

Although our long term goal is to ensure that the GRID infrastructure developed by ELeGI is broad enough to support the next generation of mobile and ubiquitous learning applications, we felt that the exercise with Finesse was a crucial mile stone in achieving this.

In the next section we detail the various ways in which we have tried to get GRID clients to run on a PDA device.

4   A FIGS Mobile Client

In developing Mobile Grid clients for the Finesse Grid Services, we have investigated three main approaches:

- Java-based Grid clients
- .Net-based Grid clients
- Proxy-based Grid clients

In this section we first introduce the key technologies of the Grid and then look at the benefits of a Grid approach for mobile devices. We then present the three implementation options that we have explored.

4.1   GRID Technologies

The Open Grid Service Architecture (OGSA) is a standards-based definition of a Service Oriented Architecture (SOA) for the Grid. The Open Grid Service Infrastructure (OGSI) is the base infrastructure specification for OGSA [2].

OGSI defines a set of Web Service Definition Language (WSDL) specifications for interfaces, behaviours and schema that fulfil OGSA [3]. It specifies mechanisms for creating, naming, managing, monitoring, grouping and sharing information among Grid Services.

WSRF[15] is a refactoring of the OGSI that is fully based on Web Services. It defines a generic and open framework for modelling and accessing stateful resources. Not all of the WSRF services are fully defined and it is very much an evolving standard.

There are several implementations of the OGSI specification for different environments and platforms including the popular Globus Toolkit for Java [5], and OGSI.NET[18] for the Microsoft .NET environment.

The Globus Alliance is a research and development project with a number of academic and commercial partners that has the objective of developing a Grid infrastructure for scientific and engineering computing. The Globus Toolkit is a collection of services, written in Java, which can be used to deploy and discover other services.

Microsoft's .NET environment is a development framework that enables people to build service oriented software solutions. .NET is common across all new Microsoft platforms and has bindings to several languages, including C# and Visual Basic. The OGSI.NET implementation is a container framework that allows .Net applications to access Grid Services.

The Grid community is currently in a transition period between OGSI and WSRF, and this is reflected in the ongoing development of Grid infrastructure and tools.

Mobile OGSI.NET is an effort to develop a cut-down version of OGSI.NET tailored to mobiles devices running the .NET Compact Framework (mainly Pocket PCs). OGSI.NET is discontinued and has been replaced by WSRF.NET [19] which is an initial implementation of WSRF on .NET.

The next release of the Globus Toolkit (version 4) will also attempt to be WSRF compatible.

OGSI::Lite [16] (an experiment in developing a Perl implementation of OGSI) is now superseded by WSRF::Lite, which is a Perl implementation of the current WSRF definition.

Other Grid frameworks have been conceived with WSRF in mind. OMII_1 [17] is an open source, robust and secure Web services platform for building Grid applications, which already supports the most stable elements of the WSRF standard . It is developed by the Open Middleware Infrastructure Institute at the University of Southampton, UK.

4.2   Mobile Grid Computing

Mobile Grid computing is about making Grid Services available and accessible anytime anywhere from Mobile Devices. A number of research efforts have investigated this domain and highlighted both advantages and limitations in extending the Grid to the mobile world [1,13,14].

Wireless mobile devices are generally characterised by several limitations and constraints. The processing power is low in most of these devices (maybe with the exception of laptops), the built-in memory is low and the storage capacity is very limited compared to desktops despite the use of external memory cards by some devices. The battery life is short and display facilities are also very restricted in size and quality. Due to mobility the network is unstable (intermittent connectivity and poor bandwidth).

These limitations make mobile devices a platform that would benefit from the Grid. The main advantages of mobile Grid computing include mobile-to-mobile and mobile-to-desktop collaboration for resource sharing, improving user experience, convenience and contextual relevance and novel application scenarios. A grid-based mobile environment would allow mobile devices to become more efficient by off-loading resource-demanding work to more powerful devices or computers.

4.3 Java for Mobile Devices

The Finesse Grid Services are developed and deployed on Globus Toolkit (GT3.2) which is a Java-based environment; therefore Java for mobile devices appeared as a natural starting point for our investigation. We tried several mobile JVMs (Java Virtual Machine) on our IPAQ PPC looking for one that could support a minimum set of APIs to invoke GT3.2 Grid services. In this process, we looked at the following JVMs for mobile devices:

EWE VM [20] is a free and open source Java-based VM for mobile devices (mainly PDAs). It comes with an SDK (Ewe SDK) and provides support for advanced features such as Infra-red, Synchronisation, Swing or applets but its API's compatibility is limited to Java 1.1 and 1.2 as well as the discontinued Personal Java. This compatibility issue made it impossible for us to use the Ewe VM.

Cre-ME and CrEme are two Java-based virtual machines from NSICOM [7]. The Cre-ME 4.0 VM is compatible with the J2ME Connected Device Configuration (CDC) 1.0 which is based on JDK 1.3.1; whereas CrEme 3.25 is compliant with Sun's Personal Java, mainly based on JDK 1.1.8. These two virtual machines are based on two separate subsets of Java, neither of which supported enough of the Java API to run a Globus client.

IBM J9 VM is another mobile Java-based virtual machine. This JVM was used as part of WebSphere Studio Device Developer (WSDD), an IBM Integrated Development Environment for mobile devices using J2ME (Java 2 Micro edition). IBM J9 which supports J2ME (CDC & CLDC, MIDP & Personal Profile), the official cut down version of Suns Java. Unfortunately J"ME is very limited and was again unsuitable for our development needs.

SuperWaba VM [6] is another Java-based VM for mobile devices (PDAs and smart phones). It is part of the SuperWaba open source platform which defines a language, a virtual machine, a class file format and a set of basic classes. It has an active community and developers can use common Java IDEs, however it does not implement the standard Java specification and its API does not support Java packages. Therefore SuperWaba could also not be used in our exercise to develop a mobile Figs clients.

In the end none of these virtual machines could support client APIs to invoke the FIGS directly from our Pocket PDA because they are all running some cut down version of the Java API. We have therefore concluded that at the moment there is also no Java-based OGSI implementation for mobile devices.

Table 1 summarises the compatibility specifications for each of the JVMs.

	JVM Compatibility
EWE VM	Java 1.1, Java1.2, Personal Java
Cre-ME	J2ME CDC 1.0
CeEme	Personal Java, JDK 1.1.8
SuperWaba	JDK 1.1+
IBM J9	J2ME

Table 1: Mobile JVM compatibility

4.4 Mobile OGSI.NET

We also investigated the use of Microsoft .NET Compact Framework and looked at Mobile OGSI.NET [1] which is an implementation of the OGSI.NET implementation for mobile devices. It is built on top of Microsoft .NET Compact Framework and based on the Pocket PC 2003 or Windows Mobile operating system. It has a three-layer architecture with a *Mobile Web Server* for handling endpoint-to-endpoint communication, a *Grid Services Module* for message parsing and multiplexing, and a *Grid Service* dealing with application logic and processing. The aim is to bring the benefits of Grid computing to mobile users while addressing resource limitations and intermittent network connectivity.

While Mobile OGSI.NET emerged as the only real attempt to provide an OGSI implementation for mobile devices, we could not further our investigation into potential interoperability issues with GT3.2 because the tool itself is no longer supported and is no longer generally available for download.

4.5 Proxy-based Grid Client

Having unsuccessfully explored the two previous options, we were left with a proxy-based Grid client as the best option taking into account resource and network limitations on mobile devices.

This is disappointing as the loosely coupled architecture of Grid Services seems ideal for mobile devices that often change their networks and neighbours. A Proxy solution relies on a connection back to a trusted server, and while Grid services can be invoked this way, it looses much of the flexibility and dynamism of a genuine mobile Grid client.
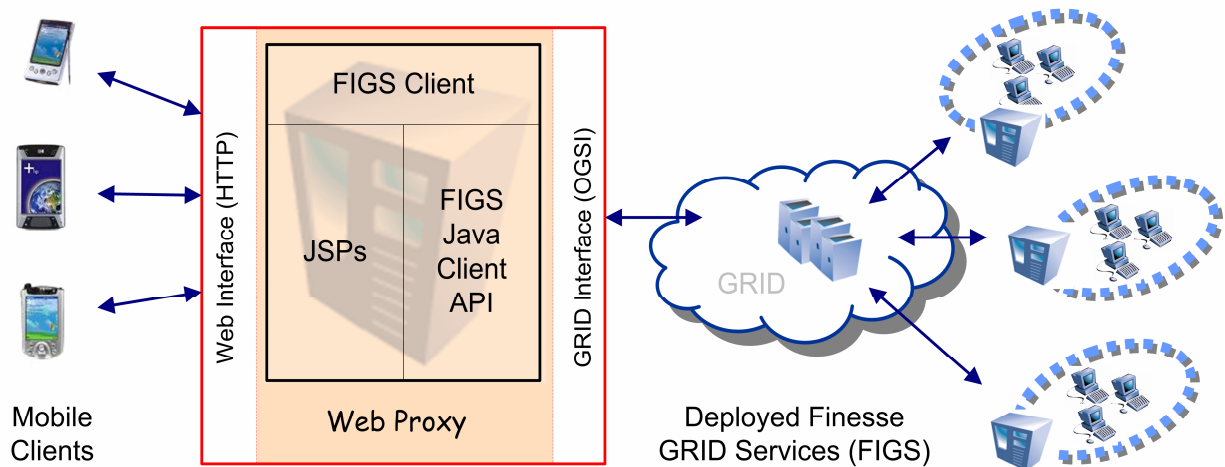
Figure 2: Proxy Architecture for Mobile Clients

Java Server Pages (JSPs) are a lightweight way in which to invoke Java code on a remote web server. So we decided to implement our mobile client using JSPs, as depicted by the architecture diagram in Figure 2.

The Finesse Grid Services are deployed on the Grid, and clients are implemented as a set of Java Beans talking to the Grid and rendering appropriate responses as JSP pages to mobile devices. Requests from mobile users are input via the mobile device's Web interface, these requests are first handled on the proxy by client beans which then issue proper Grid Services requests to the Grid. Once the proxy receives responses from the Grid, it generates and serves appropriate pages to the mobile client.

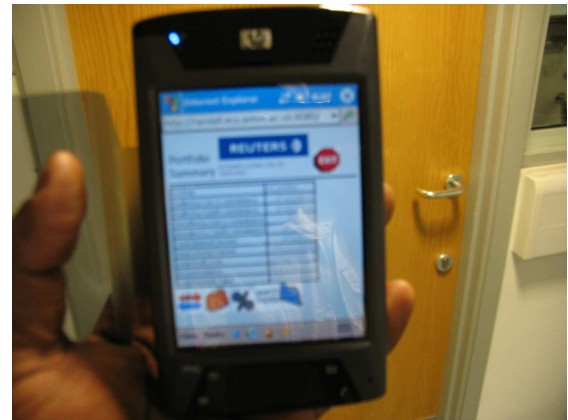Figure 3 shows the web proxy viewed on a mobile device, in this case a PocketPC based PDA.

5   Conclusion

In this paper we have detailed the justifications for pursuing a mobile GRID client for the domain of e-learning. We have looked at three options for a mobile GRID implementation:

1. The ability of the current generation of mobile Java Virtual Machines to support common GRID implementations, such as GT3 [3].

2. The ability of .NET to invoke remote GRID services from a mobile device [4].

3. Thirdly, the option of using a proxy to invoke GRID services, e.g. via a web server.

Figure 3: Web pages viewed on PDA device

We conclude that there is currently no OGSI Java API for mobile devices, and that OGSI.NET is inappropriate because it is not available on mobile devices (despite promising initiatives), has questionable interoperability with existing GRID systems such as GT3, and has been deprecated in favour of WSRF. We believe that in the future WSRF may improve support for GRID clients on mobile devices because it is fully web services based and has a simplified API, but a good reference implementation may take some time to emerge.

Our solution is to implement a web-based proxy that uses GT3 to talk to distributed GRID services, and then makes this functionality available via a web interface specially designed for mobile devices with limited display capabilities.

6   Acknowledgments

The authors would like to thank Colin Allison at the University of St. Andrews and his team for supporting the FIGS prototyping work described in this paper.

7   References

[1] D. Chu and M. Humphrey, "Mobile OGSI.NET: Grid Computing on Mobile Devices", 5th IEEE/ACM International Workshop on Grid Computing - GRID2004 (associate with Supercomputing 2004). Nov 8 2004, Pittsburgh, PA.

[2] Open Grid Services Infrastructure (OGSI), Version 1.0, available as http://www-unix.globus.org/toolkit/draft-ggf-ogsi-gridservice-33_2003-06-27.pdf (last accessed Feb 2005)

[3] M. Weiser, 'Ubiquitous Computing', Computer, v.26 n.10, p.71-72, October 1993

[4] Ian T. Foster, 'The Anatomy of the Grid: Enabling Scalable Virtual Organizations', Proceedings of the 7th International Euro-Par Conference Manchester on Parallel Processing, p.1-4, August 28-31, 2001

[5] Thomas Sandholm, Jarek Gawor, 'Globus Toolkit3 Core – A Grid Service Container Framework', http://www-unix.globus.org/toolkit/3.0/ogsa/docs/gt3_core.pdf (last accessed Feb, 2005)

[6] SuperWaba, The real Power of Mobile Computing available at http://www.superwaba.com.br/en/default.asp (last accessed Feb 2005)

[7] Cre-ME and CrEme Java-based virtual machines, available at http://www.nsicom.com/Default.aspx (last accessed Feb 2005)

[8] Power D., Michaelson R. and Allison C., 'The Finesse Portfolio Management Facility', Proceedings of the 9th Annual CTI-AFM Conference, pp119-125, April 1998 CTI-AFM Publications, East Anglia

[9] T.C. Liu, H.Y. Wang, J.K. Liang, T.W. Chan, H.W. Ko & J.C. Yang, 'Wireless and mobile technologies to enhance teaching and learning', Journal of Computer Assisted Learning, Volume 19 Issue 3 Page 371 - September 2003

[10] John Traxler, Brendan Riordan, 'Using personal digital assistants to support computing students', LTSN-ICS 5th Annual Conference, Belfast, 31 Aug – 2 Sept, 2004

[11] Rieger, R. & Gay, G. (1997) Using Mobile Computing to Enhance Field Study. In Proceedings of the Computer-Supported Collaborative Learning Conference - CSCL'97 (eds. R. Hall, N. Miyake & N. Enyedy) pp. 215-223. AACE, Charlottesville, VA

[12] Rogers, Y., Price, S., Fitzpatrick, G., Fleck, R., Harris, E., Smith, H., Randell, C., Muller, H., O'Malley, C., Stanton, D., Thompson, M. and Weal, M. J. (2004) Ambient Wood: Designing New Forms of Digital Augmentation for Learning Outdoors. In Proceedings of Third International Conference for Interaction Design and Children (IDC 2004), pages pp. 1-9, Maryland, USA.

[13] Stan Kurkovsky, Bhagyavati, Arris Ray, Mei Yang. 'Modeling a Grid-Based Problem Solving Environment for Mobile Devices', In Proceedings of the International Conference on Information Technology: Coding and Computing (ITCC'04), Las Vegas, Nevada, April 05 - 07, 2004.

[14] Bhagyavati and S. Kurkovsky. "Wireless Grid Enables Ubiquitous Computing", in Proceedings of The 16th International Conference on Parallel and Distributed Computing Systems(PDCS-2003), Reno, NV, Aug. 2003.

[15] The WS-Resource Framework, available at http://www.globus.org/wsrf/specs/ws-wsrf.pdf (last accessed Feb 2005)

[16] OGSI::Lite and WSRF::Lite - An Implementation of the Web Services Resource Framework., available at http://www.sve.man.ac.uk/Research/AtoZ/ILCT and http://vermont.mvc.mcc.ac.uk/RealityGridWorkshop2.pdf

[17]OMII_1.1.1 available at http://www.omii.ac.uk/ (last accessed Feb 2005)

[18] G. Wasson, N. Beekwilder, M. Morgan, and M. Humphrey. OGSI.NET: OGSI-compliance on the .NET Framework. In 4th IEEE/ACM International Symposium on Cluster Computing and the Grid (ccGrid 2004). Chicago, Illinois. April 19-22, 2004.

[19] M. Humphrey and G. Wasson. Architectural Foundations of WSRF.NET. Journal of Web Services Research. 2005. To appear available at http://www.cs.virginia.edu/~humphrey/papers/ArchFoundationsWSRFdotNET.pdf

[20] Ewe Virtual Machine, available at http://www.ewesoft.com/ (last accessed Feb 2005)