

AstroGrid as an e-Learning Environment

P. Contreras

School of Computer Science, The Queen's University of Belfast
Belfast, BT7 1NN, United Kingdom
p.contreras@computer.org

F. Murtagh

Department of Computer Science, Royal Holloway, University of London
Egham, Surrey TW20 0EX, UK
fmurtagh@acm.org

Abstract

We present the concept of the Virtual Observatory through the AstroGrid project, which is a Virtual Organization making intensive use of the Internet to produce a working data-grid for Astronomical applications.

AstroGrid has implemented a series of tools which facilitate collaborative work such as news, forum and discussion pages, all of them inspired by the open source and licensing philosophy. Additionally we explore how such an Astronomical Grid project can be used in the future as an e-Learning service platform and as a tool for a much wider audience than researchers and astronomers, such as students and the wider public interested in this area.

Keywords: AstroGrid, grid computing, collaborative work, e-Learning

1. INTRODUCTION

The terminology "Grid Computing" has been widely adopted in reference to the use of distributed computer resources across a network, working in a coordinated way, and sharing not just information but its processor cycles.

Grid technology is applied in a number of scientific areas, from climate simulation to finance and astronomy just to name a few. In order to achieve these different Grid networks a series of guidelines, standards and protocols are needed, and a range of working group such as those affiliated to the Global Grid Forum [7] have been established.

The astronomical area is no exception to this trend: astronomers and computer scientists are working to ensure standards for a data-grid platform. This context is where AstroGrid is playing an important role.

The paper is structured as follows: firstly we present AstroGrid and its relationship with the Virtual Observatory, and secondly the AstroGrid collaborative environment and its architecture is shown. The final section explores e-Learning scenarios for AstroGrid and e-Learning architectures for the Virtual Observatory.

2. ASTROGRID AND THE VIRTUAL OBSERVATORY

AstroGrid [2] is a £10 million project aiming to produce more economic, faster and effective astronomy through the generation of open standards and interfaces. The AstroGrid framework helps to enable different data centres across the United Kingdom to publish services and data into a data-grid infrastructure accessible on the Web. Such a framework also makes it easier for the different astronomical data centres to interact; it offers a wide range of analysis and visualisation tools through a common interface.

AstroGrid is not alone in its attempt to standardise access to astronomy information. It is part of a worldwide effort to transform astronomical data repositories into a world Virtual Observatory (VO), an organisation of which it is also a founding member. Such an initiative is led by the International Virtual Observatory Alliance [9] (IVOA) movement, which consists of around sixteen active international members.

AstroGrid itself can be considered as a Virtual Organisation, with its members working in institutions across the United Kingdom. To achieve its goals AstroGrid bases its work on collaboration, and most of its generated information, including almost all aspects of its work processes, is open to the public and accessible over the Internet.

The following section describes the collaborative approach taken in the development of the AstroGrid project – the so-called *extreme programming* software engineering model based on continuous rapid prototyping and code integration.

3. ASTROGRID DEVELOPMENT PROCESS AND COLLABORATIVE WORK

The AstroGrid philosophy is one of an open collaboration of researchers and technologists. To allow these people to work in partnership effectively, a range of web-based tools is required. In such a collaborative software development project there are different obstacles to overcome before the collaborative development can become effective. Firstly, good communication is essential in many aspects. Developers need to be aware of information regarding the project as a whole, information regarding specific sections, and they need to be able to directly communicate with other developers. The latter is of high importance in the AstroGrid case. Developers working within the project are assigned tasks in so-called development cycles which are specified by astronomers through definition of science cases. Developers will work on a certain task or project module for a few months before being moved to another area. In some cases more than one cycle or iteration is necessary to achieve a goal. In any case task assignment is under constant review by the project leaders, with a view towards reallocating human resources when and where they are needed.

The benefit of an approach like this is a highly cross-trained development team. For example in the case of a member's absence another developer who has previously worked on that area can be assigned the task. The fact that, at some point, they would have worked in the particular area will mean they can quickly re-integrate themselves into the development team for the particular project module. However, adequate facility for communication must be in place for developers to be able to exchange ideas and to make and to respond to queries, so that the development process is as efficient as possible. One important issue in relation to this cyclic development process is development by second parties. As well as communication facilities there are other ways to ensure that the collaboration is as effective as possible.

Standards play an important role in facilitating code development by second and third parties. Standard forms of software design and standard coding styles all play a part in reducing development time. Standard software design methods will aid in code integration and coding style standards will help reduce the time required for second party developers to familiarise themselves with software. Standard forms of code documentation can also greatly aid second party developers.

3.1. AstroGrid Development Process

Support for second party development in AstroGrid can be classified into three areas. Firstly, there are document repositories, news, notification, and a discussion forum to facilitate communication between members of the project. Secondly, there are the standards for the design and implementation of software generated by the project. Finally a considerable amount of documentation is required once the software is working. This documentation includes documents for the actual code, integration documents and testing documentation. Then a range of tools is used to tackle the problems posed by AstroGrid's needs for effective collaboration.

3.2. Communication

To facilitate good communication practice any person with expertise or simply an interest in the Virtual Observatory concept, Grid technologies or e-Science is free to post or comment upon news articles, participate in the forum exchanges or post documents making use of the AstroGrid portal. The portal provides an access point to the various communication tools available. The Wiki pages provide support for the posting of relevant documents for the project. Wiki is a leading-edge, web-based collaboration platform targeting the corporate Intranet world. TWiki [16] fosters information flow within an organization. Some

Wiki features are: web browser, auto links, text formatting, search, e-mail notification, structured content, attachment, templates and skins, statistics and plug-in resources. Further to the Wiki pages the AstroGrid portal provides a forum facility which provides a number of grouped discussion areas. The groups deal with site-related topics, project-specific topics and general areas such as virtual organisations and Grids. To allow developers to be kept up to date with various information about articles posted, items of news, events and polls, a news section is available. The idea of a central portal providing access to the various communication platforms is essential to effective collaboration. There is no confusion over how to contact other members as all contacts are made through the use of one central site. Further to these web-based community-oriented communication tools, one-to-one contact is necessary in order to ensure that developers can carry out their tasks efficiently. For example, in the case of second party development the ability to contact a previous developer to query some aspect of the work greatly reduces re-development time. To facilitate the necessary direct communications between project members, project contracted people keep close contact through e-mail, telephone conferences, chat programs and regular meetings.

3.3. Code

Integration Code is tackled using Current Version System [5] to store, modify and keep track of code changes. This has various uses in software development practice helping returning developers to identify and understand changes that have been made since they last worked on a particular project area. JUnit [10] Test is used to prove code integrity and ensure quality of service. JUnit is a regression testing framework for code made with Java. Usually a unit test exercises some particular method in a particular context in order to prove that a specific piece of code does what it is supposed to do. Code bugs are tracked using Bugzilla [3], a “Defect Tracking System” or “Bug-Tracking System”. Defect Tracking Systems support individuals or groups of developers to keep track of outstanding bugs in their software effectively. Once an error, misbehaviour or unexpected answer has been detected, this will be described and assigned to a developer in the form of a ticket which contains all the information related with the bug.

In order to support re-use or re-development the following methods of providing reliable high quality software are important. When a piece of code is developed, in order for further work to be effectively carried out, it is essential that the product is of high quality. To provide standard coding practices a number of tools are used in the AstroGrid project. A range of free Integrated Development Environment (IDE) tools such as Eclipse [6] and JBuilder Foundation are used during the development process. These, in conjunction with code design standards (i.e. UML), are needed to establish the basis on which to allow different people in different places to modify each others’ code easily, the philosophy being that if everybody is using the same tools to build a product the components of the product can be more easily modified or enhanced.

The main programming language used in AstroGrid is Java. Different standards like UML, XML and Web Services are used for the design process to maintain consistency. When implementing code in Java a coding standard is utilised, which allows different people to understand the work of each other, maintain coding style and consistency. This leads to a cleaner and easier coding process. AstroGrid code guideline standards are described in Vermeulen et al. [17]

3.4. Documentation

Code management and documentation is addressed mainly through the use of standard Java code documentation and Maven [12]. Maven is a Java project management and project comprehension tool. It is based on the concept of a project object model (POM) where all the Maven objects are a result of a well defined model. Builds, documentation, source metrics, and source cross-references are all controlled by the POM. Maven aims to make the developer’s life easier by providing a well defined project structure, well defined development processes to follow and a coherent body of documentation that keeps developers aware of what is happening within the project. Maven alleviates a lot of what most developers consider drudgery and lets them get on with the task at hand. This is essential in projects where there are not many people dedicated to the task of documenting and spreading the critical information about the project, which is necessary in order to dedicate resources to other critical tasks as coding and coding testing.

3.5. Security

One problem not included in the above section is security and the identification of which information is publicly accessible and which is not. As AstroGrid is an open project, almost every document is publicly accessible through the Internet. One exception is the code, where there are restrictions in place as to who is authorized to upload and integrate code. In this case SSH [15] protocol is used to allow secure communication between developers and the server repository. Though remote login is the primary use of SSH, the protocol can be used as a general purpose cryptographic tunnel, capable of copying files, encrypting email connections, and triggering remote execution of programs. Currently AstroGrid uses SSH Version 2 which operates over TCP. In its simplest mode of operation, it connects to a server, negotiates a shared secret key, and then begins encrypting the session. A username and password are passed over the encrypted session and, if authenticated, the server starts a command shell over the encrypted session.

4. ASTROGRID AS ASTRONOMICAL GRID ARCHITECTURE

In concordance with the construction of the VO, AstroGrid has a modular architecture [11], open to all contributions from either data, services or resources.

Figure 1 shows AstroGrid modular architecture at a high level which we explain in this section.

Portal: This is a server-based component which delivers an interface for accessing services in the VO. In AstroGrid all the components which need to interact with the users will do so through a *Portlet*. A portlet gives a modular and flexible layer. Any developer is able to add his/her own functionality by wrapping it on a portlet.

Community: This allows the construction of flexible communities within a group. Then once the permission has been assigned to a community administrator, he/she will be responsible for assigning rights to individuals.

Workflow: This enables the construction of complex tasks like building queries and data analysis, upload/download data and render the output in a different format like tables or images.

Registry: This contains metadata descriptive information about resources available, which is based on IVOA standards. A resource can be a data set, web service, service, information on other registries, etc.

Common Execution Architecture: This concept also called by the initials CEA is an abstraction of application components and the necessary parameters to run it.

Data Set Access: This is an interface component able to take a standard query, translate it into a dataset and execute it, and return later the result to the data centre.

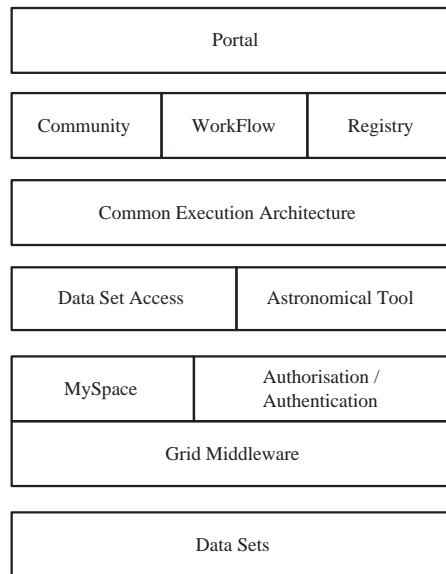
Astronomical Tool: A number of essential astronomical tools such as object catalogue builder SExtractor [14] and photometric redshift analyser HyperZ [8] (which are freely available) will be incorporated to AstroGrid through a portal wrapper.

MySpace: This defines a virtual space to allocate both temporal and long term data, such as data sets generated by queries submitted to databases. MySpace is not necessarily in a local repository, but it will interact with the user as easily if it were on a local machine.

Authorisation/Authentication: This is the component in charge of identification and granting of access to the users, as well as maintaining security in AstroGrid.

Grid Middleware: This allows AstroGrid to integrate different astronomical data centres and to share resources in a coordinated way.

Data Sets: Data allocated in a distributed way in different data centres across the United Kingdom in AstroGrid's case, and across the globe once the Virtual Observatory alliance will be fully working.

**FIGURE 1:** AstroGrid Architecture

5. ASTROGRID AS E-LEARNING CASE SCENARIO

There is no single definition for the e-Learning concept just yet. We consider e-Learning as use of any electronic device to assist the education or training process, taking advantage of the Internet or any other communication channel to connect other devices to deliver information and knowledge.

The possibility of using the Virtual Observatory for e-Learning is at the moment limited and reduced to the research level. This is mainly due to the early stages of development and because a fully usable version has not been released yet.

When working, the Virtual Observatory is expected to provide a data-grid platform for astronomers and the scientific community. But it also can provide a good chance for students to experience grid computing in a real-case scenario.

This goes beyond the use of AstroGrid accounts to access information in the VO. Students will be able to implement experiments in areas such as image processing, algorithms and data mining in computer science related areas, just to name a few. But e-Learning does not stop there: it will be possible for mathematicians, statisticians, engineers, physicist and astrophysicists to use this platform to implement experiments or to investigate theories.

The most interesting aspect would be the interaction of these individuals doing work as a group driven by common interest. Just as research projects such as AstroGrid work as a collaboration of individual researchers, students can reenact this by working in interdisciplinary groups, using the platform to learn to effectively share individual knowledge and expertise.

This scenario, however, needs adequate tools to coordinate, support and monitor such virtual groups. A starting point is to take Open Source software to generate an adequate bundle of software that will enable students to organise such a group.

5.1. e-Learning Service Architecture for the Virtual Observatory

The scenario presented above is not just a challenge for teachers and students: it also presents a challenge for the scientist working in today's Virtual Observatory.

Questions such as the following are on the agenda:

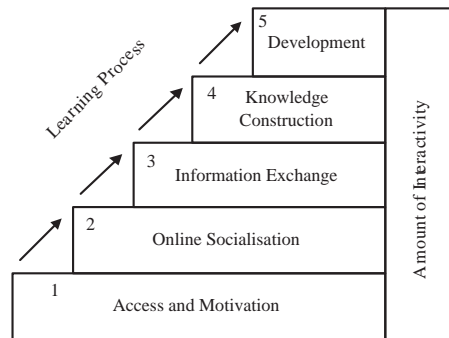


FIGURE 2: Model for understanding learning via CMC

- What kind of service would we like to offer to e-Learners?
- What resources are the Virtual Observatories willing to allocate for non-heavy scientific analysis?
- How are we going to present such information in a friendly, open way?
- How will effectiveness and efficiency of VO e-Learning systems be evaluated?
- Moreover how well prepared is our architecture of today to incorporate future e-Learning schemes?

Essentially all Virtual Observatories are based on open standards such as XML and Web Services. This makes the architecture transparent for incorporating customised plug-ins or extensions in the case of designing e-Learning modules.

The problem we find here, and must address, is that VO architectures can reach very complicated states, and the more complicated this is the more difficult to incorporate additional modules that can respond adequately to the learning process.

As our starting point we can consider Gilly Salmon's model for understanding learning via CMC [13] for implementing e-Learning modules that can be on the top of AstroGrid architecture. This model, shown in Figure 2, is derived from analysis of Computer Mediated Communication and it has become established as a course design tool. Salmon presents a five layer process leading to knowledge construction.

The advantage presented by this model is that it is rather close to the AstroGrid philosophy, since this approach is not formally but intrinsically related to the way that AstroGrid is developing today. Also it is a non-exclusive model, for example to ensure quality of services. Learning via the CMC model can interact with Allison's [1] layers of distributed learning environments.

The Salmon model helps to understand the kind of service that we would like to offer to potential e-Learners. In a more concrete scenario we can consider the implementation of the Intelligent Web Teacher Platform (IWT) as described by Capuano et al [4].

IWT is a distributed learning platform which offers external plug-in connectors that can be designed specifically to connect AstroGrid Portal. Since AstroGrid supports special designed portlets to access its resources, IWT presents a potential solution to facilitates the creation of an e-Learning environment for Astronomy.

6. CONCLUSIONS

This paper explores the current state of AstroGrid Virtual Observatory collaborative work, and the way in which the project is organized to reach its goals.

We have discussed how this kind of grid-enabled project could be used as an e-Learning platform and environment, and we present two possible e-Learning models to be implemented in the Virtual Observatory.

REFERENCES

- [1] C. Allison. (2003) *Quality of service requirement for the e-learning grid*. In *2nd International LeGE-WG Workshop on e-Learning and Grid Technologies: a Fundamental Challenge for Europe*, March. <http://ewic.bcs.org/conferences/2003/2ndlege/session2/paper3.htm>
- [2] AstroGrid. (2005) *Astrogrid portal*, <http://www.astrogrid.org>
- [3] BugZilla. (2005) *Bug-Tracking System*, <http://www.bugzilla.org/>
- [4] N. Capuano, A. Gaeta, G. Laria, F. Orciuoli, and P. Ritrovato. (2003) *How to use grid technology for building the next generation learning environments*. In *2nd International LeGE-WG Workshop on e-Learning and Grid Technologies: a Fundamental Challenge for Europe*, March. <http://ewic.bcs.org/conferences/2003/2ndlege/session1/paper3.htm>
- [5] CVS. (2005) *Concurrent Versions System*, <https://www.cvshome.org/>
- [6] Eclipse. (2005) *Eclipse Foundation*, <http://www.eclipse.org/>
- [7] G. Forum. (2005) *Global Grid Forum*, <http://www.gridforum.org>
- [8] Hyperz. (2005) *Photometric redshift*, <http://webast.ast.obs-mip.fr/hyperz/>
- [9] IVOA.(2005) *International Virtual Observatory Alliance*, <http://www.ivoa.net/>
- [10] JUnit.(2005) *JavaTM Testing Framework*, <http://www.junit.org/>
- [11] T. Linde. (2003) *Astrogrid Architecture Thesis*, <http://wiki.astrogrid.org/bin/view/Astrogrid/ArchitectureThesis#Bib3>
- [12] Maven. (2005) *Apache Maven Project*, <http://maven.apache.org/>
- [13] G. Salmon. (2000) *E-moderating: the Key to Teaching and Learning Online*. Kogan Page Ltd, London.
- [14] SExtractor. (2005) *Object cataloguer for astronomical images*, <http://terapix.iap.fr/rubrique.php?idrubrique=91/>
- [15] SSH. (2005) *Secure Shell*, <http://www.ssh.com/support/downloads/secureshellwks/non-commercial.html>
- [16] TWikiTM. (2005) *Enterprise CollaborationPlatform*, <http://www.twiki.org/>
- [17] A. Vermeulen. (2005) *The Elements of JavaTMStyle*. Cambridge University Press, Cambridge.