Integrating eSAD and VRE_SDM

ENGAGE Program

David Wallom, Segolene Tarte, Tiejun Ma, Pin Hu and Kang Tang

Background (0.5 pages)

This proposal brings together two research initiatives;

eSAD - The Image, Text, Interpretation: e-Science, Technology and Documents project, a collaboration between OeRC, Classics and Engineering within Oxford and the library and information studies school in UCL

And

VRE-SDM - The VRE for the Study of Documents and Manuscripts a collaboration between OeRC, Classics and Phonetics within Oxford.

The project links all of the expertise that the eSAD project has with image enhancement and processing with the user community that has been built up around the VRE supporting the humanities and classics. These researchers have a significant requirement for the tools developed which should enhance their productivity as well as get them up and running on the UK national e-infrastructure.

The development teams on this project were wholly based within the OeRC liaising through Dr Segolene Tarte to eSAD and Ruth Kirkham to VRE-SDM. Integration with the NGS was enabled through the close linkage between the OeRC and core part of the NGS team.

Aims and Objectives (0.5 pages)

The eSAD project uses computing technologies to aid experts in reading ancient documents, creating tools which can aid the reading of damaged texts like the stilus tablets from Vindolanda.

The VRE for the Study of Documents and Manuscripts is a project designed to complement research into damaged and illegible documents. The pilot VRE allows researchers access to image collections, supported by annotation tools and reference and support resources from around the world. The project incorporates communication and collaboration tools to allow researchers to work with widely dispersed colleagues with access to advanced imaging tools.

The aim of the project has been to enable the use of eSAD developed image processing algorithms within the framework developed within the VRE-SDM. This will mean bridging the gap between the use of the NGS for image processing and the web based access mechanisms used by this field of researcher.

The image processing algorithms, simple ones such as brightness and contrast adjustment, illumination correction, woodgrain removal, as well as more complex ones such as stroke detection, are offered as functionalities wrapped in one or several web-services and presented to the user in a portlet in the VRE-SDM application. Similarly, access and search through the knowledge base currently being constructed as part of the ISS would also be offered either within the same portlet as the image processing tools or in a different portlet in the VRE-SDM application. The underlying algorithms are wrapped in an application that has been installed on the UK NGS using the Uniform Execution Environment tools developed to ensure common locates for different software packages where they are installed on a number of different physical resources.

Before this project has been completed this was difficult due to the algorithms and tools within eSAD being developed within the MATLAB software environment and were as a consequence difficult for a non-computationally aware researcher t get access to. The main highly complex algorithms also took significant processing to run on a single system, which made them significantly cumbersome for a researcher to try multiple methods on an image and experiment with the best results. Access to distributed computing resources for example was not possible and using the open standard job submission mechanisms that are available using GridSAM means that users are also not constrained to one single system as a back end.

The interface that had been developed within the MATLAB environment was also not conducive to the long term storage and maintenance of the raw or processed images and so integration with the data model within the VRE was necessary. This allows metadata to be associated with the input and processed images and a record to be kept of the operations that were performed on the image.

Methodology and Implementation (1-2 pages)

Within the system there were in separate strands of development.

1) Development of installation systems for the algorithms fitting into the NGS Uniform Execution Environment

The algorithms used for the image enhancement and analysis for the project have been developed within the MATLAB toolkit. Due to licensing costs and restrictions the underlying MATLAB software cannot be installed onto all nodes within even a single institution let alone the whole of the NGS. It was important then that we were able to devise a mechanism by which the MATLAB compiler could be used to provide distributable binaries that were then called by the user code. This also meant that we have to distribute the MATLAB runtime libraries with the binaries. The initial building of the binary files was done on an architecturally compatible system within the OeRC.

Once compiled the resulting files and libraries have to be packaged into a simple 'tarball' that can then be copied onto the system from which they will be executed. Within the tarball there should also be packaged the developed installer script that has been developed. This not only copies all the executable files into areas on the system which are shared between head and execution nodes but also creates the soft links etc that have to be made to allow the software to operate within the NGS uniform execution environment.

A testing script has also been developed to ensure that before the user ever has to run the system then we are able to ensure that any failures will not occur because of incorrectly installed software etc.

2) Development and Implementation of the User interface Portal

The portal is an extension of the system developed in the "Building a Virtual Research Environment for the Humanities" (BVREH) project. It keeps all the functional concepts proposed in the BVREH project and delivers additional functions to facilitate secure and seamless distributed image analysis to users. Furthermore, the portal part transforms most server-side (Java) logic used in the BVREH system to "Asynchronous JavaScript and XML" (AJAX) based dynamic client-side (JavaScript) components in order to make the system much easier to integrate into different Web application or Portal frameworks; enhance the interactivity between server and client; and eliminates unnecessary page reloads at the user-end. Overall, the portal part of the system delivers a set of functions in both frontend and backend, including:

(Frontend)

- Search original images or processed images obtained from analysis;
- Display thumbnails and image metadata;
- Display single or tiled images;
- View and edit annotations on images;
- Select algorithms for analysis;
- Select region of interest for analysis;
- Run analysis on individual images or a group of images belonging to the same object view;
- View real-time GridSAM job status;
- Dually display single images.

(Backend)

- Access metadata from TripleStore;
- Manage and locate images;
- Generate JSDL job descriptions;
- Asynchronous GridSAM job submission and monitoring;
- Make tiles for processed images;
- Generate and add metadata for processed images.

Engage Data Model

The system uses the ENGAGE data model to manage images and related information e.g. analysis and annotation. The data model expands the BVREH data model (see the gray area in Figure 1) and generally classifies images into two categories: original images and processed images.

The Engage data model uses two properties: bv:azim and bv:elev to describe different light angles in taking original images. This allows not only users to know the light angle information for an original image, but also image analysis software to be able to define what algorithms to use based on the information.

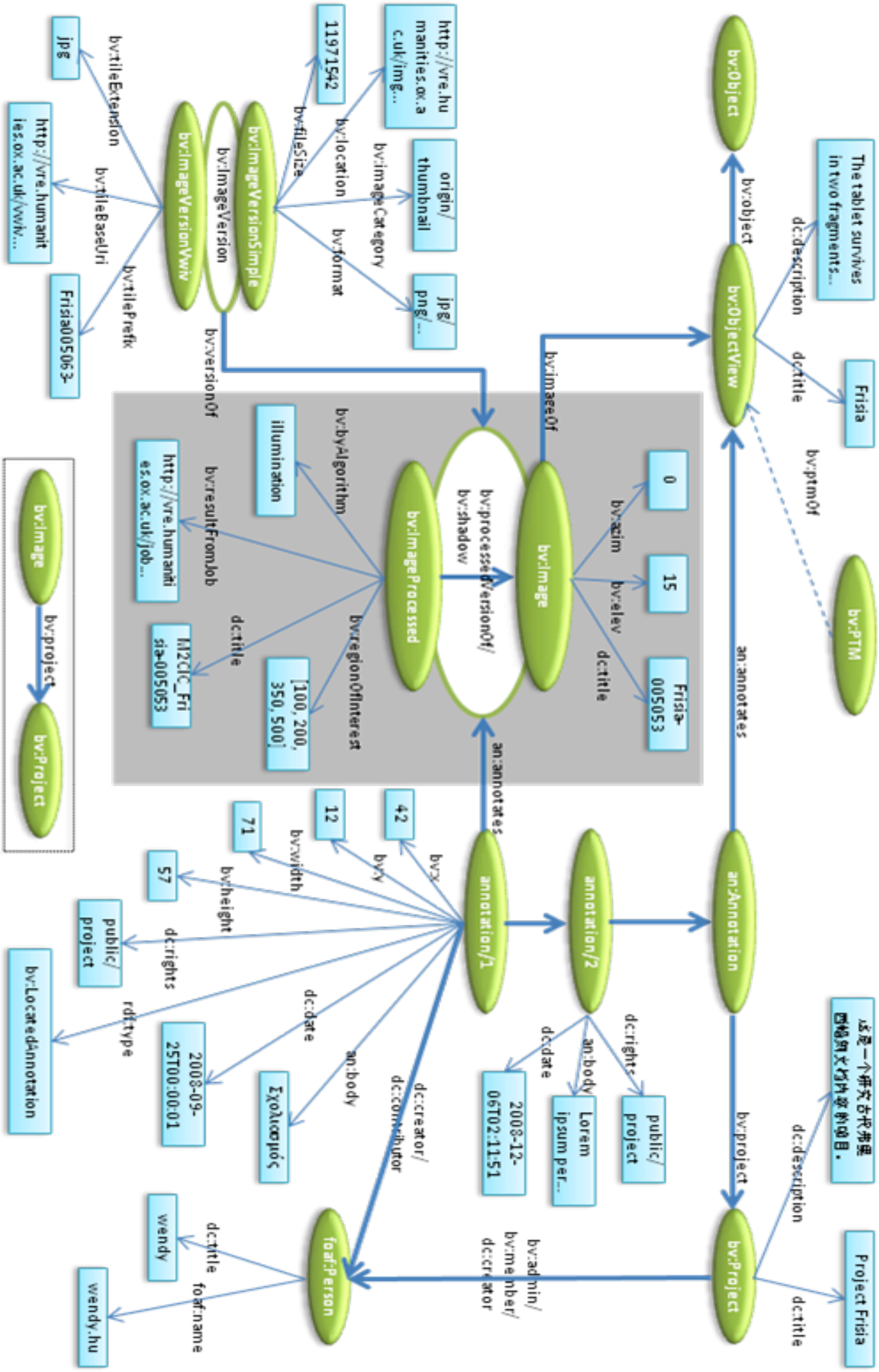

Figure 1. Engage data model

The Engage data model uses three properties: bv:algorithm, bv:regionOfInterest and bv:resultFromJob to describe what algorithm, region of interest and analysis job were used to generate a processed image from an original image or in some cases a processed image, respectively. This lets users to get analysis information of processed images and also makes processed images traceable and classifiable.

The Engage data model uses two properties: bv:processedVersionOf and bv:shadow to describe the relationship between an original image and a process image and also distinguish shadow images with other types of processed images.

In the Engage data model, other properties used to describe original and processed images are same. This is because both types of images are image. Both of them can be annotated and allocated to one or more than one specific projects. They are associated to specific object views.

Architecture and Infrastructure

The portal part of the Engage system is implemented base on a set of technologies, as presented in Figure 2. It consists of:

- a uPortal to hold all developed JSR-168 portlets and manage user accounts and access rights;
- a generic Servlet and associated classes to deliver services at the server-side for client-side JavaScripts to, for example get the information of a list of image by keyword from the Engage TripleStore and submit analysis requests to the NGS;
- a JavaScript library to deliver dynamic functions required at the client-side, such as image display, mapping of image coordinates and getting user-defined analysis parameters.
- a Direct Web Remoting (DWR) based COMET implementation to support "server data push to browsers" in order to realise such as display of real-time job status to users and instant messaging;
- a TripleStore for the storage and retrieval of image metadata in Resource Description Framework (RDF) formats;
- a Linux file system to hold all images and a Tomcat Web container to deliver the remote access to the uPortal and those stored images.

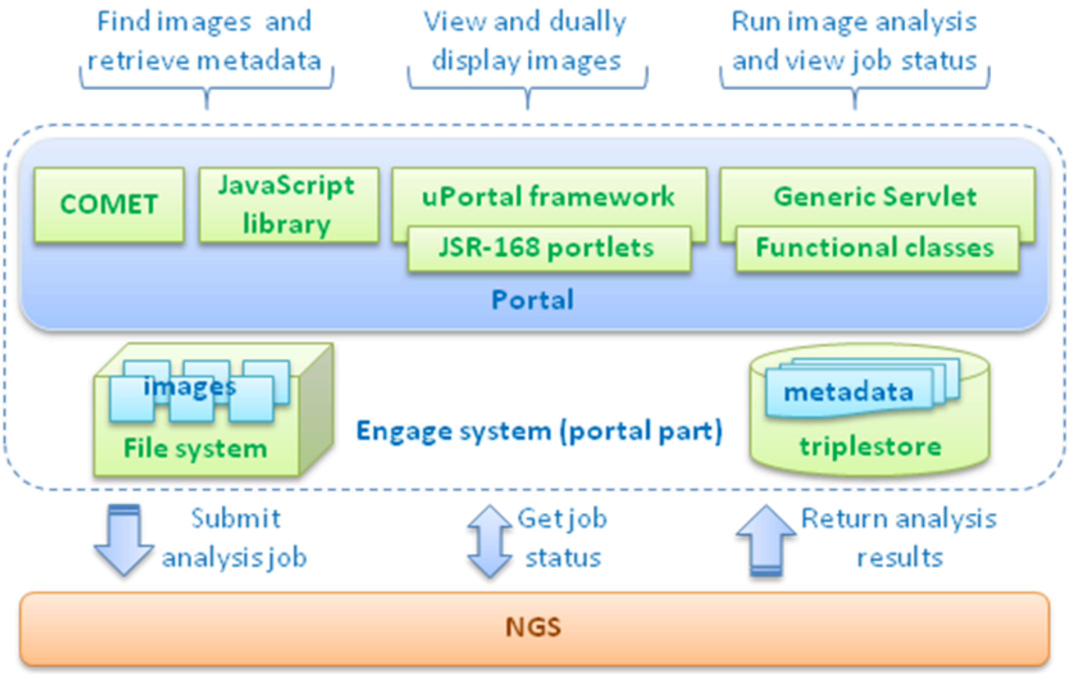

Figure 2. Overview of the portal part of the Engage system

3) Integration with NGS via GridSAM

GridSAM is an open source job submission and monitoring web service. It uses JSDL specification from OGF and supports OGSA-BES, which makes it an ideal standard-based mechanism for submitting and monitoring jobs in this project. The standard web service interface and extensive client side API it provides also helps to reduce the development effort typically required by using JavaCoG Kit.

Another benefit we found during this project when using GridSAM is that, it helps to keep a clean boundary between normal web application development and Grid development. The latter is usually considered as complicated and error-prone. By using GridSAM client side API together with the simple job manager wrapper we developed during the project, portal server can easily submit jobs and query job status without knowing any details of grid resources in the back end. At the end of the process chain on the portal, each job is represented as a JDSL description file that contains following information:

- Executable and argument
- Standard output and error files
- Data Staging in/out information
- Access information for proxy certificate

GridSAM supports various protocols for data staging, such as FTP, GridFTP and SFTP, etc. We decided to use SFTP due to security and simplicity considerations. However, at the moment when this project was carried out, the latest version of GridSAM (v2.1.14) supports only user name and password authentication when using SFTP for data staging, which means we will have to embed access information for file servers into JDSL description. The security risks can be mitigated using SSL. When setting up GridSAM service in Oxford, we have identified this potential security risk and used SSL to encrypt all the plain SOAP messages between portal and GridSAM server.

A job is submitted by calling submitJob() method on the Job Manager Wrapper object, which return a unique job ID once the submission has been completed. The job ID acts as a unique handler for each submitted job. The portal can query the current status of each submitted job by providing this job ID handler.

Once a job has been received by the GridSAM service it identifies the data staging information inside JSDL and starts to download source image files onto NGS. GridSAM will then invoke the executable on behalf of the portal when computing resources are available on NGS. During the period of job execution, the portal server will periodically query GridSAM for job status update. If a job has been successfully completed, the output images will be staged out via SFTP based on access information within JSDL. If, on the other hand, errors have occurred before a job is finished, a brief error message will be returned back to the portal.

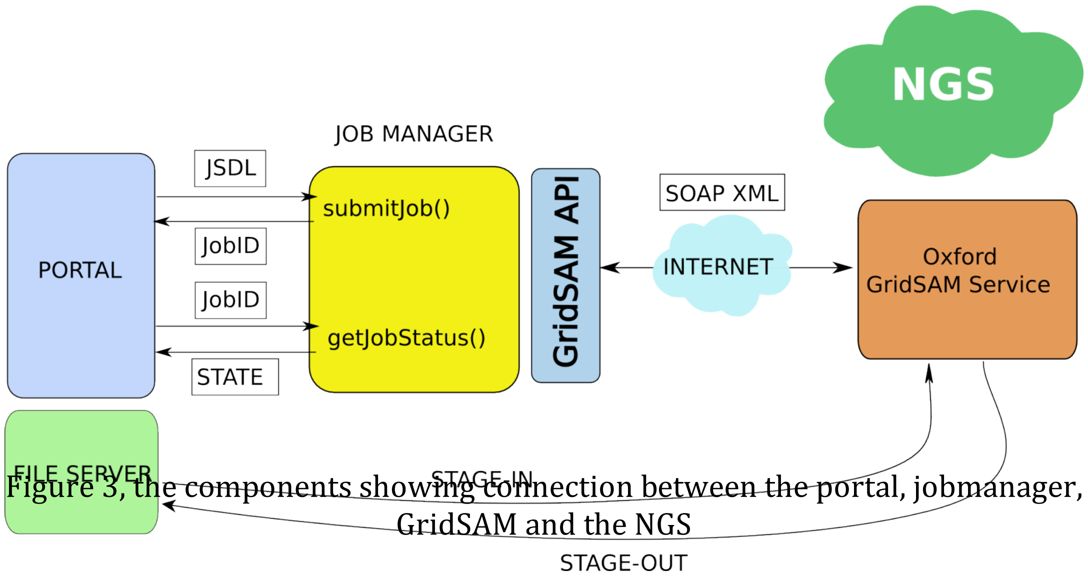

Figure 3, the components showing connection between the portal, jobmanager, GridSAM and the NGS

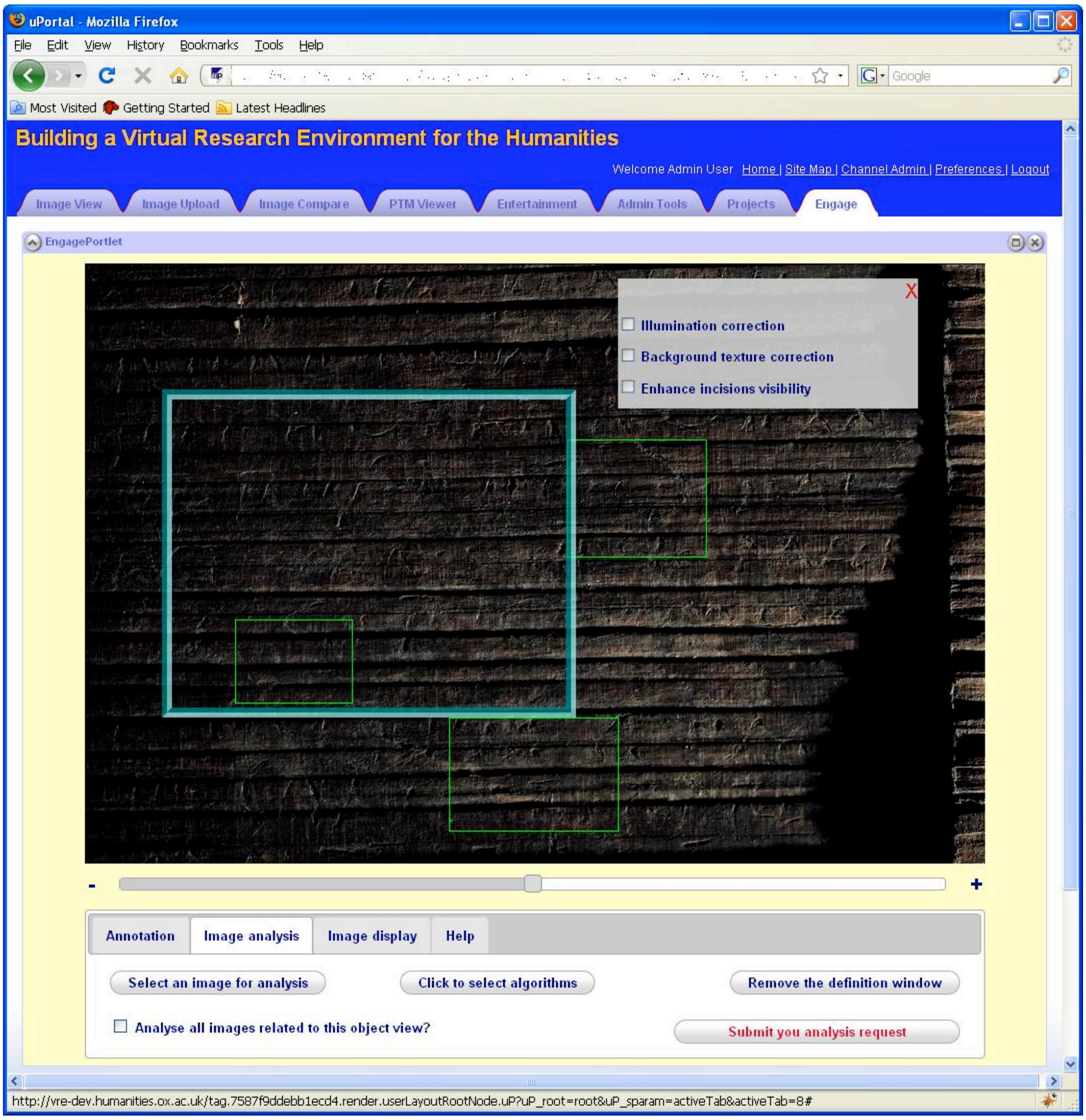
uPortal - Mozilla Firefox
File Edit View History Bookmarks Tools Help
Google
Most Visited Getting Started Latest Headlines
Building a Virtual Research Environment for the Humanities
Welcome Admin User Home | Site Map | Channel Admin | Preferences | Logout
Image View
Image Upload
Image Compare
PTM Viewer
Entertainment
Admin Tools
Projects
Engage
EngagePortlet
X
□ Illumination correction
□ Background texture correction
□ Enhance incisions visibility
-
+
Annotation
Image analysis
Image display
Help
Select an image for analysis
Click to select algorithms
Remove the definition window
□ Analyse all images related to this object view?
Submit you analysis request
http://vre-dev.humanities.ox.ac.uk/tag.7587f9ddebb1ecd4.render.userLayoutRootNode.uP?uP_root=root&uP_sparam=activeTab&activeTab=8#

Figure 4 the user interface portal for the project showing enhancements and selection of regions within an image.

What was the approach that was taken?

Overall splitting the development work between two separate developers, allowing parallel development allowed the project to work at a significantly increased pace. Through the development of simple testing and evaluation mechanisms we ensured that each of the developers were in no way dependant during the early stages on the work of the other. The only bottleneck has been with the interaction with the NGS where we have to wait for them to install software etc.

What design choices were made and why?

The key design choices were made such that extra development was minimized above and beyond the two different projects that we are integrating. This was important both because researchers have worked closely with the VRE-SDM to develop the user interface etc. The packaging of the application of the execution hosts has been predefined by the NGS with their uniform execution environment. Working with the researcher developing the algorithms allowed rapid turn around from the development of a new algorithm to having this installed as a compiled binary that is accessible using the WS interface available through GridSAM.

What was the model of interaction between developers and researchers during the project?

Bi-weekly team meetings were held as well as impromptu meetings between the researcher and developers. This allowed issues pertaining to the understanding of the algorithms, arguments needed for correct operation and of the NGS, the workflow of the image processing tasks, the design of the data model and the GUI design of the portlet.

Evaluation (2-4 pages)

The important point in this section is to get both the researcher and the developers experiences of this project: good and bad!

As always in an interdisciplinary project communication between the members of the team is important. By having people working physically close together we were able to ensure that speed of communication was always guaranteed even if there were minor issues around clarity. Demonstrations of a test interface by the researcher ensured though that these were quickly and easily cleared up with the development team.

What issues were experienced by the developers during the course of the project, and how were they resolved?

The two main hurdles that had the be overcome had to do with:

- MATLAB and licensing… The MATLAB Runtime Environment had to be installed on the NGS, and a strategy had to be developed, jointly with the NGS support team to enable the compilation of the MATLAB code in a way that is compatible with the NGS's MATLAB Runtime Environment.
- GridSAM. Some somewhat expected/unsurprising libraries conflicts occurred between GridSAM and the portlet. OMII tickets were raised, and efficient support from OMII-UK was provided.

What issues were experienced by the researchers during the course of the project, and how were they resolved?

None, in fact the researcher did not have to learn anything about what happens on the NGS side, or on how the portlet technically connects to the NGS. This allowed them to concentrate on the underlying algorithm development without stressing about factors outside their remit.

Did the model of interaction between developers and researchers work, and why?

By having the developers and researcher physically based in the same department we were able to maximize the interaction between the two groups. This meant that when there were problems or issues then they were quickly and easily worked through. This was especially important since a new developer had to come and work on the project with less than two months to goand needed to be up to speed as quickly as possible. The project met once per week whilst using the Basecamp system to keep track of the ongoing work.

One problem that we have had is with the installation of the end user code onto the NGS. This process is still extremely difficult and whilst we have two GridSAM gateways setup for the Oxford and Rutherford Lab nodes we have been unable to get software quickly onto either. This is especially important as the underlying algorithm is being actively worked on and the version that will be part of the final project output is still only a 'test' version with significant extra functionality still being built.

We have found that this interaction model is the best for working with researchers and getting them up to speed with utilizing e-infrastructure as quickly and easily as possible. We have used it both before this project and are intending to use it with new projects.

We think that we have achieved all of the original objectives for the project though with the development of new and improved algorithms the researcher will be using the developed framework to continue to make cutting edge methods available to the research community that need to use them.

Outcomes (0.5-1 pages)

The project outcomes are two fold as per the design. Firstly we have created a build and installation configuration toolkit that can be used by researchers from the eSAD project to distribute the underlying tools that they create onto the resources of the NGS. This includes all of the build scripts to create the binary products that can then be redistributed whilst protecting their IP.

We have proved that utilizing the NGS UEE scripts to produce a simple common location for application installations is useful and enables researchers to work with code installed at a number of sites

The researcher is now provided with the necessary tools and knowledge to add new functionality to the portlet, without having to worry about how the connection to the NGS happens. The implication of this for the community are that users in the community have now access to an easy to use tool, which will facilitate the dissemination of the outcomes of the eSAD project. It proves the usability of the VRE.

We have shown that it is possible to not overwhelm researchers with the technical details of how the NGS works and communicates with a web-based application! All the researcher needs to know, is that it is possible to use the NGS, and that it doesn't mean that they have to worry excessively about the details of the communication protocols between the NGS and a web-based application.

Dissemination and Exploitation (0.5-1 pages)

The work of this project has been presented a number of times. Firstly at workshops that have been organized by the eSAD and VRE-SDM projects as well as the OMII developers conference, Edinburgh. It is planned that we will submit this work as a paper into the UK e-Science AHM 2009.

To exploit this project further it will be necessary to generalize further the tools that we have created. The methodology for distribution of MATLAB based applications can be reused across the NGS with all of the associated installation framework etc.

For image enhancement and processing then it is only the case of development of a number of new algorithms and pluggin these into the MATLAB code. It is planned that the eSAD project will continue to develop these new algorithms etc over the lifetime of the project.

We would aim to put this piece of software forward to whoever is successful with the EPSRC Software Sustainability call so that it may be hardened into a standards compliant portlet able to do image processing as a product. Others could easily build on this work through the building of their own remote applications or hosting environment for the user interface.

- OMII collaborations workshop poster (c.f. OeRC website)
- eSAD Workshop 13th May (c.f. eSAD website)
- Arts and Humanities E-science projects meeting (8th June, UCL, c.f., AHeSSC website)
- IEEE/AHM paper

What is being done, and needs to be done, to exploit the outcomes of the project?

- Sustainability. Keep contact with VRE to adapt to their changes and synchronize.
- Plan to keep adding functionality as research outcomes crop up from the eSAD project.

Recommendations

It's a good idea to have a person playing the role of interface between the end-users and the developers, and to encourage face-to-face meetings (enables hand-holding and reassuring as to the use—ease of use—of the NGS). Never underestimate the fear-factor on the end-users' side when it comes to use advanced computing technology.

It is essential that there are slightly longer lead times for these projects. We have been expected to get about a years work done in considerably less time than that. We would also like to have been more ambitious with some of the aspects but are limited by both time scales and the availability of the skilled developers on this and other projects.

It is important that the outputs of these projects are picked up by OMII and supported we cannot have software being developed by projects even those funded by OMII if there is then no support for the continued utilization of the materials afterwards.

References

- eSAD website: http://esad.classics.ox.ac.uk/
- AHeSSC website: http://www.ahessc.ac.uk/image-text-interpretation
- OeRC website: http://www.oerc.ox.ac.uk/about/library/posters