

Modern Portal Tools And Solutions with Vine Toolkit for Science Gateways

Piotr Dziubecki¹, Piotr Grabowski¹, Michał Krysiński¹, Tomasz Kuczyński¹, Krzysztof Kurowski¹ and Dawid Szejnfeld¹

¹Poznań Supercomputing and Networking Center, Poznań, Poland,

[deepres, piotrg, mich, docentt, krzysztof.kurowski, dejw]@man.poznan.pl

ABSTRACT

Motivation: Science Gateways are web portal environments targeted for given community and dedicated to the specific scientific domain. End users require different set of tools, applications, data integration patterns to be able to satisfy special demands of different kind of scientific research approach. To let users benefit from computational resources and storage, a portal framework should support access to the grid and storage services. To accomplish this, PSNC [1] developed Vine Toolkit framework (in short Vine) integrated with Adobe Flex/BlazeDs technologies, offering unified abstract API for different grid middleware infrastructures. Additionally it automates integration process with portal frameworks like Liferay [2] or Grid-Sphere [3], allowing to set up a portal solution in relatively short time. The Vine project also introduces so called subprojects, which can extend core API or define new business components at low level or web applications at high level. This way Science Gateway prototyping process is definitely shortened. The architecture of the solution allows programmers to prepare building blocks which can be reused in a simple manner and applied for different gateways. This article presents different tools and solutions prepared by means of Vine framework as an illustration of success stories.

1 INTRODUCTION

In this paper we try to show different aspects of scientific applications and external grid services support within computational web-based Science Gateways. The requirements of different communities vary according to the given scientific domain, complexity of the existing problem and proficiency of the existing tools. The potential solution covers such aspects like integration of the scientific applications, analysis tools and implementation of missing components. The main basement of our work is Vine Toolkit (in short Vine) [4] [35], which facilitates building of portal environments using a modular structure and existing Vine components or prepare a new ones using integrated technology like Adobe Flex [5] and Adobe BlazeDs [6]. Vine lets developers integrate HPC resources in relatively easy way using high-level abstract API. Consequently, an easy-to-use presentation layer can be deployed together with various collaborative and visualization tools to simplify the way researchers perform intensive computing studies and share data between simulations via lightweight web interfaces. Vine APIs encapsulates capabilities for remote job submission, monitoring and control as well as data and workflow management, security and user management. Vine supports OGF standards like JSDL [7] or BES [8], so it is easier to take advantage of such middlewares like gLite [9], Unicore [10], Globus [11] or CosQosGrid [12]. Implementation work can be easily organized into reusable

subprojects, which may be reused in different projects. This paper consist of set of paragraphs devoted to different tools and solutions as examples of potential usage of Vine framework.

2 NANO-SCIENCE GATEWAY

One of the interesting example of Science Gateway based on Vine is Nano-Science Gateway prepared during PL-Grid project [13]. The solution was described in detail in one of the papers for previous IWSG conference [14]. As a short reminder - it was a joint research and development effort with researchers interested in collaborative, Web2.0 and large-scale simulation studies based on Density Functional Theory (DFT) and Many-Body Perturbation Theory. We selected a key software package called ABINIT [15] and created from scratch many new and advanced web-based applications around it. The scientists started to use portal web application instead of the command line client, which brought them many facilitations related to job and data management. The main web application is NanoClient, which enables support for nano-technology related scientific application in grid infrastructure. The solution is still being developed and maintained. End users still have more requirements. In consequence, a component called the Structure Builder was created. Its main aim is to combine functionality available in the existing commercial and open source software such as: GoVasp [16] and XCrysDen [17]. One of the key components of GoVasp is a structure builder which is an editor for building and manipulating the nano structures –the solution is commercial and it could be unavailable for small groups of scientists with small budget behind. XCrysDen is an open source crystalline and molecular structure visualization program. Therefore we tried to combine the two mentioned software packages into one open source web based tool with the use of Adobe Flash and BlazeDs technology. Of course we covered only some part of the functionality of both packages in the first version of the editor. This way it enables an ab initio creation of crystal structures, their visualization and modifications. This way nano scientists are able to perform virtually all steps related to the research process in our Science Gateway environment by designing crystal structures being subject to large computations.

It consists of visual editor (further called NanoEditor) and interface that allows users to enter input data (further called NanoBuilder). The first one is a graphical tool that enables creating and editing super-cells and generating output files that can be further used in the Abinit package. The latter generates geometry of crystallographic coordinates of at least one of the 230 symbols of symmetry group in the Hermann-Mauguin notation and lattice vectors and angles. Generated coordinates of all atoms in the cell in accordance

with established symmetry can then be used to render the cell, which may be subjected to a process of further processing in the NanoEditor. By processing we mean cell replication operations, changing positions of certain atoms or groups of atoms within the cell, the addition of atoms and defining the vacuum.

NanoBuilder was created using Adobe Flex technology. Vine introduces interesting concept of abstract data model for applications. The data model which is a serializable java bean represents all data entities exchanged between server and client components with state persistence capabilities. The developer does not have to take care of data exchange process (internally supported by Adobe BlazeDs technology) between server and client, which simplifies the overall development process.

NanoBuilder is a client of a Spacegroup application - an open source utility that produces crystal geometry for use with the Elk code[18]. This external software is managed locally by the Vine itself and is made available through standard Job Management API. The developer just prepares JSDL description to execute Spacegroup and the rest is managed by Vine components like application execution, working directory creation, input and output data management and making them available for end user web interface. The Spacegroup tool can be available locally for the portal instance but it is also possible to run it in the grid infrastructure without big changes in the code. NanoBuilder's main task is to generate files with a data model that can be loaded into NanoEditor. User friendly interface enables a step-by-step approach to creating crystal geometry by defining basic information such as: space group by its Hermann-Mauguin symbol, lattice vectors lengths, lattice angles, number of unit cells in each direction and telling whether unit cell should be found. User is also able to add atoms in a very simple way, by choosing them from the displayed period table. After selecting atoms user can define their positions in Angstrom units. All stated information is, invisibly for the user, passed to the Spacegroup software. Retrieved results are saved to the unique file. Layout of the tool is presented in the figure 1.

The screenshot shows the NanoBuilder interface with three tabs: Symmetry, Cell, and Atoms. The 'Cell' tab is active, displaying input fields for lattice vector lengths (a, b, c) and angles (ab, ac, bc) in Bohr atomic units. There are also fields for the number of unit cells in each direction (1, 1, 1) and a checkbox to set the primitive unit cell to 'true'.

Fig. 1. NanoBuilder's layout

The second tool - NanoEditor - was also created in Adobe FLEX technology. To present the crystal structure in a third dimension, Papervision3D library [19] has been used. This tool enables in easy and affordable way creation, manipulation and edition of super-cells.

Primarily user can load data generated in NanoBuilder by choosing a file from the File Window. After choosing a proper file, model is

loaded into the NanoEditor and its graphical representation is displayed (figure 2). It is possible to load models unlimited number of times.

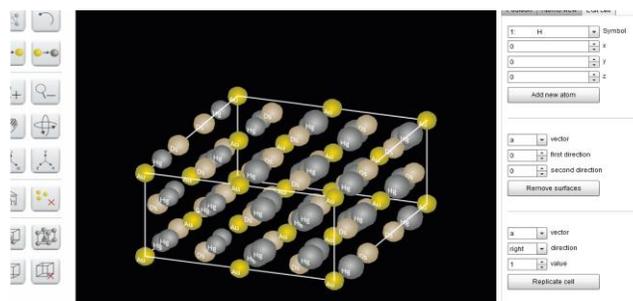


Fig. 2. Graphical representation of the crystal cell in NanoEditor

User can freely manipulate displayed crystal cell with the mouse. For a convenient use it is possible to set two modes: rotation and shift mode. In the first one user is able to rotate the model in every direction so it can be seen from different angles. In the latter user can shift whole model in every direction to set the proper view. Moreover, the tool enables selecting and deselecting all atoms or just the single one. This functionality is necessary to define atoms, which should be processed while model edition.

NanoEditor contains a few functions, that facilitate cell edition. To recognize cell directions, user can display cell axis (basis vectors). After selecting such option, three lattice vectors will be shown, with beginnings situated in point (0,0,0). For operations that require Cartesian axis system it is possible to display Cartesian axis. To easily identify crystal system, border of the crystal cell can be shown. Example of displayed cell border is demonstrated in figure 2.

User is able to switch between two display modes of the unit cell. By design, crystal cell is displayed as a primitive cell (translational asymmetric unit). But additional option is delivered – displaying a “nicely-cut unit cell” (shortly unit cell). Difference between both cells is demonstrated in figure 3. The unit cell displays additional atoms in places where multiplied cells are connected together after cell multiplication process.

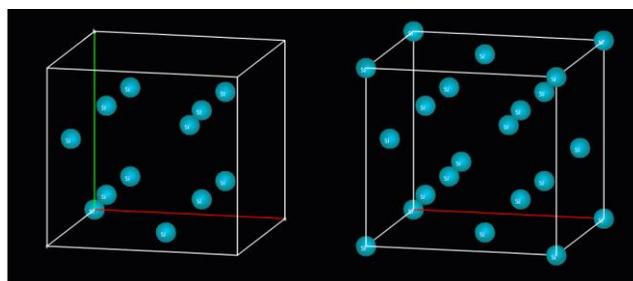


Fig. 3. Difference between primitive cell (at the left side) and the unit cell (at the right side).

A crucial functionality of the NanoEditor is the edition of the crystal cell. The tool enables the removal of any atom, and an exten-

sion of the cell content by adding new ones at the specified positions. It is also possible to modify coordinates of selected atoms by shifting them by values provided in Angstroms along every axis. Important feature is the ability to replicate the crystal cell specified number of times along selected lattice vector. It can be done in both display modes – primitive and unit.

When all desired alterations are effectuated, user is able to define vacuum of the crystal cell. It is done by specifying how many surfaces of the cell, in which direction of the cell coordinate system, need to be removed.

Because most of the users are prone to make some mistakes while providing changes to the edited cell, all described above actions, but defining vacuum, can be undone. It allows users to make alterations without worrying about irreparable consequences.

Recent features regard general information about edited crystal cell. Primarily user can display coordinates of every selected atom. There is also an opportunity to display distance between two selected atoms. But most important is displaying model parameters in the Abinit file format. It contains data that describe number of atoms, lattice vectors, cell angles, atoms types and many more. User can use it in another nano product – Nano-Science Gateway – just by copying displayed content. The returned model can be further processed and used.

The prepared editor became a really useful application for nano scientists who are able to design a nano structure without need of having access to very expansive specialist software package like GoVasp. The source code is distributed as open source so it could be further developed in the future. This is also great example of how we can use Vine mechanisms to prepare advanced graphical tools for end users.

3 NAMD WEB CLIENT

NamdClient belongs to the second generation of web components based on the Vine Toolkit framework. One of our assumptions and requirements for the Vine Toolkit itself was to give developer a complete environment facilitating the design and creation of new tools. First applications created with a use of Vine like NanoClient have been designed to solve specific problems in the area of nanotechnology (like finding the total energy, charge density and electronic structure of systems made of electrons and nuclei within Density Functional Theory). On the other hand we wanted to extend our base of reusable concepts – building blocks which could be used later to create different applications as well. Having that in mind, we based our software on a set of components, which could be reused in many different contexts.

For the NamdClient we gathered user requirements and created a general view of the new application stack. The whole system is meant to solve molecular dynamics (MD) simulation problems. Scientists would like to have access to the portal where they can define problem by creating/editing polypeptide model with the use of graphical tool or importing data from local repository as well as from the remote protein data banks. Some input files are created as a result of that process. Next step is to prepare and configure the geometry optimization (optional) and finally end up with the MD / Steered MD computations. On each stage there is a need for efficient data organization – creating a problems library with a set of features facilitating database browsing and finding the desired information.

Fortunately there is Application Registry service and UI component used also in the NanoClient which will be our base system responsible for the the data management. The following diagram depicts the NamdClient specific application/case mapping in the Application Registry structure.

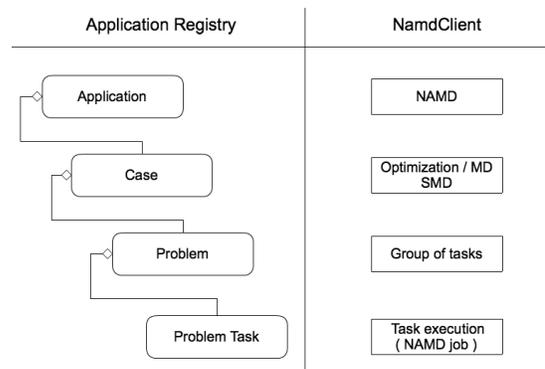


Fig. 4. Application registry mapping diagram

Utilization of the mentioned service lets user work with data in a convenient way especially when dealing with a massive amount of data objects.

The next step is to edit/configure a new case. In order to do so we provide a form based components with a graphical representation of the config file. “Advanced” mode allowing inline modification of the text input files is also available.

The final stage is the job submission which is the main added value. It could be a single job or a set of jobs, depending on how many variants user would like to examine. It is worth to mention that parameterized tasks are grouped under a “Problem” category and thus it is easier to analyze and compare the results.

Job submission and monitoring are also in a form of reusable code, so they are used without any modifications in order to schedule and observe the task execution process.

The last element of the proposed solution is a simple tool allowing for the analysis of computed results. We are working on the log scrapping mechanisms, which let us visualize the partial result in the realtime. That feature will provide user with an instant feedback and he will be able, for instance, to terminate tasks not matching experiment's requirements.

Currently we are focusing on the delivering of the core functionality – integration Application Registry, task definition interface with a Job Submission and Monitoring. Later on we will complete the whole system with an editor of polypeptide model and more visualization tools.

Reusing components gives us one more benefit, we are capable of creating a new application, which will be able to incorporate the functionality from the NanoClient and NamdClient projects. Therefore users could use both components within the context of the same software instance. For the end user it will be just a matter of choosing different task type from the application registry.

4 IRODS WEB CLIENT

Other example of advanced web application we want to describe is a iRODS [20] server web client prepared during on-going project

HPC-Europa 2 [21] we are involved in. The main concern of one of the research activities here is data management solution for scientific data with meta data support and advanced search functionality. The common problem with large scale simulations is the problem of output data after the simulations ends. If data is produced at a large scale and volume grows very fast, the only solution is backing them up in some large repository (like big central local data storage, P2P network, grid/cloud based storage) and post-process them later. The data have to be stored in a proper way and should be well described at the same time. That is why iRODS server was chosen as it supports meta data storage describing the physical data and provides advanced mechanisms to search through the data and allows for preparing so called microservices, which can do some post-process actions on the server side. Within the project we abstracted data with so called DataObjects, which organize data related to some scientific simulation within well structured directories, together with meta data attributes based on the XML schema prepared.

Searchable meta data attributes describing each data object are divided into several sections. These sections contain different kind of data object information. The general structure of the schema is presented in figure 4. Attribute called *UUID* contains unique identifier of each data object. In the *Info* section each data objects has a basic description of an experiment. Details about each science experiment as well as authors of an experiment, software that had been used can be found in *Source* section. All scientific parameters specific for experiment can be defined in *Science* section. *Data* section refers to the information about results of computations.

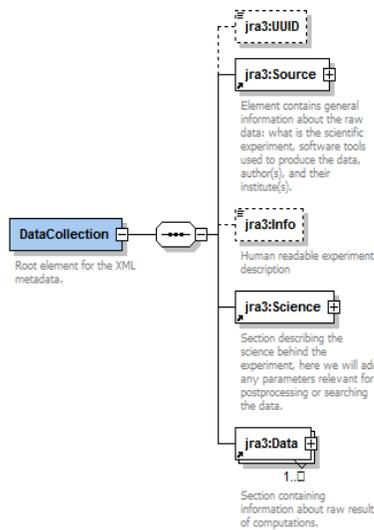


Fig. 5. Metadata xml schema root

One of the main scientific applications used within the system is ENZO 1.5 [22]. It generates a sample of twenty galaxy clusters with a large masses, simulated with high spatial resolution. Software produces large amount of data, which can be divided into outputs of the simulations like gas density, temperature and velocity, dark matter, and processed results like images or statistics. In both cases complexity, resolution and scale of data, which might in a case of one cluster reach 600GB force to use specialized software. So it is quite demanding for the data storage system. Meta

data are very important here as they enable to identify only interesting sets of data, which may be later post-processed.

We prepared a web portal to make it easy to search the data through and allow users to view some post-processed data such as visualizations exposed by the server storage system. Jargon library [23] which is used in our software has functions for managing data objects existing on iRODS servers. It also allows for querying metadata attributes. Functionality of Jargon was extended by our client in the area of building queries. The disjunction operator was added between the conditions of a query to have more flexible solution for searching methods.

Vine allows to add custom authentication modules to the portal. Such new module was developed which allows end users to authenticate against iRODS system. Vine checks if user account exist on iRODS server and creates an account on the web portal if authentication process succeeded. From that moment user has access to iRODS resources through the web application. Such approach was one of the requirements of the project, which says the user should have an iRODS account which automatically gives access to the portal system.

The web client allows for easy browsing of iRODS resources, checking available metadata information and doing searches in iRODS databases. The main view of the web client application shows the set of data existing on the iRODS resource configured within the domain selected by the user during the login process. The data object can be selected here and afterwards user can browse the detailed information connected with the object on the right side of the page. Main view allows for filtering of the displayed data sets based on common attributes. User can use wild cards like '*', '%' (representing chain of any characters) or '?', '_' (representing some one character) while setting filters. Basic filtering is based on attributes like the name of the experiment, authors name, role of the author, name of the authors institute and identifier of the experiment. Results of filtering can be grouped into pages with 10, 20 or 50 data objects. After selecting proper data object user can view more information about it by clicking on one of the tabs on the right side of the data sets list. First tab, called Info contains general information about selected data object. This information is gathered from the metadata attributes. *Details* tab (figure 5) contains information about the structure of the data object as well as the basic information about the experiment, authors and tools used within the experiment. The user can browse through the directory structure and download files by choosing proper option from a context menu of a selected file. For example the user can download XML file called *metadata.xml* to view all attributes of the data object and compare it with a different object.

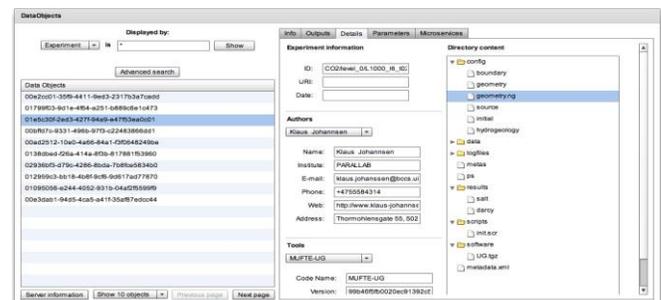


Fig. 6. iRODS web client – Details tab

The web application provides a very user friendly interface for searching through scientific data. Complex queries of the underlying database can be specified in an easy, intuitive way.

To do more sophisticated searches, the web application has an appropriate form (figure 6) available after clicking the “Advance Search” button in the main view. Using the advanced search form user creates queries based on all existing metadata attributes describing the data object. User is able to prepare really sophisticated search formula and thus find expected data.

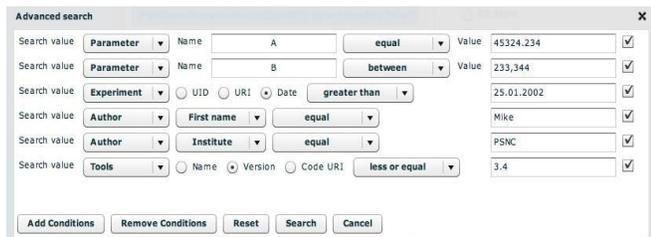


Fig. 7. Advanced search form

One of the features of the web client is that it can be connected to the service that produces images during the experiment's post-processing phase. All visualized data can be shown by the application. This kind of data is grouped into the HDF5 [24] format file list and user has got ability to browse through its internal files and variables that have been used during the experiment. Figure 7 shows the example of visualized data which are placed on one of the tabs with given data object details.

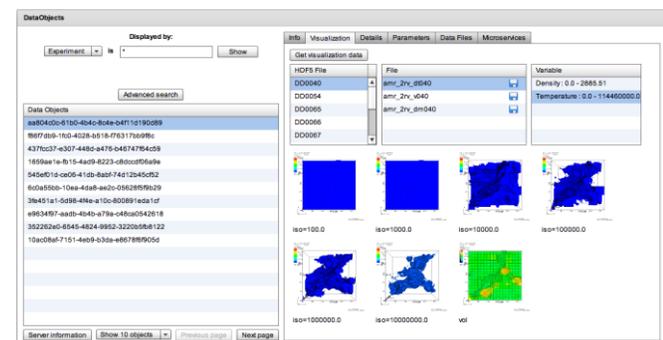


Fig. 8. Visualization tab

The web client take advantage of a dedicated web service, which retrieves internal data from select data object and generates visualization files exposing them on so called cache servers. Whole process is fully dynamic and generates graphics on demand if they do not exist. The web service responses with XML document with hyperlinks to the graphic files and parameters related to these files, so the web client just parses the document and retrieves the images.

The final outcome of this research activity is expected to be the deployment of an effective service to search, select and access huge and complex data sets, like those produced by ultimate numerical simulations. Such kind of support appears to be necessary in order to let the user exploit archived data, focusing promptly only on that data which is actually of interest and be able to analyze their graphical representation.

5 SCIENTIFIC SIMULATIONS WITH ADVANCE RESERVATIONS

Computational simulations in the grid environment typically depend on local resources usage patters and thus their availability at the given moment. This aspect is problematic for complex simulations, which require access to many resources like clusters at some strict range of time. Many simulations consist of parallel computations with high degree of dependency. In order to be able to address such complex and demanding simulations grid system should support advance reservations, which enable resources allocation. A middleware which supports such functionality is QosCosGrid (in short QCG) [25]. Advance reservations are supported in the resource broker called GRMS [26] and SMOA Computing [27] service. GRMS is an open source meta-scheduling system. It dynamically discovers resources and applies advanced scheduling methodologies, deals with resource management challenges, e.g. load-balancing among clusters and various work-load systems, remote job control or file staging support. Therefore, the main goal of GRMS is to manage the whole process of remote job submission and control. SMOA Computing acts as a computing provider exposing on demand access to computing resources and jobs over the HPC Basic Profile [28] compliant Web Services interface. SMOA Computing offers remote interface for advance reservations management, which is used by GRMS broker to allocate resources for the given simulation. This issue is very important in the scope of workflow jobs supported by the GRMS. Advance reservations allow prepare resources for efficient job execution especially these, which take advantage from different computational resources like clusters at the same time.

Any potential client should be able to reserve resources for the given computational job. In order to make it happen for end client we proposed a JSDL extension as a XML document, which can be used to specify advance reservation value. The extension is later processed by the GRMS broker in order to use SMOA Computing API [29] to reserve resources. The role of the GRMS broker is important here as it selects potential resources automatically and automates many actions related to jobs allocation, monitoring, reservation and even data processing. The illustration below shows the schema of the defined JSDL extension:

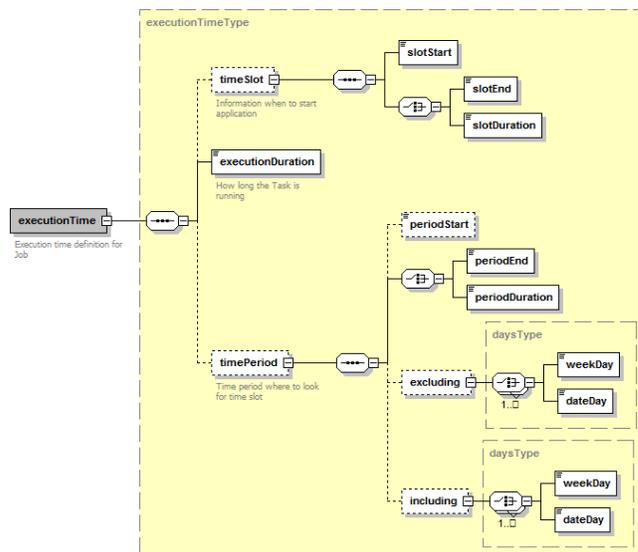


Fig. 9. JSDL extension for defining advance reservations

The example below illustrates the usage of the extension:

```
<ns2:executionTime
xmlns:ns2="http://qcg.psn.c.pl/jsdlex/2011/01/qcg-allocation">
<ns2:executionDuration>PT30M</ns2:executionDuration>
  <ns2:timePeriod>
    <ns2:periodStart>2011-03-
20T10:05:00.000+01:00</ns2:periodStart>
    <ns2:periodDuration>PT60M</ns2:periodDuration>
  </ns2:timePeriod>
</ns2:executionTime>
```

It means that the requested job will took 60 minutes in the period of time starting from 20th March 2011 from 10:05 am CET till 11:05 am. Default xml types like DateTime Data Type and Duration Data Type [30] are used here to define date, time and duration constraints.

Execution duration is a required value while timeSlot and timePeriod both are optional. Time period lets you specify the date range within the job should fit in with its executionDuration. Time slot is optional element to find small time range expressed in hours to fit in the job. For the most cases time period and execution duration elements are enough to request for resources in the given time frame. The extension is added to the Resources JSDL element.

We used castor data binding framework to create java beans for building up the allocation XML extension and to enable proper parsing and validation of job description documents going through the Vine Toolkit grid API layer. As a result, any type of client like web application or web service can take advantage of the advance reservation functionality by exploiting easy to use Vine API.

The feature was tested within the Job Manager web application which is a built-in job management web application in a Vine release. Users can add here proper extension to the XML JSDL job description in order to reserve resources. The extension was used also within Euforia project [31] components to address Kepler [32] workflow jobs to reserved resources in the grid by means of Kepler actors. Kepler actors are simple clients of the Vine web service component, which plays role of the grid infrastructure abstraction layer. By using advance reservation, Kepler workflow engine is able to allocate resources during workflow execution and run simulations more efficiently in distributed infrastructure.

The Vine Toolkit in conjunction with QCG middleware introduces interesting portal and grid service layer with advanced functionality related to grid resources management and robust performance. The built-in support for GRMS resource broker lets developers and users take advantage of advance reservation functionality without need of investigating low level middleware or batch system API.

6 STEP TOWARDS MODERN WEB SOLUTIONS

Nowadays we observe rapid development of the modern web applications (social networks, content aggregators, etc). These services have changed the way we use the web today. At present most of the web users (including scientists) utilize some kind of "online desktop" in their day to day work. Some users habituate to use

personalized web pages (e.g. iGoogle [33]), others prefer to use social networks (e.g. Facebook [34]).

The Vine Toolkit follows this modern trend by enabling all its web applications to be integrated into third part web infrastructures almost seamlessly. On account of the specific architecture of the web part of the Vine Toolkit, it is possible to embed Vine applications in almost all web based environments by using just a few lines of pure HTML code. This capability allowed us to setup and test Vine applications in a range of external web systems (starting from pure HTML pages, through Wiki, various CMS solutions, other instance of the Liferay, ending on Facebook and iGoogle).

Although it is possible to aggregate Vine applications into almost any web based system just by means of HTML, we have taken steps towards tighter integration with selected web platforms.

We have implemented *RemoteVineGeneratorApp*. This web based tool allows developers to generate ready to use integration packages for various platforms without forcing Vine administrator to play with internals of the Vine Toolkit. Administrator just checks which Vine applications need to be exported and chooses type of the export. Then, integration package is generated and saved in the Vine's Portal File System - it is ready to download and use in the external web system. Nature of the package depends on the target platform, in most cases these are standard Web Application Archives (WARs).

At the moment of writing of this article *RemoteVineGeneratorApp* supports the following: other instance of the Liferay, Facebook (integration illustrated in figure 10) and iGoogle. It is planned to cover much more services in the near future.

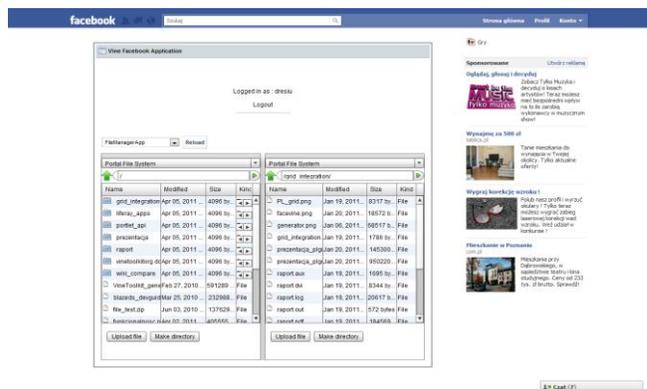


Fig. 10. Vine Toolkit's File Browser accessed from the Facebook

In addition we have implemented Single sign-on (SSO) application, which allows authenticate against the platform Vine is integrated with (account is mapped basing on the email address of the user received from the platform). Moreover, it is possible to set up Vine to automatically create an account for the new users (Single sign-up). Also in this case, data provided by the platform Vine is integrated with is utilized (user profile is automatically filled in). Both Single sign-on and Single sign-up are currently implemented for the Facebook and iGoogle.

FUTURE WORK

Our future work concerns different areas. First of all we will cooperate with our current end users to extend the tools we created so

far. For example in case of nanotechnology related science gateway we want to add support for more applications in near future like Quantum Espresso or NAMD. Secondly, we plan to start work on similar gateway solutions for different scientific domains like biotechnology or chemistry. It is worth to work in different domains as the potential set of solutions becomes more complementary. Thirdly, we want to continue working on the data management issues as the topic plays more and more important role related to large scientific simulations. Work towards integration of the Vine Toolkit into third party services will be also continued.

CONCLUSIONS

In this paper we presented different solutions built with the help of Vine Toolkit related with science gateways in general. We showed that the web technology could be used to prepare front end interface for large infrastructures and to encapsulate scientific applications in the form of well designed web application. Adobe Flex technology allows for creating very advanced applications, which might be helpful in analysis of scientific data or preparing input data like models of nano structures or chemicals. Science gateway potentially takes a central place in the end user's interaction process with large infrastructures. Cooperation between end users and developers possibly leads to the situation where all functional requirements are satisfied by the portal solution and end users do not have to use other external tools or services to conduct whole scientific process related with computations and data analysis. Another important aspect is the data flow in such solution which is automated in a large extent. Users do not have to worry about the data because system takes care of everything and holds the whole history of executions and outcomes for future analysis. For many use cases it is worth to apply such approach regarding end users' access. The Nano Science Gateway is used by the group of nano scientists in Poznan and was also successfully introduced at the NANO'2010 nanotechnology workshop which took place in Poznan in previous year. The solution will be presented also this year at the NANO'2011 nanotechnology workshop in Gdansk. The NAMD solution will be developed and disseminated within the ongoing PL-Grid project. The iRODS web interface is a result of a research activity in HPC-Europa2 and is rather a proof of concept which may be a foundation for future data management tool with metadata support. Advance reservations are applicable to any scenario with large scientific applications simulations – we plan to use it to improve the system responsiveness and give users possibility to predict their grid resource usage time. We did also a big step towards social networking and our integration mechanisms available for Facebook or iGoogle could be used to promote Scientific Gateway web components amongst the broader community.

ACKNOWLEDGEMENTS

The authors would like to thank all people whose work allowed us to provide the Nano-Science Gateway for the scientific community. The special thanks goes to Mr. Michał Hermanowicz and people at the Institute of Physics, Poznan University of Technology who provided a lot of useful feedback to user interfaces and generated main requirements.

This work has been funded by the PL-Grid project: contract number: POIG.02.03.00-00-007/08-00, website: www.plgrid.pl. The

PL-Grid project is co-funded by the European Regional Development Fund as part of the Innovative Economy program.

REFERENCES

- [1] <http://www.man.poznan.pl/online/en/>
- [2] <http://www.liferay.com/>
- [3] <http://www.gridisphere.org/gridsphere/gridsphere>
- [4] <http://vinetoolkit.org/>
- [5] <http://www.adobe.com/pl/products/flex/>
- [6] <http://opensource.adobe.com/wiki/display/blazeds/BlazeDS>
- [7] <http://www.ogf.org/documents/GFD.56.pdf>
- [8] <http://www.ogf.org/documents/GFD.108.pdf>
- [9] <http://glite.cern.ch/>
- [10] <http://www.unicore.eu/>
- [11] <http://www.globus.org/>
- [12] http://larix.man.poznan.pl/wiki/QosCosGrid_Tutorial
- [13] <http://www.plgrid.pl/en>
- [14] Krzysztof Kurowski, Piotr Dziubecki, Piotr Grabowski, Michał Krysiński, Tomasz Kuczyński and Dawid Szejnfeld, *Nano-Science Gateway development with Vine Toolkit and Adobe Flex*, Proceedings of the International Workshop on Science Gateways (IWSG2010), Catania, Italy, September 20-21, 2010
- [15] <http://www.abinit.org/>
- [16] <http://www.govasp.com/>
- [17] <http://www.xcrysden.org/>
- [18] <http://elk.sourceforge.net/>
- [19] <http://blog.papervision3d.org/>
- [20] <https://www.irods.org>
- [21] <http://www.hpc-europa.org/>
- [22] <http://lca.ucsd.edu/portal/software/enzo>
- [23] <https://www.irods.org/index.php/Jargon>
- [24] <http://www.hdfgroup.org/HDF5/>
- [25] Krzysztof Kurowski, Tomasz Piontek, Piotr Kopta, Mariusz Mamoński, Bartosz Bosak, *Parallel Large Scale Simulations in the PL-Grid Environment*, COMPUTATIONAL METHODS IN SCIENCE AND TECHNOLOGY Special Issue 2010, 47-56
- [26] <http://www.gridlab.org/WorkPackages/wp-9/>
- [27] http://larix.man.poznan.pl/wiki/SMOA_Computing
- [28] <http://www.ogf.org/documents/GFD.114.pdf>
- [29] <http://www.gridforum.org/documents/GFD.179.pdf>
- [30] http://www.w3schools.com/Schema/schema_dtypes_date.asp
- [31] <http://www.euforia-project.eu/EUFORIA/>
- [32] <https://kepler-project.org/>
- [33] <http://www.google.pl/ig>
- [34] <http://www.facebook.com/>
- [35] Dawid Szejnfeld, Piotr Dziubecki, Piotr Kopta, Michał Krysiński, Tomasz Kuczyński, Krzysztof Kurowski, Bogdan Ludwiczak, Tomasz Piontek, Dominik Tarnawczyk, Małgorzata Wolniewicz and Piotr Domagalski, Jarosław Nabrzyski, Krzysztof Witkowski, Vine Toolkit - Towards portal based production solutions for scientific and engineering communities with grid-enabled resources support, Scalable Computing: Practice and Experience, Volume 11, no. 2 (June 2010)



Krzysztof Kurowski holds the PhD degree in Computer Science and he is leading now Applications Department at Poznan Supercomputing and Networking Center, Poland. He was involved in many EU-funded R&D projects in the following areas Distributed and High Performance Computing and Grids over the last few years, such as GridLab,

inteliGrid, HPC-Europa, or QosCosGrid. He was a research visitor at University of Queensland, Argonne National Lab, and CCT/Louisiana University. His research activities are focused on the modeling of advanced applications, scheduling and resource management in networked environments. Results of his research efforts have been successfully presented at many international

conferences and workshops. The work has been funded by the HPC-Europa2 project under grant agreement number 228398, website: www.hpc-europa.org.

Piotr Dziubecki received his M.Sc. degree in Computer Science from Czestochowa University of Technology in 2006. The same year he joined Poznan Supercomputing and Networking Centre, Poland where he works as a software analyst. Before joining PSNC Mr. Dziubecki was involved in the development and benchmarking of the parallel multiple sequence alignment software. Currently his main area of interest is interface design for grid-aware portals. Since 2006, until now Piotr has participated in a number of EU funded projects such as inteliGrid, OMII-Europe, and BeInGrid. He is an author or co-author of several papers in professional journals and conference proceedings.

Piotr Grabowski obtained his M.Sc. in Computer Science in 1998 at Poznań University of Technology, Distributed Computer Systems. After his M.Sc., he joined the programmers group at PUT and worked on mobile network protocol analyzers software for Siemens A.G. and Tektronix, Inc. Since 2002 he has been working in Poznań Supercomputing And Networking Center, Application Department. In 2002 he joined GridLab project and worked on access for mobile users. Since 2005, until now Piotr has participated in a number of EU funded projects such as inteliGrid, OMII-Europe, and BeInGrid and local initiatives on enabling access from Mobile Devices and Portal technologies. The main thrust of his current research work are Mobile Devices and e-Mobility, Web Services and Web technology. He is an author or co-author of several papers in professional journals and conference proceedings. Since 2010 he leads User Interface Laboratory at PSNC.

Tomasz Kuczyński holds MSc Eng degree in Computer Science (Software Engineering) from Czestochowa University of Technology. Since 2004 employed at PSNC. He was involved in many, both EU-funded and national research projects, such as GridLab, HPC-Europa, OMII-Europe, BEinGRID, ACGT to name a few. Led users' interface WP in the ClusterIX national project. Contributed to the GridSphere portal project. Currently he is taking part in PL-Grid national project. Main areas of expertise are web technologies and security. He is credited with discovery of numerous security vulnerabilities in portal software including Liferay, Apache Tomcat, Mortbay Jetty, Caucho Resin and others. All vulnerabilities discovered by Tomasz was verified and published by the US-CERT (Department of Homeland Security of the USA).

Michał Krysiński received his M.Sc. degree in Computer Science from Poznań University of Technology in 2008. Since 2006, when he joined PSNC, Michał has participated in a number of EU funded projects such as OMII-Europe, and ACGT. He is an author or co-author of several papers in professional journals and conference proceedings.

Dawid Szejnfeld Head of Laboratory of Portal Applications of Application Department at PSNC, received his MSc in Computer Science from Poznań University of Technology in 2001. After studies he gains commercial experience during his work for ALMA S.A. and for one of the biggest commercial web portals in Poland called Wirtualna Polska. Since 2004 as PSNC employee he

has been involved in many EU funded projects in the area of web and grid technologies and web security like HPC-Europa, OMII-Europe, BEinGRID. Currently involved in HPC-Europa2 project and polish infrastructural project called PL-Grid. He is an author or co-author of several papers in professional journals and conference proceedings. Technical certification gained: MCP, MCSE and MCDBA.