

# g-Social: Enhancing Integrated e-Science Tools with Social Networking Functionality

Andriani Stylianou

Department of Computer Science  
École Polytechnique Fédérale de Lausanne  
andriani.stylianou@epfl.ch

Nicholas Loulloudes

Department of Computer Science  
University of Cyprus  
loulloudes.n@cs.ucy.ac.cy

Marios D. Dikaiakos

Department of Computer Science  
University of Cyprus  
mdd@cs.ucy.ac.cy

**Abstract**—During the last decade, the scientific community has witnessed an unprecedented deployment of large-scale, federated *e-Infrastructures* such as Grid Computing, primarily for supporting data-intensive scientific exploration and coordinated problem solving. However, practical experience and user studies have indicated that the adoption of such *e-Infrastructures* is lagging behind original expectations, a fact which is mainly attributed to the limited support that available tools provide for user collaboration and information sharing. The goal of this paper is twofold, first to lay down the foundations for building a collaboration environment in the form of abstractions and second to show the effectiveness of these abstractions through *g-Social*, an Eclipse-based, open-source environment as an extension to *g-Eclipse*, that provides a powerful, user-friendly, platform-independent toolset for users, application developers and administrators of Grid infrastructures. *g-Social* enables user collaboration and resource sharing through Online Social Networking services, capitalizing on the success that these services have.

## I. INTRODUCTION

During the last 10-15 years, a large number of research projects have focused on developing large-scale, federated computing infrastructures to support advanced scientific and engineering work. A main driving force behind these projects has been the push towards the “*Fourth Paradigm*,” a new paradigm of scientific work promoted by the late Jim Gray [1], [2]. This paradigm is characterized by data-intensive scientific exploration that relies on the analysis of huge volumes of data produced by computer simulation and scientific instruments (telescopes, microscopes, environmental sensors, DNA sequencers, etc.) [3]. Data-intensive scientific exploration is an activity that is inherently collaborative and requires the sharing of unprecedented computing resources that are hard to procure, operate or maintain centrally. Therefore, it raises the need for distributed *e-Infrastructures* aggregating large-scale computing and storage power and providing resource management and application support services. This need has been addressed by Grid computing research, which pursued the vision of developing a “wide-scale, distributed computing infrastructure for resource sharing and coordinated problem solving in dynamic, multi-institutional Virtual Organizations” [4], [5]. Currently, several Grid infrastructures are in production-mode operation throughout the world, federating numerous resource providers and serving a variety of scientific disciplines and user communities [6], [7], [8].

Despite the significant progress made by Grid computing research, practical experience and user studies show that the

adoption of Grid technologies is lagging behind original expectations [8]. This lag is attributed to: i) The high complexity of Grid middleware and protocols, which results to a rather limited usability of Grid tools and to a high-barrier of entry to Grid infrastructures [9]; ii) The limited support that current Grid middleware and tools provide for user collaboration and information sharing in the context of Virtual Organizations and beyond [9], [8]. Current Grid tools do not treat user collaboration as a first-class citizen. The few tools that do incorporate collaborative functionality are application-specific or oriented to the needs of particular communities [10], [11]. Consequently, Grid end-users, application developers and administrators have to rely on other approaches for collaboration and information sharing, from e-mailing, wikis, and blogs to crowdsourcing and online social networking services [12]. The great potential that these tools bring to online collaboration and information sharing has been demonstrated in the context of numerous, highly successful Internet services [13], [14]. Nevertheless, their adoption in the day-to-day operations of Grid users would require an extensive manual effort on their behalf and would result to an increase of the users’ information overload. Therefore, there is a clear need to integrate collaborative and information sharing functionality with existing, general-purpose Grid services and tools.

In our work, we address this need by introducing *g-Social*, an extension to *g-Eclipse*, an Eclipse-based, open-source environment that provides an integrated set of powerful, user-friendly, platform-independent tools for users, application developers and administrators of Grid and Cloud infrastructures. *g-Eclipse* represents various computing and storage entities as platform-independent abstractions, hiding many of the complexities of different middleware platforms. *g-Social* enables the sharing of user interactions with these abstractions through Online Social Networking services, capitalizing on the success that these services have.

## II. CONCEPTS

### A. The *g-Eclipse* Framework

*g-Eclipse* [15], [16], [17] is an integrated workbench framework that provides a middleware independent, unified toolset for users, operators and developers of Grid infrastructures. It is built on top of the reliable Eclipse [18] platform and follows its plug-in based software architecture. *g-Eclipse* consists of core plug-ins that enable and standardize access to Grid

infrastructures from within Eclipse, independently from any middleware running on those infrastructures. It supports a rich functionality including job and file management; application development, deployment and benchmarking; workflow programming; infrastructure configuration, monitoring and testing. The intuitive and friendly user-interface of g-Eclipse follows the Eclipse user-interface guidelines, hiding from the user the complexity of interaction with Grid and Cloud infrastructures. In turn, this results in a low-entry barrier for users who are new to these technologies, while simultaneously improving the workflow efficiency of experienced users. Moreover, thanks to the plug-in architecture of Eclipse, the g-Eclipse functionality and GUI can be easily extended and customized to support new and emerging complex user needs, offering obvious advantages over Web-based tools. The latest version provides support for the gLite [19] and GRIA [20] middleware platforms, as well as support for Amazon's Cloud Web services EC2 and S3 [21].

Consequently, g-Eclipse represents an ideal "macroscope" platform [22] for integrating social and collaborative functionalities that arise in the context of the "Fourth Paradigm" of scientific work.

### B. Motivating Scenario

Consider a group of High Energy physicists who study the impact of cosmic rays on the environment. The group runs several *projects*, each one examining the problem from a different scientific viewpoint with dedicated applications and datasets. Since HEP applications are well known for their demanding computational and storage requirements, the group employs a Grid infrastructure<sup>1</sup> for processing and analyzing large datasets containing field measurements. Such applications are submitted as single or parametric *jobs*, but nevertheless each job-run has its own requirements and performs a specific task. Requirements, usually described in a job definition file, include which executable should be run, any execution parameters and the location of input and output file(s). Traditionally, such applications and the set of results generated upon the execution of the respective jobs are only utilized during the lifetime of the corresponding project. Collaboration among group members is done via e-mail, online chat and/or the Web, wherein scientists can disseminate experiences, exchange ideas, and send notifications with pointers to datasets and new results. Consequently, most of the collaboration is done outside the software environment(s) employed to access, program and use Grid resources (computing and storage hardware, data, configuration files, etc.). Therefore, scientists need to spend a lot of manual effort transferring information (unstructured text, structured data, links, files, simulation setups) from their communication medium (e-mail client, chat application or Web page) into their software environment(s) (Grid portals, command-line interfaces, etc.). Lately, group members have joined online communities and social networking (OSN) services like Facebook and Twitter, establishing a HEP-focused

<sup>1</sup>A Cloud infrastructure can also be utilized. However the above example assumes a Grid computing infrastructure.

online community and use those services in an effort to share information and knowledge seamlessly. OSNs offer several advantages over e-mail and wikis, as they provide support for the easy establishment and management of groups, and for the automatic dissemination of notifications to selected people. Nevertheless, the propitious features that OSN services provide are not integrated into the software environment(s) used by the scientists to undertake their computational work, thus requiring again substantial manual effort to share information.

In order to enhance scientific collaboration, HEP group members would need to have the Grid client environment extended with functionality to: i) organize and manage the information related to a project (i.e. job submission, resource management, etc.) through a simple and intuitive graphical user interface; ii) seamlessly share and retrieve, through their social network, all the particulars of one or more *research work(s)* performed in the context of a project. Specifically, any member of the HEP group should be able to:

- *Define structured and standardized social meta-data.* Meta-data such as description, tags, versioning, etc., present to a social network the functions and requirements of a job. By reading these meta-data, social network users can get a clear understanding whether a particular research work is of their interest. In addition, meta-data such as the above would be highly beneficial to search engines.
- *Enrich social metadata with links to project-related resources.* Such resources include: application executable(s) and/or source code, input datasets, job definition files, job results, user/developer manuals, published articles, etc. Since, resources such as the above can potentially be dispersed in heterogeneous storage facilities (local machine, Grid, Cloud, Web), the tool should utilize mechanisms that hide from the user the complexity of interacting with the underlying infrastructure.
- *Access all the project-related resources easily.* In essence, the user project should be considered as a bundle of related resources comprising of the social networking meta-data and real project data. Since the real project data can be stored in different storage services, their access is complex and it should be done in a user-friendly manner.
- *Share project data and meta-data* to a social network with just a few clicks. Consequently, interested parties (individual colleagues or groups) should be automatically prompted with a notification on the availability of the above data. Notifications should obey the rules and constraints of any social network.
- *Retrieve shared information* easily through an intuitive user interface that includes searching and sorting facilities. Access to project resources that are protected, should automatically trigger the necessary authentication processes required by the underlying infrastructure.
- *Interact seamlessly with Online Social Networks (OSN)* such as Facebook and Twitter. Mechanisms should enable the user to share information related to a project, with

other users and groups of these OSN. Consequently, such mechanisms must respect the principles and policies defined by these platforms and enforce user privacy by invoking the necessary authentication processes. Furthermore, considering that OSN are by design web-based systems, the tool should be able to seamlessly interface with a web-browser, to ease the automated import of such shared information in the user’s workspace with just a few clicks.

### C. OSN and scientific collaboration

Scientists use traditional Internet services to disseminate and exchange information worldwide, making access to knowledge universal. But traditional, Internet-mediated communication should be improved further and extended in order to facilitate online scientific collaboration through the use of Online Social Networking services. OSNs present a high potential in that direction, because of their structure that has important implications for the spread of valuable scientific knowledge and because of the huge participation of end-users, which can provide complete domain coverage for the scientists. Popular OSN services embrace a critical mass of hundred of millions of users from all over the world and they assist their every day communication and exchange of information. Other social networks concentrate their functions on the needs of a specific group of people or interest domain and have either met a premature success or failed completely because of the low awareness and the outdated, sparse user profiles indicating the effectiveness of a global general social network. The fast, easy and free-of-charge information sharing and retrieval in OSN and the possibility to attach Internet-based resources to OSN services, facilitates the capability of using popular OSNs for sharing data and knowledge, and for supporting scientific collaboration.

### D. Main Challenges

Although today social networks like Facebook, Twitter, LinkedIn etc., have taken a form that the majority of the users accepts, finds efficient and simple to use (e.g. user profile, news feed, groups, posts etc.), it is by no means the case that a scientific social network can follow the exact same model because of its unique and distinguishable characteristics that set it apart from a general purpose social network. Specifically in building a new scientific social network we were faced with the following challenges:

- *Handling a diversity of projects:* g-Eclipse for instance is used by physicians, chemists and non-engineer scientists to execute a variety of projects. This suggests that the abstractions should be general enough to cover the multitude of projects differing in both semantics and structure.
- *Keeping the user effort low:* intended users have different characteristics compared to those of a general purpose social network such as less time and will to spend familiarizing with a new product. Thus, in this case the benefits should exceed effort and the product should be easy to learn, use and maintain.

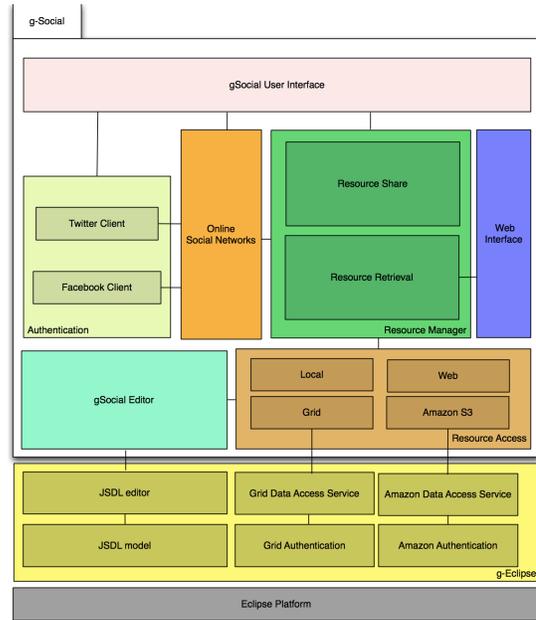


Fig. 1: g-Social software architecture

- *Ensuring the security of data:* an important challenge since scientific data may contain private or confidential information. Users may want to restrict access for example only inside a research center or a university faculty. The social network should give users confidence that their data is safe.
- *Forming scientific friendships:* since these “friendships” are based on domains of interest, they are not constant and are hard to discover. The social network therefore, should facilitate the formation of friendships and the discovery of common interests between users.
- *Handling of large amount of data efficiently:* knowing that a scientific project can contain enormous amount of resources. These resources to be useful should be searchable and the user should be able to extract pieces of resources that interest him.

## III. THE G-SOCIAL ARCHITECTURE

### A. The g-Social Abstractions

As discussed earlier, the g-Eclipse framework provides abstractions of Grid infrastructures to its users, which facilitate and coordinate any task-oriented interactions. Primarily, the *Grid Project* abstraction [23], provides a placeholder for a user to organize, in an ad-hoc manner, the files and information required for the execution of various Grid and Cloud tasks in the context of a particular research project. Specifically, it is a self-contained entity that collects and maintains all the files, job descriptions, links to remote files, connection endpoints to local and/or remote storage resources as well as complementary information (dependencies) required by jobs which were or will be submitted to an execution infrastructure.

Through the abstractions presented in this section, g-Social aims to supplement the above by enabling its users to seamlessly share and retrieve, via a social network, all the particulars of the research work performed in the context of

a real project. Following the motivating scenario described in Section II-B, below we present the supported high-level abstractions.

### Resource

In the context of g-Social, a resource refers to any file(s), which is directly or indirectly related to the execution of a Grid task specific to a scientific project. A list of such file(s) is (not limited to): an input dataset, application executable(s), source code and execution output, configuration file(s), user/developer documentation, as well as published articles that present the project findings to the scientific community. In most cases, these resources are scattered in heterogeneous storage facilities (Grid, Cloud, Web, local hard drives) and access to them might require the use of dedicated authentication procedures and transfer protocols.

By interfacing with the g-Eclipse file system [23], g-Social supports any type of resources and provides to the user seamless access to different types of storage facilities such as: i) GridFTP, DPM or the Storage Resource Manager (SRM) for Grid infrastructures; ii) HTTP or FTP for Cloud (Amazon S3) infrastructures and the World-Wide Web; iii) local storage (hard drives, removable media); iv) CVS or SVN for version control systems.

A notable feature of g-Social is the consideration of Grid Projects as individual resources. Since Grid Projects are resource placeholders, effectively they are also considered to be resources themselves. Therefore, scientists now have the capability to share their complete working environment and the state of their interaction with the infrastructure in just a few clicks. In turn, other scientists that wish to utilize or extend any of the work available, will simply have to import the shared Grid Project in their g-Eclipse workspace, avoiding the time consuming and often error prone installation and configuration procedures.

### Social Meta-data

As aforementioned, the principal idea behind g-Social is to enable its users to share, through a social network, all the particulars of the research work performed in the context of a project.

For instance, a HEP Grid Project dedicated to the study of cosmic rays, might involve a number of different applications, each one performing a different and specific task. At the same time, despite the use of correct software development practices, it is possible for one version of an application to be substantially different from another version in terms of the required input dataset, execution workflow, runtime parameters or even simply the output format. Thereby, it is obvious that due to the diversity in the context of these project particulars, there is a need for traditional descriptive meta-data, such as *name*, *description*, *version*, *license* and *tags* (for a full list see Table I) which provide to the social network and its users, the necessary and sufficient information about their purpose and function. In addition such metadata allow content to be indexed and therefore easily discoverable using the searching and filtering capabilities of social networks.

Having in mind that the majority of social networks cur-

rently available do not support file exchange among their users, it is crucial to inform the recipients of shared items for the location and how they can be retrieved. Consequently, g-Social enriches the social meta-data with links to the location of project-related resources. Specifically, these links are governed by well-defined file schemas introduced by g-Eclipse (see Section IV for details) that dictate the location, as well as, the protocols required to access them.

Name	Version	License
Function	Purpose	Design Explanations
Tags	Screenshots	Relevant Resources

TABLE I: Descriptive social meta-data used in g-Social.  
**Authentication Manager**

The authentication manager enforces the security and privacy control of users while interacting with a social network. Primarily, it holds the responsibility for ensuring the user is authorized and authenticated against a particular social network. It does so by executing the necessary authentication mechanisms required by the social network. Secondly, it is responsible to monitor the life cycle of all the authentication tokens created for accessing and interacting with a social network. The tokens are maintained only in memory and stay active until g-Eclipse is closed or after a validity period has elapsed.

### Resource Manager

The sharing, as well as the retrieval (of previously shared) resources in g-Social, is handled through the Resource Manager. In terms of sharing, the manager is responsible to complete the necessary steps in order for the resource to become available in a social network. This includes:

- calling the Authentication Manager to ensure that the user is authorized to commit content to the social network.
- collect all the social meta-data.
- encapsulate all the above information in a form that is suitable and acceptable for publication to the specific social network.
- calling the SN-specific API to publish the content.

On the contrary, in terms of resource retrieval from a social network, the manager is responsible for:

- the extraction of all the published data, including meta-data and resource location links.
- determining whether g-Eclipse authentication manager should be invoked, in order to create the necessary Grid infrastructure access tokens.
- calling the g-Eclipse file system manager for retrieving the shared resources.
- importing the resources in the g-Eclipse workspace. If the resource is a Grid Project, then a new project is created and links to all the contained resources are established.

### Online Social Network Interface

By design Online Social Networks are web-based systems, thus the primary means of accessing the content published in those is via a web-browser. Scientists who are users in such OSNs might come across to interesting posts by other colleagues or groups, which were shared through g-Social.

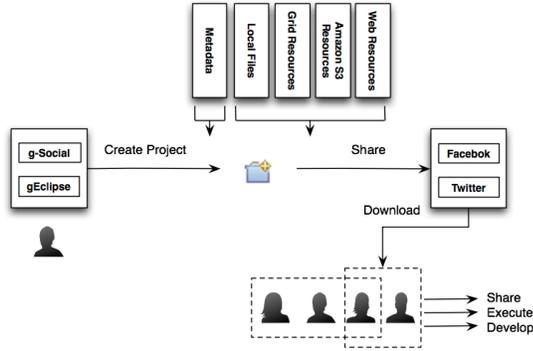


Fig. 2: g-Social Execution Flow

For this reason, the OSN-gEclipse interface serves as an intermediate between the web-browser and g-Eclipse. If the user clicks a post in the web-browser that can be handled by g-Eclipse, then Eclipse is invoked and the necessary steps are performed.

#### IV. DESIGN AND IMPLEMENTATION

##### A. The g-Social Abstractions Implementation

This section describes the implementation of g-Social as extension components to the g-Eclipse framework. The implementation is driven by Equinox, an implementation of the Open Services Gateway Initiative specification (OSGi)[24] also used in the core of the Eclipse IDE. g-Social has benefited greatly through the utilization of OSGi, since it capitalized from a robust system which manages in a dynamic manner the life-cycle of various components.

For the implementation new plug-ins were contributed to the Eclipse eco-system and existing g-Eclipse plugins were modified to support the social meta-data. Overall, approximately 8000 lines of code have been added to the g-Eclipse code base for the complete g-Social extension. The set of abstractions described in Section III are reflected in the internal model of g-Social. In the following paragraphs we describe the implementation details of the abstractions.

##### Resource

A resource in g-Social can be identified and retrieved through a unique g-Eclipse URI string as shown in Table II. This URI is valid for any resource residing in the local user environment or any remote storage locations. Each URI consists of two parts: i) a corresponding schema part that determines the protocol and the access method/mechanisms that ought to be used in order to retrieve the resource; and ii) the actual location of the resource. For the user the resource access is transparent and is presented in the implementation of resource manager.

Local File	gecl:C:/hiberfil.sys?geclslave=file
Grid Storage Element	gecl://storage.mydomain.com:2811/data/Graphs/run.sh?geclslave=gsiftp
Amazon S3	gecl://AKIAJZYMN2S6ECLJKFBA/geclipsesocial/serial_rt?geclslave=s3
Web Resource	gecl://http://www.es.ucy.ac.cy/es00as0/documentation.pdf

TABLE II: Example URIs for different storage locations. For the users the complexity of accessing many different storage architectures is hidden.

##### Social meta-data

Social meta-data are the main objects that are shared, stored and searched by g-Social users since the actual data are not accompanying the shared bundle. For that reason they need to be easily and efficiently accessed and stored in a persistent manner. Therefore, we propose an extension to the Job Submission Description Language (JSDL) [25] to permit the user to enrich job descriptions with social information. In order to sustain social meta-data a structured data model is defined by supplying an XSD Schema to the Eclipse Modeling Framework Project (EMF) [26], since JSDL is an XML content document.

##### Authentication Manager

The authentication manager in g-Social operates in two levels: (i) The authentication of a user against a storage infrastructure (e.g. Grid, Cloud) by triggering the authentication mechanisms of g-Eclipse; (ii) The authentication of a user to the OSN. The APIs of the OSN had been used both to authenticate the users and to advantage from any functionality they offer [27], [28].

##### Resource Manager

The resource manager consists of centralized web server with a back-end relational database server for storing the social meta-data of shared bundles and the local shared resources. The data are saved under the active user account. These data can be accessed later on by relational queries and the contained URIs are extracted and can be used to retrieve the resources through the mechanisms for Resource Access of g-Eclipse.

##### Online Social Network Interface

A plug-in called OSN-gEclipse has been developed to provide seamless interaction between a web-browser and a running instance of g-Eclipse with g-Social installed. The plug-in follows the Eclipse Web Interface paradigm, and adds the ability to externally invoke commands in Eclipse via a web browser. This external command invocation is realized through a simple web server (loaded on Eclipse startup), listening for HTTP requests on the localhost interface and on a pre-defined port.

Once such a request is received, g-Eclipse will sequentially: i) check if the request contains an acceptable command; ii) a handler exists for the command; and iii) forward the command parameters to that handler. In g-Social these commands are invoked when the user clicks a link in the shared post in an OSN like Facebook or Twitter. The job name contains an encoded hyperlink such as: `http://localhost:34567/?command=eu.geclipse.socialnetwork\webInteraction.openfile&path=http://my.storage/proj.zip&schema=1` that issues an `openfile` command to g-Eclipse for an archive file `proj.zip` located on `http://my.storage`. The current plug-in implements the `openfile` handler for retrieving and opening resources related to a job. The Resource Manager identifies the type of the resource (as described in Section IV-A) and calls the applicable handler for the specific file system to access it. For Amazon and Grid resources the user must follow a wizard to get authenticated and obtain the necessary access tokens.

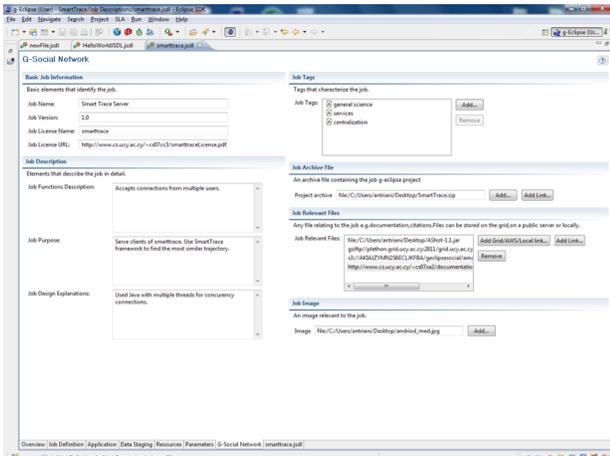


Fig. 3: Extending the g-Eclipse JSDL Editor to allow the definition of Social Meta-data

### B. The g-Social User Interface

#### Social Metadata Editor

The existing g-Eclipse JSDL multi-page editor has been extended with an additional page called “Social Metadata” (see center of Figure 3), which accomplishes the exposition and the editing of social meta-data. The page contains the necessary text/property/list fields for the input of social meta-data. Moreover, the new page provides the necessary fields that assist the user in specifying the location of resources that are related to a job. Adding a resource, invokes the g-Eclipse file system [23] dialog, which provides a unified view for accessing different types of storage such as Storage Elements, GridFTP, DPM, SRM, Amazon S3 and local media (i.e. hard disk and removable media). Once the user identifies and selects one or more related resources, the path to their location is automatically listed in the editor.

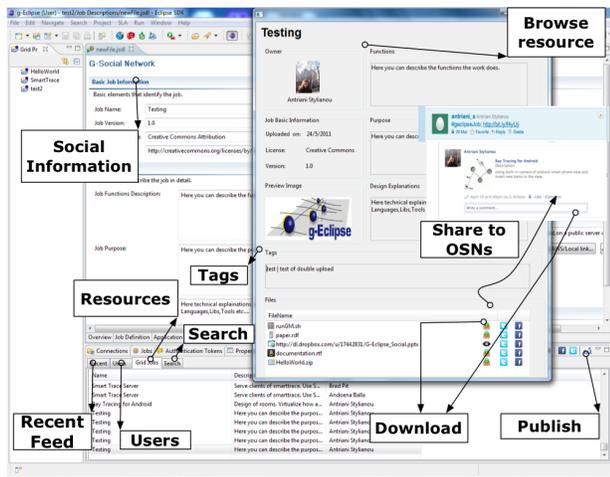


Fig. 4: The g-Social User Interface

#### g-Social View

A multi-tab view (see center-bottom of Figure 4) has been developed that enables the user to interact with a social network. The view consists of several action buttons and tabs, each with the following functionality:

- **Authentication Action Button** enables a user to initiate

on-demand an authentication procedure with an OSN. During this process, the user supplies her credentials to the Authentication Manager which in turn interacts with the authentication system of the selected OSN.

- **Share Action Button** the user can share the social meta-data of a job in a social network. Since, social metadata are embedded in JSDL documents, the wizard presents to the user a list of all files with a JSDL extension available in the g-Eclipse project. Subsequently, the selected JSDL is parsed with the aid of EMF in order to extract the social metadata. The resource is then shared to the g-Social network and can be also disseminated to OSNs as a whole or parts of it. Shared resources in the OSNs are attributed by the respective job name, description and image defined in the social metadata. Finally, the resource name that appears in the OSN contains an encoded hyperlink to be consumed by the Web Browser Interaction plug-in.
- **Job Details Action Button** presents (see Figure 5) the full details of a shared job, as they were defined in the social metadata editor. Importantly, this view provides a list of the resources related to the shared job and a link to their storage location for easy retrieval. Based on the URI schema of each resource, an icon indicates the type of the infrastructure hosting it (i.e. Grid, Cloud, Web, local server). A user action to retrieve a resource from the infrastructure, triggers the g-Eclipse authentication manager to check for the existence and validity of an appropriate authentication token - if no such token is found, the user is prompted to create one. If the requested resource is a g-Eclipse project (Grid project), then g-Social imports the project along with all of its containing resources (folder structure, JSDL files, links to storage locations, etc.) in the active g-Eclipse workspace. If the resource is any other type of file, then g-Social prompts the user to identify the location where it should be stored.
- **Recent Tab** presents, in a tabular format, summarized information concerning each job which was recently (within a user-defined time frame) shared to the social network with the possibility of sorting them using different attributes. Additional details can be viewed by clicking the **Job Details Action** button.
- **Users Tab** provides an overview of the scientific collaboration network of the g-Social user.
- **Jobs Tab** serves as a catalogue of all the jobs that have been shared in the social network.
- **Search Tab** allows the user to search for jobs that have been shared in the social network in the past. Keyword search is performed against social metadata and the job owner name. If one or more matches are found, the respective jobs are presented to the g-Social user. OSN elements like comments concerning the resource are also searched and presented.

### C. The g-Social in action

The basic execution flow of g-Social can be seen in Figure 2. The user runs a g-Eclipse instance with g-Social installed and

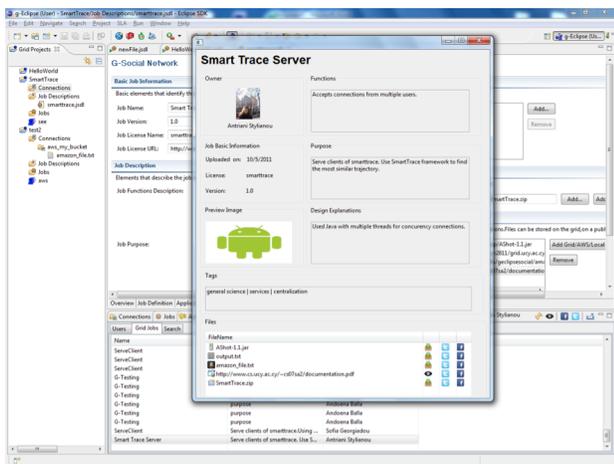


Fig. 5: g-Social UI, detailed preview of a shared job

creates a g-Eclipse project enhanced with social meta-data and pointers to remote or local resources. Next, the user can share the project to the g-Social and then choose its favorite social network to create a post with links to any of the resources or meta data. Depending on the permissions used in sharing the data different groups of users can see the shared data and they can in turn, share them again, download and execute them and/or further develop the downloaded data.

## V. RELATED WORK

The emergence and profound adoption of Online Social Networks such as LinkedIn [29] and ResearchGate [30] have propelled professional networking and serve as valuable tools for scientists and researchers to facilitate global scientific collaboration. Specifically, they enable their users to create profiles related to their profession/expertise, network with current and prospective colleagues, discuss and publish the findings of their scientific work and identify new topics of interest. Nevertheless, despite the significant features available in these platforms, they: i) are by design external to existing scientific software environments and ii) lack the ability to support resource sharing among collaborating parties.

Focused around specific application domains, the nanoHub [11] and myExperiment [10] social networks aim at providing a more immersive scientific collaboration environment and overcoming the above inabilities. The nanoHub project has established a website which serves as an online social hub for the nanoscience community, wherein field researchers come to collaborate, share resources (notes, tools, publications, etc.) and run computationally demanding simulations in order to solve real nanotechnology problems. Simulation jobs are transparently dispatched for execution to distributed computing resources (in the past at the NSF TeraGrid, currently at the Cloud) and results are rendered back to the web-based simulation tool. More recently, myExperiment [10] has provided a dedicated virtual research environment that enables scientists (primarily of the bio-informatics community) to collaborate by sharing data and know-how associated to their research work. Through a web-based portal, users are able to build their own research-

oriented social network, find other colleagues with similar interests, create or join groups, and more importantly discover and share scientific workflows. myExperiment supports the Taverna [31] workflow system, which provides a standardized high-level declarative way of describing digital methods for the analysis and execution of complex scientific experiments. Workflows are downloaded from the myExperiment portal and are consequently imported and executed inside the Taverna dedicated workbench.

However, both nanoHub and myExperiment are not generic solutions, thus making it hard for scientists from other domains to participate. In addition, by relying on proprietary social infrastructures their target userbase is required to introduce, monitor and interact with, an additional information source on a daily basis, thereby increasing the manual effort on their behalf and resulting to possible information overload.

Chard et al. [32] introduce the notion of social collaboration to the Cloud by proposing the Social Cloud scalable computing model, where users discover and share computation and storage resources by taking advantage of the pre-established trust formed through friend relationships available in existing social networks. Following the distributed model of volunteer computing, Social Cloud allows users to share between them, their personal computing or storage resources, while at the same time enabling user accountability and management by leveraging the trust existing in social friend relationships.

While the concept of Social Cloud in terms of physical computing/storage resource sharing looks indeed very promising for the future, what is highly interesting and applicable to current times is the exploitation of the inherent trust stemming from social friend relationships. g-Social capitalizes on this by providing the necessary abstractions that enable scientists to utilize their existing memberships in one or more Online Social Networks in order to establish scientific collaborations. Most importantly, in contrast to nanoHub and myExperiment, since g-Social is an extension to the g-Eclipse framework, it allows scientists from any application domain to seamlessly collaborate and share information among them within a powerful, user-friendly, platform-independent software environment.

## VI. FUTURE WORK

In the near future we are planning to propose the establishment of the extension of social meta-data in the description of a job as described in Section IV so that any project can be shared by adding a few more data fields to create its social perspective. This will also help in sharing existing projects without substantial user effort. Moreover for the next version of g-Social we are working on a more complicated recommendation system based on text mining techniques to provide a more solid user social experience and to facilitate searching and discovery of individual resources or projects.

We have distributed the final version of g-Social for testing to a group of researchers and a group of g-Eclipse developers conducting a survey that demonstrated: i) the rapidly growing need for sharing scientific data; ii) the simplicity in installation, usage and maintenance of g-Social. Since the results

of the survey are encouraging we are planning to release an official version of g-Social and make it publicly available via the g-Eclipse community update site.

## VII. CONCLUSIONS

The massive use of social networks today, has radically changed the way Internet users communicate. If the trend continues and we exploit the shared information we can hope for huge leaps in technology and society.

If the scientific community comprehends the value of the scientific prospect of social networks, they can augment the value of their work by reducing the cost of research and development and improving the quality of the work.

Virtual organizations in Grid Computing, was the first step towards cooperation between scientific groups, however they did not reach their full potential because of their concentration on technical functions for accessing and using the Grid infrastructures.

In this paper we have introduced g-Social, a mechanism that allows communication, access and retrieval of resources by adopting the simplicity and usefulness of Web 2.0. The advantages for the scientific community are obvious for developing projects but also for the quality, reuse and their preservation.

g-Social is satisfying key needs of a scientific group by:

- Defining structured and standardized social meta-data, which make a resource sharable and retrievable.
- Enriching the meta-data with links to resources that can reside in any protocol accessible local or remote location.
- Providing efficient access to all resources, because of the non-costly nature of links.
- Making sharing effortless, since sharing can be done in a few clicks to any well known social networks and can be instantly seen in the g-Social social network view or the OSN in conjunction with the resource manager which provides seamless access to any resource hiding any possible complexity.

## REFERENCES

- [1] J. Gray, "eScience: the Next Decade Will be Interesting!" Talk at ETH, Zurich, Switzerland, May 22 2006.
- [2] T. Hey, S. Tansley, and K. Tolle, Eds., *The Fourth Paradigm: Data-Intensive Scientific Discovery*. Microsoft Research, October 2009.
- [3] G. Bell, T. Hey, and A. Szalay, "Beyond the Data Deluge," *Science*, vol. 323, pp. 1297–1298, March 2009.
- [4] I. Foster, C. Kesselman, and S. Tuecke, "The Anatomy of the Grid: Enabling Scalable Virtual Organizations," *International J. Supercomputer Applications*, vol. 15, no. 3, pp. 200–222, 2001.
- [5] I. Foster and C. Kesselman, in *The Grid: Blueprint for a New Computing Infrastructure*, I. Foster and C. Kesselman, Eds. Elsevier, 2004, ch. 4: Concepts and Architecture, pp. 37–64.
- [6] M. Altunay, P. Avery, K. Blackburn, B. Bockelman, M. Ernst, D. Fraser, R. Quick, R. Gardner, S. Goasguen, T. Levshina, M. Livny, J. McGee, D. Olson, R. Pordes, M. Potekhin, A. Rana, A. Roy, C. Sehgal, I. Sfiligoi, and F. Würthwein, "A Science Driven Production Cyberinfrastructure - the Open Science Grid," *Journal of Grid Computing*, vol. 9, no. 2, pp. 201–218, 2011.
- [7] I. Bird, B. Jones, and K. F. Kee, "The Organization and Management of Grid Infrastructures," *IEEE Computer*, vol. 42, no. 1, pp. 36–46, 2009.
- [8] K. L. Kroeker, "Grid Computing's Future," *Communications of the ACM*, vol. 54, pp. 15–17, March 2011. [Online]. Available: <http://doi.acm.org/10.1145/1897852.1897858>
- [9] U. Schwiegelshohn, R. M. Badia, M. T. Bubak, M. Danelutto, S. Dustdar, F. Gagliardi, A. Geiger, L. Hluchy, D. Kranzlmüller, E. Laure, T. Priol, A. Reinefeld, M. Resch, A. Reuter, O. Rienhoff, T. Rüter, P. M. A. Sloot, D. Talia, K. Ullmann, R. Yahyapour, and G. von Voigt, "Perspectives on grid computing," *Future Generation Computer Systems*, vol. 26, no. 8, pp. 1104–1115, 2010.
- [10] D. D. Roure, C. Goble, and R. Stevens, "The Design and Realisation of the Virtual Research Environment for Social Sharing of Workflows," *Future Generation Computer Systems*, vol. 25, no. 5, p. 561, 2009.
- [11] "nanoHub," <http://www.nanoHub.org> (last accessed Sept. 2012).
- [12] J. M. Hellerstein and D. L. Tennenhouse, "Searching for Jim Gray: A Technical Overview," *Communications of the ACM*, vol. 54, no. 7, pp. 77–87, 2011.
- [13] A. Doan, R. Ramakrishnan, and A. Y. Halevy, "Crowdsourcing Systems on the World-Wide Web," *Communications of the ACM*, vol. 54, no. 4, pp. 86–96, 2011.
- [14] G. Pallis, D. Zeinalipour-Yazti, and M. D. Dikaiakos, "Online Social Networks: Status and Trends," in *New Directions in Web Data Management I*, ser. Studies in Computational Intelligence, L. Jain and A. Vakali, Eds. Springer, 2010, no. 331, ch. 8, pp. 213–232.
- [15] P. Wolniewicz, N. Meyer, M. Stroinski, M. Stümpert, H. Kornmayer, M. Polak, and K. Gjermundrød, "Accessing Grid Computing Resources with g-Eclipse Platform," *Computational Methods in Science and Technology*, vol. 13, no. 2, pp. 131–141, 2007.
- [16] K. Gjermundrød, M. D. Dikaiakos, M. Stümpert, P. Wolniewicz, and H. Kornmayer, "g-Eclipse - An Integrated Framework to Access and Maintain Grid Resources," in *Proceedings of the 9th IEEE/ACM International Conference on Grid Computing (Grid 2008)*. IEEE Computer Society, 2008, pp. 57–64.
- [17] H. Kornmayer, M. Stümpert, K. Gjermundrød, and P. Wolniewicz, "g-Eclipse — A Contextualised Framework for Grid Users, Grid Resource Providers and Grid Application Developers," in *Proceedings of the 8th international conference on Computational Science, Part III*, ser. ICCS '08. Berlin, Heidelberg: Springer-Verlag, 2008, pp. 399–408.
- [18] "The Eclipse Platform," <http://www.eclipse.org/> (last accessed Sept. 2011).
- [19] "gLite Middleware," <http://glite.cern.ch/> (last accessed Sept. 2012).
- [20] "GRIA Middleware," <http://www.gria.org/> (last accessed Sept. 2012).
- [21] "Amazon Web Services," <http://aws.amazon.com/> (last accessed Sept. 2012).
- [22] K. Börner, "Plug-and-play Microscopes," *Communications of the ACM*, vol. 54, no. 3, pp. 60–69, Mar. 2011.
- [23] K. H. Gjermundrød, M. D. Dikaiakos, M. Stümpert, P. Wolniewicz, and H. Kornmayer, "g-eclipse - an integrated framework to access and maintain grid resources," in *Proceedings of the 9th IEEE/ACM International Conference on Grid Computing (Grid 2008)*. IEEE Computer Society, 2008, pp. 57–64.
- [24] The OSGi Alliance, "OSGi Service Platform Core Specification, Release 4.1," <http://www.osgi.org/Specifications>, 2007.
- [25] A. Anjomshoaa, F. Brisard, M. Drescher, D. Fellows, A. Ly, S. MCGough, D. Pulsipher, and A. Savva, "Job Submission Description Language (JSDL) Specification, Version 1.0," Global Grid Forum, Tech. Rep., Jun. 2005.
- [26] D. Steinberg, F. Budinsky, M. Paternostro, and E. Merks, *EMF: Eclipse Modeling Framework 2.0*, 2nd ed. Addison-Wesley Professional, 2009.
- [27] "Facebook Java API," <http://code.google.com/p/facebook-java-api/> (last accessed Sept. 2011).
- [28] "A Java Library for the Twitter API," <http://twitter4j.org/en/index.html> (last accessed Sept. 2011).
- [29] "LinkedIn," <http://www.linkedin.com/> (last accessed Sept. 2012).
- [30] "ResearchGate," <http://www.researchgate.net/> (last accessed Sept. 2012).
- [31] D. Hull, K. Wolstencroft, R. Stevens, C. Goble, M. Pocock, P. Li, and T. Oinn, "Taverna: A Tool for Building and Running Workflows of Services." *Nucleic Acids Research*, vol. 34, no. Web Server issue, pp. 729–732, July 2006.
- [32] K. Chard, S. Caton, O. Rana, and K. Bubendorfer, "Social cloud: Cloud computing in social networks," in *Proceedings of the 3rd International Conference on Cloud Computing IEEE Cloud 2010*, 2010.