A Grid Credit System Empowering Virtual Research Communities Sustainability

C. Manuali and A. Laganà

Department of Chemistry, University of Perugia, Perugia (IT)
{carlo,lag}@unipg.it

Abstract. In this paper a new Grid Credit System called GCreS specifically designed for evaluating Grid Virtual Organizations and Virtual Research Communities is presented. GCreS is based on the Evaluation of the Quality of Services and Users of the considered Community. To this end, use is made of a SOA Framework called GriF and of its features useful to support the analysis of Grid activities. Its first application to the Virtual Organization COMPCHEM is also discussed.

1 Introduction

Grid empowered computational experiments, which have become nowadays indispensable for science advances, are typically performed in an open Grid infrastructure as is the case of the European Grid Infrastructure, or EGI [1]. In the work done on the Grid by Virtual Organizations (VO)s, Heavy User Communities (HUC)s or Virtual Research Communities (VRC)s, it is crucial to organize, operate and harmonize the efforts spent and the results obtained by the various members. This is of particular importance for building collaborative complex computational applications on the Grid. To facilitate the achievement of this goal, we have designed and developed a Grid Credit System (GCreS) model aimed at enhancing VRCs Sustainability by leveraging on Quality Evaluation.

GCreS is based on a new Collaborative Grid Framework called GriF [2] designed and developed by us according to a Service Oriented Architecture (SOA) approach [3]. In GriF information related to the behavior of the Grid users and to the characteristics of their job runs are gathered together and combined with the information extracted from the middleware. GriF was originally designed to help the users of the COMPCHEM VO [4] to find the section of the Grid best suited for running their jobs and, therefore, to facilitate massive calculations on the Grid. For this reason, it has been proposed among the tools for aggregating different VOs into a Chemistry and Molecular Innovation Science and Technology (CheMIST) VRC (as it has been described in the recent homonymous proposal to the FP7-INFRASTRUCTURES-2011-2 (1.2.1: e-Science environments) [5]).

Another key feature of GriF is, however, the possibility of collecting information useful to evaluate the Quality of Grid activities. This has encouraged us to utilize GriF to evaluate both the Quality of Users (QoU) and the Quality of Services (QoS) of Grid organizations and to use these information to reward the work done by the members of a VRC and enhance its Sustainability.

B. Murgante et al. (Eds.): ICCSA 2011, Part III, LNCS 6784, pp. 397–411, 2011.

Aim of this paper is, therefore, to illustrate for the first time the structure of GCreS and to present its prototype implementation.

Accordingly, the paper is articulated as follows: in section 2 the main characteristics of the Grid Framework GriF and its relevance to Quality Evaluation are described; in section 3 an overview of the main parameters used for the Quality Evaluation is given; in section 4 the Grid Credit System is illustrated; in section 5 and 6 a first formulation of the QoS and of the QoU is proposed; in section 7 an example of their application is presented. Conclusions and directions for future work are outlined in section 8.

2 The Grid Framework GriF and Quality Evaluation

The basic goal of GriF is to provide Grid users with a user friendly tool allowing them to exploit the innovative features of Grid computing with no need for mastering the low-level Grid environment. This means that there is no need for using specific Grid operating system dependent commands, as for example to establish links to the Grid Proxies (and/or to the Grid Certificates) and to manage all the other operations (as, for example, running Grid jobs, checking their status and retrieving related results from the Grid middleware) as well. In other words, GriF makes Grid applications black-box like pushing the Grid Computing to a higher level of transparency. This makes GriF a tool of extreme importance for enhancing VRC activities. Its utilization, in fact, leads to better memory usage, reduced cpu and wall times consumption as well as to an optimized distribution of tasks over the Grid platform.

For example, one of the most complex and important feature implemented by GriF is called 'Ranking'. In particular, the term 'Ranking' defines the ability of GriF to evaluate, by making use of adaptive algorithms already described in detail in ref. [6], the Quality of a Computing Element (CE) queue (considering several different variables, like for example the performance, the latency and the Grid Ranking) of a VRC running Grid jobs.

GriF also offers to Grid developers new interesting perspectives like those associated with *Service Selection* [7] (rather than pure Service Discovery) and, as already mentioned, with the evaluation of the activities carried out on the Grid. More than that, GriF is open to an user-side usage allowing so far the management of a domain-specific operation logic. This has a clear value for the development of new Workflow Design and Service Orchestration advanced features and the establishing of collaborative operational modalities in which users and providers collaborate.

Therefore, GriF leads to a more efficient exploitation of the innovative features of the Grid when building applications of higher level of complexity and workflows. GriF is, in fact, as already mentioned a Java-based SOA Grid Framework aimed at running on the EGI Grid (supporting the gLite middleware [8]) multi-purpose applications. The main purpose of a SOA and Web Service approach is to provide some functionalities implemented by a user (which could be an institution, an organization as well as a person) on behalf of all the other users. Thanks to its SOA Framework nature, in fact, GriF can support collaboration among researchers bearing complementary expertise. Because of this, GriF is enabled to articulate the computational application as a set of sequential, concurrent or alternative Grid Services by exploiting the features of SOA. Its organization (described in detail elsewhere [9, 10, 11]) allows the adoption of common standards, friendliness and ability in efficiently tracking user activities.

Another important feature of GriF relevant to the objectives of our work is the fact that it allows, at the same time, the monitoring of the activities of the users related to the utilization on the Grid of Web and Grid Services and the evaluation of the internal and external life of a VRC. In particular, for example, GriF allows to make a new distinction between the user providing the appropriate software to implement a particular service (also considered as Provider) and the user wishing to make use of a provider's Web Service (also considered as Consumer).

The information collected provide also useful indications on the behavior of the user, the paths he/she preferentially follows and enable as well to develop semantic inferences out of Grid activities. As a matter of fact, more objective an subjective information can be derived to further specify the user profile and his/her levels of trust and reputation.

These features of GriF represent an important contribution to the robustness of a VRC and to the development of a Grid Economy Model. GriF offers, in fact, the basis on which the members of a VRC could be awarded Credits (also called "terms of exchange credits", or *toecs* [12]) for the activities carried out or the resources made available on behalf of the VRC (*toecs* can be redeemed either by acquiring services or by trading them for financial resources).

This has prompted the singling out of the parameters necessary for evaluating both the QoS achieved in a Grid Services-based producing modality and the QoU associated with the users running on the Grid.

3 The Parameters for a VRC Quality Evaluation

In order to quantify QoS and QoU we have identified appropriate sets of parameters characterizing a Service and a User.

QoS relies on a whole range of parameters evaluated on the ability of a Service to match the needs of Grid users with those of the Service Providers in a competition for available Grid resources.

In this paper, therefore, by QoS we mean non-functional properties of Web Services such as [13]:

- *Availability* (S_{ava}): whether the Web Service is present or ready for immediate use. A useful parameter associated with Availability is the Time-To-Repair (TTR). TTR represents the time Web Service Providers takes to repair a Web Service that has failed. If there are no errors TTR can be considered equal to 0;
- *Accessibility* (S_{acc}): the capability of satisfying a Web Service request. There could be situations in which a Web Service is available but unaccessible. A

low value of the sum of TTRs associated with each error divided the total number of successful operations represents a good estimate of accessibility. High-accessibility of Web Services can be achieved by building highly scalable systems. Scalability refers to the ability to consistently serve the requests despite variations in their volume;
- *Integrity* (S_{int}): how much the Web Service preserves the correctness of the interaction with respect to the source. Proper execution of Web Service transactions provides the correctness of interaction. A transaction refers to a sequence of activities to be treated as a single unit of work. When a transaction does not complete, a rollback procedure is required to cancel all the partial operations already performed;
- *Performance* (S_{per}): a combination of throughput and latency. Throughput represents the number of Web Service requests served at a given time period. Latency is the round-trip time between sending a request and receiving the response;
- *Reliability* (S_{rel}): the extent of preservation of the Web Service and its Quality. The complement to the number of failures can suitably represent a measure of the reliability of a Web Service;
- *Regulatory* (S_{reg}): the scalability of the Web Service with regard to rules and laws, its compliance with Official Standards (OS)s, established Service Level Agreements (SLA)s and documentation. Strict adherence to OSs and SLAs by Web Service Providers is of fundamental importance for a proper use of Web Services by users;
- *Security* (S_{sec}): the extent of confidentiality and non-repudiation by authenticating the parties involved, encrypting messages, managing access control and authorization.

In the rest of the paper we shall consider as a Grid Service any set of collaborative Web Services of GriF running on the Grid by sharing a common distributed goal and we shall apply to them the QoS parameters mentioned above.

As to QoU, we refer in this paper to the collection and filtering of different implicit and explicit information provided by users. In this respect, Active Filtering (AF) is the method that is based on the fact that users want to share information with other peers. This method has become increasingly popular in recent years due to an ever growing base of information available on the Web. In AF users can send their feedbacks (e.g. users satisfaction) over the Grid where others can access them and use the ratings of the Grid Services to make their own decisions. There are various advantages in using AF among which the fact that the rating is given by an interested user who has actually used the considered Grid Service. This corroborates the credibility of this type of ranking. Another advantage of using AF is the fact that the users explicitly want to (and ultimately do) provide information regarding the matter dealt. There are some disadvantages in using AF, though. One is that the opinions expressed might be biased. Another is that fewer feedbacks are obtained than when using passive approaches because the act of providing feedbacks requires explicit action by the user. On the contrary, Passive Filtering (PF) collects information implicitly without involving the direct input of opinion by the users whose evaluation is instead deducted by their actions. This reduces the variability of the opinions and the pressure exerted on the users leading to more natural outcomes. These implicit filters are therefore aimed determining which are the real wishes of the user and the applications of potential interest for him/her because they rely on all the actions undertaken by users and recorded for further utilization. An important feature of PF is the use of time to determine when a user is running a program and the evaluation of semantic aspects to single out both the high-value execution paths (Execution Path Similarity) and the "goodness" of the experiments. The major strength of PF is that it does not include the analysis of certain variables that would normally be considered in AF. For example, in AF only certain types of users will take the time to rank a Grid Service while in PF anyone accessing the system would automatically do it.

PF indicators can be used to help the previously mentioned GCreS tool to evaluate user activities. For example, it could be useful to determine the ratio of the number of compilations over the number of runs, the ratio of the number of compilations over the number of successful results and the ratio of the number of successfully results over the number of runs as well as the correct utilization of the Grid middleware itself. In this respect, a preliminary classification of the users based on the different characteristics of their compiling and running activities (performed on the section of the Grid available to the COMPCHEM VO) has been already made in ref. [11] by identifying four different user profiles: the Passive User (PU), the Active User (AU), the Service Provider (SP) and the Software Developer (SD).

4 The Grid Credit System GCreS

GCreS exploits the above mentioned concepts to foster the Sustainability of VRCs by equipping them with mechanisms suited to evaluate the commitment of the users to their organizations and to reward them appropriately for the associated work. To this end, as already mentioned, GCreS was designed as a prototype tool based on GriF operating in a standard Grid scenario.

Basically, in GCreS users are rewarded for the work done on behalf of an organization by being assigned a certain amount of Credits to be redeemed via a preferential utilization of the resources (including the financial ones). Such development, in addition to leveraging on collaboration, stimulates also a certain extent of competition among the members of a VRC to produce innovative Grid Services and improve the existing ones as well. More specifically, this means also that on top of QoS and QoU evaluations, new higher level ways of managing Grid Services can be adopted:

1. by *Users*, which will be able to ask for Grid Services by specifying as keywords high-level capabilities rather than memory size, cpu/wall time and storage capacity;
2. by *GriF*, which will be able to automatically select the most appropriate low-level capabilities related to the current Grid job then enabling different

running policies (in other words, when a Grid job has to be run, GriF can make use of different system requirements in terms of memory size, cpu/wall time and storage capacity according to the class level of the related user owning the Grid job).

To this end, GCreS has been structured as a 3-Tier Architecture [14] based on a Back-end, on a Business Logic and on a Presentation layer (see left hand side of Fig. 1), respectively.

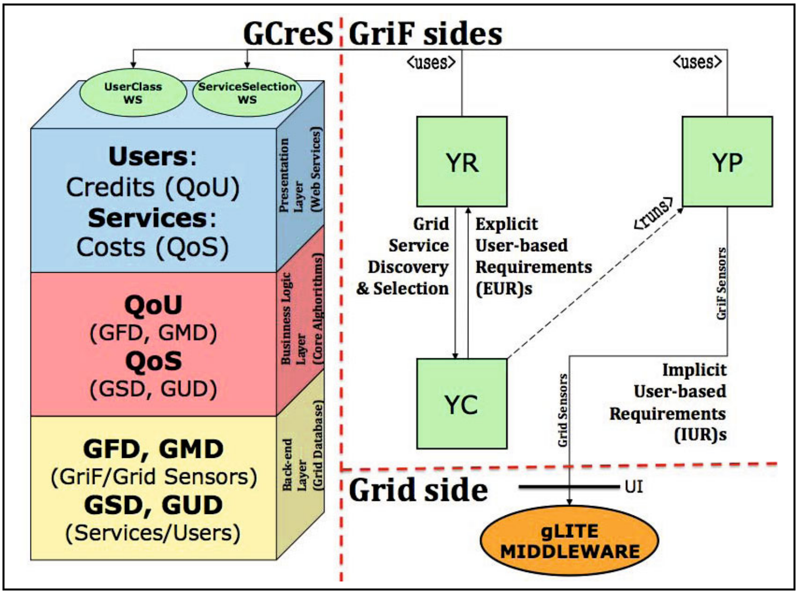

Fig. 1. GCreS and GriF Communications

The Back-end layer is devoted to the collection in a global database of all the data related to the available information on users and Grid Services. In particular, GCreS collects four types of data:

– *Grid Framework Data* (GFD), in which all the information saved in GriF (as for example the number of run and the success/failure ratio, the name of the related application and its run modality, the number of compilations performed as well as information related to the operations time (aimed at identifying execution path similarities discussed above in order to produce semantic inferences on the users behavior)), can be recorded and analyzed per user;
– *Grid Middleware Data* (GMD), in which accounting information related to the operations time, to the amount of memory consumed, to the cpu time elapsed, to the CE queue used and to the number of jobs run registered by the Grid Middleware, can be recorded (in order to integrate that of GFD) per VRC and/or per user;

- *Grid Service Data* (GSD), in which QoS parameters can be calculated and saved per Grid Service;
- *Grid User Data* (GUD), in which users feedbacks and profiles can be organized.

To collect GFD, use can be made of various off-line scripting procedures which will be responsible for the handling of the information from GriF to GCreS. Moreover, GCreS will be also able to access other different Web Services-based GFD external sources (like for example GEMSTONE [15]). To collect GMD, use can be made of various sources (like for example DGAS [16]). To collect GSD, a separate application producing detailed measurements of the QoS evaluation parameters for each Grid Service, has to be implemented. To collect GUD, at present the quite crude method of exporting the user profiles and feedbacks already recorded by GriF into the GCreS database could be used (although a dedicated procedure (or a set of them) should be better implemented in order to improve the already existing collaborative part of GriF).

The Business Logic layer (that will be fully-implemented in the final version of GCreS) paves the way to the definition of the Credits/Costs schema (then consumed by the Presentation upper layer) by implementing the core algorithms producing the QoS (for which a first formulation will be given in the next section essentially as a function of GSD and of GUD) and the QoU (for which a first formulation will be given later on essentially as a function of GFD and of GMD though, sometimes, also GUD could be used especially when self-evaluation is implicitly considered).

The Presentation layer, as typical of Business-to-Business relationships, in addition to reward users in terms of *Credits* on the basis of their calculated QoU and to produce *Costs* for Grid Services on the basis of their calculated QoS as well, is intended to interact with GriF (see upper right hand side of Fig. 1) exposing two Web Services (called `UserClass` and `ServiceSelection`, respectively) that can be consumed by:

- *GriF Providers* (originally called (YP)s in ref. [9]), which before running a job on the Grid can automatically add to their running policy specific requirements derived from the related user class level provided by the `UserClass` Web Service (on the basis of the QoU of the user owning the Grid job);
- *GriF Registries* (originally called (YR)s in ref. [9]), which before returning the YPs list matching the request for an application (made by users) can access each related Grid Service QoS (provided by the `ServiceSelection` Web Service) to return a ranked list of the hosting YPs (as is typical of the above mentioned *Service Selection* concept) instead of the unranked one (as is typical of Service Discovery).

Accordingly, we have defined, respectively, *Implicit User-based Requirements* (IUR)s the low-level requirements automatically generated by YPs during their GriF running phase (from this point onwards IURs can be adopted transparently for users allowing YPs to use them in order to run jobs on the Grid with different priority) and *Explicit User-based Requirements* (EUR)s the high-level requirements (e.g. reliability and/or performance rather than machine parameters) which can be requested by users to YR during each (Grid) Service Discovery and/or Selection phase.

5 The QoS formulation

The implementation of the above described GCreS function transforming the available information (including the QoS and QoU evaluations) into *Costs* to be paid by users for the utilization of Grid Services and *Credits* to be awarded to them users for the work provided, prompts a detailed definition of its tasks and of QoS and QoU as well.

In case of *Costs* to be assigned to services, for example, the concept of Service Discovery (in which when searching for Grid Services users receive back an unranked list of matchings) is better replaced by the already mentioned *Service Selection* in which the ranking is provided by QoS. In the case of *Credits* to be awarded to users, a careful definition of QoU is instead needed.

As a matter of fact, QoS has recently become a significant factor in ranking the success of Grid Service Providers and plays a crucial role in estimating the Grid usability because applications with very different characteristics and requirements compete for heterogeneous resources. At the same time, thanks to the adoption of standards like SOAP [17], UDDI [18] and WSDL [19] by all major Web Service players, a whole range of Web Services are being currently deployed for various purposes. Moreover, as most of the Web Services are increasingly required to comply with standards, QoS is going to become a necessary label and an important differentiation parameter.

In the (first) formulation of QoS proposed for COMPCHEM we have developed a tentative quantification of the following (already described) QoS parameters: *Accessibility*, *Integrity*, *Reliability*, *Security* and *Performance*. The overall **QoS** value is expressed, accordingly, by the following expression:

$$QoS = w_0 S_{acc} + w_1 S_{int} + w_2 S_{rel} + w_3 S_{sec} + w_4 S_{per} \tag{1}$$

where $w_{(i=0..4)}$ are the weights that a Quality Manager (QuM) or a Quality Board (QuB) of a VRC chooses for each QoS parameter while the various S parameters are defined as follows:

$$S_{acc} = 1 - \frac{\sum_{i=1}^{N_e} TTR_i}{N_f} \quad , \quad N_f \neq 0 \tag{2}$$

where N_f is the number of functions (f)s (each f corresponds to a stable Web Service call or a set of them) invoked by a Grid Service, TTR_i is the Time-To-Repair associated with each error occurred with N_e being the number of errors occurred in the time interval considered. Due to the lack of previous data to refer to, in our study N_e was determined by running a dedicated test program checking the accessibility of each function of the Grid Service every X minutes (TTR starts from 0 and is incremented by 1 at every failure reported by the checking procedure related to the same error). In our study we also found it convenient to choose the value X on the basis of the SLA of the Grid Service by taking $X < 5'$ for the low values and $X > 60'$ for the high values, respectively, for Grid applications requiring high availability and for applications accepting a reasonable amount of downtime;

$$S_{int} = k_{int} \tag{3}$$

where k_{int} is a constant indicating rollback absent (value 0), partially implemented (not fully-tested in production, value 0.5) or fully-operating (value 1), respectively;

$$S_{rel} = 1 - \frac{N_e}{N_f} \quad , \quad N_f \neq 0 \tag{4}$$

$$S_{sec} = (k_{en} + 2k_{ae} + \frac{4k_{ao}}{2^{k_{ae}}})/5 \tag{5}$$

where k_{en}, k_{ae} and k_{ao} are three constants (of value either 0 or 1) indicating if encryption, authentication and authorization are supported or not, respectively. Accordingly, $k_{ao} = 1 \Rightarrow k_{ae} = 1$. Moreover, k_{ae} has to be supported for each Grid Service (otherwise, when only encryption is enabled, the Grid Service is assumed as un-secure and it should not be released to the public);

$$S_{per} = \frac{TR - \alpha}{LT - \beta} \quad , \quad LT \neq \beta \tag{6}$$

where TR (the Throughput) can be assumed as the number of times that a Grid Service has been consumed in a time interval while LT (the Latency) evaluates the way it is conveyed. LT is most often quantified as the delay (the difference between the expected and the actual time of arrival) of the data. It has to be considered, however, that a YP with high average TR and LT can be worse for some applications (and their Grid Services) than the one with low average TR and LT. For this reason, the α and β coefficients can be used by the QuM to favor one aspect or the another by properly shifting the related scales. In Eq. 6:

$$TR = N_t \quad , \quad N_t \in \Delta_t \tag{7}$$

where Δ_t is the time interval considered, and:

$$LT = \frac{\sum_{i=1}^{N_f}(t_i - k_i)}{N_f} \quad , \quad N_f \neq 0 \in \Delta_t \tag{8}$$

where t_i is the time (say in seconds) elapsed by each f (e.g. the retrieve of the results) and k_i is the associated time constant indicating the optimal time (the same unit as t_i) for it (also depending, for some fs, by the length of the files involved). For example, we have identified 10 different fs for GP (by making use of a special variable distinguishing each f in a given Grid Service and also reporting the related elapsed time) for which the corresponding k are valued[1]: 0.006, 1.1, 0.005, 6 + 1.1 for each MB input used (or 31 + 1.1 where the Ranking feature of GriF mentioned above is not selected by a user), 0.81, 0.008, 0.007, 1.3, 2.8 and 0.385 seconds. It is worth noticing here that in order to obtain a realistic evaluation of S_{per} (and also of QoS) for a Grid Service it is necessary to apply it to a real production environment (for example more than one Grid Service hosted for each YP, several users accessing GriF and large amounts of Grid job runs) and to develop as well a dedicated program properly manipulating all the information saved in the default log files (otherwise S_{per} is *Not Applicable* and therefore taken as null).

After calculating the **QoS** of a given Grid Service, a *Cost* (and then a corresponding position in the ranked list of available Grid Services) is determined for it by the `ServiceSelection` Web Service. It is worth pointing out here that when GUD will be fully-available, the global QoS as well as each (objective) QoS evaluation parameter will be improved and refined with the (subjective) information provided by users (e.g. their feedbacks) also considering their QoU.

6 The QoU Formulation

In the first formulation of QoU, we have produced some custom indicators useful for all types of users and applications run by a specific Grid Service. In particular, some of them are also useful to corroborate the singling out of the four user profiles PU, AU, SP and SD mentioned in section 3. For example, when considering the percentage $P_{c,x}$ of runs executed as 'CUSTOM' (e.g. applications uploaded for running by users) rather than as 'STANDARD' ($1 - P_{c,x}$, e.g. applications made already available by the Grid Service considered), the $P_{c,x}$ is proportionally larger for SDs, SPs and AUs than for PUs (which typically run on the Grid platform a stable version of an application and, therefore, do not need to compile every time they execute) because, in general, 'to compile' tends to coincide with 'to deal with custom forms'. This confirms that a $P_{c,x}$ value close to 1 is more likely for PUs than for the other three profiles.

For the overall **QoU** evaluation we have adopted the following expression:

$$QoU = p_0 w_5 U_{cx} + p_1 w_6 U_{cu} + p_2 w_7 U_{cm} + w_8 U_{ge} + w_9 U_{fb} \tag{9}$$

In Eq. 9 $p_{(i=0..2)}$ are three coefficients (with possible values of 0.1, 1 and 10)[2] weighing the contribution of the different user profiles and valued as shown in Table 1. In the same equation $w_{(i=6..10)}$ are weights chosen by either the QuM or QuB for each QoU parameter (in addition to U_{cx} corresponding to the ratio between the number of compilations (N_c) and the number of executions (N_x) performed by a user already studied in ref. [6] for COMPCHEM), respectively, as follows:

[1] The values of `k` have been calculated in ref. [6] by averaging the elapsed time by each `f` during six months of activities.

[2] In this respect, it is worth noticing here that the resulting values for U_{cx}, U_{cu} and U_{cm} have different meanings depending by the profile of the user considered while those for U_{ge} and U_{fb} can be assumed to be independent of it.

Table 1. Coefficients of the U_{cx}, U_{cu} and U_{cm} parameters for the various COMPCHEM user profiles

User Profiles	p_0	p_1	p_2
Passive User (PU)	10	0.1	0.1
Active User (AU)	1	0.1	0.1
Service Provider (SP)	0.1	1	1
Software Developer (SD)	0.1	10	10

$$U_{cu} = P_{c,x}(N_b * R_{ra,N_b} + (N_x - N_b) * R_{ra,N_{(x-b)}}) \tag{10}$$

where R_{ra,N_b} and $R_{ra,N_{(x-b)}}$ are ratios between the number of results retrieved and the number of results available respectively related to the number of custom runs (N_b) and to those runs derived from applications already available under the form of Grid Services ($N_x - N_b$). Accordingly, U_{cu} gives additional qualitative information on the quantities N_x and N_c already studied (N_b is, in fact, inversely proportional to N_c). Moreover, U_{cu} favors the 'CUSTOM' running modality mentioned above since the $P_{c,x}$ is applied;

$$U_{cm} = N_r * (\psi\overline{C} + \omega\overline{M}) \tag{11}$$

where N_r is the number of results retrieved from the Grid by a user, ψ and ω are normalization coefficients, $\overline{C}$ and $\overline{M}$ are, respectively, the average cpu time (in hours) and the average virtual memory amount (in GB) consumed per "Retrieved" job by a user (in this respect, in order to integrate the quantitative evaluation of GFD one can also take into account GMD);

$$U_{ge} = GE_{user} \tag{12}$$

where GE_{user} is the Grid Efficiency GE[3] applied to a specific *user* in order to refine the QoU formulation);

$$U_{fb} = N_m \tag{14}$$

where N_m is the number of messages (e.g. feedbacks) produced by a user.

After calculating the total **QoU** (obtained by adding up the QoU related to each Grid Service used) for a user, *Credits* corresponding to the class level of the type *Low*, *Medium* or *High* are determined using the `UserClass` Web Service. Moreover, as mentioned in the case of the QoS formulation, also QoU will be refined when GUD will be fully-available on the basis of users self-evaluations.

[3] GE can be generically defined as follows:

$$GE = \frac{\sum ct}{\sum wt} \tag{13}$$

where ct and wt are the elapsed cpu and wall time, respectively. Accordingly, GE can be used in evaluating users (in this case ct and wt will be related to their Grid jobs), CE queues (in this case ct and wt will be related to the Grid jobs run on each of them) as well as to the Grid middleware itself (in this case all the jobs run on the Grid will be considered).

7 A Preliminary Benchmark

A first trial simulation of a VRC Sustainability based on Quality Evaluation (applied to a VO in this case) has been carried out by considering different periods of GriF activities (one month for the QoS example and three months for the QoU one) when offering two Grid Services (called ABC [20] and GP [6], respectively) operating within the CEs belonging to the COMPCHEM VO. Accordingly, we have collected GFD, GMD and GSD for both ABC and GP as well as for the COMPCHEM users. At the same time, partial GUD (only related to some aspects of QoU) were available. Our goal here is to illustrate the initial evaluation of the QoS for the GP Grid Service and of the QoU for a COMPCHEM PU named *sylvain* (who typically runs applications dealing with chemical processes in order to carry out realistic a priori simulations) with respect to both ABC and GP.

In the case of the QoS evaluation we have measured the above formulated parameters for GP during the month of September 2010 (in production state) without considering the management activity of the related SP user (that is expert in the handling of GP) in order to attempt an evaluation of the meaning of the reported values.

In this way we have obtained N_f = 194 and N_e = 3 (for which $TTR_{(i=1..3)}$ = (2, 0, 3), respectively). Moreover, by applying Eq. 1–8 (and choosing $w_{(i=1..4)}$ = 1 for all the QoS parameters and $\alpha = \beta = 0$) to the GSD for the GP Grid Service, a QoS value of 3.4 was obtained by the QuM (see Table 2 for details).

Table 2. An example of QoS evaluation for a Grid Service

	Grid Service: *GP*
S_{acc}	0.9742
S_{int}	1
S_{rel}	0.9845
S_{sec}	0.4
S_{per}	*Not Applicable*
QoS	**3.3587**

It has to be commented here, however, that one will be able to evaluate a truly reliable *Cost* for the Grid Service *GP* considered in this example only after comparing the resulting QoS value with those of other Grid Services of the same type (although a limiting value can be given for it as suggested by the maximum value for S_{acc}, S_{int}, S_{rel} and S_{sec}).

In the case of the QoU evaluation we have measured the above formulated parameters for the COMPCHEM PU *sylvain* during the considered three months of his activity in Grid (from August 2010 to October 2010) carried out using both ABC and GP Grid Services (summing up the related values).

To this end, by choosing $p_0 = 10$ (indeed, low values of U_{cx} are more likely for PUs because they compile less than other users), $p_1 = 0.1$ and $p_2 = 0.1$ (indeed, high values of both U_{cu} and U_{cm} are more likely for PUs and AUs because they run on the Grid more than SPs and SDs) according to Table 1, we have obtained, in total for both the Grid Services mentioned above, $N_c = 10$, $N_x = 167$, $P_{c,x} = 88.24\%$, $N_b = 162$, $R_{ra,N_b} = 0.9815$ (having 3 not-retrieved r), $R_{ra,N_{(x-b)}} = 0.8$ (having 1 not-retrieved r), $N_r = 163$, $\overline{C} = 1.2995$, $\overline{M} = 0.2873$, $GE_{sylvain} = 0.8449$ and $N_m = 12$. Moreover, by applying Eq. 9–14 (and choosing $w_5 = 0.5$ for U_{cx} because one (the GP) of the two Grid Service has the compilation function not yet completely stable resulting in a number of compilations performed by VO users theoretically lower, $w_{(i=6..9)} = 1$ for all the other QoU parameters as well as $\psi = \omega = 1$) to both the GFD and GMD (and also to a reduced set of the GUD) for the VO user mentioned above, a QoU value of 53.4 was obtained by the QuM (see Table 3 for details).

Table 3. An example of QoU evaluation for a COMPCHEM User

	COMPCHEM user: ***sylvain***
U_{cx}	0.0598
U_{cu}	143.8338
U_{cm}	258.6484
U_{ge}	0.8449
U_{fb}	12
QoU	**53.3921**

Finally, after comparing the obtained QoU value with those of other COMPCHEM users of the same type (even if in the presence of an inadequacy of the GUD estimate), we have been able to assign a *High* class level to the *sylvain* PU considered in this example for the corresponding *Credits* award.

8 Conclusions and Future Work

In the present paper the possibility of using the recently proposed Grid Framework GriF to facilitate Grid empowered calculations useful for the scientific advances of a Virtual Research Community is illustrated together with the evaluation of the parameters defining in a quantitative way the Quality of its Users and Services.

To this end, the progress made by the COMPCHEM VO of the Chemistry and Molecular Innovation Science and Technology (CheMIST) community to make Grid applications truly user friendly and composable for the assemblage of more complex computational procedures, have been considered. As a result, not only it has been possible to carry out on the Grid massive computational campaigns by spending a minimum effort and achieving maximum throughput but it has been also possible to profile some types of users. To this end, a quantitative definition of the parameters like *Availability*, *Accessibility*, *Integrity*, *Performance*, *Reliability*, *Regulatory* and *Security* have been provided for Grid Services and four types of user behavior have been singled out with respect to their working habits within the Grid.

This still provisory definition of parameters and classifications has been, in any case, of particular importance for the profiling of the applications and of the users so as to be taken as a basis for the evaluation of the work carried out in a VRC. As a result, we have been able to quantify, for the bench case described in the present paper, the QoS and QoU indicators and to formulate as well the award of *Credits* through the new Grid Credit System GCreS.

The work described here represents, therefore, the ground on which CheMIST will build an internal system of rewarding its members for the work done for the community. Moreover, it will also represent the ground on which the work for ensuring Sustainability and performing a global Quality Evaluation of a VRC in Grid (as for example in order to compete for funding within the EGI project) will be based.

This will need further developing of new quality evaluators called by us Quality of Computing (QoC) and Quality of Provider (QoP). QoC consists of an evaluation of computing-related objects of the Grid Middleware belonging to a VRC (e.g. User Interfaces (UI)s, Storage Elements (SE)s, CEs and batch systems). To this end, for example, the concept of Ranking mentioned in section 2 can be applied to a gLite CE queue (or a set of them). Moreover, also Eq. 13 can be used. QoP consists of an average evaluation of Hardware Providers offering Grid Services belonging to a VRC (the YPs). To this end, for example, YPs hardware and network characteristics should be considered. Moreover, also the two general QoS parameters (introduced in section 3 but never used), respectively called *Availability* (S_{ava}) and *Regulatory* (S_{reg}), can be applied. Accordingly, a first definition of the overall **Quality-to-Community** (*Q2C*) in Grid is proposed here to be formulated as follows:

$$Q2C = QoC \cup QoP \cup QoS \cup QoU \tag{15}$$

where QoS and QoU are, in this case, the sum of the QoS for each Grid Service and the sum of each VRC user QoU, respectively.

Acknowledgments

The authors acknowledge financial support from the EGI-Inspire project (contract 261323), the MIUR PRIN 2008 (contract 2008KJX4SN_003), the ESA-ESTEC contract 21790/08/NL/HE, the Phys4entry (Planetary Entry Integrated Models) FP7/2007-2013 project (contract 242311), the Eurodoctorate (A Framework for a Third Cycle Qualification in Chemistry, 177048-LLP-1-2010-1-GR-ERASMUS-EAM), the EChemTC-Valorisation of EChemTest Testing Centres (504854-LLP-2009-GR-KA4-KA4MP) and ARPA. Computer time allocation has been obtained through the COMPCHEM VO of EGI.

References

1. European Grid Infrastructure (EGI) (April 2, 2011), http://www.egi.eu/
2. GriF: The Grid Framework (April 2, 2011), http://www.hpc.unipg.it/grif/

3. W3C Working Group, Web Services Architecture (2004), http://www.w3.org/TR/ws-arch/ (April 2, 2011)
4. Laganá, A., Riganelli, A., Gervasi, O.: On the Structuring of the Computational Chemistry Virtual Organization COMPCHEM. In: Gavrilova, M.L., Gervasi, O., Kumar, V., Tan, C.J.K., Taniar, D., Laganá, A., Mun, Y., Choo, H. (eds.) ICCSA 2006. LNCS, vol. 3980, pp. 665–674. Springer, Heidelberg (2006)
5. Work Programme 2011 - Part1: Research Infrastructures (April 2, 2011), http://cordis.europa.eu/fp7/ict/e-infrastructure/docs/wp2011.pdf+
6. Manuali, C.: A Grid Knowledge Management System aimed at Virtual Research Communities Sustainability based on Quality Evaluation, Ph.D. Thesis, Department of Mathematics and Informatics, University of Perugia (IT) (Febraury 14, 2011), http://www.unipg.it/carlo/PhD_Thesis.pdf (April 2, 2011)
7. Karta, K.: An Investigation on Personalized Collaborative Filtering for Web Service, Honours Programme of the School of Computer Science and Software Engineering, University of Western Australia (2005)
8. The gLite middleware (April 2, 2011), http://glite.cern.ch/
9. Manuali, C., Laganà, A.: GriF: A New Collaborative Grid Framework for SSCs. In: Proceedings of Cracow Grid Workshop (CGW 2009), pp. 188–195 (2010) ISBN 9788361433019
10. Manuali, C., Laganà, A., Rampino, S.: GriF: A Grid Framework for a Web Service Approach to Reactive Scattering. Computer Physics Communications 181, 1179–1185 (2010)
11. Manuali, C., Laganà, A.: GriF: A New Collaborative Framework for a Web Service Approach to Grid Empowered Calculations. Future Generation Computer Systems 27(3), 315–318 (2011)
12. Laganà, A., Riganelli, A., Manuali, C., Faginas Lago, N., Gervasi, O., Crocchianti, S., Schanze, S.: From Computer Assisted to Grid Empowered Teaching and Learning Activities in Higher Level Chemistry. In: Innovative Methods of Teaching and Learning Chemistry in Higher Education, pp. 153–189. RSC Publishing (2009) ISBN 9781847559586
13. Mani, A., Nagarajan, A.: Understanding Quality of Service for Web Services. Improving the performance of your Web services (2002), http://www.ibm.com/developerworks/webservices/library/ws-quality.html (April 2, 2011)
14. Erl, T.: Service-Oriented Architecture (SOA): Concepts, Technology, and Design. Prentice Hall, Englewood Cliffs (2008) ISBN 0131858580
15. Grid Enabled Molecular Science Through Online Networked Environments (GEMSTONE) (April 2, 2011),http://gemstone.mozdev.org/
16. DGAS: Distributed Grid Accounting System (April 2, 2011), http://www.to.infn.it/dgas/
17. Simple Object Access Protocol (SOAP) 1.2, http://www.w3.org/TR/soap/
18. Universal Description, Discovery and Integration (UDDI) 3.0.2 (2005), http://www.oasis-open.org/specs/ (April 2, 2011)
19. Web Service Description Language (WSDL) 1.1, http://www.w3.org/TR/wsdl/
20. Skouteris, D., Castillo, J.F., Manolopulos, D.E.: ABC: a quantum reactive scattering program. Computer Physics Communications 133, 128–135 (2000)