

Achieving Team-awareness in Scientific Grid Environments

Christoph Dorn, Daniel Schall, Schahram Dustdar
Distributed Systems Group
Vienna University of Technology
1040 Vienna, Austria
dorn | schall | dustdar@infosys.tuwien.ac.at

Abstract

The vision of the Scientific Web is to enable research scientists to collaborate in a ubiquitous manner; sharing resources, data, and insights how to perform experiments. The grid provides the underlying computing infrastructure, thereby enabling workload sharing and distribution of computational tasks among millions of nodes. However, current collaboration tools in scientific grid environments do not support scientists in managing various resources, artifacts, data, experiments in a team-aware manner. In this paper we introduce an approach to enhance current grid infrastructures with awareness of team interactions. Our proposed activity model allows users to manage activities, involved team members and their expertise, as well as associated grid resources. Based on the action data captured from various sources, we establish a bipartite activity-user graph — central to team-centric collaboration metrics. We discuss the applicability of these metrics in grid environments by introducing novel adaptation algorithms enabling team-aware resource utilization.

1 Introduction

While recent advances in the domain of Grid Computing improve usability and performance of data and computationally intense systems, the interaction aspects on the human side remain largely neglected. Figure 1 depicts the gap between scientific (human) collaboration and the infrastructure in the Scientific Web. However, the Collaborative Working Environment (CWE) community focuses on improving member interaction, whereas the Grid community focuses on computing centric challenges such as performance. In this paper, we motivate and demonstrate the importance of bridging this gap by analyzing inter-human and human-grid relations to improve overall (scientific) grid utilization. Our approach promises to improve collaboration and coordination on the human side, and more effective and

efficient grid utilization on the technical side. Consider the following motivating scenario: A team of bioinformatics researchers distributed across Europe, Asia, and America collaborates on new ways of analyzing protein structures. This involves running a great number of experiments which may take up to several days or weeks. Part of the scientists' work consists of searching and selecting workflows to be used in experiments, evaluating results, optimizing workflows in terms of tuning configuration parameters, branching points, branching conditions, and repeating the experiments. Some workflows require human input for decisions that cannot be made automatically or be anticipated in advance.

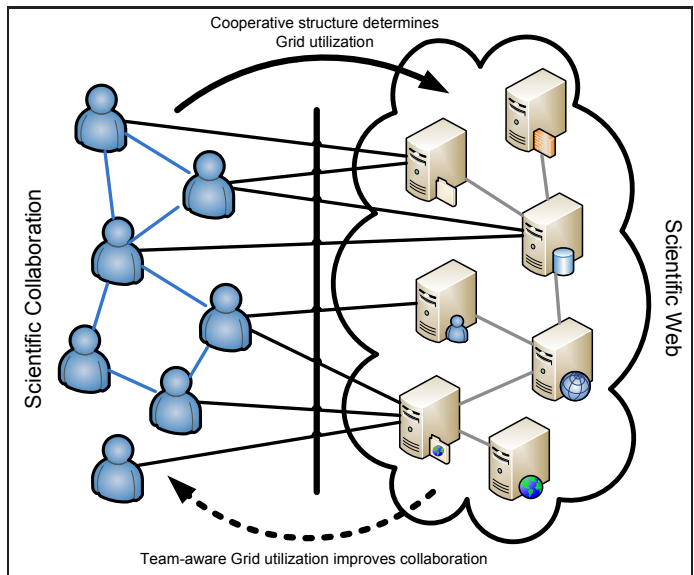


Figure 1. Unified view on human interactions in Scientific Web.

Such distributed research teams face two major challenges. First, coordination and communication between team members relies heavily on email only, sometimes supported by shared document repositories. Information ex-

change about applicable resources such as grid infrastructures, workflow repositories, or evaluation algorithms lacks tool support. Second, current grid environments do not support team-awareness during scientific workflows preparation, job progress monitoring, or experiment result evaluation. Information on who designed or updated a specific workflow instance, who to prompt for run-time input, or who to provide access to results are virtually non existing. Existing Web 2.0 platforms such as myExperiment.org enable publishing and sharing of workflows. However, they cannot provide information within a team on changes implemented by member scientists to fit their specific experiment, and whether that updated workflows actually completed successfully.

Enabling team-aware grid resources reduces the effort for team members to provide their colleagues the relevant information at the right time. Involving the right expert at the right time during an experiment reduces dead time, especially important for experiments on the critical path in a larger research project.

Our salient contributions in this paper are:

- an approach to combine human collaboration and scientific grid computing
- an activity-centric bipartite graph model to integrate interaction sources from various systems
- a novel set of collaboration metrics based on a bipartite graph representation
- exemplary team-aware adaptation algorithms to improve collaboration in scientific grid environments

The remainder of this paper is organized as follows. First, we will present our approach to capturing data describing the relevant aspect of scientific collaboration. Next, we briefly outline the key contributions of our work compared to current research efforts. After describing the underlying data sources and data formats, we continue to introduce our novel set of collaboration metrics. We will then proceed to demonstrate the benefit of these metrics applied to our scenario in more detail. Finally, conclusions and future work complete our paper.

2 An Activity-Based Approach

Team-aware grid utilization requires detailed information about the scientific collaboration structure. The range of relevant input covers global structural information such as how closely are members communicating, to interaction subsets on delegation of tasks and documents down to user-centric information about responsibility of future activities. Especially details on individual members relies on interpretation in the context of the overall research team, for example: we can only correctly interpret the amount of emails sent by a specific member, if we know the activity con-

text of that member together with the overall team activity-communication structure.

Figure 2 shows our approach to achieve team-aware grid utilization. The uptake of service-oriented computing by the Grid community (e.g., Open Grid Services Architecture (OGSA)) and the call for service oriented science [5] justify our assumption that collaboration tools will be increasingly available as services through standardized interfaces. As scientists apply collaboration services in their individual preferred ways—some favor email, other instant messaging—we need to collect data from a heterogenous set of sources (1). These sources also include data on the use of grid resources.

From this set of raw data, we extract the activity structure underlying the collaborative efforts (2). The resulting bipartite graph describes the relations between team members (including applied resources) and their joint activities.

In the subsequent step, we determine the set of novel collaboration metrics (3) to establish the nature of the research team.

Finally, combining collaboration metrics and individual activity data enables researchers to utilize grid resources specific to their team structure (4). Examples include intelligent grid management services that select the most appropriate researcher to provide input to scientific workflows, to notify all relevant members interested in the results upon experiment termination, or update access policies as members join/leave the team.

3 Related Work

Some research over the last years focused on analyzing social structures of email-based interactions [2, 11]. The Email Mining Toolkit (EMT) [13] provides interaction metrics while [4] aims at extracting the underlying activity structure. However all of these lack the notion of applied resources (e.g. documents, (grid) services). In contrast to our approach utilizing a bipartite activity-user graph, social network mining techniques provide metrics based on pure interaction graphs only.

Foster [5] as well as Hey and Trefethen [7] outline the importance of a service-oriented computing approach to science. Bringing together scientific teams and grid through collaboration, computation, and storage services provides great potential for integrating the adaptation of Scientific Collaboration and the Scientific Web.

Adaptation of grid resources is a central concern in many domains. Examples of context-aware grid adaptation such as [6, 15, 16] apply location, time, or data on scheduled jobs, but neglect the interdependencies arising from joint activities. Scientific workflows for grids [3, 8] model dependencies to some extent but cannot be applied in situations that demand for ad-hoc changes. In their survey of the the Open

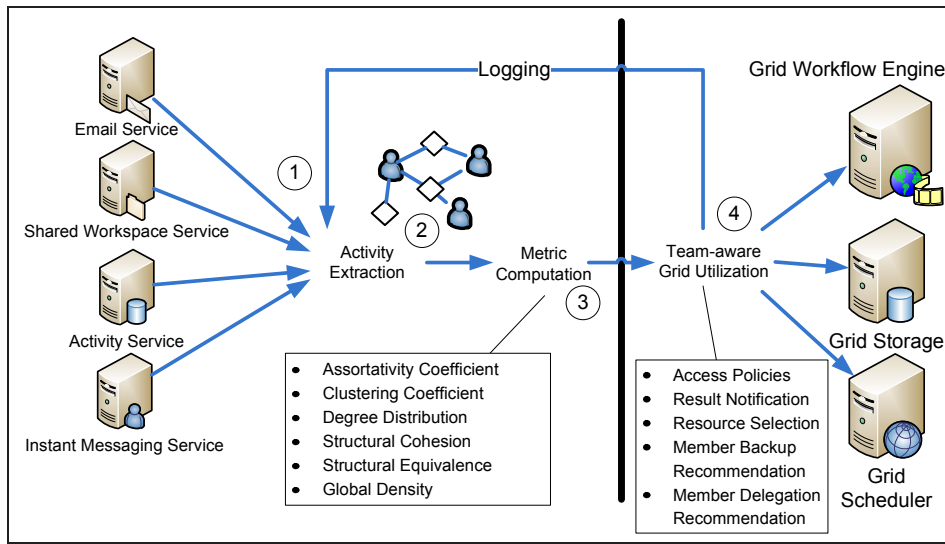


Figure 2. Approach to team-centric grid utilization.

Grid Computing Environment (OGCE), Alameda et al. [1] include also collaboration features. However the OGCE neither includes analysis of the underlying collaboration structure nor enables team-awareness of grid resources.

4 Capturing Collaboration Data

As researchers apply a variety of collaboration tools, we cannot assume to capture all relevant collaboration data from a single source. We selected email, instant messaging, Microsoft Sharepoint, and the Activity service (developed in the scope of our project inContext) as representative sources. In addition, the inContext infrastructure also provides the means to monitor and log arbitrary (grid) services applied during collaboration. Together, these sources cover the significant spectrum of coordinative, communicative, and work aspects in everyday’s team work.

4.1 Raw Logs from Human-Service Interactions

Figure 3 shows a fragment of a logged human-service interaction obtained from the shared-workspace service. Each line in the log file contains (i) a unique id (guid), (ii) timestamp, (iii) server identifier, (iv) top-level site, (v) resource on the server, (vi) user id, and additional information from the requesting client such as browser type. The given example shows a document *upload action*, captured by the `upload` resource identifier in shown in Figure 3. Based on captured raw service logs our system can automatically establish correlations between upload actions and documents that are used in collaborations and involved members, for

example, by establishing the link between sent emails and document URIs.

4.2 Email Mining

Although services play an increasingly important role in collaboration, email will remain one of the core communication means in the foreseeable future. Thus, we extract interaction data not only from well formatted messages sent between collaboration services but also email repositories. In contrast to previous email mining approaches [4, 11] we analyze explicitly attachments as they are representing major collaboration artifacts. We subsequently correlate these attachments to documents uploaded or downloaded from the shared workspace service (here Microsoft Sharepoint).

Emails contain the following significant information: (i) Sender and Receivers: the interaction participants; (ii) Attachments: the applied collaboration resources; (iii) In-Reply-To: to identify threads; (iv) Activity: available for emails sent via the inContext email service; and (v) Subject: to enable activity extraction for emails without an explicit Activity identifier.

4.3 Activity-Event Logs

The inContext system specifies and implements the PCSA (Pervasive Collaboration Services Architecture), which is based on the principles of SOA and Web services. Collaboration services registered in the PCSA are invoked by routing each service request toward an *access layer*, which is a transparent SOAP dispatcher and router. Each service interaction is captured by a logging service that implements – aside from the basic logging features –

```

guid 11:20:17 server incontext wp5/forms/allitems.aspx smoretzky Mozilla/4.0
guid 11:20:23 server incontext wp5/forms/upload.aspx smoretzky Mozilla/4.0
guid 11:21:30 server incontext wp5/t5.3 implementation/design_12.doc smoretzky
guid 11:21:30 server incontext wp5/forms/allitems.aspx smoretzky Mozilla/4.0

```

Figure 3. Example log from user interactions captured from Share-Workspace Service.

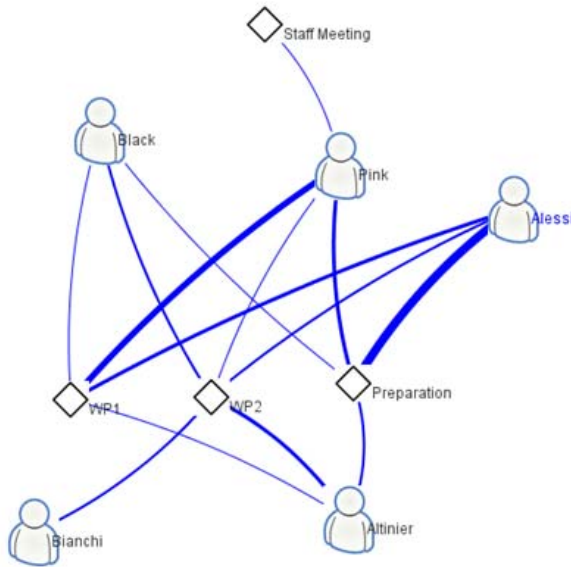


Figure 4. Bipartite graph capturing activities, user involvement, and actions. The line-thickness of each edge is based on the action count.

publish/subscribe and event brokering features. The logging service manages subscriptions of various service interaction events, among them activity-event logs including TExecutionAction, TCoordinationAction, TCommunicationAction. More details regarding the activity and action model can be found in Section 5, Figure 5.

5 Establishing Collaboration Structure

The sources introduced in the previous section deliver the required data to generate the underlying collaboration structure. The activity model (previously introduced in [12]) describes the management aspects such as responsible and involved users, temporal constraints, skill requirements, and applicable resources. The action model completes the activity design-time aspects with runtime information. This set of actions — collected as the team evolves — provides the fundamental input for deriving our collaboration metrics. We will briefly revisit the activity and action model, as some parts are vital to understand the relation to our bi-

partite graph model. Figure 5 displays only the relevant elements of the original model.

Coordination action capture any kind of management aspects such as activity updates, delegation or work progress notification. Communication action identify the members and resources involved in a synchronous/asynchronous interaction. Coordination actions and communication actions differ in their level of formality. The former results in an update of the activity structure, while the latter describes unstructured information exchange. Finally, execution actions record the use of (common) resources within the scope of an activity.

Graph Model for Activities and Actions

Let us denote the activity/action graph as $AG = (V_A, V_U, E)$, $V_A \subseteq \mathcal{A} = \{a_1, a_2, \dots, a_n\}$ defining the set of activities and $V_U \subseteq U = \{u_1, u_2, \dots, u_n\}$ the set of users involved in a collaboration. Each edge $e \in E$ holds a set of actions $\{at_1, at_2, \dots\}$ associating users with activities. This graph model is the basis for our collaboration metrics introduced in the following section. An instance of such a graph generated by Algorithm 1 is shown in Figure 4.

6 Collaboration Metrics

Literature in social network mining defines a number of social network metrics based on interaction graphs. An interaction graph describes the relations between multiple humans usually based on their communication messages. Metrics building on such type of graph provide useful data on the generic structure of large social networks but neglect the underlying activity structure. Consequently, we need to base collaboration metrics on the bipartite graph model presented in the previous section. Inspired by existing social network metrics, the following set of collaboration metrics provides greater insight and reveals hidden structures. Analyzing a bipartite activity-user graph, most metrics can be determined from an activity and user perspective with differing semantics. Thus, we present the general nature of each metric first, and subsequently describe the mapping onto our bipartite graph. In addition, we provide an interpretation of high and low metric values and their impact on collaboration.

Assortativity Coefficient AC : assortative networks consist of nodes that predominately connect to nodes of similar structure. The similarity measurement can be appli-

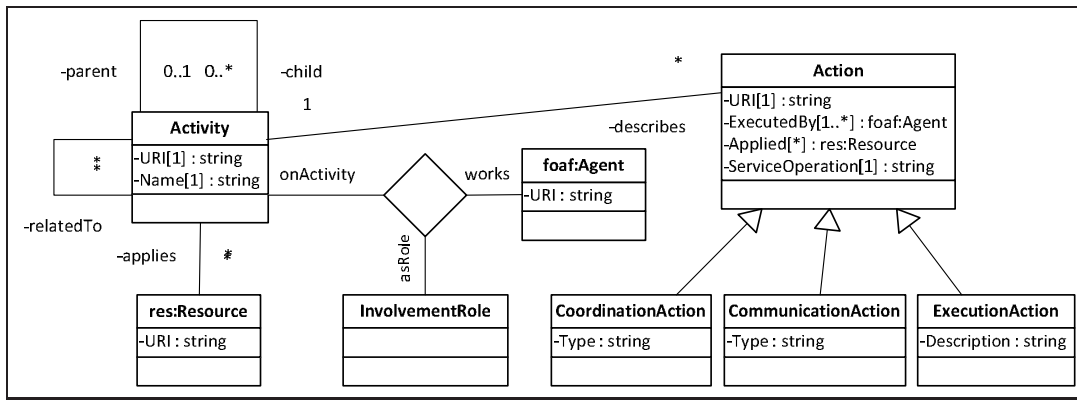


Figure 5. Activity and action model excerpt.

Algorithm 1 Create AG from action events.

```

/* Get actions from ActionRetrieval service */
A ← getActions()
for each action  $at \in A$  do
  /* Initialize a set for users involved in  $at$  */
   $U' \leftarrow \emptyset$ 
  for each user  $u \in getExecutedByFoafAgent(at)$ 
  do
    if  $u \cap V_U = \emptyset$  then
       $V_U \leftarrow V_U + u$ 
    end if
     $U' \leftarrow U' + u$ 
  end for
   $a \leftarrow getDescribesActivityURI()$ 
  if  $a \cap V_A = \emptyset$  then
     $V_A \leftarrow V_A + a$ 
  end if
  /* Add actions to  $AG$  */
  for each user  $u \in U'$  do
     $E_u \leftarrow getOutgoingEdges(u)$ 
    if  $a \notin V_A$  then
      /* Create new edge  $e = (u, a)$  */
       $E \leftarrow E + e$ 
    else
      /* Add actions to existing edge  $e$  */
    end if
    incrementActionCount( $e$ )
  end for
end for

```

cation specific or generic such as node degree. The Assortativity Coefficient AC — first introduced by Newman [10] for monopartite networks — measures the degree to which a network is assortative.

The *Activity Assortativity Coefficient* (AC_a) describes whether all users involved in an activity are involved in a similar number of other activities. A high value ($AC_a > 0.5$) does not imply that all users are heavily loaded with activities or only marginally involved in activities. Instead it only implies that users feature roughly the same involvement in other activities and thus are faced with the same level of cooperation complexity. In contrast, a low value ($AC_a < 0.5$) implies that across all activities, some of the involved members are very engaged in the project, while others are not.

The *User Assortativity Coefficient* (AC_u) indicates whether users tend to work on homogeneous activities. A set of activities is considered heterogeneous if the number of involved users varies across the set. From a high value ($AC_a > 0.5$) we can conclude that members involved in an activity with low member involvement is generally working on activities with little joint work. Also, a member assigned to an activity with many co-workers tends to participate in collaboration-intensive activities.

Clustering Coefficient CC originally introduced by Watts and Strogatz [14] measures the degree to which a graph features 'small world' properties, i.e. nodes are predominantly linked within neighborhoods. As in a bipartite graph two neighbors of a node must not be neighbors themselves, we provide following definition for the Activity and User Clustering Coefficient:

The *Activity Clustering Coefficient* (CC_a) is the likelihood of two activities having a common user, to have other common users. A high value ($CC_a > 0.5$) signifies a high number of common workers, thus suggesting high activity dependencies. In contrast, a low value ($CC_a < 0.5$) indicates a strong separation between activities.

The *User Clustering Coefficient* (CC_u) is the likelihood of two members in an activity, to be co-workers in another activity. A high value ($CC_u > 0.5$) suggests that the encountered combination of members provides a very reusable composition of skills or interests. A low value ($CC_u < 0.5$) indicates activities with pure task focused member assignments where reusable compositions are not feasible or have not been considered.

Degree Distribution DD describes whether edges are evenly distributed across all nodes — such as in random networks — or following some non-uniform distributions — such as a powerlaw function. Specifically, the *User Degree Distribution* DD_u describes the set of edges to be distributed around few or many workers. Likewise, the *Activity Degree Distribution* DD_a describes the set of edges distributed around few or many activities. We do not attempt to derive the exact distribution function but rather to detect a trend towards an equal distribution or an unequal distribution. A good indicator is the difference between the degree average of the top 10% connected nodes minus the degree average of the remaining 90% (assuming the degree average to be scaled to the interval of $[0, 1]$).

A high User Degree Distribution ($DD_u > 0.5$) reflects the existence of a few central coordinators, while a low value ($DD_u < 0.5$) suggests that no user possesses a good project overview. Similar a high Activity Degree Distribution ($DD_a > 0.5$) indicates a number of coordination activities. Again, a low value ($DD_a < 0.5$) implies that a central coordination activity is absent.

Structural Cohesion SC [9] describes a graph’s vulnerability to break into multiple subgraphs upon removal of a node. In a bipartite graph we can measure how susceptible a graph is towards removal of each type of node.

Activity Structural Cohesion SC_a is the ratio of total activities to the minimum number of activities which, if removed, would disconnect the graph. A high value ($SC_a > 0.2$) signifies that involved users of an activity are also connected through other activities. For a low value ($SC_a < 0.2$), we can deduce that removing (e.g. completing or aborting) an activity will have the involved members continue their work on separate activities, requiring no further communication or data exchange.

Likewise, *User Structural Cohesion* SC_u is the ratio of total users to the minimum number of users who, if removed, would disconnect the graph. A high value ($SC_u > 0.2$) indicates that removing one fifth of all users would still result in a connected graph. Hence, any leaving user has little effect on communication and data exchange in the team. A low cohesion ratio ($SC_u < 0.2$) indicates a lack of sufficient communication relations and activity assignments to avoid team instability in case of leaving members.

A low SC_u value results from a small number of users

assigned to a high number of activities lacking a sufficient degree of member involvement. In contrast, a low SC_a value arises from many users being distributed across a few activities. A collaboration graph featuring both low SC_a and low SC_u is especially susceptible to degenerate upon activity or user removal.

Structural Equivalence SE describes the occurrence of identical sets of vertices. For example, two users being involved in the same three activities share a common set of vertices and are thus considered structurally equivalent. In monopartite graphs, SE is significantly distinct from the clustering coefficient CC as SE compares also nodes that are not neighbors. As this is never the case in bipartite networks, we included an additional condition to clearly separate SE from CC .

Activity Structural Equivalence SE_a measures the extend to which activities of the same type have a common set of involved users. A high value ($SE_a > 0.3$) indicates that the applied composition of involved users performs well for the same activity type. High SE_a values coinciding with high SE_u values (see below) suggests a tendency towards collaboration of users with the same role, while for low SE_u values, we will find users featuring heterogeneous roles. A low structural equivalence value ($SE_a < 0.3$) indicates same type activities being predominately worked on by different user combinations. This highlights the existence of alternative user compositions.

User Structural Equivalence SE_u describes the extend to which users of the same role are involved in a common set of activities. A high value ($SE_u > 0.5$) indicates that a composition of similar users (i.e. having the same role) works well across multiple, potentially different activities. User roles that are best enacted individually result in a low value ($SE_u < 0.5$).

Global Density GD measures the ratio of user-activity involvement compared to maximum possible involvement (i.e. each user involved in every activity). While the definition is the same as in traditional, monopartite networks, the calculation is different, as we must not count ‘missing’ links between nodes of the same type. Hence, the maximum possible number of edges is $A * U$ with A denoting number of activities, and U denoting number of users.

Global Density serves as a first check whether to compute the set of metrics introduced above. If GD is very high ($GD > 0.9$) or very low ($GD > 0.1$) all the other metric have little expressiveness as they will yield measurements close to their value boundaries. Values in-between those extremes indicate that computing the other metrics will result in significant findings.

7 Applying Metrics in Algorithms

This initial set of simple, team-aware algorithms demonstrates how collaboration metrics can be applied to improve both grid utilization and research collaboration. Each algorithm covers a subset of the metrics introduced in the previous section, which includes the values for HIGH and LOW that are applied here.

User recommendation

Long-running scientific workflows potentially require user input for specific decisions but at unpredictable points in time. To reduce dead time, a researcher can elect additional members allowed to make the required decisions. Algorithm 2 recommends suitable co-scientists for this purpose. In the scope of this paper skills describe both explicitly stated expertise and implicit information in the form of previous actions.

Relevant user notification

The activity associated to an experiment lists the number of researchers to contact after the experiment has completed. Based on the collaboration structure, Algorithm 3 determines if other members outside the activity at hand should be contacted as well and which type of notification to use — e.g. multicast or unicast style messages. Here we apply the ranking algorithm introduced in [12] to establish the relevant set of members for an activity which are not explicitly assigned to the activity.

Task delegation

Efficient delegation of tasks such as scientific workflow design, experiment preparation, or result evaluation within a team, is only effective when the collaboration structure is taking into account. Selecting always the best skilled member is not a good decision when the overall team structure features bottlenecks in terms of information and data sharing. Algorithm 4 recommends which members to assign to a new activity. When collaboration becomes increasingly dependent on single activities or members, the algorithm focuses on improving the structural cohesion, otherwise on selecting most suitable members based on skill or load.

8 Conclusion and Future Work

We have presented an approach to combine research collaboration and grid utilization. To this end, we establish the collaboration structure comprising users and activities by extracting interaction information from various collaboration data sources. Without considering the activities in teams interaction analysis, we cannot uncover the fine grained structures that are influencing the significant inter-dependencies of humans and grid resources. Applying collaboration metrics based on a bipartite graph model

Algorithm 2 Recommend backup member:
getBackupMember(Activity a, User u).

```

p ← getProject(a)
/* Compute the set of required skills */
S ← getSkills(u) ∩ getSkills(a)
/* Initialize set of recommended backup members*/
Urec ← ∅
if getGlobalDensity(p) = HIGHGD ∨
getGlobalDensity(p) = LOWGD then
    Urec ← getUsersBySkill(a, S)
    return(Urec)
end if
Acand ← ∅
/* There is a high likelihood of finding additional suitable
users if no central coordinator exists and only few users
are involved */
if getUserDegreeDistribution(p) = LOWDD ∧
getDegree(a) = LOWDD then
    if getActStructEquiv(p) = LOWSE then
        Acand ← Acand ∪ getSimilarActivities(p, a)
    end if
    if getUserStructEquiv(p) = HIGHSE then
        Acand ← Acand ∪ getInvolvedInActivities(u)
    end if
else
    Acand ← Acand ∪ getParentChildActivities(a)
end if
for all activity ac ∈ Acand do
    Urec ← Urec ∪ getUsersBySkill(ac, S)
end for
return(Urec)

```

Algorithm 3 Adaptive notification type:
notifyMembers(Activity a, User u, Message msg).

```

p ← getProject(a)
U ← getInvolvedUsers(a)
if getGlobalDensity(p) = HIGHGD ∨
getGlobalDensity(p) = LOWGD then
    notifyMulticast(U, msg)
else
    if getUserClusterCoeff(p) = LOWCC then
        notifyUnicast(U, msg)
    else
        notifyMulticast(U, msg)
    end if
    /* Initialize set of relevant users (outside activity)*/
    Ur ← ∅
    /* There is a high likelihood of finding additional inter-
    ested users if no central coordination activity exists
    and few users are involved */
    if getActDegreeDistribution(p) = LOWDD ∧
getDegree(a) = LOWDD then
        Ur ← getRelevantUsers(p, a)
        if getActClusterCoeff(p) = LOWCC then
            notifyUnicast(Ur, msg)
        else
            notifyMulticast(Ur, msg)
        end if
    end if
end if

```

Algorithm 4 Recommend users for task delegation:
getDelegates(Activity a, Skills S, Role r).

```

p ← getProject(a)
/* Initialize set of potential users*/
U ← ∅
if getGlobalDensity(p) = HIGHGD ∨
getGlobalDensity(p) = LOWGD then
    parent ← getParentActivity(a)
    U ← getRankedUsersBySkill(parent, S)
else
    /* Decide whether its important to stabilize team*/
    if getActStructCohesion(p) = LOWSC ∧
getUserStructCohesion(p) = LOWSC then
        /* Add strategic links to the new activity*/
        A⊙ ← getStructuralHoleNeighbors(p)
        for each activity ac ∈ A⊙ do
            U ← U ∪ getBestUserBySkill(ac, S)
        end for
    else
        if getActAssortCoeff(p) = HIGHAC then
            /* Recommend the user with best skills if
            users are involved to same degree */
            U ← getRankedUsersBySkill(p, S)
        else
            /* recommend the user with lowest load and
            required role, if users are involved to different degree */
            U ← getRankedUsersByLoad(p, r)
        end if
    end if
end if
return(U)

```

is promising because it combines both users and activities. The discussed algorithms demonstrate how different adaptation strategies can be combined for team-aware grid utilization.

We plan to analyze metrics across multiple teams to find regularities that allow for team form clusters. Adaptation and decision could then be customized on a team form basis, rather than individually for every team. In addition, we are extending the set of adaptation algorithms to evaluate in more detail the significance of the various metrics.

Acknowledgment

Part of this work was supported by EU FP6 project in-Context (034718).

References

- [1] J. Alameda, M. Christie, G. Fox, J. Futrelle, D. Gannon, M. Hategan, G. von Laszewski, M. A. Nacar, M. Pierce, E. Roberts, C. Severance, and M. Thomas. The open grid computing environments collaboration: Portlets and services for science gateways. *Special Issue for Science Gateways GGF14 workshop. Concurrency and Computation: Practice and Experience*, 19(6):921–942, April 2007.
- [2] C. Bird, A. Gourley, P. Devanbu, M. Gertz, and A. Swaminathan. Mining email social networks. In *MSR '06: Proceedings of the 2006 international workshop on Mining software repositories*, pages 137–143, New York, NY, USA, 2006. ACM.
- [3] I. Brandic, S. Pillana, and S. Benkner. An approach for the high-level specification of qos-aware grid workflows considering location affinity. *Scientific Programming Journal*, 14:231–250, 3-4 2006.
- [4] M. Dredze, T. Lau, and N. Kushmerick. Automatically classifying emails into activities. In *IUI '06: Proceedings of the 11th international conference on Intelligent user interfaces*, pages 70–77, New York, NY, USA, 2006. ACM.
- [5] I. Foster. Service-oriented science. *Science*, 208(5723):814–817, May 2005.
- [6] T. Glatard, D. Lingrand, and J. M. and Michel Riveill. Impact of the execution context on grid job performances. In *First International Workshop on Context-Awareness and Mobility in Grid Computing*, May 2007.
- [7] T. Hey and A. E. Trefethen. Cyberinfrastructure for e-science. *Science*, 308(5723):817–821, May 2005.
- [8] Y. Long, H. Lam, and S. Su. Adaptive grid service flow management: framework and model. *Web Services, 2004. Proceedings. IEEE International Conference on*, pages 558–565, July 2004.
- [9] J. Moody and D. R. White. Structural cohesion and embeddedness: A hierarchical concept of social groups. *American Sociological Review*, 68(1):103–127, 2003.
- [10] M. E. J. Newman. Mixing patterns in networks. *Phys. Rev. E*, 67(2):026126, Feb 2003.
- [11] R. Rowe, G. Creamer, S. Hershkop, and S. J. Stolfo. Automated social hierarchy detection through email network analysis. In *WebKDD/SNA-KDD '07: Proceedings of the 9th WebKDD and 1st SNA-KDD 2007 workshop on Web mining and social network analysis*, pages 109–117, New York, NY, USA, 2007. ACM.
- [12] D. Schall, C. Dorn, S. Dustdar, and I. Dadduzio. Viecear - enabling self-adaptive collaboration services. In *34th EURO-MICRO Conference on Software Engineering and Advanced Applications (SEAA)*, September 2008.
- [13] S. Stolfo, S. Hershkop, K. Wang, O. Nimeskern, and C.-W. Hu. Behavior profiling of email. In *Intelligence and Security Informatics*, June 2003.
- [14] D. J. Watts and S. H. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393(6684):440–442, June 1998.
- [15] A. C. Yamin, I. Augustin, L. C. da Silva, R. Real, C. Geyer, and G. Cavalheiro. Towards merging context-aware, mobile and grid computing. *International Journal of High Performance Computing Applications*, 17(2):191–203, 2003.
- [16] G. Zhang and M. Parashar. Dynamic context-aware access control for grid applications. *Grid Computing, 2003. Proceedings. Fourth International Workshop on*, pages 101–108, Nov. 2003.