**Proceedings of 2012 ASME International Design Engineering Technical Conferences &
Computers and Information in Engineering Conference
August 12-15, 2012, Chicago, IL, USA**

DETC2012-70604

EVALUATING SOCIO-TECHNICAL COORDINATION IN OPEN-SOURCE COMMUNITIES: A CLUSTER-BASED APPROACH

Inna Rytsareva[1], Qize Le[2], Emma Conner[3], Ananth Kalyanaraman[1], and Jitesh H Panchal[2]

[1]School of Electrical Engineering and Computer Science, Washington State University, Pullman, WA, USA
[2]School of Mechanical and Materials Engineering, Washington State University, Pullman, WA, USA
[3]Oberlin College, Oberlin, OH, USA

ABSTRACT

In complex product development, coordination is the act of managing dependencies between artifacts. Socio-technical coordination is the achievement of coordination through the alignment of organizational structures and product structures. Socio-technical coordination is achieved in hierarchical product development organizations by aligning the organizational structure with the system architecture. However, within virtual community-based product development such as open source development, the organizational structure is not designed by a central authority. In contrast, the community evolves as a result of participation of individuals and their communication with other individuals working on the project. Hence, understanding and quantifying socio-technical coordination is particularly important in open-source communities.

Existing approaches to measuring socio-technical coordination are based on the congruence between ideal communication and the actual communication structures within communities. The primary limitation of existing approaches is that they only account for explicit communication between individuals. Existing measures do not account for the indirect communication between individuals and the shared knowledge that individuals working on a joint project possess. Due to these limitations, the socio-technical coordination values have been observed to be very low in the existing literature. We propose two alternate approaches to measuring socio-technical coordination based on clustering techniques. We illustrate the approaches using a case study from an open source software development community. The proposed approaches present a broader and more encompassing view of coordination within open source communities.

Keywords: Coordination, social network analysis, open source software, online communities, clustering

1 INTRODUCTION: SOCIO-TECHNICAL COORDINATION

1.1 Overview and the need for measuring socio-technical coordination

Coordination within an organization can be defined as "integrating or linking together different parts of an organization to accomplish collective sets of tasks" [1], or "management of dependencies" [2, 3]. Various methods, including informal communication, group meetings, email and forum discussions, and development of plans and rules can be employed to achieve coordination within an organization. Three types of coordination approaches are defined by Herbsleb and Grinter [4] as: 1) architecture-based coordination, 2) plan-based coordination, and 3) process-based coordination. Architecture-based coordination is associated with product modularization, which is applied to reduce dependencies across modules in a product. Plan-based coordination involves the development and integration of plans and timelines to ensure the accomplishment of interdependent tasks. Process-based coordination is achieved through development and modification of management processes.

A widely adopted strategy to achieve coordination is the alignment of organizational structures with product structures. The coordination achieved through the alignment of organizational structures and product structures is referred to as socio-technical coordination [5]. Recently, a measure called socio-technical congruence has been proposed in the literature to quantify the extent of socio-technical coordination [5]. Existing studies in organization science indicates that the alignment between design of organizational structures and product structure is beneficial for product development. High level of socio-technical congruence has been shown positive influence in productivity and product development rate [5, 6], while a lack of coordination has negative effects on productivity [7, 8]. Besides, existing literature also suggests that organizational structure affects the structures of products developed by that organization. The effect is commonly known as the Conway's law: "any organization that designs a system will inevitably produce a design whose structure is a copy of the organization's communication structure" [9]. Finally, it has been hypothesized that because the communication patterns between individuals are driven by the product dependencies, the social structure of organization matches the structure of product [10]. This hypothesis is known as the mirroring hypothesis [11].

Our goal in this paper is to develop approaches for computing socio-technical coordination in virtual communities involved in developing open-source products. Understanding of the socio-technical coordination is important in open source communities because it helps in determining the impacts of product structures on community structures and vice versa. Furthermore, understanding of socio-technical coordination may help in increasing the success of open source projects, particularly, the open source hardware projects.

1.2 Review of existing literature

As discussed in Section 1.1, the literature on socio-technical coordination is associated with two concepts: mirroring hypothesis and socio-technical congruence. The mirroring hypothesis states that the technical dependencies and communication patterns are mirrored with one another in the product development processes [10]. Existing studies on mirroring hypothesis are focused on analyzing the correspondence between technical dependencies and communication patterns based on empirical case studies. Different levels of correspondence between technical dependencies and communication patterns are displayed in existing empirical studies. Colfer and Baldwin [10] present a detail review of literature by comparing 102 empirical studies ranging from different levels of organizations, including a single firm, across multiple firms and open communities. Based on the studies, the authors conclude that mirroring hypothesis is more prominent in single firms, less prominent in the studies across multiple firms, and are least prominent in open communities. Camuffo and Cabigisu [12] and Hoetker [13] analyze the extent to which modular products are associated with modular organizations. Camuffo and Cabigiosu [12] study the inter-organizational relationships between the manufacturers and suppliers based on an example of air conditioning industry. Kratzer et al. [14] analyze multi-institutional communication patterns in the space industry and conclude that the informal communications play a strong role in achieving coordination between different activities.

While mirroring hypothesis is focused on correspondence between technical and organization structure during the product development process, socio-technical coordination [15] is based on the assumption that the organizational structures *should* match the technical dependencies. Cataldo et al. [5, 6] introduced the concept of congruence as a measure for coordination activities carried out within an organization. The congruence measurement is based on the comparison of matrices representing coordination requirements and actual coordination [6]. Coordination requirements refer to coordination activities that should happen due to the technical dependencies of a product. The actual coordination is the set of coordination activities that individuals are engaged in through multiple communication methods such as face-to-face discussions, emails, and online forums. Sosa [16] extends the analysis of congruence by analyzing not only the coordination requirement that are met but also unpredicted and unintended integrations. The impacts of socio-technical congruence on performance are also highlighted in existing studies. Different measures of performance are used by different authors. For example, Cataldo et al. [6] employ resolution time as a measure of performance, and show that lower resolution time, namely better performance, is observed when the congruence are high. Kwan et al. [17] apply the success of software builds as a measure of performance and show the congruence is proportional to build success by collocated teams.

1.3 Research gap and paper outline

The limitation of existing approaches for socio-technical coordination when applied to open source processes is that existing approaches compare the information exchange between teams, which are not well defined in loosely coupled open source communities. The congruence measures are calculated by analyzing the direct interactions between individuals. However, in open source processes, the number of individuals is generally very large. The values of socio-technical congruence are very small due to the fact that participants work on different aspects of the code. We have observed values less than 1% in our studies of some open source software (see Section 3.3). However, the small value of congruence does not necessary mean that the coordination is not taking place. As the number of participants increases, the coordination requirements grow at a faster rate than the actual coordination. Hence, the socio-technical congruence calculated according to direct communication actually reduces.

Additionally, the existence of dependencies between modules does not necessarily imply that the corresponding individuals must communicate. Coordination may also happen through a) shared knowledge within communities, and b) prior communication on related dependencies. Such mechanisms of coordination are not captured by the socio-technical congruence measure. Hence, the existing socio-technical congruence measure provides inadequate information about coordination in open source communities.

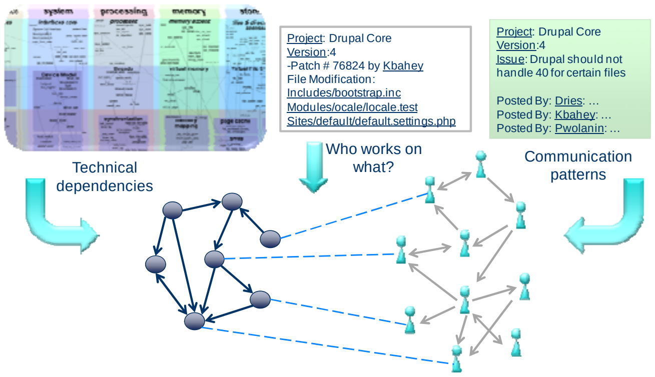
To address these limitations, we present an alternate approach to measuring coordination. The approach is based on identification of communities via clustering analysis and comparison of ideal communities with expected communities. The underlying hypothesis is that if individuals have closer communications with each other, they are part of the same community, and they share some common knowledge. Because of this common knowledge, they do not need to communicate for each and every dependency. In the proposed approach, instead of comparing networks at the individual link level, the networks are first clustered into communities and then the resulting clusters are compared to measure the socio-technical coordination.

The paper is organized as follows. In Section 2, the proposed cluster-based approach is described in detail. Our approach for modeling of product and communities as hybrid networks is presented in Section 2.1. The clustering techniques employed in this paper for bipartite and 1-mode networks are highlighted in Section 2.2. The cluster comparison methods are described in Section 2.3. In Section 3, the proposed approach is applied in an open source software community - Drupal. The data collection process for Drupal is discussed in Section 3.1. The outcomes of the clustering step are discussed in Section 3.2. The baseline results from coordination measured using existing socio-technical congruence measure and Sosa's approach [16] are presented in Section 3.3. The results from the proposed approach are presented in Sections 3.4 and 3.5. Finally, closing comments are made in Section 4.

2 PROPOSED CLUSTER-BASED APPROACH FOR SOCIO-TECHNICAL COORIDNATION

2.1 Modeling products and communities as hybrid networks

Let $V_p=\{p_1,p_2,\dots p_m\}$ denote a set of open source software developers (referred to as "people"), and $V_f=\{f_1,f_2,\dots f_n\}$ denote a set of source files (referred to as "products"). The input to our analysis contains three networks: i) people network, ii) project network, and c) people-project bipartite network. A combination of the three networks with the three types of links, as shown in Figure 1, is called the hybrid network. The files are represented as circles and the links are represented as arrows. Ideally, the communication between individuals and dependencies between files are directed in nature. However, we model them as undirected edges for initial analysis presented in this paper. The approach will be extended to account for the directed nature of edges in the future.



Project: Drupal Core
Version:4
-Patch # 76824 by Kbahey
File Modification:
Includes/bootstrap.inc
Modules/ocale/locale.test
Sites/default/default.settings.php

Project: Drupal Core
Version:4
Issue: Drupal should not handle 40 for certain files

Posted By: Dries: …
Posted By: Kbahey: …
Posted By: Pwolanin: …

Figure 1 - Generation of the hybrid networks with three types of links

i) People network $G_p(V_p,E_p)$: An edge (p_i,p_j) in E_p (undirected, weighted) represents the communication between developers p_i and p_j, and the edge weight (an integer) represents the volume of that communication, i.e., the number of times two individuals communicate with each other. This is shown as links between individuals in Figure 1.

ii) Product network $G_f(V_f,E_f)$: This network contains information about the product interdependency. An edge (f_i,f_j) in E_f (undirected, weighted) implies that there exists a dependency between two files p_i and p_j, and the edge weight represents the degree of that dependency. Two files are considered dependent if there is at least one function call between the two files. The edge weights correspond to the number of function calls across the files. Edges E_f are shown as arrows between circles (projects) in Figure 1.

iii) People-product bipartite network $G_b(V_p,V_f,E_b)$: This is a bipartite network where each node represents either a developer or a source file, and an edge (p_i,f_j) in E_b (undirected, unweighted) represents the information that the developer p_i has contributed to the product f_j. The edges are shown as dashed lines in Figure 1.

2.2 Clustering techniques for bipartite and 1-mode networks

Our first step is to individually cluster the bipartite network and two 1-mode networks. To perform clustering, there are many well known community detection methods that can be used (e.g., [18-20]). We use the Louvian method [18], which is one of the more recent widely-used algorithms. This algorithm is an adaptation of one of Newman's classical methods [19], and implements a greedy heuristic to optimize "modularity" [21] of clustering. Modularity is a measure of the quality of a division of a network. Modularity measures "the fraction of the edges in the network that connect vertices of the same type (i.e., within-community edges) minus the expected value of the same quantity in a network with the same community divisions but random connections between the vertices" [21].

The main idea of the clustering algorithm is as follows: Starting with each element in a "community" for itself, the algorithm deploys an iterative approach to migrate elements into communities so as to optimize the increase in modularity at every stage. At the start of every iteration, all elements are visited in a sequential order, and for each element *i*, a community is determined, which by recruiting *i* would result in the largest positive value increase of modularity. Note that this search needs to be restricted only to those communities with which *i* shares a neighbor and to those in which the change in modularity is positive. Furthermore, the value by which the modularity would change owing to the migration of a given element *i* to a give community can be calculated in constant time. The sequential order processing is repeated iteratively until there is no further change in the community compositions. This completes the first phase. In the second phase, a new meta-network is constructed by representing each community resulting from the previous phase by one meta-node and connecting those pairs of meta-nodes which share at least one edge in common. The algorithm terminates when communities stop changing.

Although the Louvian method does not guarantee optimality of modularity, it is effective in producing high quality clusters in practice [18], and the tool is available as part of Gephi [22], which is a popular open source graph visualization and analysis toolkit.

Modularity of the output clusters were computed differently for the unipartite and bipartite networks. For unipartite networks, modularity was calculated as per the standard Newman's formula [21], which calculates the fraction of intra-cluster edges obtained by the clustering and then subtracts the fraction as would be obtained in a random network with identical vertex degree distribution (to negate for the random factor). There are multiple ways to calculate modularity in bipartite networks [23-25]. We used the expression given by Murata [25] because it does not enforce a one-to-one mapping between clusters from both sides.

2.3 Cluster Analysis Methodology

In this section, we outline the steps carried out in analyzing the results of clustering. We present three levels of comparisons between the two individual networks (people and product) and the bipartite network. Each of these three comparative methods provides a different insight as explained below. The primary question driving our analysis is: *what is the extent of socio-technical coordination within open source communities?*

2.3.1 Link-link comparison

The first way is based on the existing work by Cataldo et al. [5, 6] and Sosa [16]. As discussed in Section 1.2, the existing approach to measure socio-technical coordination is to calculate congruence [5, 6] based on coordination requirements and actual coordination activities among individuals. The coordination requirement network consists of communication links that should ideally be present between individuals in order to resolve product dependencies. While actual coordination records the communication that is actually happening among individuals. The matched interactions [16] are defined as the number of links that occur in both coordination requirement network and actual communication network. The ratio of matched interactions to the total number of links in the coordination requirement network is the socio-technical congruence [5, 6].

Sosa's approach [16] is another approach for link-link comparison where, *matched interactions* are defined in the same way as congruence. The *potential unattended interactions* are defined as the links that present in the coordination requirements network but absent in the actual communication network. Finally, the links present in the actual communication network but absent in coordination requirements network are called *unpredicted interactions*.

Both these approaches are based on direct comparison of the links between ideal communication network and the actual communication network. In the context of our open software community data, this can be achieved by deriving the ideal communication structure between individuals based on the bipartite network and the 1-mode product network, and comparing the ideal communication structure with the actual 1-mode people network. This approach assumes a strict definition of coordination and uses direct communication between individuals as an indication of coordination. This approach provides a way to quantify the number of conserved as well as unique links at the finer-level of individual people and their connections. As discussed in Section 1.3, this does not provide any insights into the underlying community structure as it uses only the original input networks as input. The results from link-link comparison are presented in Section 3.3.

Note that a similar approach can also be applied in the other direction – comparing the ideal dependencies between products based on the communication between individuals with the actual dependencies. This can be used to check the validity of the Conway's law for this project. However, we only focus on coordination in this paper.

2.3.2 Cluster-cluster comparison

An orthogonal approach is to compare the communities detected within the people network against the communities detected from the other network. In the context of our analysis, this is achieved by generating clusters (aka. communities) for all three input networks and then comparing the clusters of each of the 1-mode (people, product) networks against the clusters obtained for the bipartite (people-project) network. The clusters obtained from the bipartite network indicate natural groupings of people and projects based on the commonality of tasks that people are working on. The individuals clustered together within the bipartite network are the ones who are working closely together on related tasks. Hence, it can be assumed that these individuals share common knowledge. On the other hand, the clusters of individuals in the 1-mode people network indicate that individuals are related with each other through direct or indirect communication. A higher overlap between clusters obtained from the 1-mode network and the clusters obtained from bipartite network indicates a higher extent of *implicit coordination*.

Specifically, based on these clusters, we ask the question: what fraction of pairs of individual entities (pairs of developers or pairs of products) have been clustered together (or separated) consistently between the two clustering results (1-mode networks and bipartite network)? We implement this comparison using a Rand Index calculation [26]. The Rand index can be expressed as:

$$\mathrm{R} = \frac{\mathrm{a} + \mathrm{b}}{\binom{\mathrm{n}}{2}}$$

where a is the number of pairs of nodes in the same cluster in both partitions being compared, b is the number of pairs of nodes in different clusters in both partitions; n is the total number of nodes. Other measurements for comparing clusters include the adjusted rand index [27], Jaccard index [28], the Wallace index [29], and the normalized Lerman index [30]. However, in this paper, we only use the Rand index due to its simplicity of calculation and intuitiveness. The comparison of clusters could lead to a better understanding of the degrees of similarities and differences underlying the community structures of both networks (1-mode vs. bipartite). Yet the obtained information, which is a Rand index, is only a summary figure.

2.3.3 Link-cluster comparison

The link-link and cluster-cluster comparison approaches are two extremes where link-link approach uses a strict definition of coordination whereas the cluster-cluster comparison uses a broader and more encompassing view of coordination. The third approach is an intermediate approach where links in one network are compared against communities detected in another network.

In the context of our analysis, we implement this approach by comparing the connections implied/induced by the bipartite network clustering among the people (or alternatively, products) against the observed connections in the corresponding input network of people (or alternatively, products). For instance, let *(p_i, p_j)* and *(p_i, p_k)* be two edges present in the input people network, such that p_i and p_j are clustered together in the bipartite network but p_i and p_k are not. This implies that $\mathrm{p_i}$ and $\mathrm{p_j}$ are both communicating with one another and are expectedly grouped together in product space; whereas $\mathrm{p_i}$ and $\mathrm{p_k}$ are communicating and yet their product affiliations do not offer sufficient basis for them to be grouped together in the bipartite setup. Henceforth, for sake of convention, we will refer to edges such as (p_i, p_j) as ***conserved*** edges, and all edges such as (p_i, p_k) as ***discrepant*** edges.

The same analysis when applied to the product network has a different meaning. Two products f_i and f_j which are both dependent (i.e., share an edge in the product network) and also get clustered together in the bipartite network implies that there is a group comprising of one or more developers who are jointly contributing to these two interdependent products. However, if there are two products $\mathrm{f_i}$ and $\mathrm{f_K}$ which are inter-dependent in the product network but not clustered together in the bipartite network that indicates a potential gap (or a disconnect) in communication among those products' contributors. In other words, this approach of comparing links to clusters to determine conserved and discrepant edges can serve as an effective way to flag anomalies as well as formulate hypothesis about the level of organization and communication in an open source software community. An alternative albeit less interesting outcome of the analysis is perhaps that the quality of the output clusters is not adequate, which in itself could be another layer of useful information. The results of the link-cluster comparison are presented in Section 3.5.

3 CASE STUDY: DRUPAL

In this section, we present the approach using an open source software project – Drupal [31]. Drupal is a content-management system used for the creation of community-based websites. Drupal has been under development since 2001. We analyze a recent major version of Drupal core (version 7.7). Drupal is well developed with over 7000 community-contributed add-ons, known as "contrib" modules. Besides, over 1000 developers are involved in the project development. Drupal is selected as a case study because of its maturity, availability of code and the availability of information about communication between participants.

3.1 Data collection

The software structure of Drupal is modeled as a weighted network where nodes represent files and the links represent function-calls between files. A documentation generator tool, Doxygen [32], is used to extract functions and corresponding function calls from the source code. The strength of the interface between files is defined by the number of function calls between two files. The community structure of Drupal is also modeled as a weighted network. The communications between participants are derived from on-line forums. The communication links between participants are determined by analyzing each post on the forum. A post on the forum contains information about the names of participants and the software version they are discussing about. Relationships are built among participants discussing on the same post.

The participation links are created by analyzing file modifications and issue/patch records, recorded on the Drupal website. A file modification indicates which files are modified to resolve a specific issue. An issue/patch record indicates who worked on the specific issue/patch. Combining the information derived from file modifications and issue/patch records, the participation links representing participants working on files are generated.

The generation of the networks with the three types of links is illustrated in Figure 1. The basic statistics of these networks is presented in Table 1. Isolated nodes have been eliminated for clustering purposes. Hence, the number of nodes in the table is based on the nodes with at least one edge. The 1-mode people network contains 5,180 nodes with 148,102 edges. The 1-mode product network contains 167 source files which are linked through function calls. The files that do not have function calls (e.g., image files, .css files, etc.) are isolates and are not captured in the 1-mode product network. The bipartite network captures all files linked to at-least one individual. The number of nodes in the bipartite network is 6,572 and the number of edges is 95,988. The edges in the bipartite network only represent affiliation relationship. Hence, these edges are not weighted.

Table 1 – Input networks statistics

Input network	No. nodes (with at least one edge)	No. edges	Weighted?
People (1-mode)	5,180 people	148,102	Yes
Product (1-mode)	167 source files (or “products”)	900	Yes
Bipartite	6,572 (=5,794 people + 1,278 products)	95,988	No

3.2 Summary of Network Clusters

We clustered the three input networks separately using the methodology described in Section 2.2. The results are summarized in Table 2. The table shows the modularity of clustering achieved for each of the three networks. In practice, a modularity of 0.3 or more is generally considered indicative of a well-defined community structure in unipartite networks [19]. As can be observed the modularity is higher for the people network (0.307) than it is for the other networks, indicating a more well-defined organization of the software developers into communities. This observation is further illustrated by Figure 2, which shows a lower degree of inter-cluster connectivity across communities in the people network (i.e., more modular) than it is for the other networks. On the other hand, the 1-mode product appears to have a lower modularity (<0.3; also see Figure 2(b)), implying a weaker organization as communities.

Table 2 – Summary of clustering results

Input network	Modul-arity (Q)	No. of clusters	Average number of intra-cluster edges per cluster (K_1)	Average number of inter-cluster edges between any two clusters (K_2)	Ratio K_1:K_2
People (1 mode)	0.307	15	4,644	980	4.738:1
Product (1-mode)	0.208	7	46	27.4	1.678:1
Bipartite	0.334	9	4,887	1,665	2.935:1

Table 2 also shows the average number of edges that ended up within clusters (intra-cluster: K_1) and between pairs of clusters (inter-cluster: K_2). A larger modularity would result in a higher ratio for K_1:K_2. And as can be observed this ratio is highest (4.738:1) for the people network. Put another way, there are 4.738 discussions happening between people of the same community, for every discussion that involves people from two different communities.

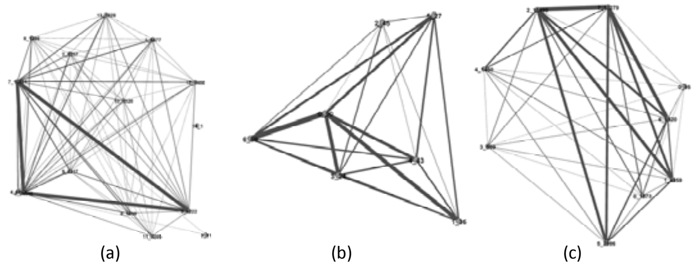


Figure 2 – Clustering results: “Cluster graphs” showing the inter-community relationships for the a) people network, b) project network, and c) bipartite network. Each node in the cluster graph corresponds to one community/cluster that was detected in the underlying network, and an edge between two cluster nodes corresponds to the presence of (inter-cluster) edges between any two vertices of those clusters. The thickness

of an edge is a relative indication of the number of inter-cluster edges between the two clusters.

Figure 3 shows the cluster size distribution for the three networks. In the clustering for the people network, the top two largest clusters accounted for almost 60% of the nodes, with the remaining 13 clusters significantly smaller. This skewed cluster size distribution coupled with the high modularity of clustering suggests the high degree of communication that characterizes people within the same cluster. As for other two networks (1-mode project network and the bipartite network), the cluster size distribution is more uniform, implying a reduced degree of community organization (as also corroborated by the smaller values of modularity).

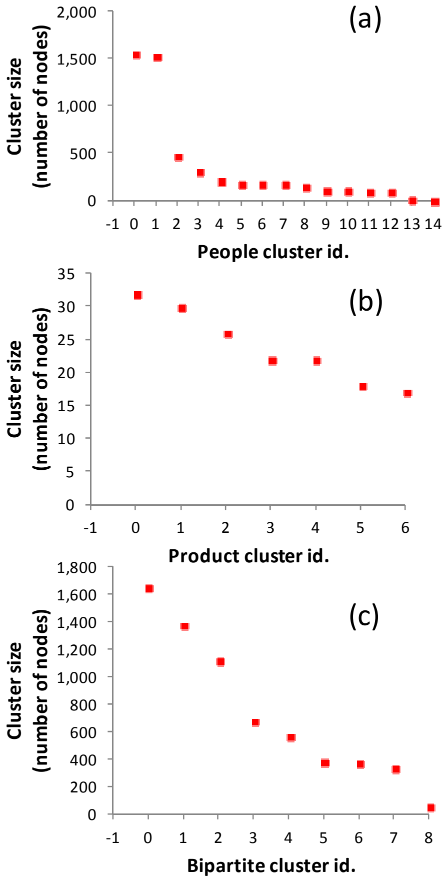


Figure 3 – Clustering results: cluster size distribution for the a) 1-mode people network, b) 1-mode product network, and c) bipartite network.

3.3 Link-Link comparison

The value of congruence measure and Sosa's approach for Drupal version 7.7 are shown in Table 3. From this table, it is observed that congruence value is 0.0099 (i.e., 0.99%), which indicates that less than 1% of the expected interactions between people are actually present in the 1-mode people network. The ideal congruence value is 1, which implies that people are communicating corresponding to all the technical dependencies. Clearly, 0.99% is a very low congruence value. Besides, the number of unpredicted and potential unattended interactions is significantly higher than matched interactions, which means the socio-technical coordination through direct communication is not a significant part in Drupal development.

Table 3 – Link-Link Comparison

Input Networks	Congruence	Matched interactions	Unpredicted interactions	Potential unattended
People vs. Product	0.0099	127,983	20,119	12,782,290

Low values of socio-technical congruence have been reported in other studies on open-source software. For example, Kwan et al. [17] observed that over 75% of software builds in an IBM project involving distributed developers have congruence value of less than 25%. Even for software builds with a congruence of 0%, the authors in [17] observed a build success of 93%. Low values of socio-technical congruence do not necessary mean a lack of coordination since coordination happens through implicit communication also. For example, by sharing workspace, individuals can get information about addressing dependencies between technical aspects. Bolici and co-authors [33] argue that within open source community, the coordination primarily happens via indirect communication, which is referred as stigmergic coordination.

Some of these limitations of congruence measure can be addressed by assuming that individuals who are within a community (cluster) share common knowledge and indirectly communicate with other individuals. One way to identify the indirect communication among individuals is through community identification. In this approach, people are clustered into communities, indicating that there are strong interactions between individuals within the community and weak interactions between communities. Instead of studying links between individuals, we assume that individuals belong to the same cluster communication with one another by direct or indirect communication. The results of cluster-based coordination are presented in Sections 3.4 and 3.5.

3.4 Cluster-Cluster comparison

We compared the clusters of the 1-mode networks against the clusters of the bipartite network as was described in Section 2.3.2. We report the Rand index for those comparisons in Table 4. As can be observed the Rand index for both comparisons are above 70%, which is a strong indication of overlap at the cluster-to-cluster level. In other words, the probability that two nodes are clustered the same way (either together or separate) in both the 1-mode network and the bipartite network is above 0.7. Note that Rand index does not take into account the actual node-level connections. It only captures the information of cluster memberships for every pair of nodes.

Table 4 – Cluster-cluster comparison: Rand indices for comparing the individual people and product communities against the communities detected in the bipartite network. A perfect agreement between the two community sets implies a Rand index of 1.

	People clustering vs. Bipartite clustering	Product clustering vs. Bipartite clustering
Rand index	0.718	0.736

3.5 Link-Cluster comparison

We compared the direct links (i.e., edges) of each of the 1-mode networks to the bipartite clustering as described in Section 2.3.3. Table 5 shows the results of this comparison.

Table 5 – Link-cluster analysis: Partitioning of the input (i.e., people or product) network edges into "conserved" and "discrepant" edge categories, based on their presence or absence within the same cluster of the bipartite graph, respectively.

Input network	Partitioning of the 1-mode network edges by the bipartite clustering		Distribution of the 1-mode network edges among bipartite clusters		
	No. conserved edges (%)	No. discrepant edges (%)	Average number of conserved edges per bipartite cluster (K_1)	Average number of discrepant edges between any two bipartite clusters (K_2)	Ratio $K_1 : K_2$
People vs. Bipartite clustering	43,779 (**30.91%**)	104,323 (69.09%)	5,086	2,748	1.850:1
Product vs. Bipartite clustering	158 (**17.55%**)	742 (82.45%)	17.5	14.8	1.182:1

Two sets of results are presented. First, the 1-mode network edges are categorized into conserved and discrepant categories (defined in Section 2.3.3). Each conserved edge is one that is present in the 1-mode network as well as induced by the clustering of the bipartite network. An edge in the 1-mode network that is not induced by the bipartite clustering is referred to as discrepant. As can be observed, a vast majority of edges in both 1-mode networks are discrepant edges. In the case of the people network, only 30.91% of the edges are conserved (i.e., maintained by the bipartite clustering); whereas for the product network this figure is further lower (17.55%). While the fraction of conserved edges may appear low, it is much higher when compared to the congruence (0.99%) observed through direct link–level comparison (shown in Table 3).

This attests to the value added to the overall understanding of community structure by augmenting 1-mode networks with bipartite information. The large fraction of discrepant edges, on the other hand, can be attributed to the lack of coordination in developing open source software.

We also measured the distribution of the 1-mode network edges among the bipartite clusters. These results are shown in the second half of Table 5. As can be observed, the frequency of conserved edges per cluster is only marginally higher than the proportion of discrepant edges across two clusters.

4 CLOSING COMMENTS

The primary contribution in this paper is a broader view of coordination within virtual community based product development. In contrast to the existing approaches that only focus on explicit communication as a means for coordination, our approach accounts for implicit and indirect communication between individuals and the presence of shared knowledge possessed by individuals within a community.

In terms of the methodology, the proposed approach is unique because it is based on clustering techniques to rather than direct comparison of links within networks. As the number of nodes (n) in a network increases, the number of possible links between nodes increases at a faster rate (n^2). Because of this, as the number of individuals grows, the congruence measures become significantly smaller. Hence, if the actual coordination increases at a constant rate with increasing number of nodes, the congruence measure would decrease. Because of this, existing measures may provide misleading information about changes in coordination over time in evolving communities. The advantage of the proposed clustering-based approach is that it is less sensitive to the increase in the network size and can be used for comparison purposes as the communities grow.

The three approaches for comparison discussed in this paper (link-link, link-cluster, and cluster-cluster) are based on different assumptions about the extent of indirect communication and shared knowledge. If the communities are strongly tied with each other and it can be assumed that there is significant indirect communication, then cluster-cluster comparison approach should be used. On the other hand, if the communities are such that coordination only happens through explicit communication, the link-link comparison is more applicable. In the intermediate scenarios, the link-cluster based comparison should be used. The availability of three approaches at different points in the spectrum provides analysts the flexibility to choose the approach which is more applicable in a particular scenario.

One of the limitations of the results presented in this paper is that we model the networks as undirected networks. More insights into coordination can be gained by modeling the communications and dependencies as directed networks and using clustering approaches for directed networks. Future research opportunity also includes utilizing different clustering algorithms for clustering the networks. In this paper, we let the clustering algorithm determine the number of clusters based on maximizing the modularity. The effect of number of clusters on the results of comparison is also an important avenue for future work.

5 ACKNOWLEDGEMENTS

This research was supported in parts by NSF grant IIS 0916463 and DOE award DE-SC-0006516 to Kalyanaraman. Le and Panchal acknowledge the financial support from the National Science Foundation through the CAREER grant # 0954447.

6 REFERENCES

[1] Van De Ven, A. H., Delbecq, A. L., and Koenig, R., Jr., 1976, "Determinants of Coordination Modes within Organizations," *American Sociological Review*, 41(2), pp. 322-338.

[2] Malone, T. W., 1988, "What Is Coordination Theory?," Coordination Theory Workshop, National Science Foundation, Report Number: SSM WP-2051-88.

[3] Malone, T. W. and Crowston, K., 1994, "The Interdisciplinary Study of Coordination," *ACM Computing Surveys (CSUR)*, 26(1), pp. 87-119.

[4] Herbsleb, J. D. and Grinter, R. E., 1999, "Architectures, Coordination, and Distance: Conway's Law and Beyond," *IEEE Software*, 16(5), pp. 63-70.

[5] Cataldo, M., Herbsleb, J. D., and Carley, K. M., 2008, *Socio-Technical Congruence: A Framework for Assessing the Impact of Technical and Work Dependencies on Software Development*, *Second ACM-IEEE international symposium on Empirical software engineering and measurement*: New York.

[6] Cataldo, M., Wagstrom, P. A., Herbsleb, J. D., and Carley, K. M., 2006, *Identification of Coordination Requirements: Implications for the Design of Collaboration and Awareness Tools*, *ACM Conf. on Computer-Supported Work*: Banff, Canada, pp. 353-362.

[7] Sosa, M. E., Eppinger, S. D., and Rowles, C. M., 2004, "The Misalignment of Product Architecture and Organizational Structure in Complex Product Development," *Management Science*, 50(12), pp. 1674-1689.

[8] Ehrlich, K., Helander, M., Valetto, G., Davies, S., and Williams, C., 2008, "An Analysis of Congruence Gaps and Their Effect on Distributed Software Development," in *Proceedings of the 1st International Workshop on Socio-Technical Congruence*, Leipzig, Germany.

[9] Conway, M. E., 1968, "How Do Committees Invent," *Datamation*, 14(5), pp. 28-31.

[10] Colfer, L. and Baldwin, C. Y., 2010, "The Mirroring Hypothesis: Theory, Evidence and Exceptions," in *Harvard Business School Finance Working Paper*. Paper Number: 10-058.

[11] MacCormack, A., Rusnak, J., Baldwin, G., 2005, "Exploring the Structure of Complex Software Desgins: An Empirical Study of Open Source and Proprietary Code," *Management Science*, 52(7), pp. 1015-1030.

[12] Camuffo, A. and Cabigiosu, A., 2011, "Beyond the Mirroring Hypothesis: Product Modularity and Interorganizational Relations in the Air Conditioning Industry," *Organization Science, in press*.

[13] Hoetker, G., 2006, "Do Modular Products Lead to Modular Organizations?," *Strategic Management Journal*, 27(6), pp. 501-518.

[14] Kratzer, J., Gemuenden, H. G., and Lettl, C., 2008, "Revealing Dynamics and Consequences of Fit and Misfit between Formal and Informal Networks in Multi-Institutional Product Development Collaborations," *Research Policy*, 37(8), pp. 1356-1370.

[15] Herbsleb, J. D., 2007, "Global Software Engineering: The Future of Socio-Technical Coordination," in *FOSE '07 2007 Future of Software Engineering*, IEEE Computer Society Washington, pp. 188-198.

[16] Sosa, M. E., 2008, "A Structured Approach to Predicting and Managing Technical Interactions in Software Development," *Research in Engineering Design*, 19(1), pp. 47-70.

[17] Kwan, I., Schroter, A., and Damian, D., 2011, "Does Socio-Technical Congruence Have an Effect on Software Build Success? A Study of Coordination in a Software Project," *Software Engineering, IEEE Transactions on*, 37(3), pp. 307-324.

[18] Blondel, V. D., Guillaume, J.-L., Lambiotte, R., Lefebvre, E., 2008, "Fast Unfolding of Communities in Large Networks," *Journal of Statictical mechanics: Theory and Experiment*, 2008, pp. 10008.

[19] Clauset, A., Newman, M. E. J., Moore, C., 2004, "Finding Community Structure in Very Large Networks," *Physical Review E*, 70(6), pp. 066111.

[20] Raghavan, U. N., Albert, R., and Kumara, S., 2007, "Near Linear Time Algorithm to Detect Community Structures in Large-Scale Networks," *Phys. Rev. E*, 76(3), pp. 036106.

[21] Newman, M. E. and Girvan, M., 2004, "Finding and Evaluating Community Structure in Networks," *Physics Review, E*, 69(2), pp. 026113.

[22] Gephi, *An Open Source Graph Visualization and Manipulation Software,* [cited 2012 February 16]; Web Link: gephi.org.

[23] Barber, M., Faria, M., and Strogan, O., 2008, "Searching for Communities in Bipartite Networks," *AIP Conference Proceedings*, 1021(1), pp. 171–182.

[24] Guimerà, R., Sales-Pardo, M., and Amaral, L. A. N., 2007, "Module Identification in Bipartite and Directed Networks," *Phys. Rev. E*, 76(3), pp. 036102.

[25] Murata, T., 2009, "Detecting Communities from Bipartite Networks Based on Bipartite Modularities," in *2009 International Conference on Computational Science and Engineering*, pp. 50–57.

[26] Rand, W. M., 1971, "Objective Criteria for the Evaluation of Clustering Motheds," *Journal of American Statistical Association*, 66(336), pp. 846-850.

[27] Huber, L., Arabie, P., 1985, "Comparing Partitions," *Journal of Classsification*, 2(1), pp. 193-218.

[28] Jaccard, P., 1901, "Étude Comparative De La Distribution Florale Dans Une Portion Des Alpes Et Des Jura," *Bulletin de la Société Vaudoise des Sciences Naturelles*, 37, pp. 547-549.

[29] Wallace, D. L., 1983, "A Method for Comparingtwo Hierarchical Clusterings: Comment," *Journal of the American Statistical Association*, pp. 569-579.

[30] Lerman, I. C., 1988, "Comparing Partitions (Mathematical and Statistical Aspects)," *Classification adn Related methods of Data Analysis*, pp. 121-131.

[31] Drupal, *Drupal: Community Plumbing,* 2011, [cited 2011 Jan. 24]; Web Link: http://www.drupal.org.

[32] Doxygen, *Generate Documentation from Source Code,* 2011, [cited 2011 Jan. 24]; Web Link: http://www.stack.nl/~dimitri/doxygen/index.html.

[33] Bolici, F., Howison, J., and Crowston, K., 2009, "Coordination without Discussion? Socio-Technical Congruence and Stigmergy in Free and Open Source Software Projects," in *Proc. Int'l Workshop Socio-Technical Congruence*, Vancouver, BC.