Self-organization versus hierarchy in open-source social networks

Sergi Valverde[1] and Ricard V. Solé[1,2]
[1]*ICREA-Complex Systems Laboratory, Universitat Pompeu Fabra, Dr. Aiguader 80, 08003 Barcelona, Spain*
[2]*Santa Fe Institute, 1399 Hyde Park Road, Sante Fe, New Mexico 87501, USA*
(Received 20 November 2006; revised manuscript received 12 March 2007; published 26 October 2007)

Complex networks emerge under different conditions including design (i.e., top-down decisions) through simple rules of growth and evolution. Such rules are typically local when dealing with biological systems and most social webs. An important deviation from such a scenario is provided by groups, collectives of agents engaged in technology development, such as open-source communities. Here we analyze their network structure, showing that it defines a complex weighted network with scaling laws at different levels, as measured by looking at e-mail exchanges. We also present a simple model of network growth involving nonlocal rules based on betweenness centrality. Our weighted network analysis suggests that a well-defined interplay between the overall goals of the community and the underlying hierarchical organization play a key role in shaping its dynamics.

DOI: 10.1103/PhysRevE.76.046118 PACS number(s): 89.75.Hc, 87.23.Kg, 89.65.−s, 05.10.−a

I. INTRODUCTION

Networks predate complexity, from biology and society to technology [1]. In many cases, large-scale, system-level properties emerge in a self-organized manner from local (bottom-up) interactions among network components. This is consistent with the general lack of global goals that pervade cellular webs or acquaintance networks. However, when dealing with human collective efforts towards a given objective, such as in a company or in distributed technology development, the situation can be rather different. Top-down decisions might dominate the structure and function in a hierarchical way; but how to distinguish between the two scenarios?

The intrinsic network organization of social interactions allows one to explore this question in depth. Many of these networks can be reconstructed by using e-mail exchanges among agents. The resulting graph provides a well-defined picture of the global community organization. By looking at its topology, we could in principle identify the presence (or absence) of self-organized (SO) or designed (top-down) patterns. Here SO refers to patterns emerging from local rules. Such a system would display global features resulting from a bottom-up dynamics. Eventually, a model of network growth can be proposed in order to explain the origin of such a pattern. An example of this is the work by Caldarelli *et al.* [2] who studied the emergence of weighted social networks. These authors showed that the structure of e-mail webs could be explained using a simple local mechanism based on positive feedback and reciprocity.

In this paper we explore the problem of how SO and hierarchy might actually emerge and coexist in a distributed community of technological developers. Specifically, we will present the first analysis of weighted open-source (OS) communities [3]. In OS communities, software is developed through distributed cooperation among many agents. These communities are known to display a large amount of distributed, bottom-up organization. Specifically, large groups of programmers are involved in building, assembling, and specially maintaining large-scale software structures. The community plays multiple roles as a design system but also as a distributed intelligence system able to accept or reject changes introduced by agents. As described, it looks like we are talking about a largely self-organized entity. Given the quality of the information available on their internal structure, OS organizations offer a unique opportunity to test if they are fully self-organized social groups [4] in constrast with more hierarchical, top-down organized social groups (i.e., large companies).

One possible test to these potential modes of community organization involves using the network of interaction between programmers working in a given software system. Software systems are themselves complex networks [5], which have been shown to display small world and scale-free architecture. Since the topological organization of software designs is scale-free, we might suspect that the community organization also displays common traits with the underlying software architecture. Previous work on engineering problem-solving networks involved in product development [6] revealed that these groups define a complex network with heterogeneous link distributions. However, these networks are unweighted and largely dominated by top-down constraints. Here, we consider a different type of engineering community where relations among agents are weighted and change in time without previously defined hierarchies.

As we will show here, OS networks (OSN) display scaling laws but also a well-defined core of main programmers defining a special subset of agents. Such finding suggests that, even in these distributed groups of individuals, emergence of hierarchy might be inevitable. Our analysis reveals the interplay between bottom-up, distributed decision making periphery in the OSN involving many agents and a top-down driven, centralized core of agents. Such rich-club structure seems to place some limits to the degree of distributedness achievable by multiagent-based technological design.

The paper is organized as follows. In Sec. II the data set is presented. In Sec. III several global network measures are presented. In Sec. IV the internal correlations are analyzed. Section V presents a nonlocal model that agrees with empirical observations. Finally, Sec. VI provides a discussion.

II. E-MAIL NETWORKS OF OPEN-SOURCE COMMUNITIES

Social network analysis depicts agents and their relationships with nodes and links, respectively [7]. Electronic exchanges allows tracking every social interaction and enables us to study highly detailed registers of human activities. Some remarkable examples of this are web surfing and e-mail communication. For example, e-mail is an important vehicle of communication and we can recover social interactions by analyzing all e-mails exchanged within a given community [8–11]. These e-mail studies recover the underlying social network by representing each agent with a node and a link indicates that e-mails have been exchanged between its end points.

We apply this methodology to the study of human interaction in the context of open source software projects, an interesting and poorly understood social phenomenon. Investigating OS social structure is useful to understand how human teams design complex engineering systems [12]. The study of OS communities is different from other studies of online communities [13]. Both communities are apparently quite similar if we look at how communication takes place (i.e., Internet-enabled communication). However, interaction in the OS community stems from the common goal of achieving a functional system, i.e., an OS software system, while communication in general web sites spans a broader range of interests and motivations.

Following Ref. [14], we have analyzed the structure and modeled the evolution of social interaction in OS communities [15]. We study a publicly available electronic database describing the e-mail activity in different open-source communities [14]. The e-mail data comes from the SourceForge (SF) web site, a large and popular OS project repository that hosts a very large number of OS software projects. This web site constitutes a centralized resource for managing software projects, issues, communication, and source code. The communication services offered by the SF store (and classify) all e-mail exchanges between project members in web pages. For example, there are web-based resources used to discuss development, software usage, and bug issues. These web pages can be searched by users to find all the previous e-mails regarding the problem they are trying to solve. From this collection of web pages, we have discarded all e-mails not directly related to the software process (i.e., personal issues, spam, etc.). We have limited our analysis to e-mail traffic associated to bug reports, which is a key feature of software development.

We have analyzed 120 OS networks corresponding to different software projects. We reconstruct the social network with the following method. For each OS network $\Omega=(V,L)$, nodes $v_i \in V$ depict community members while directed links $(i,j) \in L$ denote e-mail communication whether the member i replies to the member j. At time t, a member v_i discovers a new software error (bug) and sends a notification e-mail. Afterwards, other members investigate the origin of the software bug and eventually reply to the message, either explaining the solution or asking for more information. Here $E_{ij}(t)=1$ if developer i replies to developer j at time t and is zero otherwise. From E_{ij} we define link weight e_{ij} as the total amount of e-mail traffic flowing from developer i to developer j:

$$e_{ij} = \sum_{t=0}^{T} E_{ij}(t), \tag{1}$$

where T is the time span of software development. We have found that e-mail traffic is highly symmetric, i.e., $e_{ij} \approx e_{ji}$. In order to measure link symmetry, we introduce a weighted measure of link reciprocity [16] namely the *link weight reciprocity* ρ^w, defined as

$$\rho^w = \frac{\sum_{i \cong j} (e_{ij} - \bar{e})(e_{ji} - \bar{e})}{\sum_{i \cong j} (e_{ij} - \bar{e})^2}, \tag{2}$$

where $\bar{e}=\Sigma_{i \cong j} e_{ij}/N(N-1)$ is the average link weight. This coefficient enables us to differentiate between weighted reciprocal networks ($\rho^w > 0$) and weighted antireciprocal networks ($\rho^w < 0$). The neutral case is given by $\rho^w \approx 0$. All systems analyzed here display strong symmetry, with $\rho^w \approx 1$. This pattern can be explained in terms of *fair reciprocity* [2], where any member replies to every received e-mail. Thus we can make the simplifying assumption that the network is undirected.

However, we do not restrict our study to purely topological links. Instead, their weighted structure is also taken into account. The edge weight (interaction strength) is defined as $w_{ij}=e_{ij}+e_{ji}$, which provides a measure of traffic exchanges between any pair of members. From this weighted matrix we can estimate node strength [17] as a local measure defined as

$$s_i = \sum_j w_{ij}, \tag{3}$$

i.e., the total number of messages exchanged between node i and the rest of the community. This definition will be used below in our analysis of the weighted OS network.

III. TOPOLOGY OF OS NETWORKS

Figure 1 shows two social networks recovered with the above method. We can appreciate an heterogeneous pattern of e-mail interaction, where a few members handle the largest fraction of e-mail traffic generated by the OS community. The undirected degree distribution is roughly a power-law $P(k) \sim k^{-\gamma}$ with $\gamma \approx 2$ [see Fig. 2(b)]. However, $P(k)$ displays a hump at some intermmediate degree k_c [see Fig. 2(b)]. The hump suggests a two-level classification of nodes in the OS network: periphery nodes with few connections having $k < k_c$ and hub nodes having $k > k_c$. This desviation might be an indication of a rich-club ordering in the OS network (see below).

In order to understand the role played by hubs in OS networks, we have measured the betweenness centrality b_i (or node load [18]), i.e., the number of shortest paths passing through the ith node [19]. Betweenness centrality displays a long tail $P(b) \sim b^{-\delta}$ with an exponent δ between 1.3 and 1.8 [see Table I and also Fig. 2(c)]. It was shown that between-

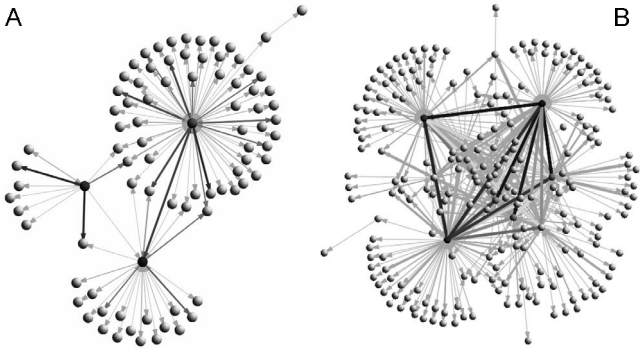

FIG. 1. Social networks of e-mail exchanges in open source communities. Line thickness represents the number of e-mails flowing from the sender to the receiver. Dark depicts active members and frequent communication. (a) Social network G_{Amavis} for the Amavis open-source community. (b) Social network G_{TCL} for the TCL (i.e., Tool Command Language) open-source community with $N=215$ members and $\langle k\rangle \approx 3$. In both networks, a few hubs (center dark nodes) route the bulk of information generated by many periphery nodes.

ness centrality scales with degree in the Internet autonomous systems and in the Barabási-Albert network [20], as $b(k) \sim k^{-\eta}$. From the cumulative degree distribution, i.e.,

$$P_>(k) = \int_k^\infty P(k)dk \sim k^{1-\gamma} \tag{4}$$

and the corresponding integrated betweenness, with $P_>(b) \sim b^{1-\delta}$, it follows that $\eta=(\gamma-1)/(\delta-1)$ [21]. The social networks studied here display a similar scaling law with an exponent η slightly departing from the theoretical prediction [see Fig. 2(a) and Table I]. The strong correlation between node load and large degree indicates that hubs tend to dominate e-mail discussions in the OS community.

In a previous work [22], we have studied different centrality measures for OS networks, including node outdegree and strength. In a weighted network, $s=\langle w\rangle k$ when there is no correlation between degree k and strength s and $\langle w\rangle$ is the average link weight. On the other hand, in the presence of correlations we will have $s(k) \sim k^\beta$ with $\beta>1$. Indeed, the latter is the case for OS networks, indicating that node strength is a better indication of node centrality than raw node degree. In the following section we will interpret this correlation in terms of a rich-club ordering of the OS network.

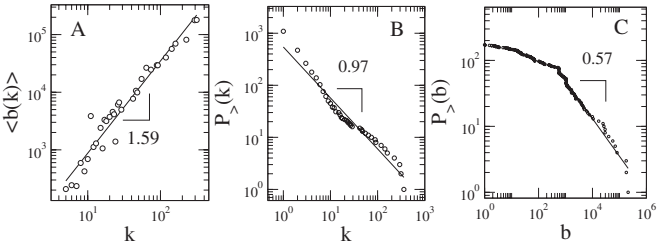

FIG. 2. (a) Average betweenness centrality scales with degree $\langle b(k)\rangle \sim k^\eta$ with $\eta \approx 1.59$ for the Python OS community. This exponent is close to the theoretical prediction $\eta_{BA} \approx (\gamma-1)/(\delta-1) = 1.70$ (see text). (b) Cumulative distribution of undirected degree $P_>(k) \sim k^{-\gamma+1}$ with $\gamma \approx 1.97$. (c) Cumulative distribution of betweenness centrality $P_>(b) \sim b^{-\delta+1}$ with $\delta \approx 1.57$ for $b > 10^2$.

IV. CORRELATIONS AND RICH-CLUB PHENOMENON IN OS NETWORKS

The above measurements provide a global picture of OSN but also suggest the presence of a two-level underlying structure, i.e., hubs and periphery nodes. In order to reveal such organization, we need to consider correlation measures among nodes having different numbers of links. We can detect the presence of node-node correlations by measuring the average nearest-neighbors degree:

$$k_{nn}(k) = \sum_{k'} k' P(k|k'), \tag{5}$$

where $P(k|k')$ is the conditional probability of having a link attached to nodes with degree k and k'. Here, the average nearest-neighbors degree decays as a power law, $\langle k_{nn}\rangle \sim k^{-\theta}$ with $\theta \approx 0.75$ for $k>10$ [see Fig. 3(a)].This decreasing behavior of $\langle k_{nn}(k)\rangle$ indicates that, on average, hubs tend to be connected to low degree nodes [see Fig. 1(a)]. That is, OS networks are good instances of disassortative networks. Moreover, the hierarchical nature of these graphs is well-illustrated from the scaling exhibited by the clustering $C(k)$ against k, which scales as $C(k) \sim 1/k$ (not shown), and consistently with theoretical predictions [23].

Following [17], we define the weighted average nearest-neighbors degree,

$$k^w_{nn,i} = \frac{1}{s_i} \sum_{j=1}^{k} w_{ij} k_j, \tag{6}$$

where neighbor degree k_j is weighted by the ratio (w_{ij}/s_i). According to this definition, $k^w_{nn,i} > k_{nn}$ if strong edges point

TABLE I. Topological measures performed over large OS weighted nets. The two last columns at left compare the observed η exponent with the theoretical prediction $\eta=(\gamma-1)/(\delta-1)$ (see text).

Project	N	L	ρ^w	$\langle k\rangle$	γ	δ	η	$(\gamma-1)/(\delta-1)$
Python	1090	3207	0.98	2.94	1.97	1.57	1.59	1.70
Gaim	1415	2692	0.98	1.9	1.97	1.8	1.24	1.21
Slashcode	643	1093	0.98	1.69	1.88	1.58	1.42	1.51
PCGEN	579	1654	0.98	2.85	2.04	1.67	1.54	1.55
TCL	215	590	0.98	2.74	1.97	1.33	2.34	2.93

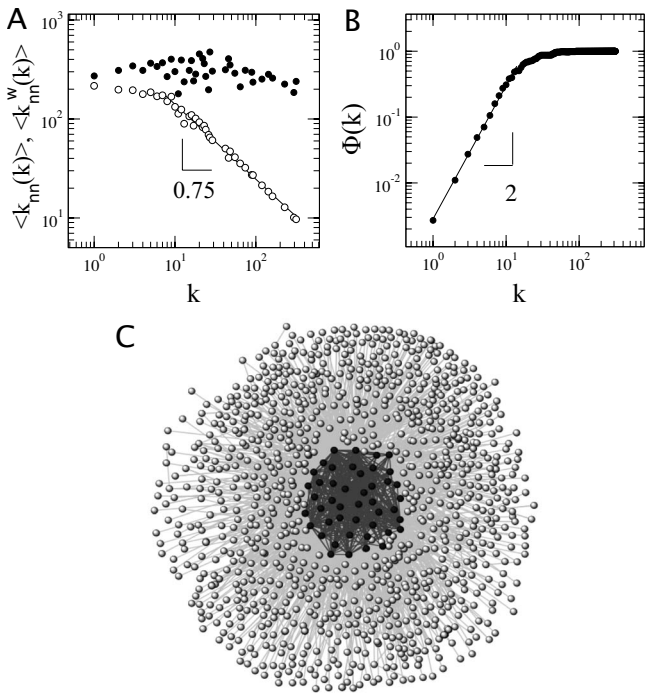

FIG. 3. Correlations and rich-club phenomenon in the Python OS community. (a) Average degree of nearest neighbors vs degree $\langle k_{nn}\rangle \sim k^{\theta}$ where $\theta \approx 0.75$ (open circles). Weighted average nearest-neighbors degree is almost constant (closed circles). (b) Rich-club coefficient $\Phi(k)$ scales with degree, $\Phi(k) \sim k^2$, for $k < k_c \approx 10$ while it is constant, $\Phi(k) \approx 1$, for $k > k_c$. The crossover k_c is consistent with the point of decreasing behavior of $\langle k_{nn}(k)\rangle$. (c) Visualization of the rich-club in the Python OS community, where dark nodes depict hubs with $k > k_c$. Peripherical nodes with $k < k_c$ (white balls) are mainly connected to hubs.

to neighbors with a large degree and $k^w_{nn,i} < k_{nn}$ otherwise. This measure captures more precisely the level of affinity between community members. Here, weighted average nearest-neighbors degree is almost uncorrelated with node degree, that is, $k_{nn,i} \approx$ const [see Fig. 3(a)]. Low connected nodes have weak edges because $k^w_{nn,i}(k)$ is only slightly larger than $k_{nn}(k)$ for small k. On the other hand, $k^w_{nn,i}(k) > k_{nn}(k)$ for large degrees, indicating that hubs have the strongest edges.

The above observations suggest the presence of rich-club ordering [24], where an elite group of highly connected and mutually communicating programmers control the flow of information generated by the OS community. For instance, during the development of the web-server software Apache, a closely-knit and small group of developers contributed about 90% of key changes, whereas the majority of developers contributed to marginal software features [25]. As we will show below, such core set of developers leaves a characteristic pattern in the social network.

The rich-club coefficient Φ has been used to assess the presence of the phenomenon in the Internet [24]. This coefficient measures when the hubs are on average more interconnected than the nodes with a smaller degree:

$$\Phi(k) = \frac{2E_{>k}}{N_{>k}(N_{>k}-1)}, \tag{7}$$

where $E_{>k}$ depicts the number of edges between the $N_{>k}$ nodes with a degree higher than k (i.e., hubs). $\Phi(k)$ indicates the ratio of the observed number of links out of all possible links between $N_{>k}$ nodes. This coefficient is an alternative measure of correlations that is non-trivially related to the average nearest-neighbors degree [see Eq. (5)]. In OS networks, $\Phi(k)$ increases for $k < k_c$ and saturates for $k > k_c$. The presence of a rich club is often associated to a monotonic increase in $\Phi(k)$ with k [26]. However, we argue that a better criteria to detect the rich club is the existence of a crossover k_c in $\Phi(k)$ characterizing the *rich nodes*. The crossover is consistent with the small hump in $P(k)$ (see Sec. III). For example, in the Python OS community, $k_c \approx 10$ [see Fig. 3(b)]. As an illustration, Fig. 3(c) highlights rich members in theOS Python community or hub nodes having more than $k > 10$ links.

We observe no rich club from the topological point of view (not shown), that is, the observed $\Phi(k)$ coincides with the expectation value $\Phi_{ran}(k)$ from maximal randomized networks having the same degree distribution [26]. However, the similarity between the rich-club ordering of the maximal random network and the OS network does not imply that OS communities lack a rich-club structure [27]. For example, the Internet contains a well-defined rich-club core despite that there is no difference between the $\Phi(k)$ measured in the Internet and the maximal randomized network [26]. The above comparison ignores link weights, perhaps discarding important information about the true organization of e-mail exchanges. Here we propose a measure that takes into account both link weights and topological features to assess the existence of a core subset of agents (i.e., the rich club) in the social network. We think that our rich-club measure is a robust method to detect the rich-club core in weighted networks.

An important difference between $\Phi(k)$ and our rich-club coefficient is that we extend the definition of the rich club to the subset of hubs, or nodes having degree larger than k, together with their *connectors*, or nodes with low connectivity that link two hubs (see Fig. 1 for a nice illustration of the hub-connector structure in OS networks). This node subset is the so-called k-scaffold graph or S_k [28]. The k-scaffold better captures the core for disassortative networks [like OS networks, see Fig. 3(a)] than the raw subset of k hubs used in $\Phi(k)$.

We define our weighted rich-club coefficient $\Phi(S_k,k)$ as follows:

$$\Phi(S_k,k) = \frac{W_S(k)}{E_S(k)\langle w\rangle}, \tag{8}$$

where $E_S(k)$ depicts the number of edges in the k-scaffold of the OS network, $\langle w\rangle = 1/E\Sigma_{ij}w_{ij}$ is the average edge weight for the full network, E is the total number of edges, and $W_S(k) = \Sigma_{i,j\in S(k)}w_{ij}$ is the sum of edge weights linking nodes in the k-scaffold subgraph [29]. The coefficient signals any deviation from a homogenous distribution of weights in the

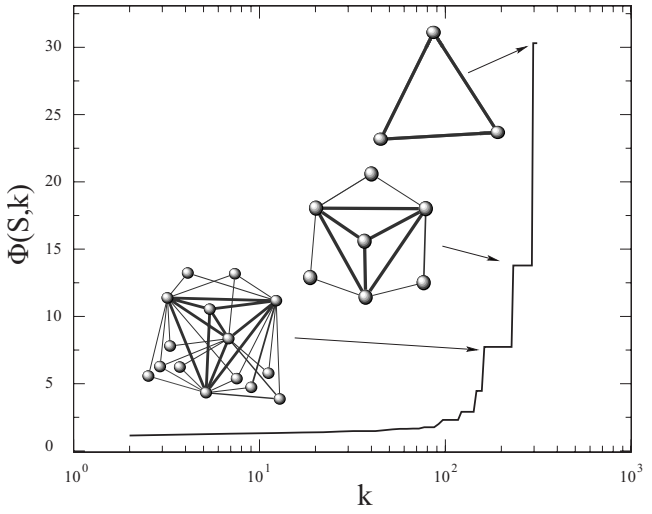

FIG. 4. Plot of the weighted rich-club coefficient $\Phi(S,k)$ against node degree k for the Python OS network. There is a significant deviation for $k>10^2$ that signals the rich-club ordering for this particular community. The subgraphs show the k-scaffolds or the predicted rich clubs for different degrees $k>100$. Line thickness indicates the weight attached to the link. We can appreciate how three nodes have a much more stronger internal interaction (i.e., exchange a larger number of e-mails) than with the rest of nodes.

k-scaffold. When weights are distributed at random then both the numerator and denominator will be the same and $\Phi(S,k)\approx 1$. However, it is easy to see that inhomogeneities in the weight distribution among edges (i.e., when large weights are clustered in the edges of some connected subgraph) yield $\Phi(S,k)\gg 1$. This seems to be the case for OS networks (see Fig. 4) where a dramatic growth of $\Phi(S_k,k)$ is observed when the core set of programmers is reached. Such divergence clearly reveals the nonhomogeneous nature of the OSN, where a large fraction of e-mails flows through a few OS hubs.

V. NONLOCAL EVOLUTION OF OS NETWORKS

Here, we assess the existence of top-down mechanisms in the evolution of OS communities. We will argue that OS networks are not self-organized systems and they require some level of centralization instead. A bottom-up system relies on local information to achieve a hierarchical organization. For instance, brains and social insect colonies are self-organized systems that operate in the absence of any central control, like a pacemaker, a leader, or an external template [30]. On the other hand, agents in a top-down driven system perform global computations. Here, we use betweenness centrality as a simple model for the computations performed by agents when selecting the target of communication.

Interestingly, our top-down model predicts the evolution and dynamics of the OS network, including the (undirected) degree distribution $P(k)$ and measurements of local correlations [see Figs. 5(c), 5(d) and 5(e)]. This model is motivated by three empirical observations. (i) There is a nonlinear relationship between node strength and degree (previously reported in Ref.[22]). In a related paper, this relationship has been explained with a betweenness centrality model [31]. (ii) Betweenness centrality strongly correlates with node strength [see Fig. 5(a)]. (iii) OS networks have a rich-club

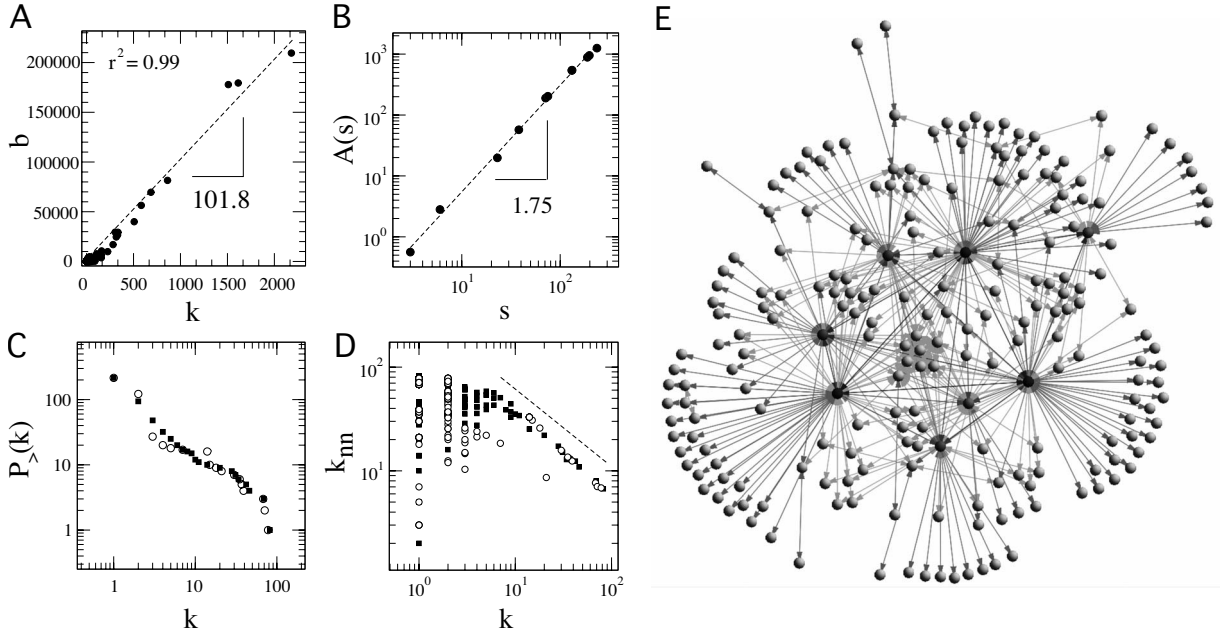

FIG. 5. Social network simulation. (a) Linear correlation between node strength s_i and betweenness centrality (or node load) b_i in the Python community. The correlation coefficient is 0.99. This trend has been observed in all communities studied here. (b) Estimation of α in the TCL community (see text). (c) Cumulative degree distribution in the simulated network (open circles) and in the real community (closed squares). All parameters estimated from real data: $N=215$, $m_0=15$, $\langle m\rangle=3$, and $\alpha=0.75$. (d) Scaling of average neighbors degree vs degree in the simulated network (open circles) and in the real social network (closed squares). There is very good overlap between the model and data for large k. (e) Rendering of the simulated OS network Ω to be compared with the OS network G_{TCL} in Fig. 1(b).

core (see above). The rich club indicates a characteristic scale in the system that emerges from an external reinforcement of core members' activities.

Core members will be more frequently e-mailed because of their importance. Key agents keep the community as a coherent system. In this context, agents exploit social cues to evaluate one another's social status [32]. A natural surrogate of social status is the number of e-mails posted (and received) by the member, i.e., node strength s_i (see Sec. II). Members earning high social status are arguably the most visible and thus they will be accessed much more frequently [33]. These key members have a global picture of the whole system, instead of being aware of just some specific parts of it. Members having a deeper knowledge of the overall system's architecture are likely to process high amounts of information. If we think in terms of agents in a network, we should expect them to canalize information flowing from many different parts of the network [34].

Taking into account the above, the algorithm for evolving the (undirected) social network $\Omega=(V,L)$ consists of the following stages. (i) The system starts (as in real OS systems) from a small fully connected network of m_0 members. (ii) A new member j joins the social network at each time step. The new member reports a small number of an average $\langle m\rangle$ of new e-mails (iii) For each new e-mail, we determine the target node by a nonlocal preferential attachment rule. The probability that new member j sends an e-mail to an existing member i is proportional to node betweenness b_i, or alternatively, to the node strength s_i (see below)

$$\Pi[b_i(t)] = \frac{[b_i(t)+c]^{\alpha}}{\sum_j [b_j(t)+c]^{\alpha}}, \tag{9}$$

where c is a constant (in our experiments, $c=1$) and betweenness b_i is recalculated before attaching the new link, that is, before evaluating the above equation. The exponent α varies from project to project (see below an empirical method to estimate this exponent from available data). Once the target node i is selected, we place the new edge in Ω, $\{i,j\}\in L$. Repeat steps (ii)–(iii) until the network reaches the target size of $N \gg m_0$ nodes.

The networks generated with the previous model are in very good agreement to real OS networks. For example, Fig. 5 compares our model with the social network of the TCL software community. The target social network has $N=215$ members and $m=\langle k\rangle \approx 3$. A simple modification of a known algorithm for measuring preferential attachment in evolving networks [35] enables us to estimate the exponent α driving the attachment rate of new links [described in Eq. (9)]. Due to limitations in available network data we have estimated the attachment kernel depending on node strength s_i instead of node betweenness b_i. Indeed, we have observed that strength s_i and betweenness centrality b_i in OS communities are linearly correlated [see Fig. 5(a)].

In order to measure $\Pi[s_i(t)]$ we will compute the fraction of links received by nodes having strength s_i at time t. This fraction [see Eq. (10) below] approximates the rate of attachment of new links, which we have hypothesized has the form described in Eq. (9) for OS networks. From the data, we compare two consecutive OS network snapshots of the same software community at times T_0 and T_1 where $T_0<T_1$. Nodes in the T_0 and T_1 network are called "T_0 nodes" and "T_1 nodes," respectively. When a new $i\in T_1$ node joins the network we compute the node strength s_j of the $j\in T_0$ node to which the new node i links. Then, we can estimate the attachment kernel as follows:

$$\Pi[s,T_0,T_1] = \frac{\sum_{i\in T_1, j\in T_0} m_{ij}\theta(s-s_j)}{\sum_{j\in T_0}\theta(s-s_j)}, \tag{10}$$

where $\theta(z)=1$ if $z=0$ and $\theta(z)=0$ otherwise, and m_{ij} is the adjacency matrix of the social network. Notice that in order to estimate the kernel we do not require any assumption about its functional form. In order to reduce the impact of noise fluctuations, we have estimated the α exponent from the cumulative function

$$A(s) = \int_0^s \Pi(s)ds. \tag{11}$$

Now, under the assumption of Eq. (9) the above function scales with node strength, $A(s)\sim s^{\alpha+1}$. Figure 5(b) displays the cumulative function $A(s)$ as measured in the TCL software community with $T_0=2003$ and $T_1=2004$. In this dataset, the power-law fitting of $A(s)$ predicts an exponent $\alpha=0.75$. A similar exponent is observed in other systems (not shown). In addition, we have estimated the α_{BA} exponent with a preferential attachment kernel, $\Pi(k)\sim k^{\alpha_{BA}}$, as in the original algorithm by Jeong *et al.* [35]. The evolution of the social networks cannot be described by a linear preferential attachment mechanism because the observed exponent is $\alpha_{BA}>1.4$ (not shown).

VI. DISCUSSION

Our analysis shows that open source communities are closer to the Internet and communication networks than to other social networks (e.g., the network of scientific collaborations). The social networks analyzed here are dissasortative from the topological point of view and assortative when edge weights are taken into account. This is consistent with the absence of topological rich club that is nonetheless detected when link weights are taken into account. The rich-club phenomenon in OS networks seems to be related to a pattern of nonlocal evolution. Such a nonlocal component appears to be related with the presence of a core of programmers that make decisions based on a global view of the system. Core programmers would both introduce a top-down control and receive a large amount of e-mail traffic from secondary members. Based on these ideas, we have presented a model that predicts many global and local social network measurements of the OS network.

We have shown that OS communities are elitarian clubs where strong hubs control the global flow of information generated by many peripherical individuals. Our conclusions are consistent with other qualitative observations of the

open-source phenomenon [36]. Quantitative evidence of elitism in distributed technological communities has been provided. The observed community organization indicates that even distributed systems develop internal hierarchies, thus suggesting that some amount of centralized, global knowledge might be inevitable.

OS communities constitute a previously unexplored example of online community. Other internet-based communities have been studied from a statistical physics perspective, including blogging [37] and bulletin board systems (BBS) [38]. Future work should address to what extent the current findings and modeling apply for other online communities. For instance, online communities have scale-free architecture, which emerges from heterogeneous members' behavior. Hub members in BBS and OS networks connect different communities in a weak manner but their links are strong [38]. On the other hand, the comparison of average nearest-neighbor degree function in the BBS network {see Fig. 2(D) in Ref. [38]} and in OS networks [see Fig. 3(a)] indicates different correlation properties. These differences might stem from the rich-club ordering of the network [27]. Core members of OS networks are externally reinforced by the rest of the community. Such an external driving of hubs might be less relevant for other online communities without a shared, clearly defined goal like in the OS networks (i.e., developing a technological product). In this context, the relative weighting of external and internal forces might account for differences in correlation properties [39] and enable us to assess the degree of self-organization in online communities [40].

A similar model to ours was presented in Ref. [31], where betweenness is recalculated only after the addition of a new node and its links. Here, recalculation of node loads represents a global process of information diffusion. The volume of e-mail traffic through any node correlates with the past experience and social context of this node, which we can compute with the betweenness-based attachment rule [Eq. (9)] [31]. We can conceive more detailed modeling approaches. For instance, we can simulate the flow of e-mails tracing shortest paths in the social network, as in some models of internet routing [41]. Packet transport-driven simulations can provide better estimations of the number of e-mails processed by any node. Still, the current model explains remarkably well many features of OS networks.

Finally, our results might be of interest in future explorations on the dynamics of so-called *computational ecologies* [42,43]. Computational ecology was defined as the study of the interactions that determine the behavior and resource utilization of computational agents in an open system. Early work by Huberman and Hogg in this area revealed that the collective behavior of such information-exchanging networks of agents can be very complex. The type of organization displayed by OSN seems to fit well with the goals of computational ecology theory. An important advantage is the explicit consideration of the real network topology reported here, which could help expand previous work on agent network dynamics.

ACKNOWLEDGMENTS

We thank Vincent Anton for useful discussions. We thank Bernat Corominas-Murtra for a careful reading of the manuscript and the anonymous referees for their useful comments and suggestions. This work has been supported by Grant No. FIS2004-05422, by the EU within the 6th Framework Program under Grant No. 001907 (DELIS), and by the Santa Fe Institute.

[1] S. N. Dorogovtsev and J. F. F. Mendes, *Evolution of Networks: From Biological Nets to the Internet and WWW* (Oxford University Press, New York, 2003); S. Boccaletti, V. Latora, Y. Moreno, M. Chavez, and D.-U. Hwang, Phys. Rep. **424**, 175 (2006); M. E. J. Newman, SIAM Rev. **45**, 167 (2003); R. Albert and A.-L. Barabási, Rev. Mod. Phys. **74**, 47 (2002).

[2] G. Caldarelli, F. Coccetti, and P. De Los Rios, Phys. Rev. E **70**, 027102 (2004).

[3] E. S. Raymond, First Monday **3**, 3 (1998), URL: http://www.firstmonday.org/issues/issue3_3/raymond/

[4] P. Ball, *Critical Mass: How One Thing Leads to Another* (Arrow Books, New York, 2004).

[5] S. Valverde, R. Ferrer-Cancho, and R. V. Solé, Europhys. Lett. **60**, 512 (2002); C. R. Myers, Phys. Rev. E **68**, 046116 (2003); S. Valverde and R. V. Solé, *ibid.* **72**, 026107 (2005); Europhys. Lett. **72**, 858 (2005).

[6] D. Braha and Y. Bar-Yam, Phys. Rev. E **69**, 016113 (2004).

[7] S. Wasserman and K. Faust, *Social Network Analysis* (Cambridge University Press, Cambridge, England, 1994).

[8] H. Ebel, L.-I. Mielsch, and S. Bornholdt, Phys. Rev. E **66**, 035103(R) (2002).

[9] B. A. Huberman and L. A. Adamic, in *Complex Networks*, edited by E. Ben-Naim *et al.* Lecture Notes in Physics Vol. 650 (Springer, Berlin, 2007), pp. 371–358.

[10] A. Arenas, L. Danon, A. Díaz-Guilera, P. M. Gleiser, and R. Guimerà, Eur. Phys. J. B **38**, 373 (2004).

[11] R. Guimerà, L. Danon, A. Díaz-Guilera, F. Giralt, and A. Arenas, Phys. Rev. E **68**, 065103(R) (2003).

[12] *Complex Engineering Systems: Science Meets Technology*, edited by D. Braha, A. Minai, Y. Bar-Yam, (Springer, New York, 2006).

[13] L. A. Adamic, O. Buyukkokten, and E. Adar, First Monday **8**, 6 (2003), URL: http://www.firstmonday.org/issues/issue8_6/adamic/

[14] K. Crowston and J. Howison, First Monday **10**, 2 (2005), URL: http://www.firstmonday.org/issues/issue10_2/crowston/; http://sourceforge.net

[15] S. Krishnamurthy, First Monday **7**, 6 (2002).

[16] D. Garlaschelli and M. I. Loffredo, Phys. Rev. Lett. **93**, 268701 (2004).

[17] A. Barrat, M. Barthélemy, R. Pastor-Satorras, and A. Vespignani, Proc. Natl. Acad. Sci. U.S.A. **101**, 3747 (2004).

[18] K. -I. Goh, B. Kahng, and D. Kim, Phys. Rev. Lett. **87**, 278701 (2001).

[19] U. Brandes, J. Math. Sociol. **25**, 163 (2001).
[20] A.-L. Barabási and R. Albert, Science **286**, 509 (1999).
[21] A. Vazquez, R. Pastor-Satorras, and A. Vespignani, Phys. Rev. E **65**, 066130 (2002).
[22] S. Valverde, G. Theraulaz, J. Gautrais, V. Fourcassié, and R. V. Solé, IEEE Intell. Syst. **21**, 36 (2006).
[23] S. N. Dorogovtsev, A. V. Goltsev, and J. F. F. Mendes, Phys. Rev. E **65**, 066122 (2002).
[24] S. Zhou and R. J. Mondragon, IEEE Commun. Lett. **8**, 180–182 (2004).
[25] A. Mockus, R. T. Fielding, and J. D. Herbsleb, ACM Trans. Softw. Eng. Methodol. **11**, 309 (2002).
[26] V. Colizza, A. Flammini, M. A. Serrano, and A. Vespignani, Nat. Phys. **2**, 110 (2006).
[27] S. Zhou and R. J. Mondragon, New J. Phys. **9**, 173 (2007).
[28] B. Corominas-Murtra, S. Valverde, C. Rodríguez-Caso, and R. V. Solé, Europhys. Lett. **77**, 18004 (2007).
[29] Here S_k denotes the naked minimal k-scaffold subgraph.
[30] E. Bonabeau, M. Dorigo, J. L. Deneubourg, S. Aron, and S. Camazine, TREE **12**, 188 (1997).
[31] K.-I. Goh, B. Kahng, and D. Kim, Phys. Rev. E **72**, 017103 (2005).
[32] D. Stewart, Am. Sociol. Rev. **70**, 823 (2005).
[33] D. J. Watts, P. Sheridan Dodds, and M. E. J. Newman, Science **296**, 5571 (2002).
[34] P. Sheridan Dodds, D. J. Watts, and C. F. Sabel, Proc. Natl. Acad. Sci. U.S.A. **100**, 12516 (2003).
[35] H. Jeong, Z. Néda, and A.-L. Barabási, Europhys. Lett. **61**, 567 (2003).
[36] J. Lerner and J. Tirole, J. Ind. Econ. **52**, 197 (2002).
[37] E. Adar, L. Zhang, L. A. Adamic, and R. M. Lukose, Proceedings of the WWW2004 Workshop on the Webblogging Ecosystem: Aggregation, Analysis, and Dynamics (ACM Press, New York, 2004).
[38] K.-I. Goh, Y.-H. Eom, H. Jeong, B. Kahng, and D. Kim, Phys. Rev. E **73**, 066123 (2006).
[39] S. Valverde, Europhys. Lett. **77**, 20002 (2007).
[40] H. J. Jensen, *Self-Organized Criticality: Emergent Complex Behavior in Physical and Biological Systems* (Cambridge Univ. Press, Cambridge, England, 1998).
[41] R. V. Solé and S. Valverde, Physica A **289**, 595 (2001).
[42] B. Huberman, *The Ecology of Computation* (Elsevier Science, New York, 1988).
[43] B. A. Huberman and T. Hogg, Artif. Intell. **37**, 155 (1987).