Self-Organization Patterns in Wasp and Open Source Communities

Sergi Valverde, *Pompeu Fabra University*

Guy Theraulaz, Jacques Gautrais, Vincent Fourcassié, *Université Paul Sabatier*

Ricard V. Solé, *Pompeu Fabra University and Santa Fe Institute*

Social network analysis shows that wasp colonies and open source software communities share statistical organization patterns. Studying these patterns reveals self-organizing processes that form social hierarchies.

In both nature and engineering, complex designs can emerge from distributed collective processes. In such cases, the agents involved—whether they are social insects or humans—have limited knowledge of the global pattern they're developing. Of course, insects and humans differ significantly in what the individual agent can know about the overall design goals. A social insect, for example, hasn't a clue about what it's contributing to the collective structure and function. In contrast, most software engineers working as part of a team understand their project's purpose and overall goal. Nonetheless, as project complexity increases, individual developers' real knowledge of the overall project rapidly shrinks; decisions become both localized and constrained by other project developments. The resulting constraints largely canalize choices, ultimately limiting the possible system-level construction rules—at least on some scales.

By viewing the complex dynamics of software development communities as a network of interacting agents involving both goals and constraints, we can compare them to other social networks and so build up evidence for basic principles of self-organization. Understanding these principles offers a first step toward quantitative reference models to explain human behavior during open source software (OSS) development. Once we have such a reference model, we'll be able to better manage the software process because we'll be able to clearly and quantitatively understand which deviations are important—and which are not. Such an understanding can benefit both software practitioners and information society in general. Existing OSS knowledge—which is based on a few qualitative studies—offers no general lessons.

We conducted a comparative study of how social organization takes place in a wasp colony and OSS developer communities. Both these systems display similar global organization patterns, such as hierarchies and clear labor divisions. As our analysis shows, both systems also define interacting agent networks with similar common features that reflect limited information sharing among agents. As far as we know, this is the first research study analyzing the patterns and functional significance of these systems' weighted-interaction networks. By illuminating the extent to which self-organization is responsible for patterns such as hierarchical structure, we can gain insight into the origins of organization in OSS communities.

Two social networks

Complex networks of interacting agents often display common organization patterns. Such regularities actually reflect common principles of organization[1] that are similar to patterns seen in nature.[2] Although social insect colonies involve simple agents with limited means of communication, pattern similarities to complex agent communities provide a basis for finding simple rules shared by both system types.

Wasp colonies

To uncover a wasp society's network structure, we conducted a set of experiments on two *Polistes*

1541-1672/06/$20.00 © 2006 IEEE
Published by the IEEE Computer Society

dominulus (European paper wasp) colonies. Within the colonies, we artificially maintained a constant number of wasps equal to $N_w = 13$. Unlike some wasps, European paper wasps are a primitively eusocial species—that is, they cooperate in the care of young; have a reproductive division of labor; and have overlapping generations (but no morphological differences among wasps), so offspring contribute to colony labor while their parents are still alive. Given these primitively eusocial characteristics, the wasps' behavior is flexible: individuals tend to adopt specialized roles determined by social interactions.

Hierarchical interactions play a crucial role in such social interactions, establishing a hierarchical structure that emerges from multiple exchanges. When wasps meet, they adopt either a dominant or submissive role (see figure 1). The dominance relationship between pairs of individuals in a colony is always stable. The entire set of colony pair-relationships forms a more or less linear hierarchy (though loops sometimes occur, in which wasps compete for dominance). Given this largely linear hierarchy, we can assign each wasp a particular hierarchical rank depending on how many wasps dominate it within the colony. In this particular species, an individual's hierarchical rank generally coincides with its birth order (and thus we see aging effects—that is, dominant wasps tend to be older wasps).

In our experiments, we removed top-rank individuals (α-individuals) in each colony and studied the subsequent reorganization of each colony's activity distribution. Each week, we removed the wasp occupying the top of the hierarchy. Seven days appears to be sufficient for a colony to stabilize a new hierarchical pattern. Our observation period lasted for 38 days (five weeks), and we recorded during two hour-long observation sessions each day. We observed the wasp's behavior visually and recorded data on a microcomputer with a keyboard customized for behavioral coding. We recorded all social contacts between pairs of individuals and measured the weight of each pair interaction in terms of the number of (directed) contacts.

Open source software community

Popular metaphors like "the cathedral and the bazaar" suggest that OSS development's distributed and unplanned nature outperforms planned schemes, such as proprietary software development.[3] OSS development advocates have also argued that decentralization leads to a distinctive organization that solves the communication bottleneck long associated with large software projects.[4]

To investigate such claims, we studied an OSS community's social network from a data set describing the email activity of 120 different software teams.[5] Our immediate goal was to understand how decentralization leads to hierarchies; we ultimately hope to understand how OSS labor divisions occur. Our test data originated from Sourceforge (http://sourceforge.net), a large open source project repository, and included communities ranging from very small networks with one or two members to large networks with thousands of members.

To determine an individual programmer's social position, we examined the email each programmer submitted to and received from the group. Because not every email message has the same influence in the software development process, we limited our consideration to email traffic associated with bug fixes and bug reporting. As other researchers have shown,[5] this email subset allows an effective reconstruction of the software community's social network.

Social network analysis

Social network analysis represents agent relationships with nodes and links.[6] Every node *i* represents an actor within the network; links (i, j) denote social ties between agents *i* and *j*. More representative social network models augment each link with the social tie's strength,[7] or the amount of information flowing through the tie. We refer to this as *link weight* ($w_{i,j}$).

Link weight analysis

The statistical analysis of $w_{i,j}$ between pairs of vertices in the social network indicates a heterogeneous interaction pattern, typically following a power law:

$$P(w_{i,j}) \sim w_{i,j}^{-\gamma}$$

where *P* is the probability of having a link with weight $w_{i,j}$. According to this pattern, a few ties are exploited with orders-of-magnitude-larger frequencies than many other social ties. However, researchers have also shown that weak ties enable fast information propagation in social networks.[7] Furthermore, heterogeneous link weight distribution might be related to a social network's hierarchical organization.[7]

Figure 1. Hierarchical interaction between wasps. When wasps meet, they adopt either a dominant or submissive role, creating a self-organizing hierarchy across the colony.

Study weights. Within the OS community's social network, nodes and links (i, j) represent email communication between members *i* to *j*, respectively. Anytime member *i* discovers a new software bug, he or she sends a notification email. Other expert members then investigate the bug's origin and eventually reply with the solution. Typically, several messages are required to solve the problem. Here, $E_{i,j}(t) = 1$ if developer *i* replies to developer *j* at time *t*, or $E_{i,j}(t) = 0$ otherwise. We also define $w_{i,j}$ as the amount of email traffic flowing from member *i* to member *j*:

$$w_{i,j} = \sum_{t=0}^{T} E_{i,j}(t)$$

where *T* is the software development time span.

In the wasp colonies' social network, nodes identify individual wasps and links represent hierarchical wasp interactions. Link weight $w_{i,j}$ indicates the number of dominances of wasp *i* over wasp *j*. Our experiments were limited by colony size. As we now describe, however, we gathered enough data to observe significant statistical correlations.

Comparison. Figure 2 compares social networks from wasp colonies and software communities, emphasizing link weight distributions, $P(w_{i,j})$. Figure 2a, for example, shows

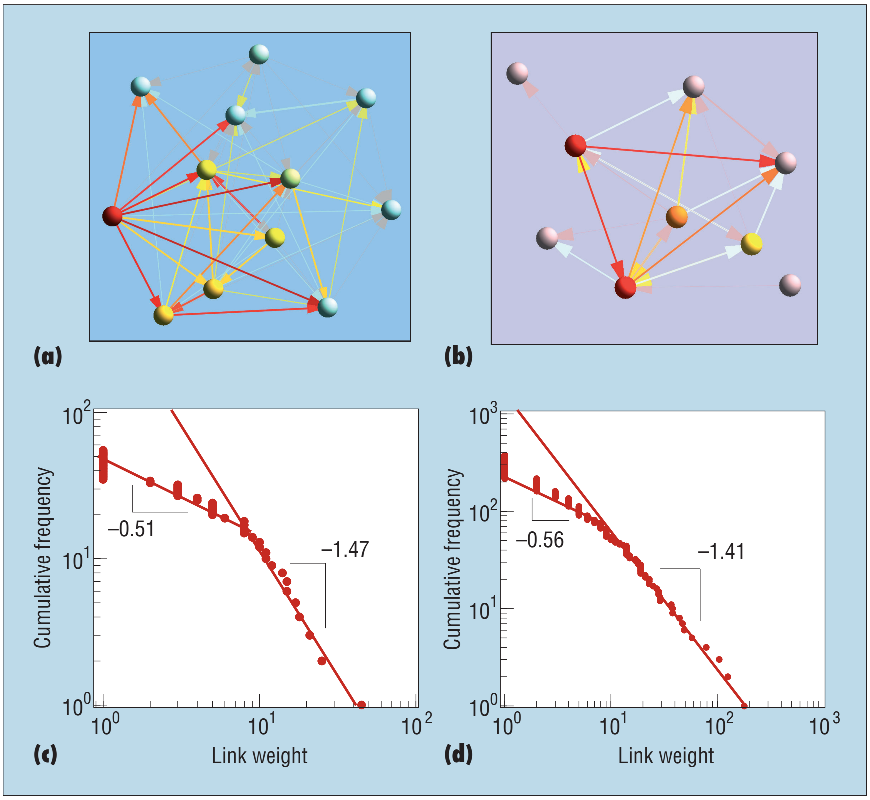


Figure 2. Heterogeneous interaction in wasp experiments and small software communities. (a) The network of hierarchical wasp interactions in a colony with 13 members. (b) Social network of email exchanges between developers in a small software community. (c) Cumulative distribution $P_>(w_{i,j})$ for a single wasp experiment. (d) Cumulative distribution $P_>(w_{i,j})$ within 12 small software communities. The tail of this distribution fits a scaling law, $P_>(w_{i,j}) \sim w_{i,j}^{-\gamma+1}$, where $\gamma = 2.41$.

a social network for a single experiment in a colony of 13 wasps. To reduce our statistical data's noise, we use the cumulative distribution $P_>(w_{i,j})$, defined as

$$P_>\left(w_{i,j}\right) \sim \int_{w_{i,j}}^{\infty} P(\omega)\, d(\omega)$$

For the standard case here—in which we observe a scaling behavior $P(w_{i,j}) \sim w_{i,j}^{-\gamma}$—we have $P_>(w_{i,j}) \sim w_{i,j}^{-\gamma+1}$.

Figure 2 shows a characteristic pattern of asymmetric interaction, in which a few strong wasps dominate the colony's activity.[8] Figure 2b shows a similar pattern for the small software community's social network. Beyond this qualitative comparison, we find significant agreement in the link weight distribution. To enable a quantitative comparison between human and wasp societies, we considered the aggregated link weight distribution $P(w_{i,j})$ in 12 small software communities with an average of 10 programmers each. Despite their small size and obvious differences, a comparison of figure 2c and figure 2d shows a significant convergence in link weight distributions between the wasp and small software communities. Interestingly, the link weight distribution in large software communities also follows a power law, with an exponent consistent with that observed in the small communities (see figure 3).

Measuring centrality: Strength and outdegree

There are limitations to what can we can understand solely by analyzing link weight distribution. A more informative system picture emerges from measuring node importance, or *centrality*.[9] If, for example, we compute a wasp's dominance index as the ratio of the number of dominances (DOM) over the total number of hierarchical interactions (DOM + SUB),[6] we get a highly reliable image of the wasp's hierarchical rank.[8]

Researchers have conjectured that many successful OS projects also display a hierarchical or onion-like organization. In many of these communities, core team members contribute most of the code and oversee the project's design and evolution. As Figure 3d shows, we can identify these core developers by assuming that members with many social ties are community leaders. Previous centrality studies of software communities[5] have focused in the node outdegree k_i—that is, the number of social ties outgoing from i. However, the outdegree might overlook important (though relatively isolated) members who connect separated subteams. We therefore use node strength,[10] s_i, as the centrality measure for weighted networks:

$$s_i = \sum_{j=1}^{N} w_{i,j}$$

In software communities, this equals the total number of emails that developer i sends. In the wasp colony, node strength coincides with DOM, so it's related to the dominance index used in biological studies of animal hierarchies.

As Figure 3a shows, in software communities, the distribution of programmer strength follows a power law $P(s) \sim s^{-\alpha}$. We can further investigate this power law's origin by measuring the dependence of node strength s with outdegree k,[10]

$$s(k) \sim k^{\beta}$$

When the exponent $\beta = 1$, the strength and outdegree don't correlate—that is, link weight $w_{i,j}$ is independent of i and j. In this case, both the outdegree and strength are equivalent measures and provide exactly the same centrality information. In software communities, however, β is significantly larger than 1 (see figure 3b), and thus node strength is a better centrality measure.

Feedback and self-organization

In animal society hierarchies, simple models of self-organization rely on a basic positive feedback mechanism, where a simple multiplicative rule reinforces successful individuals.[8] Similarly, we can define the probability of a software community's email interaction as a function of the total number of messages sent by interacting developers. As messages increase, so, too, does the likelihood of interaction. Interestingly, in many real weighted networks, a link's weight ($w_{i,j}$) scales with the product of its end nodes' outdegrees ($k_i k_j$).[10] In these real weighted network systems, we measure the dependency of average link weight with $k_i k_j$,

$$\langle w_{i,j} \rangle \sim \left(k_i k_j\right)^{\theta}$$

For the worldwide airport network and *Escherichia coli*'s metabolic network, researchers found that $\theta = 1/2$.[10] As our previous discussion implies**,** the θ and β exponents can be related. Assuming no topological correlations between connected vertices' outdegrees, $\beta = 1 + \theta$.[10] Then, in uncorrelated networks, $\beta = 1$ and $\theta = 0$. By measuring our data sets' θ exponent, we found $\theta > 0$ exponent, thus giving empirical evidence of a reinforcement mechanism in both our small wasp colony and large software community experiments (see figure 4). This is consistent with the $\beta > 1$ exponent measured in the scaling of strength with outdegree.

Comparing figures 4b and 4d suggests different reinforcement mechanisms in wasp and human hierarchies. In wasp colonies, a simple scaling law of the individual tendencies' product explains the average link weight. To reduce fluctuations and better capture the scaling exponent, we repeated the least-squares fitting with logarithmically binned data. For the wasp data set, we measured $\theta = 0.36$ exponent (figure 4b), which is consistent with the raw data set's $\theta = 0.39$ exponent (figure 4a). This simple hypothesis doesn't fit the software community's data, which shows strong nonlinearities.

As figure 4d shows, the logarithmically binned data is relatively flat for roughly two orders of magnitude, followed by a strong deviation with large $k_i k_j$. This pattern is difficult to see in the raw data set (figure 4c). Because the deviation is clear for at least two orders of magnitude, it's unlikely to result from noisy data fluctuations. This is a characteristic pattern in many software communities. The clear deviation suggests a pronounced reinforcement effect between the community's strongest members—the core developers—who have the largest outdegrees and node strengths (figure 4d).

Exploring communities of interacting systems through structural analysis of weighted networks is an open research area that requires more attention. The framework we've described for social weighted-network analysis could also play a key role in determining the mechanisms behind social self-organization. Sharing collective properties doesn't imply the same interaction mechanisms in the underlying organizations. As we've shown, in the OSS development community, reinforcement mechanisms distinguish a few core members. Arguably, these members might be qualitatively different from other community members. Selecting an appropriate model to explain these patterns remains an open research problem. The recently developed theoretical approaches for modeling insect societies[11] might offer valuable insight into many dynamic aspects of OSS software development. ■

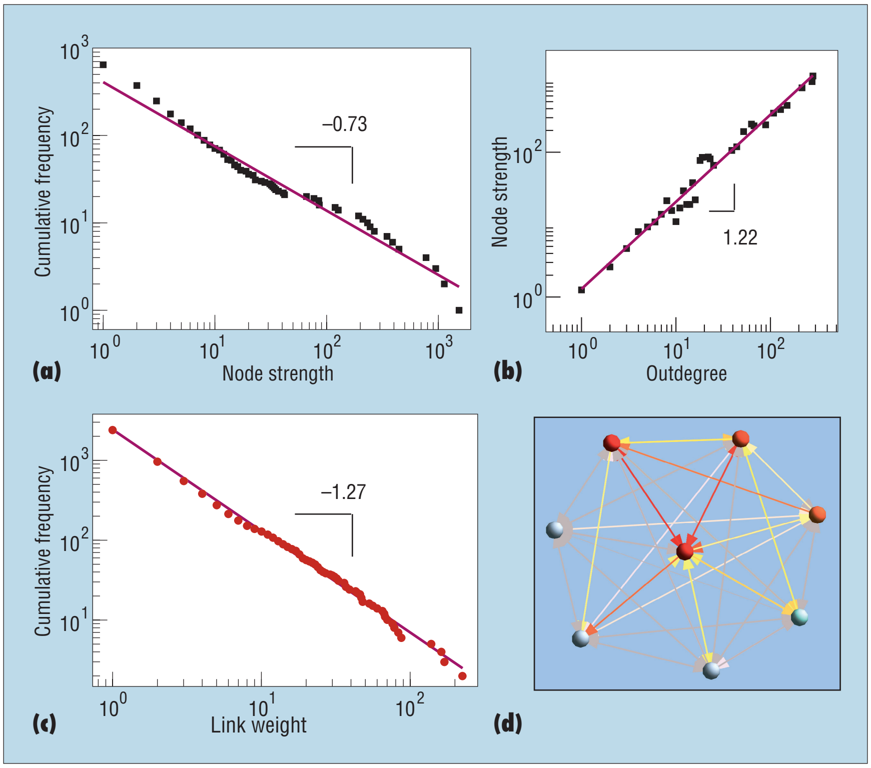


Figure 3. Analysis of a large software community. (a) Cumulative distribution for the strength $P_>(s)$, measured in the Python community, where s is node strength. The line denotes the least-squares power law fitting $P_>(s) \sim s^{-\alpha+1}$, with exponent $\alpha = 1.73$. (b) Average strength scales with outdegree—that is, $s(k) \sim k^{\beta}$, with exponent $\beta = 1.22$. (c) We can approximate the cumulative link weight distribution by a scaling law, $P_>(w_{i,j}) \sim w_{i,j}^{-\gamma+1}$, with a $\gamma \approx 2.27$ exponent. (d) A subgraph of email communication between the Python community's strongest developers (that is, those with $s > 200$ messages). The image show links between core members only; warmer nodes and links represent stronger developers and frequent email communications, respectively.

Acknowledgments

We thank Kevin Crowston and James Howison for making their software data publicly available. Our work was supported by grants FIS2004-0542; the EU's 6th Framework Program, contract 001907 Delis (Dynamically Evolving Large-Scale Information Systems) and 01194 Ecagents (Embodied and Communicating Agents); and the Santa Fe Institute. Jacques Gautrais was supported by a European community grant to the Leurre project under the Information Society Technologies Programme's Future and Emerging Technologies section, contract FET-OPEN-IST-2001-35506.

References

1. R.V. Solé et al., "Selection, Tinkering, and Emergence in Complex Networks," *Complexity*, vol. 8, no. 1, 2002, pp. 20–33.
2. R.V. Solé and B. Goodwin, *Signs of Life: How Complexity Pervades Biology*, Basic Books, 2001.
3. E.S. Raymond, "The Cathedral and the Bazaar," *First Monday*, vol. 3, no. 3, 1998; www.firstmonday.org/issues/issue3_3/raymond.
4. M.E. Conway, "How Committees Invent?" *Datamation*, vol. 14, no. 4, 1968, pp. 28–31.
5. K. Crowston and J. Howison, "The Social Structure of Free and Open Source Software Development," *First Monday*, vol. 10, no. 2,

2005; www.firstmonday.dk/ISSUES/issue10_2/crowston/index.html.

6. L. Pardi, "La 'Dominazione' e il Ciclo Ovario Annuale in Polistes Gallicus (L.)," *Ricerche sui Polistini VII*, *Boll. Ist. Entom*, Univ. Bologna, vol. 15, 1946, pp. 25–84.
7. M.S. Granovetter, "The Strengh of Weak Ties," *Am. J. Sociology*, vol. 78, no. 6, 1973, pp. 1360–1380.
8. G. Theraulaz, E. Bonabeau, and J.-L. Deneubourg, "Self-Organization of Hierarchies in Animal Societies," *J. Theorical Biology*, vol. 174, 1995, pp. 313–323.
9. S. Wasserman and K. Faust, *Social Network Analysis*, Cambridge Univ. Press, 1994.
10. A. Barrat et al., "The Architecture of Complex Weighted Networks," *Proc. Nat'l Academy of Science*, vol. 101, no. 11, 2004, pp. 3747–3752.
11. S. Camazine et al., *Self-Organization in Biological Systems*, Princeton Univ. Press, 2001.

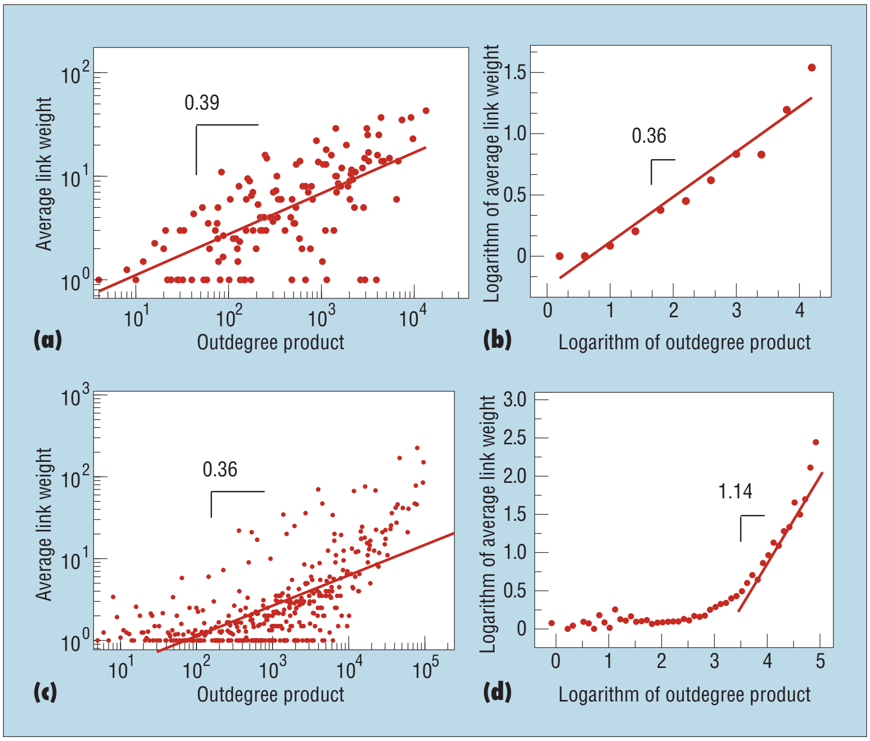
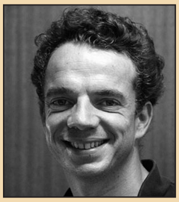


Figure 4. Dependence of link weight $w_{i,j}$ with the product of outdegrees $k_i k_j$ measured in an ensemble of small wasp colonies and a large software community. (a) In the five stabilized wasp patterns, least squares fitting of $<w_{i,j}> \sim (k_i k_j)^{\theta}$ yields an exponent $\theta \approx$ 0.39. (b) The same five patterns with logarithmically binned data to reduce fluctuations. The resulting scaling exponent, 0.36, is consistent with (a). (c) In the Python software project, the average link weight and $k_i k_j$ correlate. The straight line with slope 0.36 is the exponent for the power law fitting of the whole dataset, which shows fluctuations. (d) The Python project's logarithmically binned data is initially almost flat, followed by a persistent deviation. The dependence of average link weight with large outdegree product fits the exponent $\theta \approx$ 1.14, indicating strong correlations between highly connected nodes.

The Authors

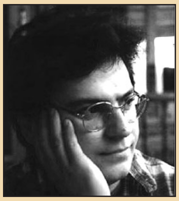
Sergi Valverde is a researcher in the Complex Systems Lab at the University Pompeu Fabra, Barcelona, and a doctoral student in applied physics and simulation in science at the Polytechnic University of Catalonia. His research focuses on complex networks and biologically based modeling of artificial systems, including the Internet and software systems. He received an MhD in computer science from Polytechnic University of Catalonia in 1999. Contact him at svalverde@imim.es; http://complex.upf.es/~sergi.

Guy Theraulaz is a senior research fellow at the Centre National de la Recherche Scientifique in Toulouse, where he heads the Research Center on Animal Cognition's research group on Collective Behaviours in Animal Societies. His research interests include collective decision making and building behavior in social insects and distributed adaptive algorithms inspired by social insects. He received a PhD in neurosciences and ethology from the Provence University, Marseille, France. In 1996, he was awarded the Centre National de la Recherche Scientifique's bronze medal for his work on swarm intelligence. Contact him at theraula@cict.fr; http://cognition.ups-tlse.fr/~theraulaz.



Jacques Gautrais is currently a researcher at the Research Center on Animal Cognition's research group on Collective Behaviours in Animal Societies at the University Paul Sabatier in Toulouse, France. His research focuses on modeling self-organized collective coordination in animals, such as schooling, behavioral synchronization, and unsupervised building. He received a PhD in cognitive sciences from the Ecole des Hautes Etudes en Sciences Sociales (EHESS). Contact him at gautrais@cict.fr.

Vincent Fourcassié is a research associate with the French Centre National de la Recherche Scientifique and works at the Paul Sabatier University's Research Center on Animal Cognition. His research interests are in orientation and decision making in animal societies. He received a PhD in biology from the University of Toulouse. Contact him at fourcass@cict.fr.



Ricard V. Solé is a research professor at the Universitat Pompeu Fabra, Barcelona, where he is the head the Institut Català per la Recerca i els Estudis Avançats (ICREA)-Complex Systems Lab. He is also an external professor at the Santa Fe Institute and senior member of Madrid's NASA-associated Astrobiology Center. He received a PhD in physics from Polytechnic University of Catalonia. Contact him at ricard.sole@upf.edu; http://complex.upf.es/~ricard.