

# Cyberconflict at the edge of chaos: Cryptohierarchies and self-organisation in the open-source movement

Athina Karatzogianni and George Michaelides

---

## Abstract

*This paper differentiates between different levels of conflict in the open-source movement and discusses the role conflict and self-organisation play in the emergence of structures of leadership emergence and the bifurcation into core and peripheral groups and soft control by cryptohierarchies; in the different levels of group polarisation and conflict between communities negotiating their identity, strategy, coordination and complexity; and lastly, in the dynamic relationships between hierarchies and networks. These dynamics are forcing open-source communities to exist at the edge of chaos, and to constantly engage in lines of flight and resistance from the system of global control, while ignoring current capitalist practices and 'growing their own' models of self-organising knowledge creation and exchange.*

---

## Introduction

In communities that exist at the interface between order and randomness (at the edge of chaos), conflict and crisis can act as a catalyst or a defence mechanism towards establishing governance structures or, failing that, disintegration. Conflict is a catalyst in the sense of enabling the morphosis of cryptohierarchies, and a defence mechanism in the sense of forcing communities to separate.

Conflict and crisis can result in different outcomes. For example, through negotiation and soft control, communities such as peer-to-peer networks can develop new structures in order to cope with conflict, creating core and periphery groups and cryptohierarchies. In another scenario, due to extreme group polarisation, the

community is unable to create new structures, but branches out and uses conflict as a defence mechanism to avoid centralisation. Or in the worst-case scenario, the community separates into two (forking the code), and there is no collaboration between original and fork, in which case conflict can be constructive or destructive depending on the evolution of the communities and groups involved.

From another angle, in this paper we differentiate between different levels of conflict and discuss the role conflict and self-organisation play in the emergence of structures, focusing on leadership emergence, the bifurcation into core and peripheral groups and soft control by cryptohierarchies (intra-communal cyberconflict); different levels of group polarisation and conflict between communities negotiating their identity, strategy, coordination and complexity (inter-communal cyberconflict); and lastly, the dynamic relationships between hierarchies and networks. These dynamics are forcing open-source communities and, more often than not, networked communities to exist at the edge of chaos, and to constantly engage in lines of flight and resistance from the system of global control, while ignoring current capitalist practices and 'growing their own' models of self-organising knowledge creation and exchange (meta-cyberconflict).

The other main purpose of this paper is to suggest, through examining issues of cryptohierarchies and the effects of self-organisation, that the open-source and/or free software movement is mistakenly romanticised as the ultimate democratic, egalitarian and horizontal system of governance, although a version of it might well replace democracy itself in the future, as the worst form of government except for the other ones that have been tried:

People often see in the open source software movement the politics that they would like to see — a libertarian reverie, a perfect meritocracy, a utopian gift culture that celebrates an economics of abundance instead of scarcity, a virtual or electronic existence proof of communitarian ideals, a political movement aimed at replacing obsolete nineteenth-century capitalist structures with 'new relations of production' more suited to the Information Age ... It is almost too easy to criticize some of the more lavish claims ... The hype should be partly forgiven ... Unlike the shooting star that was Napster, the roots of open source go back to the beginning of modern computing; it is a productive movement ultimately linked to the mainstream economy; and its developing and growing an increasingly self-conscious identification as a community that specifies its own norms and values. (Weber, 2004: 7)

Stefan Merten of Oekonux (the name is drawn from a combination of the words *oekonomie* [economics] and 'Linux'), a German mailing list discussing the revolutionary possibilities of free software, reassures us that 'conflict would no longer be built into the social system as it is today' (Merten, 2004, version 4). Similarly optimistic, Michel Bauwens (2007) of the P2P Foundation talks of peer-to-peer processes as bottom-up processes in which agents in a distributed network can freely and voluntarily engage in common pursuits without external coercion, which anyone can access, anyone can use, and any change to the commons belongs to the commons. Peer-governance-based leadership on reputational capital is the order of the day: 'Within the teams, decision making is participative and consensual, and the global coordination is voluntarily accepted and today technically feasible. Small tribes, the victims of civilizational hierarchies, are re-enabled in the new format of affinity-based cyber-collectives' (ibid. 2007). Post-monetary, post-democratic, post-capitalist modes of value and exchange embedded or not in the system are the answer and the solution to the structural crisis of contemporary capitalism (ibid., 2007). How is this system to come about? Bauwens proposes the following:

- Basic income in the private sector
- Open money systems in the sphere of the market
- Wealth acknowledgement systems (translating reputational wealth in income)
- Multi-stakeholderism (the inclusion in decision-making of participants who might be affected)
- A state retreat from the binary state/privatisation model in favour of a more neutral meta-regulatory system — a mix of government regulation, private market freedom, and autonomous civil society projects
- 'a political economy based on a true notion of scarcity in the material realm, and a realization of abundance in the immaterial realm' ... 'moving towards non-proprietary licenses, participatory modes of production, and commons-oriented property forms'.

These ideas raise many questions. What are the conditions like under peer-to-peer production? (See Dafermos and Soderbergh, this issue). What kind of subject exists and communicates under these conditions? What complexes of subjects, bodies and machines are required to accommodate the 'complex innovation and the subsequent need for creative and autonomous workers?' (See Moore and Taylor, this issue). Communities are evolving continuously, and new members cause depolarisation. On the down side,

structurelessness masks power within these communities. The equivalent in the open-source sphere could be the spectacularly skewed distribution of knowledge — the huge gap between core and peripheral developers' contributions. Is soft control by cryptohierarchies necessary in order to provide the social glue and facilitate the creation of technical infrastructures and decision-making mechanisms? If, in reality, all the ideals of the peer-to-peer movement remain unachieved, is that necessarily a bad thing? For example, unequal participation is natural in face-to-face interactions, and creates diversity in arguments and beliefs. Further, what are the acceptable levels of explicit coordination, power asymmetry and hierarchy? More and more, subsidiarity, multi-stakeholderism, expertise and reputation management are being discussed as components of the post-democratic, post-capitalist politics (see Orsi's and Arvidsson's distributions in this issue) that are emerging through the revolution underway in the governance and political economy of global communications. Is peer-to-peer revolutionary because it is sustainable and constantly self-organising?

### Main conflict plateaus

---

Weber argues that the open-source movement poses three interesting questions for political economy, and these can be summarised as follows:

- the *motivation* of individuals: why do talented programmers chose to spend time on a project for which they will not be compensated?
- *coordination*: 'how does the open source sustain coordinated cooperation among large numbers of contributors, outside the bounds of hierarchical or market mechanisms?'
- *complexity*: 'what is the nature of governance within the open source process that enables this community to manage the implications of Brooks's Law' — this states that when manpower is added to a software project, the project falls even further behind — 'and perform successfully with such complex systems?' (2004: 11–12).

Incorporating Weber's foci of analysis, the open-source community/ies and the socioeconomic and politico-economic cyberconflicts' (conflicts in computer-mediated environments) that arise therein can be categorised in the following way. First, as ultra creative, *intra-communal* conflicts between individuals in an open-source community. This can lead to much more diverse knowledge

creation or, in the worst-case scenario, to code forking. Forking, where the code is replicated and continued by another team of developers, is different to code branching. For instance, in the proprietary software Unix, different projects incompatible to each other exist (forking). In OSS Linux, official and experimental versions of software exist (branching). And next-generation software develops forking from the original one, such as Samba TNG. In most cases, technical and licence disagreements seem to cause the forks. OpenBSD forked from NetBSD 1.0; OpenSSH from SSH; and DragonFly BSD was forked from FreeBSD 4.8 by long-time FreeBSD developer Matt Dillon due to disagreements over FreeBSD 5's technical direction. Adempiere, a community-maintained fork of Compiere 2.5.3b, forked due to disagreements over the commercial and technical direction of Compiere Inc. NeoOffice is a fork of OpenOffice.org, with an incompatible license (GPL rather than LGPL) due to disagreements about licensing and about the best method with which to port OpenOffice.org to Mac OS X. On the matter of forking, the Jargon File hackers' glossary says:

Forking is considered a Bad Thing—not merely because it implies a lot of wasted effort in the future, but because forks tend to be accompanied by a great deal of strife and acrimony between the successor groups over issues of legitimacy, succession, and design direction. There is serious social pressure against forking. As a result, major forks (such as the Gnu-Emacs/XEmacs split, the fissioning of the 386BSD group into three daughter projects, and the short-lived GCC/EGCS split) are rare enough that they are remembered individually in hacker folklore. [from the Jargon File hacker glossary, cited on Wikipedia's 'Fork (software development)' page at <[en.wikipedia.org/wiki/Fork\\_\(software\\_development\)](http://en.wikipedia.org/wiki/Fork_(software_development))>]

The Wikipedia entry goes on,

It is easy to declare a fork, but it can require considerable effort to continue independent development and support. As such, forks without adequate resources can soon become inactive, e.g., GoneME, a fork of GNOME by a former developer, which was soon discontinued despite attracting some publicity. Some well-known forks have enjoyed great success, however, such as the X.Org X11 server, a fork from XFree86 which gained widespread support from developers

and users and notably sped up X development.  
[en.wikipedia.org/wiki/Fork\_(software\_development)]

What is interesting in the intra-communal conflicts are issues of personal freedom, the right to fork, ownership, leadership direction (e.g. BSD forks), competitive technical visions/ideologies, the reputational risk to the original project, and fork leader recruitment. For the purpose of this discussion, it is also interesting in respect of intra-communal conflicts to explore group polarisation, cryptohierarchies, and what Weber terms the 'winner-takes-all dynamic within certain kinds of open source projects' (2004: 160).

Secondly, *inter-communal* conflicts between different open-source communities raise questions of coordination (too much and too little), complexity (how much the community can handle), and ideology (different political visions for the open source, expressing inclinations ranging from anarcho-syndicalism to libertarianism and even to right-wing ideologies, for instance Free Software, which emphasises the freedom aspect, and Open Source Initiative, which establishes links with business. 'The relationship between the different communities can be cordial (e.g., Ubuntu and Debian), very bitter (X.Org Server and XFree86, or cdrtools and cdrkit) or none to speak of (most Linux distributions)' [en.wikipedia.org/wiki/Fork\_(software\_development)]).

In the bigger picture, there is a general conflict between the open-source community and aligned proprietary software companies supporting open-source initiatives against the Microsoft monopoly and its allies. Here, macro-organisational structures and the dynamics of the IT industry are important, as are questions of identity, strategy (framing) and structure (hierarchy vs. network or hybrid, such as in the Linux case, when Torvalds started rerouting submissions to lieutenants). Within this bigger picture, a meta-conflict occurs synchronously bringing all these different levels together and posing them in direct and intense contact and contrast to the current global system of capitalist accumulation.

---

### Cryptohierarchies, self-organisation and the edge of chaos

The 'edge of chaos' is defined as being the state of a system in which the system is undergoing a phase transition: i.e. its behaviour is shifting from one state to another. In social systems, 'edge of chaos' refers to the conceptual region between order and chaos, and refers to a system which is at a 'self organised' state. In open-source communities and possibly in other network structures, the edge of

chaos is captured in two ways in which the system can self-organise. First, open-source communities exhibit power law<sup>2</sup> distribution (see e.g. Healy & Schussman, 2003; Madey, Freeh & Tyran, 2005); and second, every successful community tends to be organised into a two-tier structure with a core and a periphery group (see Lee & Cole, 2003; Michaelides, 2006). The significance of these two forms of self-organisation in this discussion is that this is not only unavoidable, but also a necessary component to the success of the community. First, networks that follow power law distributions tend to be more robust and are more adaptable to environmental disturbances (see e.g. Barabasi, 2002). Second, the fact that communities tend to separate into core and periphery groups enables them to effectively exploit and integrate knowledge from diverse sources (Michaelides, 2006).

### Power law distributions

'Power law distributions' refers to distributions where the frequency of an event is inversely proportional to the magnitude of the event, so that  $freq(\chi) = \alpha + \beta\chi^\gamma$  where  $\alpha$ ,  $\beta$ , and  $\gamma$  are constants and  $\chi$  is the magnitude of an event. This includes phenomena such as stock market crashes, natural disasters, or the frequency of words in any text. A power law distribution of such events is typically considered to be evidence of self-organisation (Sprott, 2003).

In social networks such as open-source communities, power law distributions can be conceptualised in terms of the links and nodes of a network. In effect, the number of nodes with the same number of links is inversely proportional to the actual number of links each node has. With open source as a concrete example, the number of communities with any given number of developers is inversely proportional to that number: the majority of open-source projects have less than ten members, and only a handful have more than 100. A similar pattern exists for the connections between individuals, their contribution to mailing lists, and their contributions in terms of code (see Healy & Schussman, 2003; Madey, Freeh & Tyran, 2005; Michaelides, 2006).

In order to understand what this pattern signifies, we need to examine why there is a power law distribution to begin with. Two mechanisms are relevant here: the emergence of 'scale free networks' and self-organised criticality. The process of emergence of scale free networks (i.e. networks with power law distributions) is considered to be the result of two interlinked factors: a) that the network is constantly evolving and thus new nodes are added to the network; and b) that these nodes are added through preferential

attachment (Barabasi, 2002). In effect, this means that if newly added nodes follow a rule whereby they tend to link to nodes that are better connected, a scale free network will emerge. However, in the social network context this signifies nothing more than group polarisation. Individuals tend to preferentially attach to either the best or more knowledgeable developers, thus reinforcing the power law dynamics of the community. Unless there is some form of group polarisation to guide preferential attachment, then the community is less likely to be a scale free network.

The second mechanism through which power law distributions may emerge in open-source communities is what is known as ‘self-organised criticality’. The term was coined by Bak (1996) to explain the way order in nature manifests itself through cascades of structures. This can be easily explained using the example of pouring sand onto a sand-pile. At some point the structure will collapse, creating a smaller pile of sand next to it. What is interesting is that the distribution of avalanches as a function of the size of each avalanche is a power law. In an open-source community, it is knowledge that is ‘poured’ onto a pile of ideas. At some point these ideas may create such an avalanche, spawning a new project. Conflict is integral to this process. Often it is conflict between ideas or individuals that triggers an avalanche. This may manifest as code forking, or as the creation of sub-communities and different branches of code. Consider, for instance, the KDE community. It started out with the purpose of creating a desktop environment for Unix operating systems, then evolved through this process into a constellation of sub-communities creating a very large number of applications that are not necessarily part of a desktop environment — they range from a web-browser to an office suite and to scientific applications. This process effectively enables the community to coordinate itself and the large number of ideas that emerge as a result of the large participation. Had participation not been a power law distribution, and had the sub-projects and involvement in those project not been a power law distribution, the community would have collapsed under its own weight.

### The emergence of core and periphery

Every successful open-source community is organised into a two-tier structure of core and periphery. To this effect, it is important to examine how it emerges and why. From a simplistic point of view, a two-tier structure can be considered to be a reflection of the underlying power law distributions. Nevertheless, Michaelides (2006) found that the self-organising process that governs the



emergence of two-tier structures can be described in Kauffman's (1993, 1995) conceptualisation of a rugged landscape process. In effect, as the community evolves it reaches a point where it is no longer manageable as a single tier. Because of interconnectivity among developers, the coordination overheads scale exponentially, necessitating the creation of a second governing tier. By modelling this process as a cusp catastrophe model (see Thom, 1975), Michaelides showed that two parameters govern the question of who becomes a core developer: social interconnectivity and knowledge sharing. The amount of knowledge shared governs the asymmetry between core and periphery, while the level of social interconnectivity governs the rate of change through which a developer progresses from being a peripheral to a core member.

To this effect, Michaelides (2006) suggests that the community separates into core and periphery in order to reduce the coordination overheads; but more importantly, it separates into two different roles: those of exploration and exploitation. While there is a need to reduce the overheads that result from increased connections in the network, there is still a need for knowledge sharing. To this extent, the role of the periphery is to explore knowledge while the role of the group is to exploit it by selecting the best ideas and code (see also Lee and Cole, 2003).

---

### Group polarisation, leadership and soft control

Issues of leadership and soft control have been discussed so far in relation to self-organisation and the emergence of structure. Nevertheless, they are equally relevant to the different types of conflict occurring on the intra-communal, inter-communal and meta-conflict levels, when conflict becomes a catalyst for self-organisation or a defence mechanism against the emergence of cryptohierarchies in the form of core and periphery.

Group polarisation occurs when 'members of a deliberating group move toward a more extreme point in whatever direction is indicated by the members' predeliberation tendency' (Sunstein, 1999). Online communities tend to be more polarised: the bazaar empowers the louder and more aggressive individuals (Raymond, 1998), often exacerbating online conflicts and leaving out people who disagree while empowering people with a common cause. Having reached a critical mass, the opinion of the mediocrity gets adopted. This is directly linked to social cascades and cryptohierarchies both informational and reputational. Familiar and long-debated issues do not depolarise easily (and so in open-

source software, political/ideological issues do not depolarise easily, but technical issues do). Polarisation increases when the group defines itself by contrast to another group: when there is some sense of identity reinforcing group consensus, rather than complicating things, e.g. in the XFree86 fork X.org. On the other hand, depolarisation can occur due to external shock: new members, new arguments, new information. The lesson from group polarisation is that social homogeneity can be damaging to good deliberation, something proven by better knowledge exchange in communities in which conflict actually occurs.

Inter-communal conflicts occur, for instance with the free software vs. open-source software communities, simply because not all peer communities are the same and they do not have the same collective identities or strategies. More importantly, they certainly do not understand their ideological position (if they have one) in relation to other communities — that of the FLOSS movement as a whole, or the latter's role in the IT industry or in the global justice movement discourse — in the same way. This is also true because the FLOSS movement, or the peer revolution if you will, is a hybrid, a mosaic. It is partly a social movement, partly a formal organisation, partly a volunteer and virtual organisation, and partly a virtual community of practice (Healy & Schussman 2003; Michaelides, 2006). The stylised image of this movement is of an egalitarian network of developers free of hierarchical organisation and centralised control. However, adhering to power law distributions, participation is spectacularly stratified. Soft control, flaming and file-killing in the guise of quality control can be observed, sometimes in the guise of a 'We won't wait for your code'. Meanwhile, the distribution of projects can be skewed, and huge diversity exists even among successful projects.

Another parameter in relation to conflict in these communities and the constant threat of the fundamental right to fork is leadership. In fact, all three types of issues identified by Weber — who makes the final decision, who gets credit, and who can legitimately fork (2004: 89) — are ultimately connected to leadership (visible structure) and the core developers/cryptohierarchies (invisible decision-making) issue. Due to his excessive workload, in 1998 Linus Torvalds found himself unable to cope and to incorporate patches to the code in time. Programmers at Linux became frustrated and even started to doubt the capacity of Torvalds, their leader, to respond to them, which almost resulted in a major fork of the Linux code. A particular mirror site called VGER, set up by Dave Miller at Rutgers University, was incorporating patches that Torvalds was not, and an

argument erupted between the two, which was described by Raymond (the unofficial anthropologist of open source) as a test of the open-source movement under stress rather than as a personal battle between Torvalds and Miller or anyone else (Weber, 2004: 118). Subsequently, Bitkeeper commercial source-management software was adopted in order to resolve the workflow and organisational problems and take some pressure off Torvalds. When the main protagonists met in Silicon Valley in 1998, they agreed on what Weber calls a somewhat more formal pyramidal structure for the flow of patches and software submissions:

The key players had in effect, looked straight into the eye of a major fork and turned back from it. The heated conflict took place out in the open, on emailing lists accessible to the entire community. The resulting bargains and most of the negotiations were public. The vehemence of the conflict was de-escalated by a common language around technology. And the fight did not drag on forever; in fact the acute phase lasted less than a week. The conflict management system of then open source process was becoming more defined. (Weber, 2004: 119)

In other words, the number of patches submitted to Linus Torvalds reached a critical level in the self-organising criticality sense. As a consequence, the community had to restructure itself in order for Linus to be able to cope with the increasing number of patches being submitted. The result of this is a real hierarchy of decision-making, where Torvalds relies on 'lieutenants' who in turn rely on maintainers. It is not clear at any given time who is in which group. In BSD (Berkeley Software Distribution), governance is organised around concentric circles, while 'a small core group controls final access to the code base. This group grants (or revokes) the rights to the concentric circle, who can modify code or commit new code to the core base. These are the committers for evaluation. The boundaries of the circles are generally more definite: FreeBSD, for example, has a core of 16 and about 180 committers in the second circle' (Weber, 2004: 92).

Evidently, a strong hierarchical component is vital to successful OSS (see Jordan Hubbard & FreeBSD; Lee & Cole, 2003; Mockus, Fielding & Herbsled, 2002, 2005). Core developers are very well organised: 'not a formal organizational chart, but rather a status-based pecking order which is known to project participants and serves as a way of policing members'. OSS 'as virtual organizations' rely on mechanisms of social control and self-control, not on trust per se (Gallivan, 2001). Large-scale OSS projects are most often

staffed by professional software developers (though not always: see Netscape). Still, questions remain. How will the increased participation of commercial players influence the ideological issue of cryptohierarchies? The answer might be in the symbiosis of competitors, or in further forking and disintegration. For open-source communities to succeed, commercial players need to play with the same rules as everyone else. Only then can there be symbiotic relationships between individuals and/or organisations with different or competing interests. Lastly, is it the case that the closer a successful project is to the core of the OSS community, the more hierarchy will be found in its management style? Or is it rather a case of the larger the community, the more hierarchical? Professionalism, clear leadership and hierarchy are antithetical to the projected external image of the community, which brings us back to the hot political issue of the political romanticism related to the whole movement and the meta-conflicts this creates.

### Conclusion: Networks and hierarchies at the edge of chaos

---

What is really inspiring inside the political romanticism of cybercommunism, anarchism, libertarianism and ethical capitalism etc. is that these ideas play on the interface between hierarchies and networks, and on the increasingly dense relationships between the two. States are becoming more networked in order to deal with the current networked resistances, be they socio-political or ethno-religious, and within open source, politico-economic. These networked resistances are now more conscious of their hosting environment, reversing from networks towards cryptohierarchies in order to establish a better interface with established hierarchies ('The reversal argument', Karatzogianni, 2006).

What is also inspiring is the potential contained in network forms of social organisation as a basis for constructing resistances to repressive apparatuses and to the world system as a system of global control, yet not blind to the struggles and conflicts communities will necessarily experience when situated in the transition phase at the interface of order and randomness — when hanging at the edge of chaos (see Karatzogianni & Robinson, forthcoming 2009; Chesters, 2006). Weber (2004: 261) proposes the theorisation and comparison of instances of the current political and economic space: of the war on terror, the relationship between open-source and proprietary models of software production, and the politics between transnational NGO networks and international organisations. Both hierarchies and networks coexist,

and the interface between them makes for a fascinating study. Open-source, amongst other networked communities, has provided an empirical window onto the way the global system and its subsystems resolve problems of structural complexity, and how networks evolve, connect and create complex dynamics within diverse nodes and rhizomes. The self-organisation dynamic, sustaining this movement and creating order out of chaos, can help and is helping us to analyse networked movements, communities and resistances around the world and the dialogue between diverse systems of knowledge management, organisation, mobilisation, and leadership/decision-making structures. More importantly, despite the immaturity of its nascent modes of governance, the influence of this stubborn example of self-organisation upon the global justice movement looks likely to prove immense.

---

## Bibliography

- Bak, P. (1996) *How Nature Works* (Springer-Verlag).
- Barabási, A.-L. (2002) *Linked: How Everything Is Connected to Everything Else* (Plume Books).
- Bauwens, M. (2006) in R. Good, 'Network collaboration: Peer to peer as a new way of living — video interview with Michel Bauwens', available at <[www.masternewmedia.org](http://www.masternewmedia.org)>.
- Benkler, Y. (2006) *The Wealth of Networks* (Yale University Press).
- Chesters, G. (2006) *Complexity and Social Movements: Protest at the Edge of Chaos* (Routledge).
- Fligstein, N. (1996) 'Markets as politics: A politico-cultural approach to market institutions', *American Sociological Review*, vol. 61, no. 4, August, pp. 656–673.
- Galloway, A. (2004) *Protocol: How Control Exists after Decentralization* (MIT Press).
- Gallivan, M. (2001) 'Striking a balance between trust and control in a virtual organization: A content analysis of open source software', *Info Systems*, no. 11, pp. 227–304.
- Gereffi, G., J. Humphrey & T. Sturgeon (2005) 'The governance of value chains', *Review of International Political Economy*, vol. 12, no. 1, February, pp. 78–104.
- Gosh, R. et al. (2002) *FLOSS* (International Institute of Infonomics).
- Guastello, S. J., J. Craven, K. M. Zygowicz & B. R. Bock (2005) 'A rugged landscape model for self-organization and emergent leadership in creative problem solving and production groups', *Nonlinear Dynamics, Psychology, and Life Sciences*, vol. 9, no. 3, pp. 297–333.

- Guastello, S. J. & P. Philippe (1997) 'Dynamics in the development of large information exchange groups and virtual communities', *Nonlinear Dynamics, Psychology, and Life Sciences*, vol. 1, no. 2, pp. 123–149.
- Hardt, M. & A. Negri (2004) *Multitude* (Penguin).
- Healy, K. & A. Schussman (2004) 'The ecology of open-source software', 29 January; online at <opensource.mit.edu>.
- Karatzogianni, A. (2006) *The Politics of Cyberconflict: Security, Ethnoreligious and Sociopolitical Conflicts* (Routledge Research on Internet and Society).
- Karatzogianni, A. (2008) (ed.) *Cyber Conflict and Global Politics* (Routledge Military, Strategic and Security Studies).
- Karatzogianni, A. & A. Robinson (forthcoming, 2009) *Power, Resistance and Conflict in the Contemporary World: Social Movements, Networks and Hierarchies* (Routledge Advanced Series in International Relations and Global Politics).
- Kauffman, S. (1993) *The Origins of Order: Self-Organization and Selection in Evolution* (Oxford University Press).
- Kauffman, S. (1995) *At Home in the Universe: The Search for Laws of Self-Organization and Complexity* (Oxford University Press).
- Kuwabara, K. (2000) 'Linux: A bazaar at the edge of chaos', *First Monday*, at <<http://firstmonday.org>>.
- Merten, S. (2000, 2004) 'GNU/Linux: Milestone on the way to the GPL society'; at <<http://www.opentheory.org>>.
- Madey, G, V. Freech & R. Tynan (2002) 'The open source software development phenomenon: An analysis based on social network theory', Eight Americas Conference on Information Systems.
- Madey, G, V. Freech & R. Tynan (2005) 'Modelling and F/OSS community: A quantitative investigation', in S. Koch (ed.) *Free/Open Source Software Development* (Idea Publishing).
- Malone, T., J. Yates & R. Benjamin (1987) 'Electronic markets and electronic hierarchies', *Communications of the ACM*, vol. 30, no. 6.
- Michaelides, G. (2006) *Far From Equilibrium: Emergence, Self-Organization, and Diversity in the KDE Community*, Ph.D thesis, Business School, University of Nottingham.
- Myers, C. (2003) 'Software systems as complex networks: Structure, function, and evolvability of software collaboration graphs', *Physical Review*, no. 68.
- Osterloh, M., S. Rota, & B. Kuster (2002) 'Open source software production: Climbing on the shoulders of giants'; online at <<http://opensource.mit.edu>>.
- Peters, G. (1998) 'Managing horizontal government: The politics of co-ordination', *Public Administration*, vol. 76, Summer.
- Raymond, E. (1998) 'The cathedral and the bazaar', *First Monday*, vol. 3, no. 3; online at <[www.firstmonday.org](http://www.firstmonday.org)>.
- Sprott, J. C. (2003) *Chaos and Time Series Analysis* (Oxford University Press).

- Sunstein, C. (1999) 'The law of group polarization' (The Law School, University of Chicago).
- Thom, R. (1975) *Structural Stability and Morphogenesis: An Outline of General Theory of Models* (Reading).
- Weber, S. (2004) *The Success of Open Source* (Harvard University Press).
- Žižek, S. (2002) 'A cyberspace Lenin: Why not?' *International Socialism Journal*, no. 95; online at <<http://pubs.socialistreviewindex.org.uk>>.

---

Notes

- 1 For the cyberconflict model, and examples of ethno-religious, socio-political and cultural cyberconflicts, see Karatzogianni, 2006 and Karatzogianni (ed.), 2008.
- 2 Power law distribution in the internet literature comes up in the study of links on the internet, which is relatively new, and as Benkler points out 'if a tiny minority of sites gets a large number of links, then the vast majority gets few or no links, it will be very difficult to be seen unless you are on the highly visible site'. Not only that, but that emergent new hierarchy is becoming 'a more intractable challenge to the claim that the networked information economy will democratize the public sphere' (Benkler, 2000).