



Do penguins eat scallops?

Tony Cornford¹,
Claudio Ciborra¹ and
Maha Shaikh¹

¹Department of Information Systems, London School of Economics and Political Science, London, U.K.

Correspondence:
Tony Cornford, Department of Information Systems, London School of Economics and Political Science, Houghton Street, London, U.K.
E-mail: t.cornford@lse.ac.uk; m.i.shaikh@lse.ac.uk

Given that is in large part his words and ideas that this paper contains we include Claudio Ciborra as an author, located between us in the name order.

European Journal of Information Systems (2005) 14, 518–521.
doi:10.1057/palgrave.ejis.3000583

In mid 2003 the journal *Management Science* announced a call for papers for a special issue on Free/Open Source Software (F/OSS). The deadline for submission was 1 September 2004. As the call proposed:

Researchers in the social sciences have become aware of the very interesting characteristics of F/OSS and its development system, and a lively, interdisciplinary research community has sprung up to explore the phenomenon and its implications for management.

We (Tony Cornford and Maha Shaikh) had a long standing interest in open source and in particular in the enduring debates over the use of version control software (VCS) within the Linux kernel community, a debate that was (and still is) the basis for Maha's PhD research. The call for papers provided both of us with a general motivation to prepare and submit a manuscript. However, motivations in our case are fragile alignments that are often displaced, translated and diffused. For Claudio, then the Convenor (Chair) of our Department, our vague commitment was something to be worked on and reinforced. Indeed, as the department's Boss (a word he liked to use) Claudio exhibited a talent for an engaging (if at times enraging) form of Bossishness, and his at first gentle but increasingly insistent nagging about this putative paper disturbed us both through the early months of 2004. By the end of June 2004, just as he had departed from London for the start of his proposed Sabbatical year, he raised the heat a notch more.¹

This morning I talked to Maha about her progress on the Management Science paper. She told me where she stands. Given that the deadline is September 1st, here is how I can help. I could read Maha's material (case and abstract) to become familiar with the topic. [...] I can work on co-authoring the paper especially on the learning theory part. I think the MS opportunity is a precious one [...] It is now time for you to assess whether my involvement is necessary and advisable or if you think you can do otherwise. Now is the time to decide on this matter. Please let me know asap.

Best
Claudio

Our response was, of course, inclusive and we sent him our work so far and an outline of the paper we proposed to write

However, what was the paper we wished to write? It was, put simply (the only way we could put it at the time) an investigation of the role of version control software as a mechanism/machination for learning within the Linux community. We had data (through our analysis of the Linux Kernel Mailing List – LKML), a history and time line (e.g. of debates over use of CVS and BitKeeper, two competing VCS softwares with rather different histories), a general model of learning based on Bateson's ideas of levels and double binds ... and a general commitment to the use of ANTish

¹In this paper we have used text from a number of emails that we exchanged as the paper was written. We have in some, but not all, cases tidied up the spelling and formatting when it seems appropriate to convey better the meaning. As a general rule, if the spell checker can do it in a single click, it has been performed. If not then we have indicated the rephrasing or elisions by [].

To follow this text a few ideas and concepts need a brief introduction, although we assume that a reader has a general knowledge of open source and of Linux.

Version Control Software (VCS) – software to manage code and coding activity. Synonyms include source code management and configuration management.

BitKeeper – a closed source version control system, or in its own terms a distributed configuration management system. While targeted at open source developers, BitKeeper attracted a lot of criticism for its nonopen source license – hence spoken of as 'commercially crippled'. <http://lwn.net/1999/features/BitKeeper.php3>

CVS – an open source version control system commonly used by many open source development projects.

GPL- General Public Licence – The copyright license carried by GNU, EMACS as well as Linux granting reuse and reproduction rights to all on the condition that tools or software incorporating the code must be source-distributed on the same counter-commercial terms as GNU software (sometimes referred to as General Public Virus). The GPL is seen here as the Constitution of the collective. <http://www.gnu.org/copyleft/gpl.html>

Patch – A temporary addition to a piece of code, usually as a quick-and-dirty remedy to an existing bug or misfeature. A patch may or may not work, and may or may not eventually be incorporated permanently into the program. http://www.outpost9.com/reference/jargon/jargon_31.html#SEC38

Gateway – Part of the story of Bitkeeper – the commercially crippled VCS – was the establishment of a gateway that would allow code patches in BitKeeper to migrate through the gateway to the ideologically purer world of CVS. It did not work, was not used, and was abandoned.

thinking to explore the socio-technical assemblage – in particular through moments of translation (Callon, 1986). We also had a working title, 'The Machinations of Version Control Software to Contrive and Control Learning in F/OSS'. So far so pedestrian, some stuff, some postures, but not really a story worth the telling.

As we put it in an early email to Claudio, the paper we wanted to write was to be about learning. We could see much literature on F/OSS as an organisational form, and much on F/OSS as innovation, but the concept of learning had been seldom developed. It was often alluded to in general terms, for example, that participants have to learn how to behave, how to contribute etc., and that F/OSS communities over time 'learn' what works and what does not in terms of governance and organisation, and of course they 'learn' about their code through Linus' Law 'given enough eyeballs all bugs are shallow'. However, as we set to work on this paper, and with Claudio exercising his special ability to make studying IS interesting, funny and cumulative, we were soon displaced from our concern with learning, and led by Claudio to some rather bigger issues.

And so a new paper emerged through our email exchanges with Claudio now in Milan and in and out of hospital undergoing various treatments. It came to be based on three key ideas which Claudio with his constant playing with words/concepts called *vistas*. The first, as often in Claudio's work, was founded on a return to a classic model, in this case Galbraith's 30-year-old information processing view of the organisation (Galbraith, 1972). Seen in this way open source is a special case of organising (but not just or only), and the repertoire of actions available are, if not constrained by Galbraith's classic analysis, certainly influenced by the same dynamic. However, this is as Claudio said 'partial', we want to see further (after all this is just one *vista*), and beyond any general or conventional sense of an organisation. And so we move on to the politics of this collective and to

consider where the question of the future of the assemblage can be discussed (Latour, 2004). Finally, and again in line with Latour's work, this is seen as a grand experiment, and open source is conceived as a laboratory in which complex experiments are conducted. For Claudio the case then offered three contrasting *vistas*:

Vista 1: Formal organization of the Lab. Galbraith model, introduction of vertical info system. We have pretty much done that.

Vista 2: Community and Constitution. The whole case of reorganizing the lab with the introduction of the VCS (as prescribed by the first vista) highlights the political nature of decision making, related to the software tools too. Here ANT in a weak version as a stakeholder analysis that pays due respect to the technology actor using the scallops terms, if you like, but in a scaled down version, highlighting that BK infringes the constitution and for this is problematic. In regular hierarchical lab nobody would care about the brand of the microscope. But here it does matter.

Vista 3: the Lab as a social experiment: here it is how to go about continuing the experimentation. What needs to be changed – most probably the constitution. This should be done through the parliament of things (taking seriously the inclusion of code in making the decisions): no innovation without representation! I can see on the benches of this horrific parliament: bug fixing eyeballs; penguin patches; bugs (no, if I look carefully they are the humans!), VCS; Kernels; code modules; Linus; the lieutenants; Raymond; Stallman (no [they are] BIG BUGS!).

In other emails he returned to the question of the characterisations of open source, and the limits of the extant model of the Bazaar, reflecting first on Raymond's misleading counter-metaphor of the Cathedral.

First, Cathedrals, at least until well into the Renaissance were built by Bazaars of artisans, and were not signed by any chief architect. So, in a way they were the realization [of] OS arrangements, often without a dictator. In the Renaissance cathedrals began to be signed by chief architects and artists, and even then, huge Cathedrals like the Duomo of Florence has the Dome signed by Brunelleschi, but the tower is by Giotto

(a century before), or the Duomo of Milan – the expression of anonymous communities working across a couple of centuries.

The three vistas seemed to offer more.

What is then the Bazaar? We have seen it has elements of a formal organization buttressed by ICT. But this is a partial picture. Second, it is a community with a constitution that keeps the members glued together, gives legitimacy to the leadership, attributes power to the code, etc.

The third vista [...] is that of large social experiment (this may well explain lots of the behaviors...the good old Hawthorne effect!), the ones of which Bruno [Latour] tells we are involved in all these techno events we are immersed in from GM crops to ozone layer etc.

Even more precisely, it is a gigantic LABORATORY. And [this] is why STS studies apply to it so well: there are scientists, there are tools, there are controversies, there is the all micro ANT stuff we can apply at the scallops level. VCS is a big episode in Laboratory Life. Of course all the three vistas need to be kept together. [...] Here the Lab is on the internet no boundaries, lots of people participating etc. (That is the lab for the future, for genomics and other complex project will go through these kind of labs).

So, we have plenty of materials and insight to understand this OS lab's work, we can come up with the general model of this lab, better than Raymond's. I think the three vistas should suffice. The paper describes them, in the case of VCS, and the model is powerful enough to draw conclusions also of what Linus and his acolytes should do.

Claudio also explained how the open source community carries out laboratory-like experiments and why this route can lead to a more innovative understanding of open source and its Constitution.

The case shows: and here I am trying to get to the message of the paper, how VCS is important, both as an organizational technology (galbraith) and as a new actor (ANT analysis) that sits at this point next to Linus at the top of the hierarchy. Linux today is governed by a hybrid, socio technical entity (can we argue all that?).

Then there have been controversies. These would have no meaning in running a business firm adopting more effective tools, instead are of vital moment for the Linux community. Why? Because they touch its Constitution (issue of the revolt against the commercially crippled software BK)

[...]

There are signs and tensions that the Constitution is ripe for reform – (such as tolerating the immigrant/stranger: the commercially crippled software). But which is the body that should do it, the Parliament? The human members of the community, the hackers, plus the dictator and his lieutenants or,...an enlarged parliament composed of the hackers plus the spokespersons of the code,: ie of the bugs, of the patches, of the kernel, of the network systems, of the VCS...or in their (Linus & Co) practice the two things ALREADY coincide?

The question of a technology embedded into software work was, of course, nothing new to Claudio, something he had written about a decade earlier in introducing the formative context (Ciborra & Lanzara, 1994). However, as alluded to above, conventional ANTishness was not his style. Claudio was not a natural ANT proponent and insisted we must avoid 'another copycat ANT paper, a

theory [that has been] pulled out of circulation'. Not that he was without sympathy to such new sociotechnicalisms, but certainly the ways in which it had been used in Information Systems had failed to impress him.

Here is my problem with ANT. I went through a number of papers and they are unsatisfactory; applications in IS are [mechanistic and] lead quickly to [anthropomorphisation of] technology, exposing them and whoever does them to attacks of [all] sorts. [...] We apply it just to argue and show how considering vcs [as] just a tool is sooo narrow and misleading: try to drop THAT sort of tool and the whole Kernel development comes down. Linus at this point can be disposed of. A VCS not.

But I think we should also take seriously that the authors of ANT have call[ed] it back, possibly in front of the horrific applications seen around. As I read in [many papers] it is [just] a stakeholder analysis where the role and constraints posed by the technology are given the due importance.

This did not mean that he could not enter into an ANTish spirit:

Probably patches do not care about the commercially crippled software Bitkeeper, they want recognition and fast. Which means that if in the parliament of things we listen to their voice the GPL would be much less orthodox and Stallmanish that it is now. Patches, as is their nature I guess are programmed to be compromises, at least that is what their names says. A patch is a patch and behaves like a patch. Is not pure, it is bricolage at the end of the day, even if might be a very elegant piece of code.

That is also why the gateway has fallen into disuse. The gateway might be an obligatory passage point imposed by the fundamentalists of GPL, 'We need to have a pure way that fully conforms to the GPL'. Again patches do not care and they de facto do not go through the gateway: another sign of the power of the code (patch) over the power of the orthodox. Again another showing of Linus' wisdom of 'letting the code decide'.

What we can suggest to Linus is to formalize this power of the code/patch whatever, and celebrate it by holding a new session of the new parliament including humans ([...the purists who are so voiceful) confronting the spokespersons of the code/patches. On the agenda among other things: the revision of the GPL [...]. Probably the outcome will be something more hybrid, even more than current imbroglio in the text of the GPL, and a recognition of the now deeply socio-technical nature of the upper level of the Bazaar. It is Linus, the benevolent dictator, to a piece of software, who as things human often are, and as patches always, are [...] not pure race, but a wonderful metisse² [...]. In English you say what? – racially mixed.

Once our paper was almost written there arose the question of a title. This took a number of iterations to refine and included 'Do Penguins eat Scallops?' among those considered. The final title chosen was, 'Hierarchy, Laboratory and Collective: Unveiling Linux as Innovation, Machination and Constitution.' Machination had survived from the earliest draft! It was not easy to convince Claudio that the word collective better expressed the open source assemblage (better than community) although he did finally agree:

²Perhaps the best translation is as half breed, or echoing Harry Potter, half-blood.

Collective it turns out is THE term. Constituency has too much on electorate and too little on cooperation. Community is too strong. Collective is more assemblage but retains community AND cooperation. Finally is the term adopted by Bruno [Latour] to talk about his assemblages of humans and nonhumans.

Concluding

The paper was submitted, very late, but it got reviewed. Did the referees like it?

Yes! At the least they recognised the ambition

This paper offers an intriguing proposition: that version control software can act as agent in an actor network of open source software development. [...] The use of actor network theory also appears to be a fresh approach to studying open source 'collectives,' which are more typically conceived as communities or bazaar-like markets. ANT draws greater attention to the process of translation and its political implications, particularly the role that nonhuman agents play.

And

The questions the paper addresses are important, and the answers so far proposed by other researchers are not very

*satisfactory, thus this paper addresses a clear and present need. The conceptual formulation which draws from political science, and which builds on important insights (such as the suggestion that a volunteer organization working on a non-proprietary enterprise should not necessarily be constructed in accordance with theories developed primarily as descriptions of non-volunteer organizations whose *raison d'être* is the creation of proprietary advantage), is more much more imaginative than many other analyses of the same or similar questions.*

And No! There is such a thing as too much ambition, they said and the paper in its submitted version did not convince. The decision was not to accept it.

Our last communication with Claudio about the paper came after the referees had pronounced, and was upbeat, written by hand on the 6 February 2005

I think our paper is still very good (AND NEW), but requires lots of work.

More of this later

Best

Claudio

Alas, there was no more.

About the authors

Tony Cornford is a Senior Lecturer in Information Systems at the London School of Economics. His research interests include health informatics, open source software and systems implementation. He is the co-author (with Steve Smithson) of *Project Research in Information*

Systems: A Students Guide (London: Palgrave), a second edition of which has just been published.

Maha Shaikh is a Ph.D. student at the London School of Economics. Her research addresses issues of learning and organising within the Linux collective.

References

CALLON M (1986) Some elements of a sociology of translation: domestication of the scallops and the fishermen of St Brieuc Bay. *Power, action and belief*, J. Law London, Routledge & Kegan Paul, pp 196–233.

CIBORRA CU and LANZARA GF (1994) Formative contexts and Information Systems: Understanding the dynamics of innovation in organizations. *Accounting, Management and Information Technology* 2(4), 61–86.

GALBRAITH JR (1972) Organization Design: An Information Processing View. *Organization planning: cases and concepts*, J. LORSCH and R. LAWRENCE, Homewood, Ill, Irwin, pp 49–74.

LATOUR B (2004) *Politics of Nature: How to Bring the Sciences into Democracy*. Harvard University Press: Boston.