A Slice of PIE: Participation; Inclusion; Empowerment

Wiki and the *Agora*

Antony Bryant[1]

‘Looking back over the events of 2004, it is striking how many of the year’s disasters could have been avoided with better information and communication. For tens of thousands of people, disaster arrived suddenly, unannounced.’ ICRC Report 2005[2]

The ICRC Report drew attention to the informational and communicational aspects of disaster relief, planning and the like. The title of the report ‘**Information: a life-saving resource’** might be taken to imply that what is needed is a strategy to ensure that the internet and accompanying technologies reach across the world in ever more pervasive and accessible ways; then everything will be – if not perfect – then certainly better. The authors of the report, however, are at pains to point out that their concerns do not begin and end with the internet; the type of information, its form, and the ways in which it is gathered and broadcast are of paramount importance. They note that in many instances the formal, official, centrally-controlled information channels are far less effective than the informal, dispersed, locally-generated ones. They also stress that education, preparedness, trust and motivation are key factors affecting the ways in which people communicate with and understand one another – regardless of the technology.

The perspective that views information wholly or predominantly as a resource and a commodity neglects these key features. If information is viewed in this manner then the policies that follow are centred on concerns such as the amount of information – the more the better; ownership of information – and its value on the market; control of information – who can have access to it, who can disseminate it; and so on. The ICRC report notes several examples where informal, locally-based information proved to be far superior – on any measure – to that emanating from *official* sources.

> Vijayakumar Gunasekaran, based in Singapore, heard of the tsunami’s devastating impact on the radio early on the morning of 26 December. He phoned a warning through to his family in Nallavadu on the eastern coast

[1] Professor of Informatics, Leeds Metropolitan University, UK – a.bryant@leedsmet.ac.uk

[2] This may appear at first sight a strangely formulated observation, afterall disasters such as the Asian Tsunami of December 2004 were hardly *avoidable*; except in the sense that the aftermath could have been dealt with more effectively or the consequences significantly minimized with better planning and more resilient infrastructure. Yet despite all the horror stories associated with Hurricane Katrina and its aftermath, the magnitude of the destruction, disruption, and fatalities was significantly lower than that associated with the earthquake in Pakistan.

> of India, in time for villagers to evacuate all 3,630 residents to safety.
>
> In Tamil Nadu, the Indian state hit hardest by the tsunami, local civil society groups formed a coordination cell to capture people's priorities across 100 disaster-struck villages and report back on what aid officials were planning. Maintaining communication with affected people is a crucial way in which aid organizations can promote transparency, accountability and trust.

The ICRC report gives many further examples in addition to these two. Although the report does not mention the paradigm example from the Asian Tsunami: the case of Tilly Smith – a subject to which I shall return.

Policies or strategies for using ICTs have to take the factors mentioned above into account – that much is common knowledge amongst those involved in community-based ICT projects in areas such as Southern Africa. Many of the people reading this essay will have relevant first-hand experience, and so it would not be appropriate for me to offer insights and advice to such a readership. Yet what might prove useful is some consideration of pertinent issues from a slightly unfamiliar perspective. In this way it may be possible to generate further discussion with the result that new ideas might emerge about the ways in which ICTs can be harnessed in civil societies – many of which are far too obviously lacking in any form of civility.

The literature on ICTs in the 21st century is inherently bound up with the onward march of globalization. The term itself is highly contentious, but in this context it needs to be understood to involve free-floating share-holders having been released from any mutual pact with specific groups of workers or employees, since there is no longer any basis for anchoring a factory, warehouse, depot, or whatever to a particular location. If the costs and factors of production or distribution are more attractive – i.e. cheaper, less regulated, non-unionized – elsewhere then there is little reason to postpone relocation. The bonds that tied 19th and 20th century capitalists to a specific location and a local labour force have been broken. The rich and poor are no longer locked together by a mutual dependency that necessitates co-location; a situation that required physical interaction, compromise and agreement (see Bauman, 1998, p306). The new rich are free-floating.

This is a key feature of globalized production, and in essence it exemplifies the double-edged nature of globalization itself. This has been termed *glocalization* – a flexible and accommodating, global stage for a few; a constrained localized existence for the many. For the former the world is available to them in both real and virtual forms. Space matters less and less since they 'live in a perpetual present, going through a succession of episodes hygienically insulated from both their past and their future' (Bauman, 1998, p.306). The latter, the vast majority, are all too firmly anchored to physical space. Bauman has also characterized this division as one between *tourists and vagabonds*. The tourists can chose to travel, and if they do so they inhabit a world that is both *nowhere* and *anywhere* – the airports, the hotels, the restaurants and shops, the technology are designed to be almost entirely independent of any specific location; indeed this vacuous universality is often a key feature of the appeal of such amenities, and is extolled in their own marketing. The vagabonds, on the other hand, do not travel from choice, but from necessity. If the employment prospects or life-chances are better elsewhere, then those consigned to a locality may be tempted or impelled to move elsewhere – although increasingly such mass movements are actively discouraged as governments instigate policies to protect their domestic labour force and restrict immigration.

This is a summary of the negative aspects of globalization. A world where goods and services can travel the world during their production, distribution, consumption, obsolescence, and extinction; but where those fundamentally involved in the processes themselves are tied down to their locality – and where the attraction or repulsion of that locality is largely at the behest of forces outside the control of local or even national political forces.

One temptation is to seek refuge in the concept of *community*; seeing enforced localization as a good thing in itself, from which some new or reconstituted form of solidarity may emerge, facilitated by the potentialities of ICTs: Hence *community informatics* [CI]. Yet the problems with this aspiration arise from the very ubiquity of the term *community* itself. In the aftermath of just about any major, and all-too-often tragic event, there is near unanimous journalistic recourse to stories about how the 'local community comes together'; examples are never hard to find. Yet the very fact that such overt and repeated statements have to be made ought to act as a warning that perhaps the term itself is losing its meaning and its reference. This resonates with the observation attributed to Ralph Waldo Emerson – 'The louder he proclaimed his honour, the faster we counted the spoons.' With regard to communities in the 20th century, the historian Eric

Hobsbawm has observed, ‘never was the word *community* used more indiscriminately and emptily than in the decades when communities in the sociological sense became hard to find in real life’ (1994, p.428).

In the opening decade of the 21st century it is hard to demur from the argument that flows from such observations. The negative forces of globalization have exacerbated the community-destroying forces of the 19th and 20th centuries. Bauman has pointed out that there are two tendencies that have ‘accompanied modern capitalism throughout the whole of its history’, each aimed at creating communities in the aftermath of the massive social upheavals that destroyed the earlier forms of social solidarity. The first was ‘a consistent effort to replace the *natural understanding* of bygone community, the nature-regulated rhythm of farming and the tradition-regulated routine of the craftsman’s life by an artificially designed and coercively imposed and monitored routine. The second tendency was a much less consistent (and belatedly undertaken) attempt to resuscitate or create *ab nihilo* a *community feeling* this time within the framework of the new power structure.’ (2001, p.34). The first tendency was embodied in the awful reality of early factory production, and later resulted in the assembly-line and Taylorism. The second, largely in response to the first, led to efforts to build model villages around and integrated with factories – e.g. Saltaire, Port Sunlight, Bournville in England, and Robert Owen’s projects at New Lanark, Scotland and New Harmony, Indiana, USA. As Bauman notes, these communities did not survive, and the idea that workers might require a more meaningful existence in place of the inhuman routinization of Taylorist factories only re-appeared in the 1930s following Elton Mayo’s work and the emergence of ‘the human relations school’ of industrial sociology. Crucially the success of this latter approach was guaranteed once it became clear that ‘job satisfaction and a friendly atmosphere might go further than strict rule enforcement and ubiquitous surveillance in promoting efficiency at work’ (Bauman, 2001, p.37); in other words it turned out to be both more productive and less costly.

But this is not to allow that such human-centeredness might itself lead to a revival in communities, for Bauman’s argument is that modernity allows no trust in the spontaneous emergence of community; no alternatives can be permitted that might undermine the stability and order ‘designed using the power of reason and maintained by day-to-day monitoring and management’ (p.38). In the globalized context of 21st century modernity this monitoring and management can be done increasingly at a distance, and as was pointed out before, those doing the monitoring need no longer act *in loco parentis*; they can simply move on, leaving despair, destitution and disintegration in their wake. As Sennett has remarked, domination from the top has become shapeless while losing nothing of its strength. Moreover order can be maintained and authority exerted by the winners, but with a strategy of 'secession of the successful'.[3]

All these trends have come together to result in a view of the world that allows no alternative to the command-and-control model. The rich and powerful continue to exert their influence and domination in the sense characterized by Crozier in his analysis of bureaucracy (1964). They maximize their freedom of manoeuvre and opacity, while imposing strict rules and routines on subordinates. Those in control act as the prime sources of uncertainty of those under their control. This is the pessimist's view of globalization.

Yet there is a more optimistic view, one based on a model of community, or rather on cooperation: An approach that shouldn't work but appears to do so; albeit in specific, and for the most part virtual, contexts. As was stated earlier the issues of organizing through incorporation of ICTs are complex, and extend well beyond the specific technological artefacts themselves.[4] There are issues of co-ordination, management, perception, motivation and the like, but the lessons to be drawn from the realm of ICT are not limited simply to looking at how the technology can best be deployed. There are some exemplary lessons to be drawn from the ways in which the software development community itself has developed.

Borrowing something from software developers is in many regards a token return on a far larger conceptual loan. Computer engineers and particularly software engineers are some of the most experienced and successful of conceptual magpies; borrowing or stealing other people's ideas and applying them to their domains. This is often accomplished so successfully that the earlier meaning of concepts is effaced by the new ones; examples include *input*, *editor*, *desktop*, *program* and *icon*.

The term software engineer itself – and a whole host of associated terms such as requirements, specification, development project, life-cycle, design, construction, maintenance, and so on – have been lifted from the domain of general engineering and applied to aspects of software development. In some cases the application has been illuminating and beneficial; but there are others where it has proved far more ambiguous, if not a downright hindrance. This

[3] The phrase was coined by Robert Reich, and is used by Bauman as a chapter heading – see Bauman, 2001.

[4] It would in fact be more accurate to argue that the technololgical artefacts should themselves be seen as far more extensive and complex, not simply the stuff you can kick and kick-out – but that is an argument to be developed for another occasion.

even includes the term *engineering* itself, which when applied to software appears to have become part of the problem rather than offering a solution. F.P. Brooks – a true software guru – pointed out 20 years ago that software was better seen as nurtured and cultivated rather than built (1986). The construction metaphor had been an important conceptual breakthrough when first used; prior to that software had been thought as something one *wrote*; i.e. similar in many respects to a lone author working on a novel. The construction metaphor brought with it a panoply of issues and concerns that previously had perhaps not been at the forefront of software developers' priorities. As Brooks (1986) summarized it –

> I still remember the jolt I felt in 1958 when I first heard a friend talk about *building* a program, as opposed to *writing* one. In a flash he broadened my whole view of the software process. The metaphor shift was powerful, and accurate. Today we understand how like other building processes the construction of software is, and we freely use other elements of the metaphor, such as *specifications*, *assembly of components*, and *scaffolding*.

In place of the engineering or construction metaphor, Brooks proposed the metaphor of nurture and cultivation. More than that he offered an outline of how the development process might be seen as an incremental strategy leading from the simple to the complex.

> Let us turn [to] nature and study complexity in living things, instead of just the dead works of man. Here we find constructs whose complexities thrill us with awe. The brain alone is intricate beyond mapping, powerful beyond imitation, rich in diversity, self-protecting, and self-renewing. The secret is that it is grown, not built.
>
> So it must be with our software-systems. Some years ago Harlan Mills proposed that any software system should be grown by incremental development.[5] That is, the system should first be made to run, even if it does nothing useful except call the proper set of dummy subprograms. Then, bit by bit, it should be fleshed out, with the subprograms in turn being developed--into actions or calls to empty stubs in the level below.

[5] H.D. Mills, 'Top-Down Programming in Large Systems', in *Debugging Techniques in Large Systems*, R. Ruskin, ed., Prentice-Hall, 1971

In arguing for a move from viewing software development as a large-scale engineering project – almost inevitably managed in a command-and-control manner – to one that treats such endeavours in a far more unordered and emergent fashion, Brooks gave his readers a glimpse of an alternative model; albeit something more wraithlike than substantial. Yet within ten years this aspiration began to take form with the emergence of the open-source movement of software development.

The open-source movement actually predates Brooks' article, since its founding figure, Richard Stallman, set up the *Free Software Foundation* in 1985; having previously worked on several co-operative software projects in various US universities and organizations. But the real breakthrough came in the 1990s with the work associated with the *Linux* operating system, inspired by Linus Torvalds. Stallman is still regarded as the 'hacker in chief'; so famous within this group that he is known simply by his programming acronym RMS.[6] Torvalds is the key figure associated with the Linux model of development and collaboration. Eric Raymond's writings might then be regarded as the manifesto of the movement, in particular his paper which introduced the contrast between the *cathedral* and the *bazaar* (1997).

For Raymond the cathedral model is one relying on the efforts of 'individual wizards or small bands of mages working in splendid isolation'; the entire edifice or product needs to be completed and fully guaranteed or secured prior to its 'release'. The alternative 'resembles a great babbling bazaar of differing agendas and approaches (aptly symbolized by the Linux archive sites, who'd take submissions from *anyone*) out of which a coherent and stable system could seemingly emerge only by a succession of miracles'.

Raymond's argument centres on software development, more specifically on software debugging – the process of locating and fixing problems in software-based systems: A process that is truly endless in all commercial systems. The cathedral model relies on a small group of proficient developers working in splendid isolation, only releasing their software to users after extensive and thorough testing – all of which takes time and effort. In stark contrast stands the bazaar-like model, whereby disparate groups and individuals with differing agendas and approaches somehow produce a coherent and stable outcome. The Linux philosophy is encapsulated in Linus Torvalds' philosophy as stated by Raymond – 'release early

[6] The term 'hacker' is used here in its original meaning of someone who *hacks code* – i.e. someone who produces and modifies computer software. So hacker is term of approval, indicating that the person is skilled and experienced at the complex task of software development.

and release often; delegate everything you can, be open to the point of promiscuity'. The result ought to be chaotic and anarchic, a configurational nightmare with contending versions of software proliferating to the consternation of developers and the despair of users and customers. Yet precisely the opposite has occurred. Linux has survived, thrived and continues to flourish. Moreover the model has been adopted by others, and Raymond's paper – later republished as the title chapter in his book – can be taken as a manifesto for this type of co-operative venture; albeit that he crucially undersells his own analysis as will explained later.

At a more general level the lesson of the bazaar approach is that complex development projects reach effective and sustained outcomes when involvement is voluntary and collaborative. There is control and management, but it is enacted in a distributed and semi-autonomous fashion. People take on tasks and responsibilities because they feel motivated to contribute and exchange their views, their ideas and their efforts. The development of open-source software – specifically Linux – has come about on a model based on several principles that at first sight ought not to prove effective or successful; but they have, and they do. The question at this point is can such principles of operation be abstracted from the realm of software development and applied elsewhere?

Raymond offers a number of maxims or aphorisms – nineteen in all – to summarize this approach. Not all of these translate readily to the domain of community informatics, but some certainly are useful in helping us to understand how the open-source model might offer insights and a grounding for CI projects and initiatives.

Raymond's first maxim is: *'Every good work of software starts by scratching a developer's personal itch.'* The key issue here, and one that comes through very clearly in Raymond's description of his own participation in the open-source community, is that involvement must at least owe some of its initial impetus to the motivation and enthusiasm of the participant. This is not to subscribe to the simplistic appropriation of the concept of volunteering based on an all-too-often willful misunderstanding of Drucker's purported maxim that 'in the knowledge society we are all volunteers'. It should be obvious that most of the working population are not engaged in their particular employment voluntarily. The largest private employer in the US is Wal-Mart, and few of their 1.2 million employees would consider themselves 'volunteers', although the company itself uses the term 'associate' for all its employees whether CEO or checkout assistant.[7]

[7] For an interesting series of statistics about Wal-Mart see (http://www.pbs.org/wgbh/pages/frontline/shows/walmart/secrets/stats.html)

Drucker's maxim has a crucial corollary: 'Everyone in the knowledge economy is a volunteer, *but we have only trained our managers to manage conscripts*'.[8] For our present purposes the key points are developed by Drucker himself in an article for Forbes Magazine in the late 1990s.

> What motivates workers -- especially knowledge workers -- is what motivates volunteers. Volunteers, we know, have to get more satisfaction from their work than paid employees precisely because they do not get a pay check. They need, above all, challenge. They need to know the organization's mission and to believe in it. They need continuous training. They need to see results. … Implicit in this is that employees have to be managed as associates, partners -- and not in name only. The definition of a partnership is that all partners are equal. It is also the definition of a partnership that partners cannot be ordered. They have to be persuaded. (Drucker, 1998)

As stated, Drucker's argument is clearly and dangerously fallacious; at the very least he fails to distinguish between 'ought' and 'is'. So it may be the case that these concepts of partnership, challenge and the like *ought* to motivate workers; but it all too obvious that this has little relevance to the vast majority of people's experience. Moreover statements such as this lend themselves too easily to superficial incorporation. So Wal-Mart's use of the term 'associates' might have been derived from Drucker; whether their employment practices and employee development policies are similarly inspired and grounded is far more doubtful.

On the other hand Raymond's characterization of the open-source model demonstrates many of the features to which Drucker refers. The participants are motivated by the challenge, and the capacity to act as partners. A similar motivational mix needs to be on offer with regard to CI initiatives, and so the initial impetus must have some foundation within the community itself. The issue of what it is that actually constitutes a community remains problematic as was pointed out earlier; but for the moment let us assume that there is some basis to the concept, and some link with social reality.

The issue is then how best to encourage the emergence of such motivations and enthusiasms; how beyond that to foster them, taking some of them on to materialize as specific projects,

[8] The quote is attributed to Drucker by Snowden, although it is not found in any of Drucker's writings – stress added.

objectives, and achievements. ICT can provide a forum and a mechanism for this. Indeed Raymond's entire discussion of the development of the open-source movement is premised on ICT, specifically the internet, as a constitutive and indispensable feature. Without such resilient, extensive, and virtually effortless communication the open-source community simply would not have been able to develop as far as it did. There might have been some small-scale, geographically constrained interchange, but nothing resembling the development of Linux. Linux burst upon the scene in the early 1990s, precisely the point at which the internet, as a key component of everyday life for a significant and rapidly growing proportion of people, was taking up its central role as the communications technology *par excellence*. Raymond's contrast between the cathedral and the bazaar is the difference between 'individual wizards … working in splendid isolation', and 'a great babbling bazaar of differing agendas and approaches'. The former requires little or no communication and co-operation, the latter demands it as perpetually available; and as shall be seen, the form taken by this combination of communication and co-operation is critical.

Raymond recognizes the central and vital role of the internet; particularly in providing a safeguard and counter-force to Brooks' dictum that 'adding more people to a project that is running late, makes it later still'. Brooks was making the point that the addition of more people might seem to be a good idea, on the basis that 'many hands make light work'; but this is more than outweighed by the necessity to spend effort bringing the neophytes up to speed, and then coping with the increased communication demands – 'too many cooks spoil the broth'. Raymond counters what is now called Brooks' Law as follows: *'Provided the development coordinator has a communications medium at least as good as the Internet, and knows how to lead without coercion, many heads are inevitably better than one.'* [#19][9]

So far so good: Volunteers are better than conscripts; but note that more is at stake here. There is a need for people to feel motivated and enthused by some concept of mission or purpose. In the commercial world this is often taken to mean that any venture must have a Mission Statement – capital 'M', capital 'S'. A quick search on the internet demonstrates that there are several packages and tutorial outlines for how best to develop a mission statement. There is even a mission statement generator, found on the

[9] The importance of the internet is not that it allows individuals to talk to each other on an individual basis, but that it affords a forum for exchange and co-operation; with both asynchronous and synchronous interactions.

Dilbert.com website.[10] But mission statements are rarely the result of volunteers giving succinct expression to their overwhelming motivations and inspirations. They are usually imposed from above, possibly derived from the CEO's 'personal itch'; but also likely to have been generated with regard more to impact, and enhancement of brand recognition, than as an expression of widely felt and deeply held commitment.

This somewhat sweeping condemnation may not apply as readily to the NGO sector, since in this case branding, visibility and presence will be important; but not in quite the same sense as in the commercial realm. There may then be a case for NGOs developing mission statements; although a quick study of the mission statements of, for instance, ICRC, UNICEF, WHO, and a few others indicates that they are far too long-winded and lacking in impact and immediacy. So I little reason to modify my belief that the only mission statement that really 'works' is the one for the *Starship Enterprise* – 'to boldly go …'.

In any case mission statements will be neither necessary nor sufficient; at best they may serve as a rallying call or provide enhanced visibility and recognition. More critical is the extent to which people feel that they are being treated as associates; open to persuasion and believing in the mission as something with what Raymond terms 'plausible promise'. He develops this theme in some of his other maxims, including the following; *'Treating your users as co-developers is your least-hassle route to rapid code improvement and effective debugging.'* [#6] *'If you treat your beta-testers as if they're your most valuable resource, they will respond by becoming your most valuable resource.'* [#10]

Note that Raymond does not use the term *community* in either of these, or any of his other maxims; although elsewhere in his article he uses the term frequently – e.g. Linux community, co-developer community, open-source community. His preferred terms in his aphorisms are far more specific to the task in hand, co-ordination for software development; hence *co-developers*, *beta-testers*, *users*, and so on. In this way he makes no claims for being in the business of community building, and simultaneously relates his ideas strictly to the matter-in-hand. This neatly avoids the issue of any simple recourse to the concept of some imagined community.

[10] Anyone keen to develop a mission statement can download a package from http://www.sound-business-practices.com/rationale/mission-statements.htm. The website states that 'Our Mission Statements Formulation Package contains several example Mission Statements, is easily customizable in almost any word processing software, and most of all, there is no waiting!'
The Dilbert mission statement generator, complete with ready-split infinitives can be found at http://www.dilbert.com/comics/dilbert/games/career/bin/ms.cgi

There is a grouping of sorts involved here, but it is one intimately associated with a specific task or project; and it may or may not have some more durable existence beyond those confines. The key point for our purposes is that there needs to be a participative orientation from all sides, which is what Raymond reminds us of when mentioning the important roles played by *co-developers* and *beta-testers*.

Again, this takes us a little further, but perhaps more guidance can be offered with regard to the ways in which the efforts and enthusiasms of *associates* or volunteers can best be harnessed and sustained. Raymond offers two pertinent observations. *'The next best thing to having good ideas is recognizing good ideas from your users. Sometimes the latter is better.'* [#11] *'Often, the most striking and innovative solutions come from realizing that your concept of the problem was wrong.'* [#12] And even more insightfully he states that – *'Given a large enough beta-tester and co-developer base, almost every problem will be characterized quickly and the ?x obvious to someone.'* [#8]

What we have here is a call for a truly co-operative venture; whether or not we call it a community. Furthermore it is one owing its continued existence to the widespread availability of ICTs. This raises the issue of the extent to which we might start to develop community informatics projects along the lines suggested by the examples of open-source, and particularly Linux collaborations. Their experiences appear to offer an alternative and provocative blueprint for CI projects. Yet anyone reading Raymond's work might, at this point, raise what appears to be a major objection. Raymond states that 'one cannot code from the ground up in bazaar style'; in other words you cannot build your project from scratch along the lines of a bazaar, something must already be in existence. Yet this is exactly the point; community informatics must also build upon what is already there, but must do so in a fashion that mimics the open-source experience rather than the command-and-control bureaucratic one. Raymond stresses that

> When you start community-building, what you need to be able to present is a plausible promise. Your program[11] doesn't have to work particularly well. It can be crude, buggy, incomplete, and poorly documented. What it must not fail to do is convince potential co-developers that it can be evolved into something really neat in the foreseeable future.

[11] Raymond refers here to computer *programs*, but the point is still relevant if the paragraph is read as referring to CI programs.

The two examples mentioned at the start – the phoned warning from someone in Singapore to his family in India, and the people in Tamil Nadu co-ordinating their communications – both built upon existing structures and relationships, together with appropriate use of ICTs. In the former case the radio report was relayed by phone, and then spread around the village in India. In the latter case a network developed based on the locality in Tamil Nadu, incorporating both ICTs and word-of-mouth. What is common to both is that it was not simply communication that mattered, but communication based upon personal relationships and existing networks, using ICTs.

Again Raymond provides an observation that can be readily revised for our purposes. *'Good programmers know what to write. Great ones know what to rewrite (and reuse).'* [#2] For CI this can be restated as - *Effective CI activists know what is required. Great CI activists know what is already in place or available.*

At this point we need to consider in some more detail what is actually in place, and how best to utilize or incorporate it. Raymond's concept of the bazaar, with its babble of different agendas, is somewhat misleading. It evokes an image of a local market place, with many vendors all displaying their wares and competing for custom. To an extent this may have some relevance, although Raymond's characterization of the open-source mode of operation is distinctly not one of 'doing business'; there are no monetary transactions, nothing is bought or sold. So perhaps the term bazaar is ill-advised if not ill-chosen. In fact Raymond seems to have used the term because he is following the logic of Adam Smith with his concept of *the invisible hand*, whereby people pursue their own individual interests but thereby 'promote that of the society more effectually than [anyone who] really intends to promote it'. In other words, although each and every person is indeed following their own agenda, somehow the end result is not a zero-sum but a win-win.

> The Linux world behaves in many respects like a free market or an ecology, a collection of selfish agents attempting to maximize utility which in the process produces a self-correcting spontaneous order more elaborate and efficient than any amount of central planning could have achieved. (Raymond, 1997)

But although this is Raymond's own description of the open-source community, it is not in fact a very accurate summation of how the open-source community operates even as Raymond describes it himself. He seems to see the weakness in his own reasoning and to some extent seeks to remedy it by noting that:

> The "utility function" Linux hackers are maximizing is not classically economic, but is the intangible of their own ego satisfaction and reputation among other hackers.

But this raises as many questions as it answers. Who are the other hackers? Does each individual hacker wish to garner the praise of certain big-hitters or simply to amass the admiration of as many other hackers as possible? Are ego satisfaction and reputation assessed by *quality* or *quantity*? Does one gain admiration – 'brownie-points' – because one achieves some specific result or because one takes large risks? How is success measured? How is success balanced by non-success or downright failure?

Raymond himself notes that hackers contribute as 'volunteers in an anarchist's paradise'. The context is wholly co-operative, and not easily subsumed under some risk-reward calculus or utility function. The term bazaar in fact is misleading; and there is a far more appropriate one which Raymond himself uses in a final remark he appended to a later version of the original paper.

> Finally, I must admit that I very nearly called this paper "The Cathedral and the Agora", the latter term being the Greek for an open market or public meeting place. The seminal "agoric systems" papers by Mark Miller and Eric Drexler, by describing the emergent properties of market-like computational ecologies, helped prepare me to think clearly about analogous phenomena in the free-software culture when Linux rubbed my nose in them five years later.

Miller and Drexler's characterization of the *agora* sees it as inherently market-oriented and exchange-oriented. This is a common, but constrained and restricted, view.

> This line of investigation leads us to propose what may be called the *agoric* approach to software systems. *Agoric* stems from *agora*, the Greek term for a meeting and market place. An agoric system is defined as a software system using market mechanisms, based on foundations that provide for the *encapsulation* and *communication of information*, *access*, and *resources* among *objects*. Each of these

> notions plays a role in supporting computational markets. (stress in original)[12]

What Miller and Drexler really focus on is an argument for applying a model oriented around 'decentralized market co-ordination' to the development and sustainability of complex computational systems. Raymond clearly feels that the open-source context is akin to this, hence his reference to '*agoric* systems'. But he also offers a glimpse of something different and distinctive: In fact something far more like the original meaning of the term *agora*.

The *agora* in Ancient Greece was a specific location. Initially it was the place for public assemblies, and only later was it also used as a market place. The *agora* as a concept, however, is distinctly different from the concept of the market. The market is a space of exchange, where everything has to be available as an exchange-value, and agents can buy and sell commodities at whatever is deemed to be the going rate. By definition and design the market is mechanistic – hence market *mechanisms*; and it is non-human in the sense that it does not operate on the basis of people's beliefs and priorities unless they can be translated into prices and exchanges. In contrast the *agora* is distinctively human and collective; the space between the private realm, the *oikos*, and the public realm of the state, the *ecclesia*. Bauman defines it as

> the space neither private nor public, but more exactly private and public at the same time. The space where private problems meet in a meaningful way – that is, not just to draw narcissistic pleasures or in search of some therapy through public display, but to seek collectively managed levers powerful enough to lift individuals from their privately suffered misery; the space where such ideas may be born and take shape as the 'public good', the 'just society' or 'shared values'. (1999)

For Bauman, the history of modern societies has been a long war of attrition 'launched against the agora from the side of the ecclesia'. In other words the state has sought to curtail or eradicate this space in which issues pertaining to the collective, the shared, the communal, can be raised and discussed. In the 1980s this goal of curtailment found its expression in the infamous statement of Margaret Thatcher, at that time the British Prime Minister, that 'there is no such thing as society … there are individual men and women, and there are families'. In fact the sweeping changes associated with Thatcherism, such as privatization and

[12] The Agoric Papers are available at www.agorics.com

encouragement of market freedoms, intensified the assault. At the same time as the state sought to undermine any claims for legitimacy for the social and communal, it was also enacting policies that resulted in the expansion of the operation and calculus of the market flooding into all aspects of human existence – social and personal, the pubic and the private. The novelty of this was in its intensity rather than its actual occurrence, since, as Hannah Arendt (1998) pointed out, the pressure from the *ecclesia* often took the form of efforts to transform the *agora* 'into an assemblage of shops like the bazaars of oriental despotism'.

The open-source community is perhaps an example of a way in which this process might be reversed; a glimpse of the *agora*. The Linux community appears to exemplify the emergence of a group whose exchanges are not market-based but oriented by something more akin to 'the public good'. Raymond does not see in these terms since he seems intent on seeing all interchanges as ultimately economically-oriented utility functions. Thus he says that the Linux community might be thought of as operating on the basis of altruism, but that 'altruism is itself a form of ego satisfaction for the altruist'. This seems to be an unwarranted affront to a more noble state of things, but more critically it obscures an important feature of the ways in which open-source development actually operates. Moreover this feature is not only to be found in the world of software development. It is, for instance, also present in the efforts of the *Wiki* movement, particularly and notably *Wikipedia*.

The term *Wiki* seems to have several meanings and derivations. The word itself means 'quick' or 'fast' in Hawaiian, and the slogan *WikiWiki* is used by one of shuttle bus companies at Honolulu International Airport. It is also claimed that *Wiki* is an acronym for 'What I Know Is'. Hence the term has come to denote collaborative efforts where people come together to pool their knowledge and expertise with a minimum of fuss and formality. There is a large and growing literature on *Wiki* principles and philosophy; naturally with various contending camps and positions emerging. In many regards the *Wiki* principles are more easily understood from stating what the *Wiki* movement is not, rather than what the *Wiki* movement actually is. Hence the following headings from the Wikipedia entry on Wikipedia itself;

> What Wikipedia is not
> 1.1 Wikipedia is not a paper encyclopedia
> 1.2 Wikipedia is not a dictionary
> 1.3 Wikipedia is not a publisher of original thought
> 1.4 Wikipedia is not a soapbox

> 1.5 Wikipedia is not a mirror or a repository of links, images, or media files
> 1.6 Wikipedia is not a free host, blog, or webspace provider
> 1.7 Wikipedia is not an indiscriminate collection of information
> 1.8 Wikipedia is not a crystal ball
> 1.9 Wikipedia is not censored for the protection of minors
> What the Wikipedia community is not
> 2.1 Wikipedia is not a battleground
> 2.2 Wikipedia is not an experiment in anarchy
> 2.3 Wikipedia is not a democracy
> 2.4 Wikipedia is not a bureaucracy[13]

In fact what the organizational model relies on is a sufficient number of people feeling motivated and enthused to contribute and participate; exactly the same prerequisites that Drucker identifies for volunteers or associates, and that Raymond describes for the Linux participants. Moreover what the *Wiki* philosophy has in common with both open-source and Brooks' ideas is that such endeavours are best seen as *cultivation* as opposed to *construction*. Brooks mentions the move from seeing software as being *built* to seeing it as *grown*. Similarly, astute readers might have noted that in the extract used earlier, Raymond sees the Linux world behaving 'like a free market *or an ecology*'. So what these all share is that at any one time they are less organized than perhaps they might be; but they are also more extensive, more accessible, more visible, and more speedily updated and corrected than standard command-and-control centralized systems. What open-source ventures demonstrate is that there are good grounds to challenge the general arguments that justify the need for command-and-control management. Raymond goes even further in stressing that the overheads required for these forms of management cannot be justified; they do not even deliver what they are meant to do.

Raymond approvingly quotes the 19th-century Russian anarchist Pyotr Alexeyvich Kropotkin as follows:

> Having been brought up in a serf-owner's family, I entered active life, like all young men of my time, with a great deal of con?dence in the necessity of commanding, ordering, scolding, punishing and the like. But when, at an early stage, I had to manage serious enterprises and to deal with [free] men, and when each mistake would lead at once to heavy

[13] http://en.wikipedia.org/wiki/Wikipedia:What_Wikipedia_is_not

> consequences, I began to appreciate the difference between acting on the principle of command and discipline and acting on the principle of common understanding. The former works admirably in a military parade, but it is worth nothing where real life is concerned, and the aim can be achieved only through the severe effort of many converging wills.[14]

So here is further evidence that the open-source experience is offering a glimpse of a new way of organizing; one that throws a glance of dim recognition at the *agora*. But given the context of community informatics, and the issues of emergency planning and disaster preparedness that provide a context for this discussion a further observation is required, and it relates to the existing basis upon which CI and the like have to build. The open-source movement built upon tools and expertise that was already in existence. As was pointed out previously, Raymond stressed that the bazaar was no basis for development from scratch. Furthermore several of his maxims reinforce this point; focusing on re-use of what is already available, reliance on others (e.g. beta-testers), and recognizing good ideas from elsewhere. He also discussed tools: *'Any tool should be useful in the expected way, but a truly great tool lends itself to uses you never expected.'* [#14] For software development it is assumed that there is a tool-box of tried-and-tested devices ready at hand. Raymond is making the point that great tools are both available and flexible. People can add to the tool-box as well as using and adapting what is already there. A similar lesson needs to be learned for CI, although the nature and availability of these 'tools' is a matter for further discussion at some later point.

By way of conclusion we return to the extract from the ICRC report with which we started, and refer to an exemplar of the way in which disaster was 'avoided with better information and communication'. When the Asian Tsunami struck one of the early reports in the UK press mentioned the case of Tilly Smith. At the time it seemed noteworthy[15], and in the ensuing months her story has taken on epic proportions as any internet search will indicate.

Tilly Smith was on the beach in Phuket with her family when she noticed that 'the water started to go funny'.[16] She had been taught

[14] *Memoirs of a Revolutionist* – quoted by Raymond, 1997

[15] I wrote a brief comment about it on 11th January 2005 – available at; http://www.lmu.ac.uk/internat/reflects/jan05/jan11.htm

[16] The original story can be found at www.telegraph.co.uk/news/main.jhtml?xml=/news/2005/01/01/ugeog.xml&sSheet=/portal/2005/01/01/ixportaltop.html; such is her subsequent fame, however, that a plethora of stories and awards followed - http://news.bbc.co.uk/1/hi/uk/4229392.stm ;
and the ultimate accolade, an entry in *Wikipedia* http://en.wikipedia.org/wiki/Tilly_Smith

about tsunamis a few weeks previously in a geography lesson, and so understood what it meant when the sea receded, and boats on the horizon started to bob violently up and down. She told her mother that there was going to be a tsunami. Her parents alerted those around them, and by evacuating the beach and the hotel many people survived who otherwise would probably have perished.

This anecdote reinforces and develops the two examples quoted in the opening section of this paper. Many others saw what Tilly saw, but failed to act appropriately. Fortunately she had paid sufficient attention in her geography lesson to have understood it, and then later been able to recall the pertinent information. Moreover there was a relationship of trust and mutual understanding between her and her mother: Not everyone would take the words a ten-year-old seriously. What we have here is a succinct example of some of the essential prerequisites for beginning to realize the potential of ICTs for community informatics; nurture, growth, co-operation, willingness to learn, trust, and some concept of the public good.

Bibliography

Arendt, H., 1998, *The Human Condition*, University of Chicago Press, originally published in 1958

Bauman, Z., 1998, ‘Globalization and the New Poor’, reprinted in Beilharz, P., (ed.), *The Bauman Reader*, Blackwell, 2001

Bauman, Z., 1999, *In Search of Politics*, Polity

Bauman, Z., 2001, *Community: Seeking Safety in an Insecure World*, Polity

Brooks, F.P., 1986, ‘No Silver Bullet: Essences and Accidents of Software Engineering’, first published in the Proceedings of IFIP 1986, H. J. Kugler, (ed.), *Information Processing*, Elsevier Science; now republished in F.P. Brooks, *The Mythical Man-Month*, 20th Anniversary Edition, Addison Wesley, 1995

Crozier, M., 1964, *The Bureaucratic Phenomenon*, University of Chicago Press

Drucker, P., 1998, ‘Management's new paradigms’, *Forbes Magazine*, 10.05.98

Hobsbawm, E., 1994, *The Age of Extremes*, Michael Joseph

Raymond, E., 1997, ‘The Cathedral and The Bazaar’, first given at the Linux Kongress, 1997; subsequently republished and revised on several occasions and in several places; the version referenced here is the title essay in Raymond’s book, *The Cathedral and The Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary*, O’Reilly, 2001