Against networks: an ethology of non-human humans?

Christopher Kelty, Rice University

> With the emergence of the Internet as *the* platform of computation, this assumption [that computers will act as instructed] can no longer be taken for granted… We cannot simply expect each computer on the Internet to faithfully follow the designed protocols or algorithms.[1]

Internet killed the Network

Since about 1998, the word *network* has become unfashionable for aficionados of Actor-Network Theory. M. Latour, for instance, has put it succinctly: "double-click information has killed the last bit of critical cutting-edge left in the notion of 'network.' I don't think we should use it anymore."[2] The point, he suggests, of the term network has been confounded by promises of transparency, instantaneity and undeformable information, when what network *really* implied were translations and transductions and irreducibilities and rhizomatic connections across heterogeneous ontological domains. Now it just means porn and mp3s.

I am not so sure. I am also against networks—but in the double sense of that phrase. As a method, useful for studying scientific and technical practice, it has been diffused and defused: like *paradigm* before it, *network* threatens to become a kind of conceptual kudzu—blanketing otherwise discernable conceptual practices in an unstaunchable thicket of good intentions. Increasingly, it is easy to find *network* lazing about in the wild, occasionally working like it used to, but more often preferring, like Bartleby, not to work.[3] And yet, everywhere one goes

---

[1] *Algorithmic Mechanism Design* by Noam Nisam and Avir Ronen. (Proc. 31st ACM Symp. on Theory of Computing, 1999, 129-140). Emphasis in original.

[2] Bruno Latour, "On Recalling ANT," in John Law and John Hassard eds, *Actor Network Theory and After (*Oxford, UK ; Malden, MA, USA : Blackwell/The Sociological Review, 1999: 15-16).

[3] See, especially, Annelise Riles, *The Network Inside Out* (Ann Arbor: University of Michigan Press,

today, one comes up *against* actual networks—they drape the globe in wires and waves, they permeate our minds and relations in links and nodes, referrals and endorsements, they dangle from sockets in our office walls; our wireless networks sprout up everywhere, mimicking the human densities around them; we talk of social networks and community networks, and ad-hoc networks, and p2p networks and Real™ Networks. On the one hand, *network* is a method co-opted, disseminated, diluted perhaps, by well-meaning but misinformed researchers (but who? It is not clear); on the other it is an ecological explosion of forms of networking, devices, software, protocols etc, which demand the very same critical attention putatively eviscerated by the dissemination of the method.

Network may have been the wrong word, but it is not, and has not been, the wrong *concept*. It demands renewal and renovation, not mothballing or recall. Over the last two decades, *network* has been used to understand wide ranges of a great many things, but ironically, not networks. For most studies in STS, *network* is a way of describing scientific knowledge in the making—not a functioning network like the Internet or a mobile-phone network. Associated terms like *black box, mediator, immutable mobile, actant and actor* etc. are generally used to help analyze contests over techno-scientific pursuits, where the outcome (fact, equation, device, article) is usually the starting point for the investigation. The concept is useful as a method because it creates a specific frame in which to understand changes of scale—relationships between local scientific investigations and their global reach. It gives specificity to the assertion and spread of "universal" scientific truths.

But what role does the network play? In particular, what role does the Internet play--that one, singular network that everyone knows about—the one that is routinely confused with the World Wide Web, with email, with Ethernet, or with myriad other familiar technologies that rest upon it? The one, as the epigraph above puts it, that is *the* platform of computing. How should one

---

2000), which argues for the surprising appearance of network analysis in the wild.

go about investigating networks in their specificity, as *networks*? Should they be understood as techno-scientific achievements, as infrastructure, as medium, or as all or none of these?

I have explored these questions in a longer work, in particular, with respect to the proper way to get at the substance of the heterogeneous Internet of people, devices, software, laws and practices—and how these practices are related through free software and open source software, to the practice of contemporary science and technology.[4] Here however, I want to focus more specifically on the nature and quality of the *non-human actors* that inhabit the Internet—and ask what role they play in contracting or extending networks or *networks*.

Agnosticism in method was what originally granted agency to non-human actors in science studies: treat all components of the network as equal, with respect to actions, interests and outcomes; do not give human brains more active power than microbes and machines. Such an approach is not intended to make non-humans more like humans—by granting them special human-defining qualities like selfhood, intention or language use—it is intended only to de-hierarchize with respect to agency. Humans, on this view, are demoted, rather than (or in addition to) non-humans being promoted.

On the one hand the Internet—"where nobody knows you are a dog"—provides one of the now classic examples of this: whether you want to talk about cyborg selves or avatars or autonomous agents or bots. But these popular figures are not necessarily the place I would direct our attention. Rather, I would suggest that we look at the practice of computer scientist researchers who are 1) taking non-human agency very seriously and 2) find these non-humans to be fundamentally different in the context of the Internet than previously.

This paper proceeds then by introducing two fields of modern CS research where non-humans play a role, and explores how the practice of science goes about investigating them, making

---

[4] Christopher Kelty, n.d. *Two Bits: Free Software and the Social Imagination after the Internet*.

use of them, and most importantly, "implementing" them.

Which non-humans, which non-machines, which non-animals?

The Internet is obviously saturated with non-humans—the question of the humanity of geeks, hackers, elves or orcs notwithstanding. The so-called physical layer, made up of wires, fiber optics, satellites, cables and hubs, are obvious candidates—they are wily and resistant objects that must be buried, shot into space, insulated and twisted, they can disappear, get dug up or tripped on, fall to earth and otherwise wreak havoc with a network.

Likewise the ecology of complex machines: desktop computers, routers, web servers back-office rackspace filled with multiple processor RAID-arrayed hotrods are also important non-humans, also wily and resistant, and more importantly, horrendously complicated modularized heterogeneous collections of devices whose success or failure depends on the constant ministrations and ritual attendance of system administrators world-wide.[5]

But physical infrastructure and machines are in fact *so ubiquitous* and *so abundant* that for many Internet professionals (geeks hackers, engineers, computer scientists, entrepreneurs) they are no longer *things*, but a substrate upon which *things* flow: bandwidth, CPU cycles and memory. For many of the people I am concerned with here, the Internet is primarily a *milieu*—a teeming quasi-human space characterized by highly formatted and specific modes of interaction. It is not built not only of physical machines administered by underpaid technicians, but of ideas, behaviors and implementations. Software, protocols, and standards are the substance of this world, not merely its form. It is, for an increasingly large number of

---

[5] Cf. Greg Downey "Virtual Webs, Physical Technologies and Hidden Workers"Technology and Culture 42, April 2001. Susan Leigh Star and Geof Bowker *Sorting Things Out: Classification and its consequences* MIT Press, 1999.

practitioners, indistinguishable from societies, markets, or publics—or to put it another way, these older collectives have become indistinguishable from the *network*.

As the opening epigraph makes clear, CS researchers are not in doubt that the Internet is *the* platform for computation. For Distributed Computing researchers today, there is no other option. It is important to understand the radical nature of this claim: it is somewhat akin to a molecular biologist saying "the natural environment is *the* platform for life, leave the lab behind and experiment on *the planet*." Ironically, the radical nature of this switch has gone largely unrecognized by CS researchers, for whom the Internet is not quite yet reality, but just a really fantastic and free laboratory. But the flip side of this non-recognition is the recognition that Internet "users" are not predictable, controlled variables in an experimental set-up. And yet they are not quite humans either. As Joan Feigenbaum et. al. hedges: "participants in an Internet algorithm are economic actors as well as computational processes."

The non-humans I am concerned with, therefore, are the non-human *humans*—the *conceptual personae* invented by scientists, engineers, hackers, geeks etc. who create ideas, behaviors, implementations, and who study the implementations and behaviors of others, as they exist in this milieu—the primary locus of which is the Internet. These *personae* are drawn largely from mathematical and formal economic, political science, and evolutionary psychology research. *Personae* such as *homo economicus*, the dilemma-addled Prisoners of Game Theory, the decision-makers of operations research, the resource maximizing insects of socio-biology, the genetic mutants of evolutionary psychology, or the utility functions of economics. Which is to say: these personae are not exactly invented by computer scientists, they are instead *researched, behaved and implemented* by them. The way these implementations and experiments look, and the outcomes they have, may or may not mesh with the expectations or beliefs of the social scientists responsible for the invention or refinement of these *personae*. Nonetheless the outcomes are meaningful for computer scientists who use the data and outcomes to refine the experimental software, protocols, and algorithms they design.

There is a tendency, amongst academics who do not work with such *personae*, or who seek to critique them (I am thinking primarily of anthropologists here) to refer to them all as a single type: the rational actor. The critique often leveled at these rational actors is that they are either 1) inaccurate because humans rarely act rationally or 2) inaccurate because human behavior is not autonomous but largely determined by various environmental, cultural, historical factors. I would suggest that neither of these critiques make much difference either to the social scientists creating these *personae* or the computer scientists implementing them. I would venture to say that neither group of researchers cares about accuracy, only about precision. Indeed, there appears to be a certain proliferation and differentiation of *personae* in these disciplines—and if one understands them not as taxonomic or natural science disciplines, but as *design disciplines* (cf. "sciences of the artificial" as Herbert Simon called them) then the proliferation of *personae* is not just a question of models and tests, but a question of modes of organization and formats for social interaction—from government policy on the one hand, to peer to peer networks on the other.

Conceptual Personae give form to certain functions which can be easily 1) manipulated and calculated and 2) implemented as algorithms. They are like models in that computer scientists can use them to think about the social and collective phenomena they are researching *and designing*, but they are unlike models in that they presumed to be *uncontrollable* in an experimental sense—they exist in a unorganized, non-hierarchical, and unpredictable Internet. *Conceptual personae* are, in some sense, the opposite of a black box, something with agency that emits a signal based on an input, but the mechanism for which (the insides of the black box) are in fact unknown, inaccessible and arguable (as opposed to known and settled).

> **A Note:** the term *conceptual personae* is somewhat blatantly misused here. For Deleuze et Guattari, only philosophers work with conceptual personae—they have no existence in the lives of the everyday, and they are not allowed to be thought or understood by

scientists or artists.[6] Scientists are granted only functives, functions and prospects by D&G, while artists are given "aesthetic figures" to work with. While the general attempt to rescue a distinctiveness for philosophy is the one thing that I think makes D&G's work so urgent and interesting to so many people, I hardly think such distinctions hold water—they amount to a kind of self-important policing of the philosophical compound, through which identity and authenticity will be verified only by the true philosophers on the inside—even as the rest of the world moves on without them. What if, however, the people engaged in the creation of social science concepts, and the people creating machines, software, mechanisms and organizations, are engaged in the same kinds of conceptual creation? What if they have taken over the game of concept creation and merged it with the worldly demands of scientific experiment? D&G do say that what scientists construct are objects using functions (..made of functives and prospects—the terminology is arcane, and difficult to parse in a short space), but that this does not imply their concepts—which must be discovered (by philosophers). So perhaps I am seeing the work of social scientists and computer scientists through the lens of philosophy, in order to give some consistency to the *personae* they create, the functions they are granted, the lives and machines they live out or through, and the implications for that other, important agent-like non-human: the collective.

Between Actual and Virtual Networks: Some stories from Computer Science

The research I draw on here is based in conversations and interviews with computer scientists at Rice University.[7] There are two domains of research that I will focus on: one is experimental research into peer-to-peer network overlays and applications and the other is computer security research related to electronic touch-screen voting machines. Both of these areas are highly politically charged from the get-go: peer to peer by its implication into the global war over the meaning and status of intellectual property and cultural production; and

---

[6] Gilles Deleuze and Felix Guattari, *What is Philosophy*, tr. Hugh Tomlinson, Columbia University Press, 1996.

[7] The EPIT Project- Ethics and Politics of Information Technology. All of the transcripts cited here are available online, along with commentary and annotation by the researchers. http://frazer.rice.edu/epit/

voting by virtue of being the core practice of democracy. Both areas are, on the one hand, characterized by highly abstract research questions germane to concerns in modern computer science; but on the other hand, they are both also pursued in a context that is not restricted to the laboratory or the cloistered research machines on campus, but takes place "live" amongst actual systems for voting and actual instantiations of the global internet. Because they blur the line between simulation and implementation, of the line between laboratory experiment and installed systems—they also raise new questions about the status of the non-human humans that permeate them. These non-human humans, these conceptual personae, do not have the same kind of existence as avatars, or virtual selves, they are not second selves or puppets of users, malign or otherwise. Rather they are conceptually worked-out *personae* around which mechanisms are designed. They are, for example, versions of *homo economicus* around which mechanisms for sharing and exchanging data (or data representing money) are constructed and formatted.[8]

Michel Callon, for one, has suggested that a "peculiar anthropology" might study these personae (which he calls "calculative agencies") as they exist in the world; places where economics in particular, formats and constrains the behaviors of flesh-and-blood actors in markets, auctions, trading floors or acts of consumption.[9] For Callon, the question of whether or not humans are rational, irrational, or whether economics or political science has *accurate* understandings of the human, are uninteresting: designed mechanisms turn non-calculating flesh-and-blood humans into *calculative agencies*—assemblages of human and non-human that actually can calculate as demanded by the design of the mechanism—and can do so with ever-

---

[8] "Mechanism design" which is introduced in more detail below, is in fact a mixed sub-domain of computer science and social science that aims to marry the problems of economic analysis of behavior with the computational tractability of algorithms. Two recent papers: Feigenbaum et.al. 200? and Nisam and Ronen 1999

[9] Callon *Laws of the Markets*, 1998.

increasing precision, which helps account for things like the growth and differentiation of demand, or the differentiation of markets and goods. For the social scientists who manipulate these personae, this description would be considered convoluted: they see their role as designing mechanisms that achieve a desired outcome, given the statistically tractable behavior of humans. Callon might be arguing that that predictable rational behavior is in fact created by the mechanisms, whereas other social scientists interpret it as pre-existing.[10]

The examples I look at complicate the notion of a calculative agency by focusing on programmed environments of exchange on the Internet, or amongst humans and digital machines in settings that involve the Internet (voting machines themselves are not directly connected to the Internet, but this is exactly one of the issues in debate). Whereas Callon suggests that a particular economic mechanism *formats and constrains* the behaviors of humans in "locales" such as markets or auctions, I am looking at cases where the behaviors of non-human humans are literally created, formatted, and behaved (in a transitive sense) by computer scientists and hackers *on the Internet*. Furthermore, these systems waver between actual and virtual networks. Some, like peer to peer overlays, appear and disappear with nearly infinite speed, involving a wide range of people and machines; others, like voting machines are already in circulation in society even as they are studied, transformed, hacked and reformatted by computer scientists.

Peers, nodes, and other humans

> "Ideally, we would like to design a system where **nodes**, acting selfishly, behave collectively to maximize the common welfare."[11]

---

[10] In particular, one can look to the field of "implementation theory" in economics and political science, to see where this assumption holds its strongest sway. Jackson, M, "A Crash Course in Implementation Theory", *Social Choice and Welfare*, 18 2001, p655-708.

[11] Peter Druschel, "On designing incentives-compatible peer-to-peer systems" (with Tsuen-Wan

Research into peer to peer overlay networks has its origin in two different places: one is the tradition of "distributed computing" in academic computer science. Distributed computing as it was practiced throughout the seventies, eighties, and much of the nineties consisted primarily of theory and experiment conducted by researchers on a network of machines under their control. That is to say, researches created their own distribution of computers on which to test out ideas about distributed computing—they did not, generally speaking, employ any existing widespread networks as test-beds for their research. The other origin is in the 1990s: the emergence of hacker-created Internet-based protocols for sharing resources across existing computers connected to the network. The most famous of these, of course, was Napster—but more commonly referenced examples include KaZaa, Grokster, Gnutella, or MojoNation. These two origins combine here in the person(s) of modern distributed and peer to peer researcher Peter Druschel (and his students) at Rice University. His research projects on peer to peer overlays raise a variety of interesting questions about the status of both real users, and imagined collectivities he works with.

Druschel is fond of reminding everyone that peer-to-peer protocols for sharing resources are what make the Internet the Internet (i.e. the TCP/IP protocol is in essence a peer-to-peer resource sharing protocol). And so he refers to his projects as "overlays"—networks on top of networks—applications that use existing Internet interconnectivity to create overlay networks that change, add, or reformat the capabilities of the underlying networks, and which are easy to turn on and off, relying, as the Internet does, on extremely heterogeneous, but also extremely clearly defined infrastructure underneath.

What makes peer to peer network research interesting, with respect to the question of non-human humans, is that they are not concerned primarily with "users"—or with user interfaces, human computer interaction, or any other paradigm in which autonomous human actors have

---

"Johnny" Ngan, Animesh Nandi, Atul Singh and Dan Wallach). In *Proceedings of the 2nd International Workshop on Future Directions in Distributed Computing (FuDiCo II)*. Bertinoro, Italy, June 2004.

their way with semi-autonomous, bounded machines at their disposal (or vice versa). Rather, they are concerned with the system—they sometimes say "society" or "community"—of what they alternately refer to as nodes, peers, autonomous parties, pre-existing trust relationships, adversaries, conspiracies, participating organizations, pseudonyms, attackers, neighbors, instances, or participants.

In our research project, I spoke to Peter, along with his graduate students and mine, about the kinds of non-humans they employ. It quickly became clear that what Peter's group was attempting to study were human behaviors, but in the absence of humans. This recognition took considerable work to make: we expected to discuss the relationship of real flesh and blood users to the theories of self-organization and algorithmic design they employed. After all, we entered as *anthropologists* expecting and expected to be experts on *humans*. And indeed, Peter and his students used words like "user", "participant", "agent" and often used the pronouns "you" and "I" when discussing their work. However, what they were discussing were not flesh-and-blood users, and not artificial intelligences that stand in for them, but "peers" and "nodes"—by which they implied some kind of non-metaphorical representativity—they were not models exactly, but nor were they intended to be interchangeable with a human user. They were non-human humans. Our recognition of this kind of lateral abstraction took weeks to arrive. We continued to query them, asking questions such as: but what if a person does this or a person does that, or how do you think about the dynamics and meaning of people sharing this space on the internet? They continued to answer by saying that "nodes are selfish" or "peers might be involved in complicated double dealing" without every quite answering our questions. They were talking about different *things* than we were—different humans.

It's not that they wouldn't prefer to use humans—only that they have chosen to, so to speak, simulate the experiment using non-human humans, for the time being, instead of performing it on the more familiar flesh-and-blood participants as researchers might in, say, a clinical trial:

> "We'd love to have a large deployed user base, because that would allow us to actually derive data. Many of those questions we just

> merely speculate right now. In the absence of that data... we ask questions of how do you make this secure, how scalable is it? We can sort of touch on these questions via simulation…"

But the prospect of actually deploying the system—creating a real application that flesh-and-blood humans use to share real digital objects—creates a kind of conundrum: if it is so deployed is it any longer an experiment?

> What would it take to get a large user base? On the one hand, you sort of have to provide, obviously the support to make this available and usable to a large community of naïve users, which is one side; the other thing is that you would have find an application that would attract that many people, and then of course the obvious thing is you get into file-sharing, but then you get into these legal issues and then you'd really get phone calls from Rice Lawyers.

According to Peter, at least one of his "competitors, or fellow researchers" has gone this route—raising a host of ethical and legal questions about experimentation: not the least of which is "is this experimentation on humans or not?" But this question of "real" experimentation aside, the field of peer-to-peer research proceeds quite successfully without flesh-and-blood users precisely because so many of the assumptions about what a distributed computing environment looks like have changed. As Peter says:

> "We have to really rethink a lot of the concepts in distributed systems that actually existed a long time before. Because nothing seems to quite fit here, which makes it exciting of course."

Peer to peer research is surprising—and exciting—to CS researchers because of this changed milieu: whereas previous generations of researchers worked on systems that "always assumed that there were preexisting trust relationships," ordered in a hierarchy, designed on a drawing board and implemented by contractors, peer to peer researchers are confronted with a "live" system: the Internet. In the case where the Internet replaces the conventional distributed system, networks and computation appear to grow "organically" or "spontaneously"—in short they start to look like markets and societies (or more commonly, following a socio-biological

penchant held by Peter and his students, like ant or termite colonies, zebra stripe formation or slime mold). The pullulate and coalesce, they light up like crystal arrays or disappear catastrophically.

But as he also suggests, in the absence of an actual peer to peer system-experiment, CS researchers are constrained to imagine what "social" or "economic" behavior (though they only uneasily use these terms) looks like in more conceptually abstract ways. They invent mechanisms for constraining it, and design systems around these imaginations. They make assumptions about human behavior, largely conjured out of conventional wisdom and common sense.

Now, there are two ways one could view this activity. The first is the way Peter and his group see it: (non-human) humans have behaviors, evolved and stable, around which mechanisms can be designed and tested in order to study outcomes. Questions like: how do you decrease the number of free-loaders in a society, or how do you incent humans to share extra resources rather than hoarding them? The provisional research on these questions is conducted through the creation of software that makes use of the real Internet, but is essentially a simulation of use of the internet—not a "real use" such as Napster. On this view, human behaviors are simple sets of rules—it is the resulting complexity and emergence of a spontaneous or organic system which the researchers are interested in understanding and eventually describing in mathematical detail.

The second way of understanding this activity follows Callon: designing mechanisms is a way of formatting the behavior (calculative or otherwise) of agents in a system. On this view behaviors are reflexive and strategic, not ingrained and evolved. The "spontaneous emergence" or the evolved complexity of a system is not a *result* of simple rules that govern human behvior, but a difficult, contested outcome of a design that constrains and formats behaviors in the name of producing this emergence. The difference lies in the fact that what emerges may not be at all the same as what mechanism designers *intend* to emerge. The

question of intention shifts from the agents in the system, to the designers of mechanisms. The outcome—whether goals or met or society and humans are transformed—appears to hang on your view of the nature of human behavior. And yet no humans are yet involved.

One might expect, then, that the CS researchers imaginations of these non-human humans are incredibly impoverished. Indeed, this is usually the critique leveled by humanists and anthropologists—that humans are wilier and less predictable than these models assume. While this may be true—since models and mechanisms are by their very nature meant to be simplifications, it doesn't seem to mesh with the manifest creativity of CS researchers. Indeed, of all the subfields of Computer Science, security research is perhaps the most "socially" imaginative of them—even as it remains one of the most mathematical and abstract. The very activity of imagining the "threat model" necessary in order to make an assessment of the security of some system involves imagining (and assuming) all manner of social and personal values: safety, privacy, legality, morality, fiscal security, confidentiality, anonymity, pseudonymity, identity, authentication etc.

In the case of peer to peer research, these security concerns all focus on the central non-human human: the node. In peer to peer research nodes can be altruistic, obedient, faulty, rational, irrational, corrupt, self-interested, strategizing, trustworthy, relevant, computationally limited, and malicious. They make excuses, they pick on the new guy, they fudge the books, they manufacture evidence, they kill auditors, they deceive as groups, they have local honesty and collusion, they elude detection and perhaps most interestingly, they are reincarnated—which is also known as repeated elusion.[12] Research papers in this field are filled with strangely anthropomorphic statements, such as:

> We show how requiring nodes to publish auditable records of their

---

[12] Seth James Nielson, Scott A. Crosby, and Dan S. Wallach, A Taxonomy of Rational Attacks, Fourth International Workshop on Peer-to-Peer Systems (IPTPS '05) (Ithaca, New York), February 2005.

> usage can give nodes economic incentives to report their usage truthfully.[13]

And they cite sources that range from Game Theory textbooks, to the Enron Ethics Manual. Quite opposite the recognized dismality of formal economics and political science in these areas, CS seems to have imported some of its characteristic humor, self-mockery and cleverness. Their non-human humans don't take themselves quite as seriously as the non-human humans in economics. They tend to be carnival-goers and comedy-troupes, not Prisoners and thieves and narcs embroiled in dilemmas.

CS and social science come together explicitly in the very new field of Distributed Algorithmic Mechanism Design (DAMD). The research in this field combines the research on economic actors—actors presumed to choose according to utility functions, and within constraints set up by markets, states or organizations—and algorithmic processes. The "distributed" in DAMD refers to the attempt to calculate theses choices and processes across a very large set of independent machines or software—rather than having any central authority compute each agent's choice and the outcome of the algorithm.[14] The combination is exciting to CS

---

[13] Tsuen-Wan "Johnny" Ngan, Dan S. Wallach, and Peter Druschel, Enforcing Fair Sharing of Peer-to-Peer Resources, 2nd International Workshop on Peer-to-Peer Systems (IPTPS '03) (Berkeley, California), February 2003.

[14] A colloquial way to understand this distinction is through the difference between first-geneartion Napster and secong generation peer-to-peer protocols like Gnutella or KaZaa. Napster worked by keeping a very large database, at one or two central locations, administered by the Napster corporation, which millions of users could connect to. They didn't house the MP3s themselves, but they did keep records of who had what and how to connect and download it. Second generation protocols dispensed entirely with the central accounting, and instead provide only the service of massing search requests from computer to computer. If an invidivual computer receives a search request it can fill, it would then connect the two. Other p2p systems take this a step further by chopping up the files themselves and distributing the chunks to millions of computers—so that no individual computer possesses the entire file until which time they make a request for it, and the software helps them find the various pieces on various computers and put it together for them.

researchers primarily because it opens up a new field of questions about computational tractability, for which there are a well-developed set of mathematical and logical tools, and a robust general interest in the discipline. But there are also a host of issues that are not part of this core formal concern. Researchers are fascinated by the challenge of creating mechanisms that achieve a desired goal when used by thousands or even millions of users; the conventional problems of computer security get a new lease on life: problems of authenticity, anonymity, authentication, pseudonymity are renewed in the milieu of the Internet; issues of privacy management, censorship and anti-censorship pique researchers' interests.

All this explains why the "mechanism design" in DAMD is not just about creating software—but about organizing the action and behavior of non-human humans at large. DAMD mechanisms might attempt to achieve the obedience of nodes through coercion "out of band" (i.e. by "friends", corporations, academic institutions, or government agencies) or strictly within the network. There are both genuine and artificial incentives—there is the threat of loss (of property, of identity and of participation) and there are invented mechanisms of measuring credit and debt, such as the mandatory publishing of auditable data about a node's actions. DAMD research regularly refers to freeloaders, leeches, free-riders, and the tragedies of various commons. Reputations are soiled or enhanced. The truth is forced out. All of these mechanisms mix up the characteristics of nodes and humans, even as researchers see no such confusion.

Most often, these mechanisms are based implicitly or explicitly on the standard social science *personae*: *homo economicus*, the prisoners of game theory, the decision makers of operations research and decision theory, or the rational chooser of political science. Some work explicitly references literature in social science (e.g. "incentives-compatible mechanisms" or "the evolution of cooperation"), but to date the range of these references are quite thin (Robert Axelrod's work on cooperation is a frequent reference, as is Garett Hardin's "Tragedy of the commons"; naturally work in information economics, such as Kenneth Arrow's work or Robert Aumann's are referenced, as well as contemporary work by J. Tirole). Most such mechanism

designs employ conventional wisdom, or common sense notions of rationality and self-interest. In all cases of the research, however, the non-human humans play a central role.

Schneidman and Parkes (2003) for instance, suggest that existing peer to peer network research assumes "that users will follow prescribed protocols without deviation."[15] This is an assumption that, as they put it, "ignores the user's ability to modify the behavior of an algorithm for self-interested reasons." The "users" are not humans, but non-human humans: nodes, peers, rational "agents" in the sense in which the insurance literature uses that term: not the principal but the agent who acts on his or her or its behalf. Schneidman and Parkes propose thinking of these agents as possessing utility (i.e. having ranked preferences for possible outcomes) and acting "rationally" (defined as pursuing self-interest, that is, making choices that maximize utility), and they use notions like "bounded rationality" and terminology from the economics of auctions. They reference work in game theory—but suggest that for computer scientists, "mechanism design can be thought of as *inverse* game theory—where game theory reasons about how agents will play a game, MD reasons about how to design games that produce desired outcomes."(2)

These non-human humans are not flesh and blood actors—they might also be called non-human agents, since they are complexes that exhibit choice and agency, rational or irrational (indeed, increasingly complex), but they are nonetheless intended to represent *humans*—i.e. they are not often understood as autonomous objects that are the subjects of research, and not as artificial intelligences for instance. There is a lurking humanism here—even though these researchers go about inventing a teeming city of conceptual personae—of non-human agents that can do, be done to, and respond in all kinds of ways, they are not intended to be only

---

[15] J. Shneidman and D. C. Parkes. Rationality and Self-Interest in Peer to Peer Networks. In Proc. 2nd Int. Workshop on Peer-to-Peer Systems (IPTPS'03), 2003. http://citeseer.ist.psu.edu/shneidman03rationality.html

agents, but in some ways to *speak for* humans. Here the humans become actants, while the conceptual personae become actors.

Hackers as peers

All this might be dismissed as so much, as the phrase goes, intellectual masturbation. Theories upon theories about the behavior of humans that have little effect upon the real world. Indeed, in the case of mechanism design in economics this might largely be true; up to the point where mechanisms become policy, the conceptual personae of economic and political science theorizing are harmless claims waiting to be borne out or tested by (or formatted by, *pace* Callon) real policies and institutions. In a related set of interviews with Moshe Vardi, a CS researcher whose work includes software verification and "reasoning about knowledge," he raises this question apropos of his own research in CS:

> This stuff has application to philosophy, to computer science, but also to economics. Because if you think about what economics is, it's about what happens when you have systems with lots of agents and they trade and they do things together. It's all about multi-agent systems. And we had meetings to bring all these things together, which were fascinating meetings. I remember sitting with a well-known economist from Yale and I said: "but this doesn't really describe people. Don't you see this as a problem." And he says: "Don't *you* see this as a problem?" I said: "I deal with computers, I deal with artificial reality." [Laughter]. I don't have to think about epistemology, it's just a way of talking about it.
>
> **CK:** But you're suggesting that [these research areas in multi-agent systems] won't work for humans but that it *will* work for wires. You say at the beginning of your book…
>
> **MV:** It will work in the sense that it gives me a way to design systems. Ok? So, if I think that my agents should have some properties that they may not be aware of, some of the possibilities, then I can build it into my theory--but it's in an artificial world, it's a make believe world.

The "make-believe" world is the world of non-humans: the world so many CS researchers take

incredibly seriously because it allows them to be creative in design, to ask questions in ways that startle and often bemuse outsiders: wires that have properties they are not aware of and agents in a computer chip that bargain with each other etc. But Vardi suggests that this is make-believe, an "artificial world" with no implications for real people. Such a view *might* be true for the lofty theoretical work of logicians like Vardi and Economists from Yale, but it is less true, or less "make-believe" for experimentalists like Peter Druschel. Indeed, the very impetus for modern DAMD research is not the design of artificial systems, but the surprising existence of very real hacker-created peer to peer networks that really do involve the actions, property and money of flesh and blood humans. For CS researchers these real p2p systems are both inspiration for research and potentially real experiments generating real data about computation and exchange on the Internet. The fact that Druschel's experiments simulate users is only a choice he makes to protect himself from liability—but it has not stopped these researchers from forging connections with, and attempting to collaborate with the hackers building the "live" p2p systems in order to answer the questions they cannot.

But hackers have different problems: *they have users*.

> **PD:** [Collaborating with hackers has] actually been somewhat difficult. I can't say that there has been a lot constructive collaboration and it is partly because these folks are a sort of special-breed of mostly self taught programmers or computer scientists, who don't really appreciate actually what we are trying to do, or further our fundamental understanding of how these things work. They're focused on making their systems work today without necessarily having the desire to understand fundamentally why it works or what it means for the future, how it extends. I think they may be driven by a desire to do something that has an impact in their world.

Hackers' goals are often much more prosaic—they need flesh and blood humans to use their software, and the software needs to be only as good as it needs to be to meet this requirement. For Druschel, there are right and wrong ways to design peer to peer systems that are related to the standard CS criteria of security, robustness, efficiency, and computational

tractability; but for hackers these demands are secondary to their own needs: "making an impact" or making money, or being the first on the block—a whole other set of criteria that make hacking a valued activity.[16] Nonetheless, there are still exchanges between CS researchers and hackers:

> [But] there are some interesting exchanges; these folks, for instance, have data derived from the actual operation of large systems that we don't have and we would love to have those things because it helps us to design our systems better. We also have a lot of technology that we could give to them. And there has been some success in them adopting things, but by and large, it hasn't been as successful as one might think.

Druschel implies that the exchange would consist of technology for data. The exchange of getting real data about the experience of these systems in a real world setting for technology that makes the systems "better"—or at least that makes them more robust, more secure, more fault-tolerant, and more efficient, and more computationally tractable, the things which computer scientists care about. Indeed, the potential exchanges might even be the more or less surreptitious implementation of systems that CS researchers are interested in testing in exchange for the data that is acquired in the large-scale deployment of these systems.

But the independence of the hacker gets in the way for reasons all too familiar to science studies:

> **PD:** It is partly also interesting that these guys are very independent, in those few cases where they have picked up technology from our research community, they have used them in the form of re-implementing their own software using some ideas we had demonstrated, rather than actually using the software that came out of a research project. So, that also sort of gives you a certain level of

---

[16] cf. E. Gabriella Coleman, *The Social Construction of Productive Freedom*, PhD. Thesis, University of Chicago, 2005.

> uplift of course, [but] in general they don't give credit very well. So it is actually not easy for the public to track where these ideas came from.
>
> **CK:** They talk an awful lot about credit and reputation. I guess it is not surprising that they don't do a very good job of actually crediting…
>
> **PD:** I think it is partly because they are not aware, they don't think like we do in terms of intellectual property. They don't publish anything, they don't write papers. You've got to look [directly] at their software. In fact you can't even get them to read papers [laughter].

Hackers are content to forgo the institutional milieu of science; though they might like the idea of being credited by (or hired by) academics, only rarely have they been indoctrinated with the ethical and normative rules of scientific production. By the same token, Druschel and his group can't understand the kinds of motivations and rewards that hackers value. The lack of inclusion in the system of citation and publishing means that hackers neither give credit nor get it for their creation of p2p systems. And yet, amongst hackers, the names of those hackers identified with being first, or with creating the most successful programs are well known.[17] Their activities are constrained by different demands.

The point of this interesting interchange is that it stands in for the very questions of motivation, incentive and rationality that supposedly determine the behaviors of the non-human humans populating the Internet-based peer to peer networks. Are Hackers nodes? Do they act selfishly? Should we be able to design a mechanism (re-design the institutional relation between science and entrepreneurialism) that achieves the outcomes of steady cooperation? What does Hacker want?

---

[17] The creator of Napster, Shawn Fanning; the creater of FreeNet, Ian Curtis; and especially Bram Cohen, creator of BitTorrent are often much more well known (amongst both hackers and scientists) than the scientists are amongst hackers. Indeed, Druschel's group at one point sought to hire Cohen.

Such questions are not just metaphorical: the hacker is in fact a paradigmatic non-human human around which the activities of security, control, confidentiality, anonymity and robustness are designed, formatted and implemented. The fact that "real" hackers are often the best source of insight on what non-human hackers do is not surprising. In the next section, the case of electronic voting machines brings together two non-human humans: Hacker and Voter.

Code the Vote, Hack the Vote

The widespread adoption in live election use of electronic touch-screen voting machines (EVMs) in the United States followed the contested 2000 election, and the contentious Supreme Court decision in *Bush v. Gore*. Their implementation and use are widely perceived as a constructive response that corrects the failures of previous faulty machinery and methods (such as the now infamous "butterfly ballot"). EVM manufacturers (the most famous, or infamous of which is Diebold) were given a tremendous boost by the "Help America Vote Act" which required local election officials to upgrade their equipment and to purchase, where possible, electronic voting machines.

Not everyone agreed with the premise that computerizing the vote would eliminate any more problems than it would create. Stanford Computer Science researcher David Dill, for instance, was instrumental in organizing security researchers into a kind of activist movement that called for something Computer Science professors might never be expected to demand: more paper and less computerization. The movement has been remarkably successful in agitating for what they call a Voter Verifiable Paper (or Audit) Trail (VVPT/VVAT). The idea being that the idea of holding a "recount" given only EVMs was absurd precisely because they are so accurate—they would simply return the same number, regardless of whether that number was ill-gotten. Paper audit trails, the group argued, could be used to independently verify both the Voter's choices at the Ballot box, and be used in case of a recount to independently verify the totals.

Dan Wallach, another Computer Science professor at Rice, was instrumental in investigating

the security characteristics of voting machines. Dan sees himself very clearly as an advocate for flesh and blood voters generally, a CS professor who speaks on behalf of a fair voting system, but not on behalf or any particular voter or party.

But whereas the section above, on peer to peer networks, stressed the invention of non-human humans, and the investigation of mechanisms to control them, Wallach's work might better be understood as involving non-object objects, or non-machine machines. In a very widely cited paper, "Analysis of an Electronic Voting machine", Wallach and collaborators attempt to show how a Diebold AccuVote TS Voting Machine could be compromised. Their trick, however, was to do this without any actual "flesh and blood" Diebold AccuVote TS Voting Machine.

Wallach et.al analyzed source code that they presume is at work in these machines—source code which they "found" on the internet:

> the CVS source code repository for Diebold's AccuVote-TS DRE voting system recently appeared on the Internet. This appearance, announced by Bev Harris and discussed in her book, *Black Box Voting* [14], gives us a unique opportunity to analyze a widely used, paperless DRE system and evaluate the manufacturer's security claims. (4)

The discovery of the code and its analysis took place in a legally nebulous space: a repository of code publicly accessible on the Internet (though presumably intended to be kept "secret" by the corporation) analyzed as a proxy for existing machines that Diebold refused to give to the researchers unless they signed a strict Non-Disclosure Agreement that would effectively prevent them from publishing any security flaws they discovered.

The case of this research inverts the example of peer to peer research. Whereas Druschel might desire to "actually implement" a p2p system in order to derive experimental results from the systems he designs, and that people use, Wallach's example takes a technology that *has already been actually implemented,* and creates a simulation that can be tested only with respect to imagined machines. Both cases risk falling afoul of the law—the former by creating a file-sharing network that facilitates copyright infringement but that would test actual flaws

with the theory, the latter by infringing a publicly traded company's trade secrets in order to demonstrate potential flaws in an already deployed system. Both risks are related to the Internet as a research milieu: in Peter's case, a place to run an experiment, in Dan's case, the place where source code is stored.

Amongst the ecology of *conceptual personae* deployed in social science, the Voter is perhaps one of the most frequently modeled non-human humans; political scientists study both the behavior of voters at the ballot booth, and their ability to make their own interests mesh with their votes; psychologists are interested in the "human factors" of voting machine as well the cognitive capabilities of voters; economists are troubled by the relation of incentives to votes, and by the cost of voting itself. Voting, in general, is perhaps one of the longest standing and most eminent areas of "mechanism design"—the problem of designing a system that can achieve a clearly defined goal (elect a government official), but for which the best mechanism is constantly at stake in technical and political battles that range from the design of voting machines, to systems of voting (Condorcet vs. Proportional vs. First Past the Post) to battles over re-districting and demographic re-definition.

In broad terms, what Professor Wallach's research points out is that, in the case of EVMs, the corporations creating these machines are essentially re-designing the mechanism of the American voting system in an unaccountable, unverifiable, and potentially fraudulent manner. His attempts to show the potential flaws in these systems are attempts to point out why this mechanism might not be as good as another—in particular, why it might not be better than the paper-ballot system we have relied on to date, and why the VVPT is a necessary addition to the mechanism if one wants to assure the fairness of an election.

As I mentioned above, Wallach's research on the Diebold Machine was carried out in the absence of an actual Diebold machine—which in some ways might be seen to compromise their results, since it makes it easy for Diebold to proclaim that their system is not the same as (i.e. is better, or more secure than) the one Dan and colleagues have proven to be flawed. At the

level of technical issues of security (e.g. can it be hacked, if so how, what known bugs does it have, what kinds of software does it use and is it using the most up to date and secure version of it?), this may actually be true—but as Wallach points out it is impossible to find out in any legal way:

> At the time I had challenged Bill Stotesbury [the CEO of a competitor to Diebold] by saying "I would like to audit these machines, I would like to read their source code." He said "only under a non-disclosure agreement." I said "No way," He said, "No deal." I mean they would be happy to let me look at their source code under a non-disclosure, but that means I can't tell anybody else the results and that violates the point of me wanting to look at it, which is to, that the public should know about what they're voting on. That's the whole point of doing an audit, it seems to me.

The strategic use of non-disclosure agreements referred to here is standard practice amongst software and equipment manufacturers, meant to protect their trade secrets from the prying eyes of competitors. For Wallach, however, this is an abuse that redounds upon "voters" and "the public" for whom he speaks, and who he asserts should have the right to see these machines. But the story does not end there. US Voting machines are in fact certified by independent auditors—but there is a catch:

> It turns out that these independent testing authorities that I mentioned, *their reports are classified*. Not military kind of classified, but only election officials get to read them. They are not public. So you joe-random-voter are told "Just trust us! It's certified." [Smiles] Not only can't you look at the source code, but you can't even read the certification document. So you have absolutely no clue *why* it was certified, and you don't even know *who* you are supposed to trust. You're just told "trust us." Needless to say, this sets off all kinds of warning bells in the back of my head.

The "security" issues involved therefore extend considerably beyond the simple technical issues of whether the non-machine machine that Wallach et.al. investigated could be proven to be flawed (it is, or was), and into the realm of policy and regulation concerning the election and voting system at large, and the role of corporations and public election officials in general.

The *target* of Wallach's research, therefore, is less the machines themselves, than the entire *process* of mechanism design in democracy—a process he asserts is broken and as a result cannot be trusted to achieve the stated goal of fairly electing a candidate to public office.

Part of the problem comes from the assumption amongst corporations that the electoral mechanism is not the same as the voting machine—whereas Walllach asserts that it is—or that the "machine" extends considerably beyond the device one can touch to the process of designing, programming, implementing, buying, selling, advertising and promoting said machines. Seen from Wallach's perspective, it is difficult to distinguish the "design" that Congress perpetrates in passing The Help America Vote Act, or regulations about certificiation, from the "design" that a humble software programmer perpetrates in creating a program that can or cannot be hacked. "Security" research thus extends far beyond the technical lock-and-key cryptography concerns of researchers to the level of mechanism design at large.

What then is the object of Wallach's research? On the one hand, it is clear that his research concerns the design and security of software, and that this software may (or may not) be installed on computers that perform, as the saying goes, "mission critical tasks." On the other hand, Wallach's research extends so far beyond the arcane questions of buffer overflows and software backdoors, into the realm of election processes, regulatory loopholes, congressional acts, certification systems, and touchable voting machines that one might well ask: as a *computer* scientist, what possible right does he have to speak about these issues?

I would suggest that it is the curious role of non-human humans that grants Wallach these rights (i.e. the implementation of the conceptual person called the voter, the demands that voting be made secure and fair, and that errors of all kinds be reduced). To the extent that the domain of expertise of computer scientists extends not to computers, per se, but to computation as it is carried out in any form… and to the extent that computation is allied with issues of economic or social action in the form of questions about rationality, decision, incentives, mechanism design and then the kind of research Wallach performs potentially extends as much throughout

the social and political fabric of the planet as computation and network-mediated interaction does. If we trust the computer scientists (and we should) when they say that the Internet is *the* platform of computation—then the meaning not only of computation, experiment, and theory but also of security, of network, of applications and of software is transformed. Non human humans walk the earth.

Conclusion

What kinds of non-humans make a difference to our investigation and understanding of the Internet? Not only the cables and wires and routers and servers and satellites that make up this heterogeneous object, but the software, the protocols, the commodity-flows of bandwidth, CPUs and memory as well. Understanding how *the* network—the Internet—transforms the meaning and practice of computer science can best be understood by looking at non-humans that are not necessarily machines, animals or devices—by looking at the conceptual personae that roam the papers and the protocols of computer scientists—as well as the Internet.

The two examples reported here illustrate the transformation of computer science research by the Internet. In many ways, it mirrors the early days of computer science research, in which the struggle to discover, to define, to theorize what the *computer* would be. Today's researchers are engaged in a struggle, or contest, to define what the *network* will be. In at least a historical sense, the appearance of non-human humans should therefore not at all be surprising. Whether you consider the paper machines of Turing, the neurons of Mcculloch and Pitts, the grammar-obsessed machines of Chomsky, the frogs of Rosenbluth and Lettvin, the games of Morgenstern and von Neumann, the circuits and relays of Shannon, or any other feature of early cybernetics, computing research, cognitive science, or economics, these non-human humans are there because of an anti-metaphysical bias, an urgent urge to rid scientific thinking of the humanistic and the behaviorist, of the special pleading of the human. The non-humans of modern economics, mechanism design and behavioral science may not be much

different from these attempts—more precise, more refined perhaps, but essentially imaginary creatures. Economists can hold out hope of catching a glimpse of the Sasquatch or Santa Claus, but most people do not need these missing missing links. But the non-human humans of peer to peer research and computer security research are much more likely to be encountered in the wild—they are not as fantastic, they are not as precise, but nor are they simply humans—they are nodes, peers, participants, users, agents, devices, processes. These non-human humans are not only being theorized, they are being implemented in all kinds of settings: they are called "collective"; they are called "social"; they are called "economic"—what will have been the way these communities and actors should be defined? How should academics whose putative interest is in the "social study of science and technology" understand the emergence of "social software" or of peer to peer communities, or of other non-human human collectives?