Steps Towards an Ecology of Infrastructure:

Complex Problems in Design and Access for Large-Scale Collaborative Systems

Susan Leigh Star
Community Systems Lab
Department of Sociology
326 Lincoln Hall
University of Illinois
Urbana, IL 61801 *and*
Institute for Research on Learning
Tel: 1-217-333-7596
E-mail: slstar@ux1.cso.uiuc.edu

Karen Ruhleder
Department of Management
Worcester Polytechnic Institute
100 Institute Road
Worcester, MA 01609
Tel: 1-508-831-5573
E-mail: ruhleder@wpi.wpi.edu

ABSTRACT

This paper analyzes the initial phases of a large-scale custom software effort, the Worm Community System (WCS), a collaborative system designed for a geographically dispersed community of geneticists. Despite high user satisfaction with the system and interface, and extensive user feedback and analysis, many users experienced difficulties in signing on and use, ranging from simple lack of resources to complex organizational and intellectual trade-offs. Using Bateson's levels of learning, we characterize these as levels of infrastructural complexity which challenge both users and developers. Usage problems may result from different perceptions of this complexity in different organizational contexts.

KEYWORDS

Infrastructure, collaboratory, organizational computing, participatory design, ethnography

ETHNOGRAPHY IN SYSTEMS DESIGN

"What can be studied is always a relationship or an infinite regress of relationships. Never a 'thing.'"—Gregory Bateson

What is infrastructure? Common metaphors present it as a substrate: something upon which something else "runs" or "operates," such as a system of railroad tracks upon which rail cars run. Infrastructure in this image is something built and maintained, sinking then into an invisible background. Such a metaphor is neither useful nor accurate. Following Jewett and Kling [19], we hold that infrastructure is fundamentally and always a *relation*, never a thing. This can be seen via what Bowker [4] calls an "infrastructural inversion": a figure-ground gestalt shift in studies of large scale technological change [17,18]. This inversion de-emphasizes things or people as the only causes of change, and focuses on infrastructural relations (e.g. between railroads, timetables, and management structures in bureaucracies). It inverts traditional historical explanations and reveals how choices and politics embedded in such systems become articulated components. Substrate becomes substance.

Traditional methodologies for systems development and deployment are often based on a set of rationalistic or "mechanistic" ideas about artifacts and infrastructure. They assume that tasks to be automated are well-structured, the domain well-understood, and that system requirements can be determined by formal, *a priori* needs-assessment. Careful adherence to methodologies will lead to system acceptance and success; failure can be traced to ineffective organizational champions or willful user non-compliance — witness the many pages MIS textbooks devote to techniques for overcoming "user resistance."[1] Infrastructure forms at best a passive substrate, usually discussed as physical artifacts, e.g. a network for linking computers. This formal, linear approach to systems development and deployment is exemplified by the "waterfall" or "life-cycle" models taught in software engineering courses, where phases neatly align, or spawn off smaller versions. All work can be observed and routinized, all information codified. Users are sources of requirements, and eventually become systems recipients. These methodologies form complex *myth*ologies of systems development and use in "real-world" domains. Their rationalistic assumptions have been challenged since the late

CSCW 94- 10/94 Chapel Hill, NC, USA

---

[1] There is an analogy here with medicine, viz. studies of "patient compliance" which overlook the infrastructural and political features of medicine itself.

1970's by careful empirical analyses, which repeatedly demonstrated that even "simple" and "well-defined" tasks could only be understood as part of complex organizational contexts [11, 21]. Participatory design approaches emphasized the complex, embedded and historical nature of usage and adoption e.g., [14]. Techniques from approaches such as hermeneutics shed new light on the characteristics of organizational communication and various forms of media [3].

The foundation of these emerging models is relational: no artifact, computer-based or otherwise, is a discrete entity, a standalone *thing*. Its development and use are defined by complex relationships. Recent collaborations building on these ideas link members of traditional "technical" disciplines, such as computer science, with social scientists. Goals include: how to develop technologies to support complex knowledge work characterized by ambiguity, incomplete information, and collaboration by diverse individuals [33, 13, 16, 27, 15]; how to analyze the design process (*cf.* [9, 10, 12, 20]); and the nature of technological impacts (e.g.the emergence of e-mail and organizational communications systems, *cf.* [7, 8, 31, 26, 34]).

We develop aspects of these several models by examining system integration into a complex environment for knowledge work. We start with an infrastructural relation: difficulties encountered by scientists trying to "hook up" or "sign on" to a system designed to support their work. We see these problems not in terms of "user resistance" or "system success/failure." Rather, they are organizational and learning challenges, as Gregory Bateson characterized learning, messages and problems. He proposed three levels of conceptual complexity [1]. Problems become intractable when a "double bind" arises, as when different groups involved in system development or use address issues at incommensurate conceptual levels, leading to communication and usage failures.

THE WORM COMMUNITY SYSTEM (WCS)

The Worm Community System (WCS) is a customized piece of software for biologists studying the genetics, behavior and biology of *c.elegans*, a tiny nematode [25, 29]. It is of the genre of systems being developed for collaborative scientific work ("Collaboratories"), and involves geographically dispersed researchers. WCS is a distributed "hyperlibrary," affording informal and formal communication and data access across many sites. It incorporates graphics of the organism's physical structure, a genetic map, formal and informal research annotations (including a quarterly newsletter, the *Worm Breeder's Gazette*), directories, a thesaurus, and a database, *acedb*. Much of the system is hypertext-linked. It was developed with the close cooperation of several biologists; user feedback and requests were incorporated into the system over a period of several years.

The community consists of about 1400 scientists in over 100 laboratories. They are close-knit, considered extremely friendly; until recently, most people were first or second "generation" students of the field's founders. Recently, *c.elegans* was chosen as the "model organism" for the Human Genome Initiative, said to be the largest scientific project in history. Senior biologists hoped that WCS would help maintain intimacy in the face of increasing size and visibility.

The work of c.elegans biologists can most succinctly be captured by the notion of solving a jigsaw puzzle in four dimensions, across considerable geographical distance. The data are structured heterogeneously and must be mapped across fields; for example, a behavioral disorder linked with one gene must be triangulated with information from corresponding clone DNA fragments. Labs working on a particular problem, e.g. sperm production, are in frequent contact with each other by phone, FAX, conferences, and email.

The worm itself is remarkable both as an organism, and as a component of a complex pattern of information transfer integral to the biologists' work. It is microscopic and transparent (thus easier to work with than opaque creatures such as humans!). It is a hardy creature, and may be frozen, mailed to other labs via UPS, thawed out, and retrieved live for observation. Worms and parts of worms travel from one lab to another as researchers share specimens.Worm strains with particular characteristics, such as a mutation, may be mailed from a central Stock Center to labs requesting specimens. Tracking the location and characteristics of organisms thus is an important part of information work.

Computing use and sophistication in the labs varies widely. In labs most active in trying out WCS, there are 1–2 active, routine users. In many, computing is confined to email, word processing or the preparation of graphics for slides. In most labs one "computer person," often a student, is in charge of new programs and databases to track information.

We worked with the system developers on the project as ethnographers, analyzing WCS and other computing use, routine lab work and information-sharing, careers, competition and collaboration. We did semi-structured interviews and observations at 25 labs with more than 100 biologists over the last three years,[2] and fed back to developers both specific suggestions ("so-and-so found a bug") and general observations ("such-and-such would violate community norms"), many of which were incorporated into development.

Most respondents liked the system, praising its ease of use and its understanding of the problem domain. On the other hand, most have not signed on; many have chosen instead to use Gopher and other simpler net utilities with less technical

---

[2] Names have been changed to respect anonymity.

functionality. Obviously, this is a problem of some concern to us as system developers and evaluators. *Despite good user feedback and user participation in the system development, there were unforeseen, complex challenges to usage involving infrastructural and organizational relationships.* We examine this phenomenon closely in the following analysis.

SIGNING ON AND HOOKING UP

One classic CSCW typology distinguishes tasks as synchronous/asynchronous; proximate/long distance; and dedicated user groups vs. distributed, fluctuating groups [6]. This was useful for characterizing emerging technologies; however, it offers no assistance in analyzing the difficulties of implementation, or with integration into a particular group or community, as Schmidt and Bannon [30] cogently discuss. It also does not deal with the relational aspects of computing infrastructure and work, either real time "articulation work" or aspects of longer-term, asynchronous production tasks. We encountered many such issues in the worm community in their process of "signing on" and "hooking up"— tasks related to finding out about the system, installing and learning it.

Consider the set of tasks associated with getting the system up and running. WCS currently runs on a Sun Workstation, or remotely on a Mac over the NSFnet, or, with less functionality, remotely on a PC (a stand-alone Macintosh version is in progress). Setup includes buying an appropriate computer; identifying and buying the right windows-based interface; using communications protocols such as telnet and ftp; and locating the remote address where you "get" or operate the system. Each of these means that people trained in biology must acquire skills taken for granted by systems developers, whose interpersonal and organizational networks help them obtain necessary technical information, and who also possess a wealth of tacit knowledge about systems, software, and configurations. For instance, identifying which version of X Windows to use means understanding the X Windows class of software, installing it, andconfiguring it with the immediate or remote link. "Downloading the system via ftp" means understanding file transfer protocols across the net, *and* knowing which issue of the *Worm Breeders Gazette* lists the appropriate electronic address *and* knowing how ftp and X Windows work together.

These common issues of shopping, configuration, and installation are faced in some degree by all users of computing. But solving these "shopping" and informational issues will not always suffice to get work done smoothly. For instance, deciding to buy a SPARC station and run it on a campus which has standardized itself on DOS machines may bring you into conflict with the local computer center, and their attempts to limit the sorts of machines they will service. Or there may be enough money to buy the computer, but not enough to support training for all lab staff; in the long term, this disparity may create inequities of usage and data access. And so on.

In the case of WCS, we discovered many such instances. Because they are common to many sorts of system development efforts and types of users, all are interesting for the design of collaborative systems. With the advent of very large scale systems such as the US National Information Infrastructure, they become pressing questions of equity and justice, as well as organizational efficiency.

LEVELS OF COMMUNICATION AND DISCONTINUITIES IN HIERARCHIES OF INFORMATION

Bateson [1], following Russell and Whitehead, distinguishes three levels of communication.[3] At the first level are straightforward statements about Newtonian reality, i.e. "the cat is on the mat." A discontinuous shift in context occurs as the statement's object is changed to "I was lying when I said 'the cat is on the mat'." This second level statement tells you nothing about the location of the cat, but only something about the reliability of the first level statement. A shift to the third level would involve a *meta-message* placing the statement "I was lying...." within a broader context which may also change the mode, making the message humorous, metaphorical, etc. Bateson says there is a gulf between context (or metamessage) and message "which is of the same nature as the gulf between a thing and the word or sign which stands for it, or between the members of a class and the name of the class... context *classifies* the message, but can never meet it on equal terms." (p. 249)

Theorizing this gulf, Bateson and others went on to classify levels of learning with similar distinctions and discontinuities. There is a first and second order difference in learning something and learning *about* learning something; the difference between using a tool and learning to choose among categories of tool. Between second and third are the even more abstract differences of learning to choose among categories, and learning that there are theories of categorization itself. As the epigraph to this paper indicates, of course the regress upwards is potentially infinite. For our purposes here, however, we identify here three levels of issues that appear in the process of signing on and hooking up, and discuss each with respect to the worm community and WCS. As with Bateson's levels of communication and learning, the issues become less straight forwardly Newtonian or material as you move up. This is not an idealization process (i.e. they are not

---

3. He also discusses "zero learning," an a-contextual signal or stimulus, "the immediate base of all those acts (simple and complex) which are not subject to correction by trial and error" (p. 287; generally, pp. 279ff)

less material and more "mental"), nor even essentially one of scope (some widespread issues may be first-level), but rather questions of context.

Level One are issues which may be solved with a redistribution or increase of extant resources, including information. Examples would be answers to questions such as: What is the email address of WCS? How do I hook up my SPARC station to the campus network? Rationalistic approaches rarely move beyond this level; resource-related solutions, however, may be embedded in more complex second level issues, discussed below. **Level Two** are issues which result from unforeseen or unknowable contextual effects, perhaps from the interaction of two or more first-level issues. An example here is given above: what are the consequences of my choosing Sun instead of Mac, if my whole department uses Macs? If I invest my resources in learning WCS, are there other more useful programs I am neglecting? Again, these issues may form a component of more complex discussions taking place on the next level. **Level Three** issues are inherently political (broadly speaking); they involve permanently disputed problems whose resolution is contingent upon social or cultural norms. These would include: which aspects of genetics are emphasized in system graphics? Is competition or cooperation more important in privacy concerns? Complexity or ease of use more important in interface design? Such questions may arise from an interaction of other issues, such as trade-offs between platform choice, sophistication and training ease.

In this sense, infrastructure *is* context for both communication and learning within the web of computing [21]. That is, computers, people and tasks have some systemic properties which jointly contribute to message effectiveness. Bateson notes: "The separation between contexts and orders of learning is only an artifact... maintained by saying that the contexts have location outside the physical individual, while the orders of learning are located inside. But in the communicational world, this dichotomy is irrelevant and meaningless...the characteristics of the system are in no way dependent upon any boundary lines which we may superpose upon the communicational map."(p. 251)

Information infrastructure is not a substrate which carries information on it, or in it, in a kind of mind-body dichotomy. The discontinuities are not between system and person, or technology and organization, but rather between contexts. Here we echo much recent work in the sociology of technology and science which refuses a "great divide" between nature and artifice, human and non-human, technology and society.

What constitutes the dichotomies has the same conceptual importance for the information infrastructure that Bateson's work on the double bind had for the psychology of schizophrenia. If we, in the CSCW implementation process, are effectively building conflicting messaging systems which do not recognize the discontinuous nature of the different levels of context, we end up making work itself impossible in a manner precisely analogous to schizophrenegensis.

First Level Issues

The first level issues in this setting center around the installation and use of WCS, and include finding out about it, figuring out how to install it, and making different pieces of software work together.

Informational Issues. Potential users need to find out about the system, and need to determine the requirements for its installation and use. "Shopping" involves decisions about hardware and software, and may also involve agreements with other departments to share resources or funding; at one major lab, the "worm" people have WCS loaded onto a server owned by the "plant" people on the floor above them. This may also have its drawbacks: "The WCS and *acedb* are really on a machine upstairs, it belongs to the plant genome project people.... We can only use it evenings, weekends."(1:9:3, Brad Thomas, PD); "You can access acedb through the Suns downstairs, but it's not convenient. You can only do it after hours. People just won't use it."(1:4:1, Eliot Red, PD, p. 17)

Issues of Physical Access. In some labs, physical access is a critical issue. WCS may be located in an overcrowded and noisy room, it may be stuck into the corner of a lounge, it may be on a different floor of the building altogether. Recall above, according to their deal with the "plant" people, the "worm" people may only use WCS in the evening or on weekends. In many cases, systems use was anything but physically convenient: "Our computing is good compared to other labs. I finished up a Ph.D. at UCLA, they had one VAX, some PCs, you had to walk to another building to use the VAX. " (1:9:3, Brad Thomas, PD, p. 4)

When asked whether, in a future system, it would be desirable to replace lab notebooks with small palmtops or digitized pads, most researchers were dubious. Respondents at one cramped lab in an urban high-rise, simply noted that there was no place to put another computer — they did not even have space for all the necessary lab equipment.

Baseline Skills and Computing Expertise. Computing skills were quite unevenly distributed within the labs, and many seemed out of date. One senior researcher was not aware that databases were available without fixed-length fields; a PI made category errors in discussing operating systems and applications (equating "a Mac" and "a UNIX"). In general, PIs thought that the degree of skill was rising due to undergraduate and graduate training, but empirically this did not seem to be the case. Although there were certainly a few highly skilled people, and one or two with serious computing skills, these were not clustered in either the graduate student or postdoc categories. Sometimes there was a special "computer person" in the lab.

Skill-related issues can be cast as an access issue just as much as space or location. First level issues in this arena certainly include not only learning to use WCS software, but understanding the platform on which it runs. WCS itself is designed to be extremely user-friendly, and can be effectively used without much difficulty. The typical user is a graduate student, post-doc, or principle investigator with enough knowledge about both domain and community to read a genetic map and recognize the importance of the Worm Breeder's Gazette. One user comments: "I just turned it on, pushed buttons."(1:9:3, Ben Tullis, PS, p. 4).

In fact, most users find WCS to be fairly easy and intuitive. The platform on which it is based, however, is not (for biologists). WCS runs under UNIX, and both the operating system and software such as X-Windows or Suntools requires special expertise: "UNIX will never cut it as a general operating system. Biologists wont use it, it's for engineers. (Someone in the lab) had a printing question, took him three months to get something to print."(1:9:4, Bob Gates, GS)

Furthermore, many respondents were unclear about carrying our other kinds of computer tasks, such as uploading and downloading files from mainframe to terminal. This made it difficult for them to integrate WCS use with email correspondence, word processing files, and other Internet information spaces. Training often took place in a very haphazard way, and depended on everything from luck to personal ties: "The person who was the systems administrator until February was a good friend. Got a lot of push and shove from him, a lot of shared ideas."(1:8:3, Jeff Pascal, PD, p. 71-72) No lab offered special training in computing, although some students had taken classes at local computer centers. Several said that they only learn "exactly enough to suit what you have to do." (1:6:2, Carolyn Little, p. 45)

Second Level Issues

Second level issues can be analytically seen either as the result of unforeseen contextual effects, such as resistance to UNIX in the biological community, or as the collision of two or more first level issues, such as uncertainty during shopping combined with lack of information about how to hook up the system.

Technical Choices and a Clash of Cultures. Shopping and selection interact not only with training and ease-of-use issues, but with cultural issues within the worm community. For example, five people independently mentioned being put off by UNIX, usually in the context of comparing it favorably with the Mac. One PI mentioned having no base of UNIX knowledge available from the local computer center, although he had taught himself enough to run a SPARC station (1:1:1, Joe White, PI, p. 5). Others express similar sentiments: "It's a big problem. Biologists are Mac people, and UNIX is an evil word. Most people are afraid of it, and refuse to use it. "If it's not on Mac I don't want it." There are a lot of problems getting people to use it, rather than delegate the use of it". (1:8:2, Harry Jackson, GS, p. 67)

Yet UNIX is the language of computer scientists who support and maintain university computing. It becomes a basic "cultural" and training issue. Several people even had a theory that there are "two types of scientists—love or hate the computer," and that "the only way they'll ever do it is by force"(1:8:3, Jeff Pascal, PD, p. 71-72). They attribute computer use to "some kind of natural affinity"(1:4:1, Eliot Red, PD, p. 17).

Paradoxes of infrastructure. This second level issues refers to the contextual effects of skill and local configurations in relatively rich and relatively poor labs. One of the poorest labs, for example, which was still running outdated IBM PC-XT equipment, was actively using the system, had developed its own databases, and tracked strain exchange with a degree of sophistication unparalleled in the community. The richest lab, on the other hand, which had just received a substantial grant from the Human Genome Initiative to completely "hook up" the entire biology infrastructure on campus, was unable to operate the system through a combination of bureaucratic tangles. In this lab, getting hooked up required a little physical initiative, according to one of the graduate students:

> "No one will put the wires in, though. It's a huge problem. There is a one year wait for Ethernet connections. So we made a deal with the network people that we'd run wires and they'd connect it up. {*What do you mean, the network people? Who are they?*} The deal was with network services. They manage all the campus networks. So J____ has been the one who has dealt with Sun, though."(1:9:2, Steve Grenier, GS, p.100)

The PI, Linda Smith, confirms that they did, indeed, have to string their own cables; at the time of the interview, they were waiting on the delivery of the SPARC stations, expected within the week after which she anticipates having to spend a lot of time to "get the software underway".

In contrasting these two labs, we learned that the Principal Investigator of the poorer lab was an old crystal radio hacker, and loved to "play around" with software, wires, and communication at a distance. He had spent hours of his spare time designing software and learning to use the Internet, including designing custom software for the lab. In this context, the infrastructural limitations of the lab became interesting challenges, and disappeared under his skilled usage at solving second level problems.

Infrastructural issues arise in another sense as well. Only one lab praised their local computer center; most felt that the computer centers were uncooperative or overburdened. In addition, they did not feel that the computer centers were knowledgeable about, or even interested in, relevant applications packages. Institutions that have outstanding computer support, an extremely knowledgeable and capable technicians (true for at least two very technically-oriented schools), may have no interest, and offer no organizational mechanisms, for translating that expertise to highly domain-specific questions, applications, and issues.

> "Computing support s**** at (this institution). I called the center for help with installing WCS on the Sun and they basically told me, find a UNIX guy, buy him some pizza. If we have problems with the network or programs they support, they do it. If you didn't buy your hardware from them, forget it. If they don't support your software, forget it. It's handled on a department by department basis. Biology has no infrastructure."(1:9:4, Bob Gates, GS, p. 3)

Who "owns" a problem or an application is locally-determined, and attribution of problem ownership makes a great difference in individuals' ability to get the help they need. It may take an order of pizza, or it may take more carefully developed linkages with computer science or other on-campus resources in order to bring expertise to bear on local problems. These issues are of especial concern to post-docs looking to start up their own labs with increasingly limited funds. WCS is seen as a tool of richer labs: "Most of the upper tier has the WCS" (1:9:4, Harry Markson, GS, p. 6). A graduate student who wants to have his own lab one day describes WCS in its current implementation as "a rocket" when "we need a Model-T" (1:9:4, Marc Moreau, GS, p. 8); a post-doc with plans to start his own lab within the next two years complains that "half a system for everyone is better than a really great system for just a few labs," and adds:

> "Look, we had to hire (a computer specialist affiliated with another lab). Even the computer guys here (two graduate students) worked on it three weeks, and they couldn't load the (WCS) system. Its oriented to big labs."(1:9:4, Jay Emery, PD, p. 7).

He adds, "If it's not on a Mac or IBM, it won't get to people," and suggests, "you need a modular system, you need to be able to have parts of the database running on the Mac, *reach the small labs*" (emphasis added).

Tensions between a Discipline in Flux and Constraints as Resources. On a different note, what might be seen as constraints that could be overcome with technology may become resources from a different perspective. We proposed that it would be trivially easy to make *The Worm Breeders Gazette* available on a continual-update basis. On the one hand, continual updates serves the needs of a very fast-moving community: "The faster the (WCS) update, the better. ... You do it though the *Gazette*, you contribute regularly. You're competing (with other labs) on the same gene."(1:9:3, Brad Thomas, PD, p. 3)

Yet other respondents objected strongly to this option, even though they worked in the same competitive environment. Objections centered around the utility of community-imposed deadlines on structuring work, both in terms of *submitting* and *reading* articles: "I would run the newsletter *exactly* how it's run now. Just leaving it open ended is not good. If there is infinity there is never a time to review things. And no deadlines." (1:1:3, GS, John Wong, p. 9); "There is something to be said for deadlines. Even six times a year, and it becomes background noise. (1:7:1, PI, Gordon Jackson, p. 56). The deadline, in ethnomethodological terms, was both constraint and resource.

Third Level Issues

Third level issues are those which have been more commonly identified by sociology of science as problem-solving disputes, disputes between schools of thought, paradigms, and such. These permeate any scientific community, as all are interdisciplinary and heterogeneous. Third level issues may not be immediately recognized by members of the community as such. Nevertheless, they have long-term implications.

Triangulation and Definition of Objects. Different lines of work in the worm community come together in sharing information, including genetics, molecular biology, statistics, etc. One person explains, "I came from (another lab) where I was working on frogs (1:9:3, Brad Thomas, PD, p. 1). Another person describes himself as "really a developmental geneticist," and adds that a few years ago, "the field was smaller; ... now many people are coming from outside, from mammals, protein labs"(1:9:4, Harry Markson, GS, p. 5). Many people move into the worm community from other areas after graduate school. Differences may fall along the classical lines of organismal biology vs. molecular or genetic research: "I am more of a wormy person. That's true of the community in general. Sometimes you choose a system that's more organismal."(1:4:9, PD/RS, Jane Sanchez, pp. 32-33)

Collaboration may take place across disciplinary boundaries:

> *{Are you collaborating with anyone?}* "I'm collaborating with people in the worm and non-worm community. Mostly immunologists in the non-worm community, people interested in the immune system. In the worm community, I'm collaborating with (a person on another state), on (a particular gene)."(1:9:4, Harry Markson, GS, p. 2)

Disciplinary origin and current area of work affects the kinds of information individuals need, and the tools and data sources with which they are familiar. Those studying the organism for its own sake differ in their information needs from those using it exclusively as a model organism; others, in turn, have their own expectations for WCS data: "What you'd want is a parts list, a list of cells. ... If it's a neuron, its connections with other neurons. ... That's for neurobiologists."(1:9:4, Harry Markson, GS, p. 5); and "You need more options, especially for sequencing.... We need to work with subsets of sequences, examine them in more detail."(1:9:3, Brad Thomas, PD)

If respondents identify the system with a particular sub-line of work and not as a general utility, there may be barriers to usage.

*Multiple Meanings and Data Interpretation.*The nature and character of the community is changing as more people enter the "worm world" from other disciplines. Currently it has between 600 and 700 members: "It's neat that it's exciting now, but it's also strange to have so many people... (1:4:9, PD/RS, Jane Sanchez, pp. 32-33)

The multiple meanings or interpretations which a particular communication has turns out to be important at all levels. For example, we suggested to respondents that it might be useful to have a "who's working on what" directory in the system. This seemed like a good idea to some, but we were surprised to find that even the idea of announcing what you are working on can be problematic, especially for postdocs about to come onto the job market, or graduate students not yet sure of their dissertation topic. While most did favor having some sort of directory, or perhaps extending the address book, some would hesitate to put in certain kinds of information, or wanted announcements delayed until "they had findings":

> There's always a problem you're going to get scooped. You always walk a very fine line. There's a lot of people working on my problem. ... if you publish in the Gazette you can lay claim to it. People would respect it. There have been some clashes, some labs trying to glom on to how much they can. It's going to be a struggle from here on out.
>
> It's complex with the claim staking. That's why you want to get into it far enough so you can get ahead—before you announce it. If you could preface it with "wild speculation" (laughs)... well, there's a lot of times those can have a big payoff. But then again if five people jump on it, and in the meantime you're scooped... that's not so good! (1:8:1, GS, Mike Jones, p. 62)

So even a simple directory can be a signal, a revelation, or a flag, and the timing of disclosure of the problem can be important. Similarly, the meanings carried by the different communication channels are important. "You can be wrong with no stigma" in the newsletter, pointed out one graduate student, but actually "broadcasting" something over the system would be another matter altogether, and much more scary. One PI, talking about a graduate student who was just beginning to use WCS for annotations, explained:

> "People are reluctant to do annotations. ... It's the fear of putting yourself on the line. Making a commitment to what you're doing. It means being wrong in the eyes of your colleagues. "(1:1:1, Joe White, PI, p.6)

One post-doc suggested the implementation of a personal level and a public level of annotation (1:9:3, Brad Thomas, PD, p. 3); another PI, however, became angry at this idea. From his perspective, local annotation would work directly against WCS's commitment to *community-wide* sharing of information and turn WCS into a local tool.

Trust and reliability of information is a concern for scientists: articles in the *Gazette*, annotations, etc., have well-developed conventions about data quality, undeveloped as yet for electronic media. There are sometimes no clear-cut "answers" to questions, especially in a community populated with multiple viewpoints. In general, "there is no right or wrong, ... you have to reach consensus on things, you have to look at labs, which labs you trust more"(1:9:3, Brad Thomas, PD, p. 4). One of the scientists suggested some sort of screen upon booting up, with "some sort of caveat about citations." (p. 7) He notes wryly that people will cite you as a foil when you've said something incorrect in any event, however, and that there's no way to prevent this. All these instances of data meaning different things under different circumstances—who notifies whom and when, what medium is used, who makes an annotation, or why a particular citation is and isn't included—require knowledge of the community that is not captured in any formal system.

Network Externalities and Electronic Participation. The notion of externalities originates in economics and urban planning; a city may be said to afford "positive externalities" of cultural resources. For an artist, New York's externalities outweigh those available in Champaign, Illinois, although other amenities such as cost of housing and safety may be greater in the latter. A network externality means that the more actors actively participate in a system or network, the greater the potential, emergent resources for any given individual. Externalities may be negative in that eventually, not being "hooked up" may make it impossible to participate effectively within a given community of work or discourse.[4] For instance, the telephone network became a negative externality for those businesses without telephones some-

---

[4] This is distinct from the notion of "critical mass, "which focuses on the number of subscribers/users at which system use becomes viable.

time in the early 20th century; electronic mail has recently acquired a similar status in the academic world. For some purposes, standards (as in information standards) form important aspects of network externalities - i.e., users of non-standard computing systems are at a disadvantage as network externalities become intertwined with particular operating systems and data interchange protocols.

One goal of the system is democratization of information — the facilitation of access to critical data through a uniform interface. Yet the more central WCS becomes to the community either as a whole, or as defined by key labs, the more those who cannot sign on along with the others will suffer. The "politics of reinforcement" suggest that rich labs—either in terms of extant computing infrastructure or in their ability to procure it using internal resources—will get richer as network externalities become more dense [22]. This issue may be receding in importance as alternatives to WCS emerge via data available at ftp sites and through gophers; much of the information available via WCS can now be "pulled from the net." Nevertheless, WCS is superior in its possibilities for graphical representation, and some forms of data analysis need such tools.

Issues of participation abound. For instance, a key repository is the genetic map, which represents the relative positions of genes on the chromosomes; another is the physical map, which represents cloned fragments of worm DNA and how they overlap to form the chromosomes [29].

> There's a time problem. You want experts doing this, but you want to do your own stuff, you don't want to maintain a database. If you want this to serve a global community, you have to get the data properly defined. (1:9:3, Brad Thomas, PD, p. 4)

> There are data that should be on the (physical) map, but they are buried in labs all over the world... When it was fragmented, people sent in clones. Now it's filled in, more coherent. The need to communicate back broke down. There used to be a dialogue, now there's a monologue. They don't bother telling Cambridge they've cloned genes. ... With the genetic map there's still dialogue. (1:9:3, Ben Tullis, PS, p. 5)

Some of this is an issue of time; two attempts at an electronic bulletin board "died out within two weeks due to lack of contributions" (1:9:4, Bob Gates, GS, p. 3). Annotation and updating takes work, and "it's not of immediate profit"(1:9:5, Sara Wu, PD, p. 6). Competition is also a factor. Someone who overheard the question on dialogue breakdown contributed the following comment: "Yeah, like (one very well-known) lab, ... not sending in a note (on *y*)). And (another well-known) lab, they don't publish things when (they) are close to a gene they're working on". (1:9:3, Kyle Jordan, PD, p. 5). A graduate student in the same lab echoes a similar view of data-sharing: "The people who really need to know already know."(1:9:4, Bob Gates, GS, p. 3)

WCS does not maintain the databases or publications featured in these discussions, but it does provide uniform access and an easy-to-use interface to them (once the system is up and running). It derives a significant part of its own value from community participation in their upkeep and maintenance. Without community commitment to the maintenance and upkeep up these materials, WCS has neither value nor legitimacy as a system that fosters either communication or collaboration.

Tool Building and the Reward Structure. Finally, the role of tool building and tool maintenance may be undergoing a shift as computer-based tools become more prevalent. The tension between traditional notions of work and tool-building, and new opportunities for the same, have already been observed in at least one other academic community [28]. Many of the people interviewed could list a number of tools (from techniques, to compilations of targeted information, to analysis software) that they would like to see added or perfected. One person was there in the early days of acedb and says he still contributes, sending e-mail about bugs and suggestions for graphics. Others construct local tools, such as annotated gene lists (a project carried out part-time over the course of a year), using data from WCS. Yet another person, mentioned above, is teaming up with a computer scientist to develop tools for data visualization. The difficulty is that there are no clear rewards for this kind of work, except for the contributions the tool makes to one's own work. The biologist working with the computer scientist doesn't get any "credit" for this within his own discipline (he anticipates having tenure by the time this project begins). As one postdoc put it, a comment appropriate for both sides, "... there are a hundred things that are useful, but you don't get a Ph.D. for it"(1:9:4, Jay Emery, PD, p. 8).

DOUBLE BINDS: THE ELECTRONIC TRANSCONTEXTUAL SYNDROME

> "Double bind theory is concerned with the experiential component in the genesis of tangles in the rules or premises of habit. I... assert that experienced breaches in the weave of contextual structure are in fact' double binds' and must necessarily (if they contribute at all to the hierarchic processes of learning and adaptation) promote what I am calling transcontextual syndrome." [1], p.276[5]

Until now we have simply followed Bateson's typology for learning in categorizing infrastructural barriers and challenges. Bateson's levels of learning originated in communication theory and cybernetics. The formal statement of the

---

[5] Quotes here taken from [1], unless otherwise noted.

problem is expressed logically, following Russell and Whitehead. In "The Logical Categories of Learning and Communication" (pp. 279-308), he notes that a category error such as confusing the name of a class and a member of that class will create a logical paradox. In the world of pure logic, this appears as a fatal error, because such logical systems seem to exist outside of time and space. In the real world, particularly the behavioral world, however, time is an important factor. When a paradox of this form appears, people cope with it as best they can by working within multiple frameworks or "world views," maintained serially or in parallel.

We have adapted Bateson's levels of communication and learning below as a theory of learning and organizational change, emphasizing the complexity of infrastructural relations and communication in organizations and between designers and users [5]. **Zero learning** is a kind of primitive stimulus-response. **Learning 1** is change in the specificity of response, and places the stimulus within a particular, broader context; it implies the ability to assume a repeatable context. We use it to address instances of people learning about the system, such as how to access it. **Learning 2** is change in the *process* of Learning 1(context broadened to include choices about alternatives); "learning to learn" here means involving the recognition of contextual patterns of discontinuities. We apply this to cases where two forms of Learning 1 "collide," requiring an awareness of the next contextual level for resolution. Finally, **Learning 3** is "a change in the system of sets of alternatives from which the choice is made" (p. 293) or learning "to perceive and act in terms of the contexts of contexts" (p. 304). In the case of WCS, Learning 3 addresses issues that cross organizational and disciplinary boundaries, and may involve paradigmatic disputes.

Bateson's famous theory of the "double bind" in schizophrenia extends the idea of the specificity of communicative levels to the family as a communicative system. The double bind is an instance of what he calls the "transcontextual syndrome," and may be answered with either schizophrenia or creativity. The syndrome occurs when a message is given at more than one level simultaneously, or an answer is simultaneously demanded at a higher level and negated on a lower one. In the family-schizophrenia scenarios he uses, the parent may say, for instance, "go get your coat, it's cold outside," while closing the door to the cloakroom and standing in front of it. Attempts to point to the contradiction are met with denial, "of course I want you to get your coat; didn't I say so?" These double-binds occur in organizational contexts as well; middle managers in rapidly-changing environments, for instance, are frequently caught between the goals and expectations articulated by senior management and their actions with respect to budget allocation and performance evaluation [23]. Bateson notes: "There may be incongruence or conflict between context and metacontext." (p.245) Over time, schizophrenia may result, where the child insists on seeing the literal level and ignoring context, or inappropriately seeing context literally. The often-noted poetry in schizophrenic language is a result of this refusal—good poets deliberately play with transcontextual double entendres. Formally, this ignores or trangresses the gulf between message and metamessage.

People attempting to hook up to complex electronic information systems may encounter a similar gulf. The rhetoric surrounding "hooking up" to complex systems, including the Internet, makes "signing on" sound remarkably straightforward, totally Level One. Why, then, do so many problems arise, and how can we characterize them? We identify below several *infrastructural transcontextual syndrome*.

1. *The gap between designers and users*. What is simple for one group is not for the other, so what from the point of view of designers appears to be a Level One message contains a double bind within it for users. For instance, when asked about getting onto the system, designers of WCS might say, "Just throw up X Windows and ftp the file down." The tone of the message is clearly Level One, but for the relatively naive user, it is the functional equivalent of the parent blocking the cloakroom door: What is an "X Window,"and where do I get it; what does it mean to ftp a file down? Even more experienced users need to know what password to use (they vary) when obtaining a file via ftp, or need to know how to find this out. A Level One instruction becomes a complex set of Level Two questions, closely related to the user's own level of expertise.

Another part of this type of double-bind is an infinite regress of barriers to finding out about complex electronic information systems. If you don't know already, it's hard to know how to find out, and it isn't always clear how to *abstract* knowledge from one system to another. There is no single book that can tell you from scratch; the only way in is to switch contexts altogether and work more closely with designers. This may account for the power of the participatory design, in which designer and users work together to develop a shared context. It may simultaneously account for the difficulty of explaining or popularizing the model outside of Scandinavia, the working context of which differs greatly from elsewhere.

2. *The gap inherent in discussions within the worm community*. Within the worm community itself there exists a Level Two/Level Three double bind. Just as Level One statements can engender Level Two questions, so Level Two discussions can open up issues at the third level. Discussions about package or platform choice among some members of the community become discussions about resource allocation, data interpretation, and network externalities. Take, again, the case of "ftping a file down."A discussion of learning

about ftp, about alternatives such as gophers, etc., moves to questions of access across labs, of database maintenance and data reliability, and of norms and standards within the community for contributions to the database.

These issues are particularly poignant ones for "older" members of a fairly new community, who recognize that technical choices and decisions made at the second level—evaluations of the options for responding to Level One signals—have the ability to dramatically affect third level issues. In the worm community, the concerns involve changes in the composition of the community as "outsiders" join, and what this means for data interpretation and tool construction. The concerns also center around the multiple roles that research on the organism plays: end in and of itself vs. model organism for the Human Genome Initiative. Tools aimed at Level Two problems affect deeply the options open to the discipline when addressing Level Three questions and setting broad conceptual directions, often in contradictory ways.

3. *The gap between routines and rapidly-growing infrastructure.* The seeming paradox of why many of our respondents chose to use Gopher and other simpler, public-access systems rather than WCS involves another sort of double bind. To take on board the custom-designed, massively convenient and powerful WCS is to suffer a massive inconvenience at another level—the meeting of work habits, computer use, and lab resources. With respect to computer use, scientists are very busy people. Simpler, less powerful systems may be picked up more quickly and interfere less with other work habits than those which require substantial investment in changing habits and infrastructure. Similarly, less expensive options (such as those available via the Internet), while less complete or less elegant (especially in terms of supporting graphical data presentation), may be far less disruptive of resource allocation patterns. Gophers can be accessed from the terminals and connections supplied by most campus computing facilities; workstations must be purchased specially by the lab. Internet computer support is supplied by most academic institutions; support for WCS must be acquired at the expense of other lab personnel.

From one perspective, WCS fits scientific information needs: links between disparate pieces, graphical representations, layers of detail, etc. But within the larger context of infrastructure, it may conflict. Science is an integrative domain. The construction of WCS, while it integrates a large number of materials, does so in a very customized fashion. Lab notebooks, by way of contrast, are extremely open and integrative documents. At the same time, computing infrastructures, including gophers, ftp sites, etc., while still "primitive," fit more closely with this integrative model.

We encountered a persistent idea among respondents that they were "just about to" be hooked up with the system, and that the barriers to hooking up were in effect trivial. Sometimes this even caused them to say that there *were* using the system, whereas observations and interviews in fact showed that they were not. For instance, when trying to find a site to observe in a large city with several universities and several labs listed as user sites, one of the authors spent almost a week tracking down people who were actually using the system. No one she talked with was using it, but each person knew of someone else in another lab who supposedly was. After following all leads, she concluded that no one was really using the system, though they all "meant to," and figured that it would be available "any day now."

This is not difficult to observe ethnographically, but presents a real difficulty in administering surveys about use and needs. It is also a good example of a response to a double bind. It is clear that this representation is not mendacious, but a common discounting of what seem, from a distance, to be trivial "plug" difficulties. The above observations of the difficulties associated with hooking up and getting started, coupled with infrastructural limitations would suggest that these issues are not trivial at all. In fact, these issues turn out to be lethal as they become both chronic and ubiquitous in the system as a whole.

4. *Double levels of language in design and use.* Finally, there may be double binds in those aspects of the system which are self-contradictory, between formal system properties and informal cultural practices. The language of design centers around technical capacity; the language of use around effectiveness. Robinson [27] notes that for CSCW systems, only applications which simultaneously serve both the formal, computational level and the informal, workplace/cultural level are successful. Gasser [11] similarly identified a variety of "workarounds" developed to overcome the rigidity of a transaction processing system; Star and Gerson [32] showed that users of an insurance claims processing system developed elaborate informal procedures for workarounds.

These problems/solutions can be expressed as evidence of a double bind. The "message" of the designer focused on the technical representation of a particular set of data (i.e., customer records), and the efficiency of processing them to meet a particular goal (i.e. claims processing). The "interpretation" of the user focused on the need to mediate between conflicting viewpoints (i.e., doctors vs. representatives for large customer groups), and the need to develop effective workflow. Orlikowski discusses more narrowly the conceptualization of software design methodologies and tools as languages [24] and, together with Beath, examines the consequences of non-shared languages or superficial user participation in [2]. This double-bind is captured in the discussion of the Mac vs. UNIX, and what it means in terms of a clash of cultures between biologists and computer scientists. On one level a discussion about operating systems, on another it is representative of two world views and sets of values with respect to the relationship between technology and work, and the relationship between the tool and user.

Addressing Double Binds and the Role of Multi-Disciplinary Development Teams

Having identified different instances of double-binds, we are left with the problem of identifying a set of solutions or recommendations. One of the key difficulties in resolving double binds is *recognizing* them in the first place: individuals involved in the situation itself may not be able to identify instances of this transcontextual syndrome. The other key difficulty, once this kind of double bind is identified, is in articulating it in a way in which the other party will recognize or accept. Recall the coatroom example above, in which pointing out the incompatibility in messages met with denial; similarly, a parent might reject affectionate behavior on the part of the child, then, when the child withdraws, accuse the child of not loving the parent. The child has no capacity for analyzing and correcting this inconsistency.[6]

A computing-related analogue would be the denial on the part of developers or system administrators that technical difficulties really mask higher-order conceptual problems centered around work practices and community standards, and a failure on the part of users to recognize the complexity of their work domains, their hidden assumptions, and the various motivations of the stake holders involved. If we expect designers to learn about the formal and informal aspects of the user domain, to learn to "speak their language," we must ask users to meet designers "halfway" by learning *their* language and developing an understanding of the design domain. If designers are at fault for assuming that all user requirements can be formally captured and codified, users are often equally at fault for expecting "magic bullets"—technical systems that will solve social or organizational problems.

The "fault" really lies in neither camp, but in the general misunderstanding of the nature of miscommunication and the double-binds of information, context and meaning. The emergence of multi-disciplinary development teams may help to alleviate aspects of the transcontextual syndrome identified above, with ethnographers helping users and designers bridge the contextual divide. "You can ftp that from such-and-such a site" might well give way to "I can give you the ftp address, but the kind of data you'll get won't be detailed enough for what you want to do with it." By sharing an understanding of both the formal, computational level (traditionally the domain of the computer programming and systems analyst) and the informal level of workplace culture, double binds may be more easily identified and resolved as all members of the team learn to correctly identify the various orders or levels to which a message might belong.

ACKNOWLEDGMENTS

WCS was supported by NSF Grant #IRI-9015407. Co-PIs Bruce Schatz and Sam Ward, and developers Terry Friedman and Ed Grossman were extremely generous with their time, comments, and access to data; we also thank our anonymous respondents for their time and insight. An earlier version was presented at the *Conference on Computing in the Social Sciences,* 1993. Ruhleder thanks Michael Elmes for an interesting discussion on double binds in organizations. Geof Bowker, Tom Jewett, Alaina Kanfer, Rob Kling and Stefan Timmermans provided helpful insights and comments. Star's work was also supported by the Program in Cultural Values and Ethics and the Advanced Information Technologies Group, University of Illinois, and by the Institute for Research on Learning.

REFERENCES

1. Bateson, G., *Steps to an Ecology of Mind.* New York:Ballantine, 1972.

2. Beath, C. and W.J. Orlikowski, The contradictory structure of systems development methodologies: Deconstructing the IS-user relationship in information engineering. *Information Systems Research,* forthcoming.

3. Boland, R., Jr. and W. Day, The experience of system design: A hermeneutic of organizational action. *Scandinavian Journal of Management* (1989) **5**: 87-104.

4. Bowker, G., Information Mythology and Infrastructure, Pp. 231-247 in L. Bud, Editor, *Information Acumen: The Understanding and Use of Knowledge in Modern Business.* Routledge: London, 1994.

5. Brown, J.S. and Duguid, P. Toward a unified view of working and learning, and innovating. *Organization Science* (1990) **2**: 40-57.

6. Ellis, C.A., S.J. Gibbs, and G.L. Rein, Groupware: Some issues and experiences. *Communications of the ACM* (1991) **34**: 38-58.

7. Eveland, J.D. and T. Bikson, Evolving electronic communication networks: An empirical assessment. *Office: Technology and People* (1987) **3**: 103-128.

---

[6] This example is analyzed in more detail in [1], pp. 217–218; see pp. 212–221 for further examples.

8. Feldman, M., Constraints on communication and electronic messaging. *Office: Technology and People* (1987) **3**.

9. Forsythe, D. Engineering knowledge: The construction of knowledge in artificial intelligence. *Social Studies of Science* (1993) **23**: 445-477.

10. Forsythe, D. and B. Buchanan, Knowledge acquisition for expert systems: Some pitfalls and suggestions. *IEEE Trans.Systems, Man and Cybernetics* (1989) **19**: 435-442.

11. Gasser, L., The integration of computing and routine work. *ACM Transactions on Office Information Systems* (1986) **4**: 205-225.

12. Goguen, J. and C. Linde, Methodology for requirements elicitation, (1991), Oxford University Computing Lab, Centre for Requirements and Foundations.

13. Gorry, A., *et al.*, The virtual notebook system: An architecture for collaborative work. *Journal of Organizational Computing* (1991) **1**: 223-250.

14. Greenbaum, J. and M. Kyng, eds.. *Design at Work.* London: Erlbaum, 1991.

15. Heath, C. and P. Luff. Collaborative activity and technological design: Task coordination in London Underground control rooms. in *ECSCW'91. Proceedings of the Second European Conference on Computer-Supported Cooperative Work.* Amsterdam: Kluwer Academic Publishers, 1991.

16. Hughes, J.D., D. Randall, and D. Shapiro. Faltering from ethnography to design. in *CSCW.* '92Toronto: ACM, 1992.

17. Hughes, T.P., *Networks of Power: Electrification in Western Society, 1880-1930.* Baltimore: Johns Hopkins University Press, 1983.

18. Hughes, T.P. The Evolution of Large Technological Systems, Pp. 51-82 in *The Social Construction of Technological Systems,* W.E. Bijker, T.P. Hughes, and T. Pinch, Editors. MIT Press: Cambridge, 1989.

19. Jewett, . and RT Kling, The dynamics of computerization in a social science research team: A case study of infrastructure, strategies, and skills. *Social Science Computer Review,* (1991) **9**: 246-275.

20. Jirotka, M., Ethnomethodology and Requirements Engineering (1991), Centre for Requirements and Foundations, Oxford University Computing Lab.

21. Kling, R. and W. Scacchi, The web of computing: Computing technology as social organization. *Advances in Computers* (1982) **21**: 3-78.

22. Kraemer, K. and J. King, *Computers and local government.* New York: Praeger, 1977.

23. Mishra, A.K. and K.S. Cameron. Double Binds in Organizations: Archetypes, Consequences, and Solutions From the U.S. Auto Industry, in *American Academy of Management.* (1991).

24. Orlikowski, W.J., Using Tools and Forging Reality: Exploring the Cultural Implications of Systems Development Methodologies and Tools, (1990), Sloan School of Management, Massachusetts Institute of Technology:

25. Pool, R., Beyond databases and e-mail. *Science* (13 August 1993) **261**: 841-843.

26. Rice, R.E., Electronic emotion: Socioemotional content in a computer-mediated communication network. *Communication Research,* (1988) **14**: 85-108.

27. Robinson, M., Double-level languages and co-operative working. *AI and Society,* (1991) **5**: 34-60.

28. Ruhleder, K. and J.L. King, Computer support for work across space, time and social worlds. *Journal of Organizational Computing,* (1991) **1**: 341-355.

29. Schatz, B., Building an electronic community system. *J. of Management Information Systems,* (1991) **8**: 87-107.

30. Schmidt, K. and L. Bannon, Taking CSCW seriously: Supporting articulation work. *Computer Supported Cooperative Work (CSCW)*, (1992) **1**: 7-40.

31. Sproull, L. and S. Kiesler, *Connections: New Ways of Working in the Networked Organization.*, Cambridge, MA: MIT Press, 1991.

32. Star, S.L. and E.M. Gerson, Analyzing due process in the workplace. *ACM Transactions on Office Information Systems,* (1986) **4**: 257-270.

33. Suchman, L. and R. Trigg, Understanding Practice: Video as a Medium for Reflection and Design, Pp. 65-89 in *Design at Work,* J. Greenbaum and M. Kyng, Editors. Lawrence Erlbaum: London, 1991.

34. Yates, J. and W. Orlikowski, From Memo to Dialogue: Enacting Genres of Communication in Electronic Media, Sloan School of Management, MIT, 1993.