

# Issues in Infrastructure Development: Proposed interventions for addressing third order issues

**Ingbert R. Floyd**

Grad. School of Library and Information Science  
501 E. Daniel Street, MC-493, Champaign, IL  
61820-6211 USA  
ifloyd2@illinois.edu

**Michael B. Twidale**

Grad. School of Library and Information Science  
501 E. Daniel Street, MC-493, Champaign, IL  
61820-6211 USA  
twidale@illinois.edu

## ABSTRACT

In nascent research laboratories, researchers interested in collaborating often have only vague ideas of the kinds of research problems large-scale collaboration can enable them to address, but little idea of how to collaborate. In many cases we have observed, researchers have to negotiate shared values, understandings, and practices in order to begin to collaborate successfully, yet such negotiations can be seriously stalled by third order issues in infrastructure development. In this paper, we propose interventions for addressing third order issues and argue that CSCW system designers need not regard such issues as outside the scope of consideration.

## Author Keywords

PD, patchwork prototyping, CSCW, laboratories.

## INTRODUCTION

One of the major difficulties in creating effective CSCW systems for newly forming groups is the challenge of developing collaborative practices where none have previously existed (e.g., [7]). This is especially true in research collaboration because many interdisciplinary fields that are currently being targeted by collaborative development efforts have little established research collaboration practice, and what practices exist cannot be scaled to the large-scale collaboration which is desired. Thus collaborative practices must be developed concurrently with the cyberinfrastructure.

There are many difficulties in developing collaborative practices, including academic reward structures (both in local institutions and in obtaining general recognition in the field), publishing norms, ideals of how research ought to be conducted (e.g., the lone researcher making brilliant

discoveries in the lab), etc. However, one of the most significant barriers is a lack of effective communication between researchers from different academic traditions.

In order to articulate the various levels at which infrastructure can fail to support collaborative work [9], proposed a three-tiered classification of issues in infrastructure development:

*First order* issues may be solved with a redistribution or increase of extant resources, including information.

*Second order* issues stem from unforeseen or unknowable contextual effects, perhaps from the interaction of two or more first-order issues.

*Third order* issues are inherently political or involve permanent disputes. ([9], pg. 11)

[9] suggest that major miscommunication often occurs when people involved in the design or use of infrastructures make category errors between these orders. For example, a person may talk about a certain issue as if it was a second order issue, when in reality his concerns are about third order issues but he does not know how to express his concerns in third order terms. When his listeners interpret his concerns literally, then significant miscommunication occurs. [9] term this phenomenon a *double bind*. This paper focuses on the three orders of issues as well as double bind miscommunication, both between researchers with different values and backgrounds, and between researchers and system designers.

## CSCW DESIGN AND THIRD ORDER ISSUES

Experience observing collaborative development suggests that CSCW designers and researchers often focus on the first two orders of issues in infrastructure development. A designer's primary focus is usually on resolving first order issues because they are concrete, and lend themselves to concrete solutions. Designers also acknowledge second order issues because these are issues which either serve as design constraints that must be worked around, or are issues which can be (theoretically) overcome by teaching, training, or building out new kinds of infrastructure support. However, designers often give up on third order issues,

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

CSCW 2008, November 8–12, 2008, San Diego, California, USA.  
Copyright 2008

because they are seen as outside the scope of the design work.

There is some justification for this reluctance to consider third order issues. These issues typically arise from larger structural features of the environment, or from different conceptualizations of the work that the CSCW system is intended to support held by members of different user communities. Certainly it is not the designer's place to inflict third order solutions, and user communities would reject such efforts if they tried. However, that does not mean the designer can play no role in their resolution.

Most CSCW researchers who are interested in designing systems approach their work from a computer science (CS) perspective. A major component of CS design is managing complexity. Thus CSCW designers typically focus on what they can directly affect and control, even if they do not limit their scope solely to computing applications and hardware. However, the design of a research collaboratory requires a true sociotechnical design perspective where the goal of the design activity may in part be to create an environment where the users of the system can become aware of and resolve some of their third order problems themselves.

While sociotechnical design has historically drawn on cybernetic systems metaphors to describe the object of attention, we feel that [6]'s ecological metaphors are more accurate. By emphasizing the fundamental difference in kind between the different components of sociotechnical ecologies (computing & analog technology, building layout, geography, policy, culture, human cognition, etc.) and seeing them as [1]'s *places of intervention*, we believe a richer understanding of the mechanics of the sociotechnical as well as a better understanding of how to intentionally influence those mechanics can be obtained. By extension, this metaphor suggests that sociotechnical design might better be conceptualized as guiding evolution or enabling evolutionary pathways within a sociotechnical ecology. Thus, true sociotechnical design involves fostering emergent activity. This allows factors like third order issues to be considered that are outside the direct control of the CSCW designer, but still will be affected by his or her work. The question then becomes, how can CSCW designers enable users to undertake this work?

### **ENABLING DESIGN FOR 1<sup>ST</sup> AND 2<sup>ND</sup> ORDER ISSUES**

Our experience with patchwork prototyping [3, 4] suggests an interesting mechanism for allowing users to resolve many third order issues. However, to understand our vision for addressing third order issues, it is important initially to understand how patchwork prototyping addresses first and second order issues.

Patchwork prototyping is an emergent method of participatory design (PD) observed while studying various research collaboratory development projects. It involves rapidly iterating (1 week or less per cycle) prototypes constructed via patching together open source software

(OSS) and web APIs with minimal glue code, with the entire process facilitated by extensive collection of feedback by project leaders who are leaders of the user communities. What is interesting, is that the incorporation of these prototypes in the daily activity of the users allows the users to prototype collaboration behavior and develop collaboration practices where previously none existed. Thus, not only are requirements for technological infrastructure developed, but social norms and practices are also developed.

### **First Order Issues**

First order issues are typically identified when users run into trouble with the current prototype. The feedback loops ensure that these issues are quickly discovered, and then responses can be provided reactively as issues emerge. Software issues can be resolved by iterating the prototype in appropriate ways. Information, baseline knowledge and computing expertise issues can be resolved by improving the interface, running training sessions, or providing temporary expertise when needed. Access issues can be resolved by reconfiguring space or providing new space.

In essence, first order issues are identified and resolved by bypassing formal requirements capture, but for good reason: researchers cannot know their requirements prior to using the prototype to start real collaboration. Designing cyberinfrastructure to create new forms of collaboration is an inherently wicked [8] or equivocal [2, 10] problem. It is different from more traditional CSCW efforts that try to automate existing workflows (historically, paper-based workflows) because in these cases the new collaboration that the systems are intended to support is much richer or deeper than any kind of collaboration that was possible before. The goal of these efforts is to use technology to change scientific practice, and it is impossible to know *a priori* what new forms this scientific practice will take.

### **Second Order Issues**

Second order issues are also identified via attending to the use of the prototype. However, identifying them takes considerably more skill. Many second order problems masquerade behind complaints which are seemingly first order in nature. Thus, cultural preferences for certain technologies (e.g., operating systems [9]) can masquerade behind complaints about interface problems or dissatisfaction with a prototype. However, this can also go the other way, where second order complaints hide first order issues: we observed people dismissing wiki technology wholesale because they had been exposed to a particular wiki that had a horrible interface. Teasing apart these kinds of double binds to uncover the underlying issues is vital, because otherwise the wrong "fix" will be applied. Yet patchwork prototyping allows for this kind of inquiry to happen: by switching one wiki technology with another (since both were open-source packages this was relatively easy), and providing a bit of information seeding and teaching, we were able to overcome the objection to

wikis and turned a hated technology into one participants got excited about and started to innovate with.

Similarly, patchwork prototyping can expose differences in opinions regarding which technological options to create. Thus, it can help resolve issues such as the [9] example of the tension between people who favored putting *The Worm Breeder's Gazette* online and automatically updating it with new articles as they were submitted, versus keeping it paper, or at least limiting how often new articles were published. By creating many different spaces where researchers are engaging in collaboration-forming activity, the researchers themselves find ways of proposing new features, and then creating dialog around those proposals, allowing different views to be expressed, and consensus to be reached. While there is no guarantee that consensus will emerge, at least the discussions of these issues become public. When no resolution occurs, the public nature of the conversations facilitates exposing the likely double bind and reveals the issue as a third order issue, accompanied by the appropriate conversational shift. Where the designers come in, is in moderating the discussion, and providing insights into how the technologies can be configured so that the conversation does not become triple-bound: where first order issues obscure the second order and third order conversations that need to happen.

Patchwork prototyping can also reveal paradoxes of infrastructure. While it typically uses web-based software, thus reducing some of the infrastructure problems referred to in [9], when unexpected resource constraints limit use, patchwork prototyping can reveal the phenomenon, and either more resources can be obtained, or the prototype can be reconfigured to work within the resource constraints. Thus the second order issue can be translated into a first order issue. Similarly, by collecting feedback and using log data, cases of "any day now" syndrome can be uncovered and treated, either by supplying the extra support that is needed, or by reducing or eliminating the gap between what resources are needed, and what resources actually exist.

### **IDENTIFYING AND ADDRESSING 3<sup>RD</sup> ORDER ISSUES**

Fundamentally, third order issues are issues that various groups in the user community of a particular cyberinfrastructure must work out for themselves. They are not easy issues to resolve, as often fundamental values can be in opposition to one another. In other cases, reward structures and resource allocation decisions can be profoundly affected. Third order issues are both resistant to intentional efforts to effect their resolution, and can be resolved without any intention due to changing features of the environment. Finally, third order issues are often incredibly complex, and it is often difficult if not impossible for any one person to get a deep understanding of all the nuances of such an issue. Taken at face value, it seems impossible to "design" a solution to these issues. However, for research collaboration to occur, these issues

must be resolved, as they can otherwise be fatal to the endeavor.

The first major hurdle for addressing third order issues is identifying them. The second is addressing them. CSCW designers typically cannot address third order issues. Whether a third order issue is deemed resolvable, whether efforts are made to resolve it, whether it is ignored in the name of progress, or whether the project is abandoned are decisions which various user group stakeholders must make. However, all too often, a project gets stalled because third order issues remain unrecognized and unaddressed. It is our argument that the CSCW designer can contribute to identifying issues as being third order in nature, and can suggest multiple options for addressing those issues which would allow a particular project to go forward.

In much the same way as first order issues can obscure second order issues, second order issues can obscure third order issues [9]. Second order conversations which are about ways of doing things can often be motivated by higher, third order issues such as value conflicts, resource competition, incompatible cultural practices, etc. In the same way as with the first order/second order double bind in the patchwork prototyping example, CSCW designer can play a vital role in identifying hidden, third order issues which are present in second order conversations, in order to appropriately raise the level of the conversation.

A classic example of this kind of third order issue is the negotiation of shared data standards by scientists with different disciplinary practices. While studying a scientific collaboratory in its formative stages, one of the authors was witness to a debate between two scientists studying nitrogen concentration in soil. The debate originated in a disagreement about whether to standardize measures of nitrogen concentration. Each scientist used different measuring techniques for assessing nitrogen concentration, and numerical results of these techniques were incommensurable. Thus any standardization would inevitably invalidate at least one if not both of the measuring techniques. Yet, as the conversation continued, the underlying third order issue revealed itself: one of the scientists felt that the ability to share data between researchers which standardization would supply was worth the cost of narrowing the number of acceptable methods, even if her method of choice was not selected as the standard. The other scientist felt that it was more important to ensure the survival of a diversity of methods for measuring nitrogen, because he felt that each method was problematic, and it is only when a student is confronted by the diversity of techniques that they begin to examine their measurements critically, and recognize the limitations of their measure. In this particular conversation, the third order issue was revealed because one participant in the debate was very reflective on his epistemological stance. In other debates, however, we have observed scientists getting caught in second order debates without realizing the

underlying third order issues that were motivating the disagreement, thus we see opportunity for intervention.

Similarly, different groups of users can utilize the same terms with very different associated understandings of what those terms mean. This difference in interpretation can lead to serious misunderstandings which can remain undetected.

[5] has identified a particular case of multiple meanings which leads to systematic misinterpretation of data. Crystallographers regard the creation of complex ribbon diagram computer models of organic molecule structures to be the initiation of being a professional in the field. The process of building your own model drives home the subjectivity inherent in how models of organic molecules are created, provides an understanding of the limits of the models, and reinforces the understanding that the model which is produced is but a snapshot of a constantly vibrating, "breathing" structure. Yet among crystallographers, molecular biologists are notorious for interpreting the ribbon diagrams as being canonical representations of the molecule, because they consider these molecules to be rigid structures. This error can lead to serious misuse and misinterpretation of the ribbon diagrams when used in molecular biology, yet there is little the crystallographers can do to challenge these errors because they are generally required to upload their ribbon diagrams into shared data repositories, and it is impossible to convey the "breathing" of these molecules in the metadata which they are able to provide. Thus, in their research both groups may be talking about same ribbon diagram, but the associated meanings are very different, and likely problematically so. Thus, another role the CSCW designer can play is to be aware of such communication barriers, and alleviate them both via conversations, and via developing systems which have richer means of communication (in this case, richer metadata descriptions).

CSCW designers can also affect awareness of the advantages and limitations of network externalities. Often, researchers are unaware of the extent to which their visions of collaboration in general and data sharing in particular will be constrained by the volume of extra work needed to encapsulate sufficient meaning that shared information can be effectively utilized. Designers must be aware of the necessity of this work, so that they can be an advocate for long-term sustainability of the collaboration, and they must be aware not only of how some of this information can be automatically generated, but also of which advances in automatic processing may be just on the horizon, and which are simply wishful thinking.

## CONCLUSIONS

Despite the justified reluctance of CSCW designers to address third order issues in infrastructure development, we believe there are some actions which they can take to raise awareness of the existence of these issues so that different user communities can negotiate resolutions to these issues.

If these issues are unaddressed, collaborative development efforts can either grind to a halt or facilitate miscommunication and misunderstandings which can lead to flawed research being conducted. Thus, we advocate taking a true sociotechnical approach to design, where the object of design is not just the technology, but the entire sociotechnical ecology, and the intervention may be better conceptualized as guiding or enabling. Design still has a significant role to play, but its scope is limited to [1]'s places of intervention.

## REFERENCES

1. Buchanan, Richard (1992). Wicked Problems in Design Thinking. *Design Issues*, 8(2), 5-21.
2. Daft, R. L., & Lengel, R. H. (1986). Organizational information requirements, media richness and structural design. *Management Science*, 32(5), 554-571.
3. Floyd, Ingbert R.; Jones, M. Cameron; Rathi, Dinesh; Twidale, Michael B. (2007). Web Mash-ups and Patchwork Prototyping: User-driven technological innovation with Web 2.0 and Open Source Software. HICSS 2007: 40th Annual Hawaii International Conference on System Sciences, 2007.
4. Jones, M. Cameron; Floyd, Ingbert R.; Twidale, Michael B. (2007). Patchwork Prototyping with Open-Source Software. In *The Handbook of Research on Open Source Software*, St. Amant, K. and Still, B. (Eds), Idea Group, Inc., PA.
5. Myers, N. "Animating Mechanism: Animation and the Propagation of Affect in the Lively Arts of Protein Modeling." *Science Studies*, Special Issue: The Future of Feminist Technoscience. Vol 19 (2): 6-30.
6. Nardi, B. & O'Day, V. (1999), *Information ecologies : using technology with heart*. Cambridge, Mass.: MIT Press.
7. Orlikowski, W. J. (1993). Learning from Notes: Organizational Issues in Groupware Implementation. *The Information Society*: 9(3), 237-250. Reprinted in Kling, R. (ed.) (1996). *Computerization and Controversy: Value, Conflicts and Social Choices*. Second Edition. San Diego, CA, USA: Academic Press.
8. Rittel, Horst W. J.; Webber, Melvin M. (1973). Dilemmas in a General Theory of Planning. *Policy Sciences*. 4, 155-169.
9. Star, S. L. and Ruhleder, K. (1996). Steps toward an ecology of infrastructure: Design and access for large information spaces. *Information Systems Research*. 7(1), 111-134(8?).
10. Weick, Karl E. (1979) *The Social Psychology of Organizing*. Second Edition. Reading, MA, USA: Addison-Wesley.