Hierarchy in Flux

Designing Context as Multiscale Design

Evan Barba, Christopher Miller, Yasir Majeed, Georgetown University, Washington, DC, USA

Abstract

The role of context has been an important focus for Human-Computer Interaction research since the beginning of the Second Wave of HCI. While different theoretical frameworks within the HCI community have different approaches to analysing context, they do so always with the object of understanding its effects on human-machine interaction, often with the larger goal of generating insights into future designs. The forces that shape context itself are typically ignored in these analyses because they are not considered relevant to the interaction itself, which is the focus of HCI. Yet if these forces were to create different contexts for interaction those changes would be relevant to HCI research. This suggests that HCI might benefit from techniques that analyse and design for the creation of the institutional structures that constrain human-machine interactions. We present the notions of multi-scale analysis and multi-scale design as terms which describe approaches that seek to engage the different disciplinary proficiencies that create the context for interaction. In doing so we make the case for a new kind of design education that strives to create multi-disciplinary designers capable of harnessing the dynamics of systems at different levels of abstraction to achieve outcomes that exceed what we might expect from HCI alone.

Introduction: The Role of Context in Human-Machine Systems

Those who have been trained in the principles and practices of Human-Computer Interaction (HCI) over the past two decades are well versed in techniques for studying the context of human-machine interactions. These have been productively applied in HCI research and the epistemological changes they have engendered within the field have propelled it forward. However, it was not always true that context was given such a prominent place in HCI research, and it may be the case that further progress requires us to look more deeply at how context itself is shaped both as a phenomenological entity and as an object of analysis.

Some of the early comparative work investigating context in regard to sociotechnical, systems can be found in Nardi's (Nardi, 1996) discussion of Activity Theory (AT)(Kaptelinin & Nardi, 2006), Situated Action (SA) (Suchman, 1987), and Distributed Cognition (DCog) (Hutchins, 1996). Nardi compares the differing ways these theories conceptualize context, as well as how they differently approach the equivalence of humans and artefacts, define their units of analysis, and account for persistent institutional structures and processes. She finds that, for Situated Action, context is a moving target. She notes that SA uses the term 'arena' to denote the environment in which interactivity takes place and the term 'setting' to describe the subset of that environment's features which the user finds salient. But she also notes that the more stable institutional structures that constitute the environment are not considered as part of SA's unit of analysis, which is defined as "…the activity of persons acting in a setting,"(Lave, 1991; Lave & Wenger, 2014). SA gains something important by focusing on the details of how a user improvises with artefacts in a given setting, but it loses its ability to critique the setting and arena in their own right. For SA the systemic forces that shape context are not a part of the analysis, only the effects they have on the user and situation are relevant. Another way to say this is that the scale of the unit of analysis is the user and the setting, while the higher scale context, the arena, is not examined. This might be effective for HCI, which cares deeply about the user's actions, but it loses the ability to critique, and therefore changes, the organizational structures that create context.

While Nardi is at times quite critical of SA, she is less so of AT. For her the ability of AT to robustly formulate the role of context is one of its great benefits. In AT the unit of analysis, is an *activity*, which is the context for all the events and artefacts that occur in an AT analysis. An activity consists of a human *subject*, a goal (called the *object*), *actions* the subject takes to accomplish the goal, and smaller *operations* which are rote and routine tasks carried out in service of some action. A simple example activity like writing an email would be deconstructed this way in AT: The human subject has the object (goal) of writing an email to ask for the day off, the email consists of sentences that each support a particular reason for wanting the day off, writing each of these sentences are actions that support the overall goal, while the pushing keys on a keyboard to type a sentence are rote procedures that support the actions. AT shares the dynamism of context with SA in that it is enacted through the procedures and processes of the individual parts of the activity system. She says, "People consciously and deliberately generate contexts," while also noting that, "Context is both internal to people…and, at the same time, external to people, involving artefacts, other people, specific settings," making the point that context is both internal and external in AT. While this statement has some truth to it, especially as a description of the concerns of AT, it also deemphasizes the role of the institutional structures that support activities. Most workers would be fired if they ignored the institutionally defined contexts of their work and opted to deliberately generate their own contexts. I have no doubt that adherents of AT would acknowledge this point, they are concerned with real-world uses of technology after all, however simply acknowledging the constraints imposed by context does very little to influence the forces that shape it. Even though context can shift dramatically in AT as subjects move from one task to another, and these shifts are a part of AT analysis, there are forces outside the activity system that play an important part in shaping context, and these are not engaged in AT.

Contrastingly, DCog does not suffer from the issue of ignoring the physical and procedural structures of a given institutional context; it relies on them to define the purpose of its unit of analysis—the cognitive system. As Nardi explains, 'Distributed cognition is concerned with structure—representations inside and outside the head—and the transformations these structures undergo.' While Nardi does give a good description of the important elements of DCog, she loses the plot somewhat by not discussing how it defines and operationalizes context. The closest mention one can find to 'context' in her description of DCog is that 'The cognitive system is something like what activity theorists would call an activity.' Reading into her description somewhat, one could surmise that DCog treats context as its unit of analysis. If one is focused on understanding the interactions between people and artefacts, as HCI researchers like Nardi are, then the context that matters is the one which is immediately surrounding those users and their artefacts.

Nardi is right then that SA does not adequately account for the context itself and instead focuses on only the aspects of that context relevant to the immediate situation. She is also correct when describing AT as more deeply and explicitly considering the context of the interactions of people and artefacts within the activity system. DCog then takes a similar stance including context in the analysis as the functional goal of the cognitive system and treating people and artefacts with equal care. The problem here is that once you move the unit of analysis to internalize context (make it a part of the analysis) you need to supply an external context for that unit analysis. Cognitive systems, like activity systems, do not occur in a vacuum. SA does not suffer this problem because it makes it clear that it knows interactions take place in a context, it just doesn't care that deeply about it. AT can perhaps get away with ignoring a context external to the activity system because it focuses on the system itself while simply including the immediate context in the analysis. In other words, AT looks with a wider lens increasing the boundaries of the analysis to include context while still focusing on the individual mind. DCog however examines the system itself, ignoring the inner workings of the individuals' minds in a way that AT and SA do not. In the example of Hutchins's work cited by Nardi, the "successful completion of a flight" is stated as the goal of the cognitive system, but this is an impoverished context. It tells us nothing about the training of the crew, the history of the artefacts, or the reasons for flying. All of these are potentially impactful institutional structures that Hutchins treats quite deeply some of his accounts and provide the needed context for understanding why the cognitive system is structured the way it is. In fact one of the major criticisms of DCog is that it has limited use outside of these highly structured process-driven and mechanistic environments. DCog, like AT shifts the boundaries of its analysis to the higher scale context of the sociotechnical system, but unlike AT it also shifts the scale of the unit of analysis along with it.

Nardi's treatment is one of the most thorough comparative discussions of the three most highly regarded frameworks for examining interactions in sociotechnical systems (see Halverson xxxx for another), and the only one which takes a deep dive into context. I mention it here not to pick it apart but to point out that the role of context in HCI is theorized only insofar as it pertains to that discipline's object of inquiry–the interactions between humans and machines. Thus all of these approaches have limitations when it comes to designing context itself. Nonetheless, context itself is a designed artefact it comes about due to many influences and creates 'stable institutional structures,' as Nardi notes, that have real implications for the design of human-computer interaction. Yet there is little theory that can explain how we are to move between these frames of analysis in a coherent or systematic way. This is partly due to the differences in expertise needed. The dynamics that govern interaction between humans and machines are very different than the dynamics that govern the machines themselves, and also very different from the dynamics that govern the institutional structures that create context for HCI. We need ways to bridge these divides.

Studying Context in the Context of Multiscale Design

Context was not always a focus of HCI research. Earlier work in the so-called First Wave of HCI research treated the user and the machine as coupled black boxes whose performance was to be optimized; context never really played into it. This approach grew out of the early introduction of technology into the cockpits of fighter jets (Bannon, 1991), but the consistent introduction of computers into other kinds of work required the development of new approaches.
The second wave of HCI was an initial attempt to broaden the scope of HCI methodology to accommodate the expansion of computing technologies. Bødker (Bødker, 2006) summarizes it this way:

> In the second wave, focus was on groups working with a collection of applications. Theory focused on work settings and interaction within well-established communities of practice. Situated action, distributed cognition and activity theory were important sources of theoretical reflection, and concepts like context came into focus of analysis and design of human-computer interaction.

So it within this second wave that context becomes key and where the frameworks used to analyse context discussed above become part of the HCI cannon. Bodker also describes the impetus for third wave HCI (Bødker, 2015):

> In the third wave, the use contexts and application types broadened, and intermixed, relative to the second wave's focus on work. Technology spread from the workplace to our homes and everyday lives and culture. Research in the third wave challenged the values related to technology in the second wave (e.g., efficiency) and embraced experience and meaning-making. Early on in the third wave, second-wave methods such as participatory design were questioned, due to the perception that they were dealing only with existing (work) practices and not with emergent use.

Context has not disappeared in the third wave, it has however been broadened and the focus of investigation has been shifted to examine "emergent use," which we may take to connote the unpredictable things that people due with machines when neither are no longer chained to their desks. The idea of emergence though is a systems concept, referring to phenomena that exhibit collective behaviours at higher scales than the components of the system that give rise to them.

While HCI researchers will no doubt differ in their acceptance and the perceived usefulness of this "wave theory of HCI" it is clear that some movement has occurred in HCI theory and practice. Movement from coupled human-machine systems and human factors, to multi-faceted work environments, to populist emergent uses of technology. The inclusion of context and different ways of considering it, as well as the change from proscribed uses to emergent and dynamic ones have been a large part of this progression. Throughout this progression the concept of scale has been crucial, and there is a way to reconceive of these shifts in HCI culture in terms of how it has moved the scale of its unit of analysis and how it has coped with the effects of that change in focus. In the following section we present an example of how the different waves of HCI might examine one particular scenario, and how they might be combined together to provide a new approach to design that includes HCI but which goes beyond it to also encompass systems engineering and organizational management as well as some of the more technical areas of computer science and application development.

The Apollo Missions

Ars Technica has a series of articles detailing the early NASA Apollo missions (Hutchinson, 2012) that feature interviews and analysis by former NASA flight controller Sy Liebergot. We highly recommend these articles, they are the most accessible descriptions of how NASA conducted those early missions, and are fascinating from historical, technical, and organizational perspectives. While there wasn't really anything like HCI at the time of the Apollo missions, what is described in those articles resonates with first wave concerns. Each computer "console" was built to optimize the fit between it and its controller, although it would probably be more correct to say that each controller was trained to operate each console as efficiently as possible. Consoles were far more expensive than their users at that point in history, and so it was the user's job to adapt to it, for the most part.

Each console and flight controller was responsible for a different discipline, electrical engineering, communications, etc., and NASA still uses the language of "multi-disciplinary engineering" today to denote the specialized modules that comprise a missions staff and their responsibilities. However each console was also relatively isolated from the disciplines around it. There was little formal communication between the console operators and a firm hierarchy and modularity existed between every console and division. The reason for this becomes obvious when you consider that doing anything in space is inherently dangerous. The entire sociotechnical system is engineered to be predictable, all the mission parameters are known and planned for well before the mission launches. You cannot afford to have unpredictable information enter into the system; it could cause the entire system to fail. So everything that happens is checked and monitored by multiple parties to ensure that it is all within predictable limits. There is a kind of emergence within this system too, but not necessarily the kind that Third Wave HCI deals with. The system is engineered to achieve a kind of "weak emergence" where the collective acts in a coordinated and distributed manner to achieve a result that you could (and should) be able to predict by looking closely at those interactions. The kind of "strong emergence" which is unpredictable collective action, the kind Third Wave HCI looks for, portends disaster for the more fragile Apollo system.
The Apollo system also lends itself rather well to Second Wave analysis using any and all of the frameworks outlined by Nardi. The focus of the system, accomplishing a work-related task (albeit a rather grandiose one) is the same kind of scenario that Second Wave HCI aimed to unpack. The

system could easily be thought of as an activity system: multiple subjects engage in actions at their consoles by repeating multiple operations in effort to achieve the goal of spaceflight. Likewise, by following the different representations of the state of the system across multiple console positions we could produce a rather robust analysis of the Apollo mission as a distributed cognitive system. Situated Action could also be used to aptly describe the actions of each flight controller at their console. They respond to salient information about their environment, and make split-second decisions based on that information.

A Third Wave analysis of the Apollo system might also be done, but the tightly controlled "weak emergence" would likely render it rather uninteresting. NASA engineers were consummate professionals with a serious job to do and were not using their consoles to play games or chat on social media, and so strongly emergent uses would be rare to spot. The one exception might be when the system was thrown into an unpredicted state, as in the events of the Apollo 13 mission depicted in the film of the same name. In cases like this the system would need to reconfigure itself to adapt to the new information and new situation and emergence would be necessary. Interestingly though, as Liebergot explains, unless there was some catastrophic failure, only directly affected parts of the system would be reconfigured. The trajectory in a system like this is to get everything back to a known state as quickly as possible, stopping everything because one console starts flashing red throws everything into disarray and makes matters worse. The tight modularity and control built into this system keeps both problems and solutions as isolated as they can be, preventing strong emergence.

Ashby's "Law of Requisite Variety" (Ross Ashby, 2011) states that every stable system must have a number of states equal to that of its environment. It must be able to produce enough variety to counter the variations in its context. Apollo 13 is a good example of this law in action, as the environment perturbs the system into an unstable state through the introduction of an unpredicted equipment malfunction, the system rushes to reorganize itself into a new configuration, it effectively tries to match variety with variety and stabilize itself. This can be difficult for tightly engineered systems, but is much easier for more flexible ones. The earlier discussion of SA, DCog, and AT suggests that these three approaches have something to say about this as well.

SA is the one most closely aligned with the notion of an adaptive system. The user adapts to information from the environment in an improvisatory way. In more rigidly constrained systems this might be simple pattern recognition—the user identifies a pattern they have seen before and reacts in ways that they know are likely to achieve the outcomes they've seen before. In response to new environmental perturbations they react in much the same way, a tight feedback loop ensures that they are constantly adjusting until they see a stable situation they recognize. There is no reason this approach could not be increased in scale to describe an entire system. This is much of what DCog does, but with a slightly different spin. As discussed above DCog does increase the boundary and scale of the analysis when compared to SA. It shifts focus away from the individual actor and toward the system as a whole, but it also still looks within the system for structured interactions that resemble "cognition" rather than for improvisatory adaptation to contextual changes outside the system. AT does seem to present a kind of middle ground, it increases the boundary of the analysis to include context in a way that SA does not, but it does not shift the scale of the analysis upward the way DCog does, it still focuses on individual cognition when considering interaction with technological artefacts. To bring these three approaches together we need to think a bit differently a bit more holistically about how humans and machines act together in context.

SA, AT and DCog, are quite good at deconstructing and analysing sociotechnical systems, they each have their particular quirks, and there are trade-offs in using each, as we have described. However, when it comes to design, they have proven much less useful, and not for lack of trying. There are many who have attempted to use these frameworks to aid in the design of technological artefacts and the results are mixed at best. Analysis and synthesis may be two sides of the same coin, but they are different processes. Nonetheless there are lessons to be learned from these frameworks when it comes to designing the kind of flexible sociotechnical systems that we describe above. While we stop short of deriving actual design principles, recommendations, and processes, as most of our predecessors in this endeavour have as well, what we do offer here is a new approach to design that treats context itself as a designed artefact rather than as an environment. We refer to this approach as *multiscale design* and it can be summarized very simply as any combination of design processes that examine two (or more) scales simultaneously both in isolation and in terms of their interconnections.

As a complement to this approach, combining SA and DCog would be an example of *multiscale analysis*. DCog can be used to describe the overall system and how it achieves its goals within a given context, while SA can be used to describe the lower-scale activity of individual actors within that system who's goals are smaller and more modest than the system itself. Again the Apollo system serves as a good testbed for this kind of procedure. The entire system can be described as a cognitive system with processes aimed at achieving the mission objectives. Each flight controller and their console however act in a way best described by SA. Combining these two levels of analysis yields complementary results and a more robust analysis of the system. AT seems to do this naturally. It's constructs form a hierarchy of scales, there is the overall top-level activity, the smaller scale actions and the even smaller scale operations. The system itself achieves it's goal, while the actors within it accomplish conscious actions using lower-scale automatic unconscious operations that could, presumably, even be offloaded to a machine one day.

This seems to suggest that multiscale analysis is being done under other names, but what about multiscale design? In some sense the answer is an obvious 'yes,' after all the Apollo missions required the design of an organizational hierarchy, of man-machine console teams, and of the technologies themselves. That is design operating at three scales. This process must have been coordinated, by a team of mission planners at NASA most likely, who were effectively doing multiscale design, and so in that sense this approach is nothing new. Yet, multiscale design as an approach in own right has never been named and its practitioners are never explicitly trained in its principles and practices. The multiscale design of the Apollo system took deep knowledge of at least three disciplines, organizational management, HCI, and computer engineering. This is different than the multi-disciplinary engineering that was described earlier. Simply organizing different disciplines into modules in a way that enables them to work together to achieve some weakly emergent goal is only a part of what multiscale design is about. The other part is integrating disciplines in a way that allows them to incorporate each other's strengths and mitigate their weaknesses to achieve strongly emergent outcomes. At a minimum, multiscale designers would be able to create the organizational structure necessary to support the adaptation of the system, create the interfaces and procedures that give human users the flexibility and information they need, and to engineer the technology that is required for getting the job done, whatever it turns out to be.

As mentioned above this requires an understanding of three very different sets of dynamics: those that govern the organizational structure and therefore create the 'context' as HCI defines it; those that govern the interactions between people and machines; and those that govern the machines themselves. These are three largely unrelated disciplines with very different expertise, languages, and priorities. Mastering all of this is a tall order, no doubt, and we are not suggesting that anyone could or should take on designing a NASA mission on their own using a multiscale approach; although, one could see how multiscale training might benefit such an endeavour. For us, the issue is that designers are not trained to think and analyse at multiple scales, at least not explicitly, and they are not trained to look for the synergistic opportunities that make multiscale systems work. By providing the language and aims for a multiscale approach we hope to take a small step toward recognizing the need for designers trained in multiple disciplines that don't typically overlap horizontally, but that do influence each other vertically across scales. After all, what designer hasn't been frustrated by the seeming absurd constraints placed on them by 'stable institutional structures'? What manager has not been annoyed by designers who 'just can't see the big picture'? These are conditions that persist in any organizational environment. They stifle communication, generate conflict, and prevent innovation. If nothing else multiscale design promises to provide a bridge between worlds that don't speak the same language; not because they exist across any chasm, but because they exist directly on top of each other.

References

Bannon, L. J. (1991). From Human Factors to Human Actors. *Design at Work: Cooperative Design of Computer Systems.*

Bødker, S. (2006). When second wave HCI meets third wave challenges. *4th Nordic Conference on Human-Computer Interaction …*, (October), 14–18. http://doi.org/10.1145/1182475.1182476

Bødker, S. (2015). Third-wave HCI, 10 years later---participation and sharing. *Interactions*, *22*(5), 24–31. http://doi.org/10.1145/2804405

Hutchins, E. (1996). *Cognition in the Wild (Bradford Books)*. The MIT Press. Retrieved from http://www.amazon.com/Cognition-Bradford-Books-Edwin-Hutchins/dp/0262581469

Hutchinson, L. (2012). Going boldly: Behind the scenes at NASA's hallowed Mission Control Center. *Arstechnica*. Retrieved from http://arstechnica.com/science/2012/10/going-boldly-what-it-was-like-to-be-an-apollo-flight-controller/

Kaptelinin, V., & Nardi, B. (2006). Activity theory in a nutshell. *Acting with Technology: Activity Theory and Interactional Design*, (1978), 29–73. http://doi.org/10.1037/0022-3514.90.4.644

Lave, J. (1991). Communities of practice. *Systems Thinker*, 63–82. http://doi.org/10.1017/CBO9780511803932

Lave, J., & Wenger, E. (2014). Jean Lave , Etienne Wenger and communities of practice. *Web*, 1–18. http://doi.org/10.1111/j.1365-3156.2011.02794.x

Nardi, B. A. (1996). Context and consciousness: Activity theory and human-computer interaction. Cambridge, MA: MIT Press.

Ross Ashby, W. (2011). Variety, constraint, and the law of requisite variety. *Emergence: Complexity and Organization*, *13*(1–2), 190–207.

Suchman, L. (1987). *Plans and Situated Actions*. *Cambridge University Press*. Cambridge University Press. Retrieved from http://www.amazon.ca/exec/obidos/redirect?tag=citeulike09-20&path=ASIN/0521858917