

Using Distributed Cognition to Unpack Work Practices: A Position Paper for Cognition and Collaboration: Analyzing Distributed Community Practices for Design

Christine A. Halverson

The Social Computing Group
www.research.ibm.com/SocialComputing/
IBM T.J. Watson Research Center
Yorktown Heights, NY 10598

Intro

I have found distributed cognition to be helpful in a number of different domains including air traffic control (Halverson 1995), auto financing, help desks, and aviation. Hutchins' *distributed cognition theory* (Hutchins 1995), is focused on how cognitive systems are organized and operate, thus re-situating cognition in its socio-cultural context. In this sense it is part of the recent recognition that societies and organizations demonstrate cognitive properties that are different from those of individuals (Salomon 1993; Hutchins 1995; Clark 1997).

Distributed cognition theory serves as a useful conceptual framework for the analysis of human organizational systems, and gives us a way to approach the processes concerned with information use and reuse. Looking at the specifics of how information is represented, and where it is represented, highlights how both storing and retrieving information can be affected. Generalizing across domain examples (Hutchins and Klausen 1992; Halverson 1995; Hutchins 1995; Hutchins 1995; Hollan, Hutchins et al. 2000) there are three dimensions across which cognition tends to be distributed: people, time, and representations. All three can impact information reuse. In this position paper I discuss how I used distributed cognition in a help desk domain to gain insights into the direction of technology development and possible impacts.

Help Desk Setting

A help desk is an important part of any organization that must support products or services. These services can range from personnel matters (Ackerman and Halverson 1998; Ackerman and Halverson 1999; Ackerman and Halverson 2000) to technical matters at a variety of levels. (Halverson 2004) reports on an extremely sophisticated help desk staffed by consultants and for consultants. This paper focuses on a more classic technical help desk organization. While in some cases a help desk may be one person, more commonly it consists of a staff (often called analysts or agents) with varying skill levels. These analysts are supported by their own expertise, as well as a range of technology including phone, call tracking systems, and tools and repositories, in order to answer the customers question—whether by phone or more recently by chat.

Analysts do not have identical expertise however, so help desk staffing tends to be in layers. First response analysts have less expertise than higher levels. Along with increasing expertise there is increasing cost. “Cost is governed by the ‘rule of four,’ which says that the cost of treating the problem on the first contact is multiplied by four if the problem is forwarded to the next layer. Thus, a problem-fixing cost of \$50 in the first layer increases to \$200 in the second layer and to \$800 in the third.” (Delic and Hoellmer 2000)

Currently, especially for first contact, help desks are often located in call centers. These put collections of people in one vast location, using remote, real-time contact to deliver customer service over the telephone. In the case of the Mountain Tech¹ Call Center studied, this service is technical support for a number of outside clients, in addition to Mountain Tech's own employees. This support is delivered by live agents on telephones using a variety of resources. There is a team of analysts dedicated to client. Particularly at the first level, answers are assumed to be somewhat repetitive and thus can be saved in a repository (database) to be called up at will by the call center agent. Thus, there are two main foci of development and operations costs: human staff, and technology.

¹ Not the real name of the company

Technologically the focus tends to be on reducing the time for finding the appropriate answer and delivering it, while answering it in the first tier. Approaches to improvement have varied from how the knowledge is structured, to algorithm development for decreasing search times and increasing hit rate.

However, technology is only part of the cost and benefit. Human staff is a crucial and so far a largely unalterable variable. Currently call centers tend to be in places where the fixed costs (buildings, utilities, salaries and so on etc) are relatively low. However, the cost of such large scale infrastructure is probably not low enough. Outsourcing to locations with ever cheaper costs is an ongoing issue.

The Problem

As costs increase and profits lessen companies are looking to ways to provide help desk support efficiently at lower cost. Most of the effort for improvement has focused on technology to reduce people costs. One of the latest is the hope that anyone could answer technical questions on any account. This would allow a potentially smaller and more efficient workforce. If agents could answer any call they could be rapidly switched from answering calls of one client company to another, making the overall system more adaptive to fluctuating call volumes. And, with rapid globalization of many service functions it is not a far step to thing that any help desk call could be routed anywhere across the globe to be answered by whomever is currently available.

The presumed technical foundation that could enable these notions is to leverage knowledge that is common across accounts into a common knowledge repository. In theory this will allow a broader diversity of individuals to handle these common problems.

The research problem was to look at a help desk with an eye to developing the technology that would support the ability for an agent to successfully answer questions on any account. Knowledge-based systems (KBS) were seen as a way to ensure the solution is on first contact. The value of a KBS is based on its coverage and accuracy, and the assumption is that this translates to speed of problem solving. Another assumption is that because a KBS is constantly being updated and added to, its coverage and accuracy will consistently improve. However this has been largely untested. Delic and Hoellmer (2000) did a rough impact study, merging the data from the call tracking system with the data requested from the KBS. Based on call duration they conclude that the KBS provides a 10% decrease in call length and the cost savings that implies. They compare this with a Gartner group study that claims a 22% impact on call resolution efficiency. The implication is that quick search within a KBS, location of the right response, and its delivery are what reduce the call time and increase the call resolution efficiency. A further implication of this is that expenditure in improving search will pay off in quicker more effective calls.

Approach

Given the cost pressures on a help desk we were only allowed a brief amount of time for direct observation—one week. We took a multi-pronged approach. We had a team of three people with different expertise:

- the author—experienced in ethnographic observations, video analysis and distributed cognition;
- a technical expert in KBS and machine learning;
- and, a systems architect.

With only a week we melded a number of data collection techniques and methods.

1. We interviewed managers across a number of different teams about their experiences;
2. We observed and videotaped analysts in 3 different teams as they answered calls
3. We observed and videotaped analysts being trained in bringing a new customer site help desk online
4. We were briefed about the current technology and had several interview discussions about possibilities and expectations of the new technology.

We also brought a range of prior experience to the table. Two of us had had multiple years of experience with help desks—either studying them (the author) or conceiving of and building technology for them (the KBS expert). Our third member had been dealing with the technical aspects of the current problem and site for about 9 months.

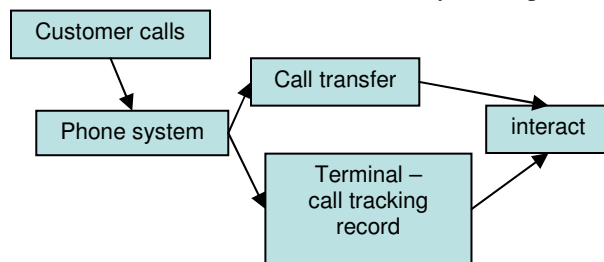
Analytically our approach was two fold. I transcribed and analyzed the videotape and notes from a distributed cognitive perspective, while the two engineers wrote up their insights. Then we had meetings – either all together, or in pairs—to brainstorm, talk through data insights and implications, and write up our conclusions. (This process is still ongoing). The results so far include models of the help desk process, several presentations, and a report in progress.

Why Distributed Cognition

My analysis is driven pragmatically in a way very similar to grounded theory approaches. I approach the data collected from the perspective that the categories that count, as well as the units of analyses that are appropriate, are only going to be determined from understanding the setting itself.

For example, at the start the developers are concerned about the scope of the technology system – one that is expected to be generalized across Mountain Tech (the whole company) and thus across each of the client company’s help desk operations. In contrast, the practitioners—both the agents and the team coordinators—realize that there are specifics for each different account. Furthermore, the performance expectations between coordinators and agents, not to mention the outside client, all have a different impact on the call.

Pulling these kinds of distinctions out of the data however, is only the first step. Each one suggests different bounds for different units of analysis. Within each unit I tend to look at a progressively more detailed scale. Even when I don’t intend to use distributed cognition as an analytical framework, I am continually drawn by how the “outcome” of each unit of analysis is a piece of cognitive work that couldn’t be done by any one piece



or player individually. The more micro scale directs my attention to understanding how the pieces fit together and what their importance is within the current scope of practice. It is that understanding that helps me work with technology developers to help them see possible outcomes.

In the sections that follow I will highlight a few of our observations across the range of granularity moving from macro to micro and then back to macro level.

Macro level: Scoping the unit of analysis

Our multi-strand approach meant that we gathered information across several dimensions.

- Helpdesk hierarchy: agents and team coordinators;
- Roles: developers and architects on the technology side versus technology managers, coordinators and agents on the practice side;
- Broader Organizational hierarchy: overall company goals, technology team development goals, coordinators goals based on milestones, and individual practitioners.

Not all of these areas were obvious going into the field. Each of these areas ended up describing opportunities for analysis at multiple levels. Not all areas had the same quality or amount of data. For example the bulk of our detailed data was at the level of agents and team coordinators within the helpdesk. Because the other team members were part of a technology effort we had some additional information about the technology expectations and process—including artifacts, personal notes and reflection of team members, and their insights. At the broader organizational level we had bits and pieces of data that we connected with experience in other situations. Coming up with this range of dimensions arose from the data we collected as well as our discussions around that data.

Macro to Micro level: Basic call process

From the beginning a call seemed to be an appropriate unit of analysis. But experience—from the past and at Mountain Tech—showed us that what constituted a call varied significantly. I did micro analyses of all calls observed over a three day period. Comparing these, and brainstorming with team members, we came up with an overall process of 6 steps: answering the call; validating the customer; extracting the problem; locating the solution; working through the solution with the customer; and then, writing up the call.

Through successive analyses this simple step model gets broken down into progressively more detailed explanations. One advantage is we can then see that these steps work for simple calls – such as a password reset—as well as more complicated calls that may go through a subset of steps multiple times before ending.

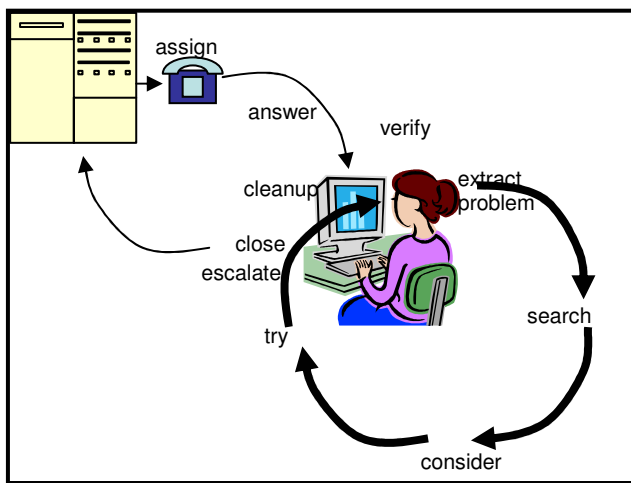
For example, the first step—validating the customer—includes a number of different parts which involve a number of technical interactions. The phone itself is connected into both a voice response system and a call tracking system that automatically brings up caller information on the agent’s screen as they answer the call. They begin to navigate through the customer record, confirming the customer’s identification and confirming

that they have access to the services. They also look at the customer's call history—both to see if they are currently have an open problem, and to see what kinds of problems they have in the past. This happens at the same time that the customer is beginning to tell them what the problem is.

At a very low level I begin to sketch out how the representational states are moving between people and technology. Then I begin to trace exact representational states as they move through the cognitive system I am sketching out. Doing this for multiple calls allows me to build back up to a more macro level. In this case it is a more generalized schematic that applies across multiple calls.

Micro to Macro: A More Complete Model

Detailing each call at a micro level shows us the variations and commonalities. This helps to refine our understanding of what is involved in terms of practices. One outcome of these comparisons is building back up a model of the system. In this case the model is a description that shows aspects within the system that are repeated reliably, or if they aren't repeated we can understand why—usually because steps are combined. The figure below shows such high level representation. The basic artifacts and actors are represented, as well as high level descriptions of their actions.



Seeing a schematic at this level is useful for seeing where a particular technology is being targeted. It is also useful for talking to developers. Details from individual calls can be brought in to talk about specific issues as they arise, helping the developers to see unexpected implications of technology solutions.

From both this more Macro level model and the numerous micro level details we brainstorm about insights. I include a couple below.

Expectation mismatches

Looking from the top down, developers and the managers driving the technology development have assumed that a better knowledge base is the solution to providing quicker answers. Knowing the technology the developers say a piece of knowledge should only appear at one place in the organizing taxonomy. The agent should take the time to locate the right information from the customer in order to execute the proper search and locate the right answer.

There are several problems with this. Agents are hired based on the 'soft' skills needed to communicate with the customer, rather than technical expertise. Due to the breadth of questions at this level it would be difficult to have a complete amount of technical expertise. However, it is not always easy to actually get to the root of the problem. Often it is only through an iterative cycle of defining the problem, searching for a solution and trying it out with a customer that they are able to finally uncover the real problem. Having the answer in only one place in the taxonomy often prolongs this experience. In addition, agents work within a set of measurements and expectations based on the particular customer they work for and the contract under which Mountain Tech provides service to that company. Agents are measured on length of call and customer satisfaction that the call is completed. Taking a long time to find the right piece of information, without this iterative interaction, may find the answer but can leave the customer fuming.

Going down the garden path

The definition of the steps and the constant record keeping make this domain appear to be very structured. In addition, collaboration between agents happened less frequently than we expected. However, collaboration between agent and caller happened all the time. Many of the agent's actions were driven by the customer from very early in the call. A misunderstanding on the part of the caller about what the problem is often leads to the agent pursuing a set of searches that persist in misidentifying the problem through several cycles of searching for and trying potential solutions. But while the problem is caused by their communication it is also repaired by the same means. We saw several cases where the experiences while on the garden path created the insight in either caller or agent, which then got them on the right path. The formal structuring of both the process and the taxonomy was almost unimportant.

Conclusion

I have given only a taste of the issues involved in this analysis. While this benefited from analyzing data from a distributed cognition perspective it also benefited from other approaches. Drawing on team members with prior experience looking at help desks helped us compare the data we collected with insights from other situations. We could not analyze and compare the data directly because we often didn't have the other data. However, those comparisons helped us see larger organizational issues, both within Mountain Tech and its customers. For example, different customer accounts have different levels of concern about the security of their proprietary knowledge. Applying those concerns to our data it makes it evident that not all knowledge can be put together into a common knowledge store.

With the one on one nature of these calls, collaboration is not evident. However, with a distributed cognition analysis it quickly becomes apparent that the real collaboration is between people who may never meet: the architects of the KBS and the content suppliers and maintainers interacting with the agent who is interacting with the caller. In defining the units of analyses and moving from micro to macro scale participants that would otherwise be hidden—people and artifacts—become apparent.

References

- Ackerman, M. S. and C. A. Halverson (1998). Considering an organization's memory. Computer Supported Cooperative Work, Seattle, WA, ACM.
- Ackerman, M. S. and C. A. Halverson (1999). Organizational Memory: Processes, Boundary Objects, and Trajectories. *32nd Hawaiian International Conference on Systems Science*, Maui, HI, IEEE.
- Ackerman, M. S. and C. A. Halverson (2000). "Re-Examining Organizational Memory." *Communications of the ACM* **43**(1): 58-64.
- Clark, A. (1997). Being There: Putting brain, body, and world together again. Cambridge, MA., The MIT Press.
- Delic, K. A. and B. Hoellmer (2000). "Knowledge-Based Support in Help Desk Environments." *IT Pro - IEEE*(Jan/Feb): 44-48.
- Halverson, C. (2004). The Value of Persistence: A Study of the Creation, Ordering and Use of Archives by a Knowledge Worker. HICSS 04, Waikaloa, HI, IEEE.
- Halverson, C. A. (1995). Inside the Cognitive Workplace: New Technology and Air Traffic Control. Cognitive Science Department, San Diego, University of California.
- Hollan, J. D., E. L. Hutchins, et al. (2000). "Distributed cognition: toward a new theoretical foundation for human-computer interaction research." *ACM Transactions on Human-Computer Interaction* **7**(2): 174-196.
- Hutchins, E. (1995). "How a Cockpit Remembers Its Speeds." *Journal of the Cognitive Science Society* **19**.
- Hutchins, E. and T. Klausen (1992). Distributed cognition in an airline cockpit. Communication and Cognition at Work. D. Middleton and Y. Engestrom. Beverly Hills, CA., Sage Books.
- Hutchins, E. L. (1995). Cognition in the wild. Cambridge, MA., MIT Press.
- Salomon, G., Ed. (1993). Distributed cognitions. Learning in doing: Social, cognitive, and computational perspectives. Cambridge, Cambridge University Press.