9

Creating One View of the Customer

The challenge of consolidation is to do explicitly, on purpose, and externally what is usually tacit, haphazard, and internal: develop a sense for a whole customer population from particular instances and events. At this point in Contextual Design, particular instances of customer experience have been captured through interviews and interpretation sessions. Affinity diagrams and consolidated work models show how individual examples of work practice are instances of overarching patterns that define the whole population, and they provide concrete representations of those patterns.

Affinity diagrams and consolidated work models have different forms and reveal different issues, but a similar thought process underlies them all. They are all built by *induction*, reasoning "from the particular to the general, from the known to the unknown" (Fowler 1876). The goal of consolidation is to generate new insights about customers and how they structure their work. You can't develop new insight by applying existing rules and concepts to the data; all you'll ever discover is more detail about the things you already know. The consolidations we build in Contextual Design use induction to bring together many instances from individual interviews, building up structure from detail to reveal new concepts and patterns. These form the understanding of the customer and provide the challenge for design.

Reveal the customers' story by seeing the pattern behind the instance

We don't create consolidations from rational arguments of what must be true. It is easy to make decisions about the work that are based not on what you saw, but on logic. For example, it's only logical to suppose that, faced with a system problem, a system manager would try to figure out what's wrong. In fact, observing system managers at work suggests otherwise. Often system managers start by applying a few techniques that fix most problems (of which rebooting the machine is the most notorious). Only if these fail, do they do any real diagnosis. And it often doesn't matter if they never discover the actual cause of the problem—making it go away is good enough. So designing for logically deduced behavior would not be as effective as designing to support trying a few standard actions quickly. Stepping out of the work to think about it increases the probability of making work more rational than it is. So never depend on theoretical arguments to decide what's true. Decide what's true by induction from the data.

Variation across customers exists within a structure —it isn't random

Because the structure is built up out of the detail, consolidations naturally accommodate variation among customers. Where designers might previously have seen only random differences between customers, bringing these instances together with induction reveals that differences are variations on a theme. If one person prefers key commands and another prefers the mouse, we can see these as alternative strategies for controlling the system appropriate to different cognitive styles. If one person prefers to write an outline before starting a paper and another just talks out her ideas, we can see these as different ways of clarifying thought and structure before starting the writing. New variants can be recognized and positioned within the structure—so someone who wrote lots of different rough paragraphs and then went back to rewrite them could be recognized as achieving the same intent of clarifying his thoughts in a new way. Variations exist within a structure.

We support induction by creating external representations of work practice. Without such representations, people base their design on their unarticulated sense for the common patterns of work derived from individual experiences or customer interviews. When the designer is good, the work practice is simple, and the system is small, this works well enough. The designer can hold all the different aspects of work in her head, can maintain all the implications of a small system, and can keep control of a project with few people on it. But once a problem gets complex and the team gets large, an explicit representation of the work to respond to becomes critical, for several reasons.

First, the sheer complexity of the problem requires a representation. Just as anyone can multiply single-digit numbers in their head but needs physical props to multiply six-digit numbers, as soon as the problem starts to grow designers need to write their understanding down. In fact, nearly all design thinking demands props. A sketch of your thinking provides something to interact with, something to push your ideas against. By representing the work practice of a customer population externally, Contextual Design takes part of the design conversation out of the designer's brain and puts it on the wall as a model. The designers then respond to it as an external entity. It holds the memory of the customer and forces designers to be accountable to the customer data. It becomes not just a prop, but a partner in design. (In fact, one team convinced their management to give them an extra office to act as a team room on the grounds that the customer voice lived and breathed and deserved its own room.)

Work models become a partner in design by holding work complexity

Second, the design is owned not just by one person, but by the cross-functional design team. They have to get the design out of their brains and on the wall just so they can act as a team—so they can share their thinking, take advantage of each other's points of view, and all contribute to the one design. Any one person is stuck in his own point of view; externalize that point of view and everyone on the team can see and modify it. If the extended team is too large for one design meeting, the models hold the thinking so different groups can interact with it. Contextual Design provides both external representations and team processes to use them to encourage the team working together and building on each other's ideas.

Finally, building up a sense of the market instance by instance works against a real shift in perspective. It works against the creative leap that might produce a next-generation product or radical business process improvement. When faced with one new piece of customer data, people assimilate it; they modify their entering conceptions just enough to account for the new piece of data. They say, "Look—we can handle that with just a small fix over here." A creative leap comes not from such small adjustments, but from seeing the large cumulative effect of lots of little pieces, which forces designers to abandon existing assumptions and come to the data from a fresh perspective. In Contextual Design, we encourage this by making consolidation a separate step. Instead of

Consolidating all models at once challenges entering assumptions looking at each piece of data individually and assimilating it, we combine all the data together so it has the maximum impact. (And along the way we use tricks, such as forbidding old terminology, to prevent our entering assumptions from showing through, which we will talk about in the next chapter.) We do it fast—a day for each model and a day for the affinity. Doing it slowly would encourage assimilation; doing it fast swamps our old paradigm with new data. Doing it slowly would encourage point solutions to each problem in turn; doing it fast encourages broad, systemic responses to the whole work practice of the whole customer population. The consolidated models and affinity become the statement of the customer that forces us out of our rut. They drive the designer to make a creative leap.

Consolidation is the inductive process of bringing all the individual data together and building one affinity diagram and one set of models that represent the whole customer population. It's a process of inquiry—looking at details from specific customers and asking how each detail informs the team's focus. Then the parts can be brought together based on meaning to reveal structure across customers. Though it's applied differently for each kind of model, this same thinking process is used in all consolidation. We'll start with the affinity to see how to do the thinking and then look at the other models to see how it is applied to each type of consolidation. We will unpack the thought process in detail to reveal how this kind of inquiry works.

Inductive reasoning is the key to seeing pattern

The affinity diagram

The affinity diagram organizes the individual notes captured during interpretation sessions into a hierarchy revealing common issues and themes (Figure 9.1). The affinity shows the scope of the customer problem: it reveals in one place all the issues, worries, and key elements of work practice relevant to the team's focus. It also defines the key quality requirements on the system: reliability, performance, hardware support, and so forth. The hierarchical structure groups similar issues so that all the data relevant to a theme is shown together, creating stories about the customer relevant to the design problem. By reading the affinity, a designer not only

Create a bottom-up hierarchy of key points to see issues

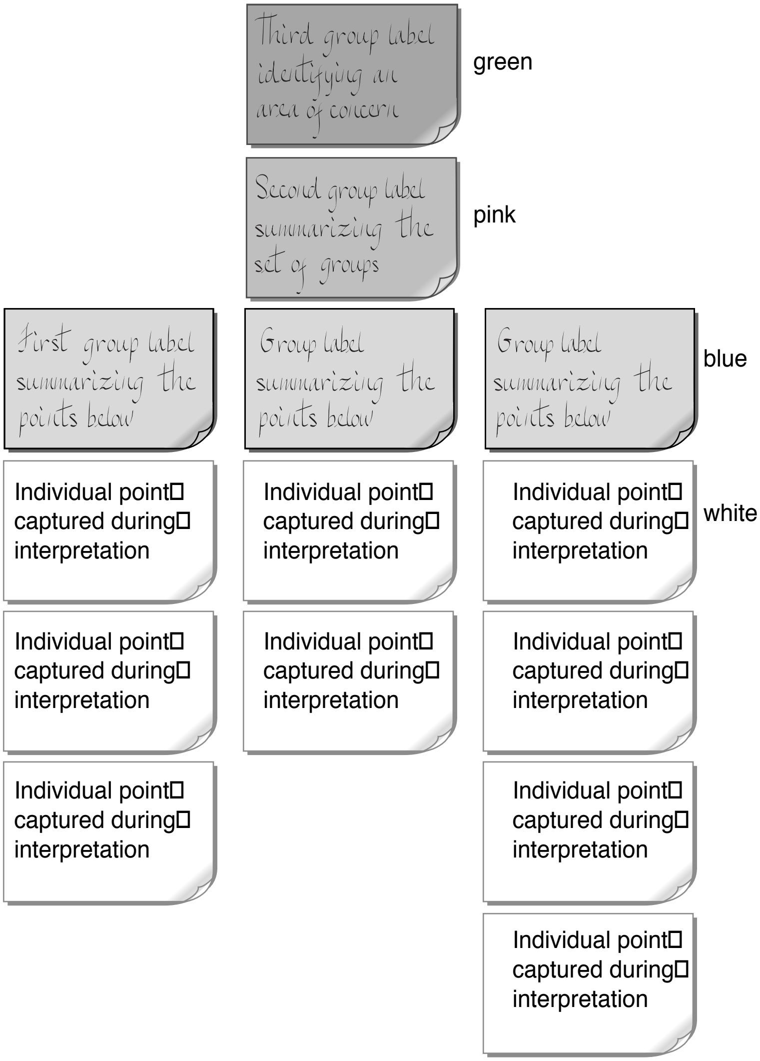


Figure 9.1 Structure of an affinity diagram.

learns the key issues, but can see the exact data that contributed to identifying each issue in the work.

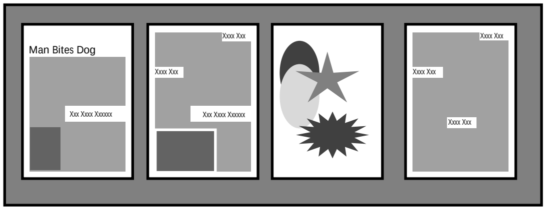
The affinity process was introduced as one of the "seven quality processes" from Japan (Brassard 1989; also known as the K-J method in Kawakita [1982]). In the quality community, affinities on the order of 200 notes are usual. We have optimized the process to handle much larger affinities, typically about 1500 notes. We build the affinity after a good cross section of users has been interviewed—usually 15–20 customers at four to six work sites, with 50–100 notes from each customer. We always prefer to finish the affinity in a single day because it's simply too exhausting to allow it to drag on. This is possible if we have one person per 100 notes to build it. If our team is smaller than that, we invite others who are interested in or affected by the design to participate.

The affinity is built bottom up, by raising common structure and common themes out of the individual notes captured during the interpretation sessions. We do not start from a predefined structure or set of categories such as "UI issues" or "Quality." Starting from such a set of categories reduces building an affinity to a sorting task; each note goes in its own bucket, and at the end you know no more than you did before. Instead, we allow the individual notes to suggest categories they might belong to. We intentionally resist using categories that might be familiar to the team, suggested by their experience instead of by the customer data. We even ban words the team is too familiar with; for example, a configuration management group was not allowed to use the word "version." Banning the term forces the team to say how the concept is relevant to the problem and helps them to come at the problem with a fresh perspective.

Ban words to force rethinking old concepts

The affinity is the first consolidation step, and it teaches the thinking for all the rest. Building an affinity is inductive reasoning at its purest. The basic process is to put up one note, then for everyone to look for other notes that seem to go with it. There's no need to justify *why* they go together—just as you can feel an affinity for a friend without justifying why. But we do push for a certain kind of affinity: two notes have an affinity if they are saying similar things about the work as it relates to the design focus of the team—they are expressing a similar intent, problem, or issue in the user's work. So deciding if notes go together is the result of an inquiry into the meaning of the words on the note to understand the work issue they represent. When it's not clear how to interpret the words, the team can appeal to the interviewer to check whether an interpretation is valid. The team is responsible for ensuring that the data will support the claim they wish to make.

Here are some examples of using the data captured on a note to infer meaning for the work. Each example gives some of the context




```
12.□□□□□□□□□                                 U6□
Searches for desired text by turning pages in full page□
view—big headline was the distinguishing feature
```

Figure 9.2 Capturing a search strategy.

(which the team would be aware of) and shows how to look at the data from a particular focus and see implications for work practice and design. If these insights occurred to team members during the interpretation session, they would be captured in separate notes; otherwise the affinity process gives the opportunity to consider the data again. These notes are all taken from an interview with a user of a page layout tool.

Inquire into the design significance of each note

The note in Figure 9.2 describes how page designers identify their pages. Even though full page view makes the page too small to see any detail, it's still possible to identify the desired page by its overall pattern and by large elements that show up even at reduced size. The work implication is that page designers, concentrating on the layout and look of pages, find it more natural to search by look rather than by text on the page.

The note in Figure 9.3 describes a UI issue, but inquiry provides deeper insights about how these users conceptualize their work. The product provides a box to contain text, but the characters in that box don't stay strictly within its bounds—risers stick up past the top, and descenders can stick out the bottom. The "snap to" guides snap the box boundary to the guide, which isn't what the page designer wants. Page designers want to align the tops of the risers, the tops of the small letters, the center of the small letters, the bottoms of the small letters, or the bottoms of the descenders. Those are the distinctions that matter to the page designer—the box is a construct that has no


```
124.                                              U8
"Snap to guide" snaps to the top of the text box, not to the
tops of the letters that stick up past the top of the box
```

Figure 9.3 Capturing a UI issue.

meaning in their work. Even product ideas such as fixing the top of the box so it coincides with the tops of the risers misses the point. A more general solution would build knowledge of the alignment points for text into the product.

The meaning a designer reads in a note and the way he groups them together is driven by the project focus. A single note will often suggest different aspects of customer work. The designer wants the meaning that will give the affinity the most insight, allow it to tell the best story about the customer for the focus. For example, consider the notes shown in Figure 9.4, collected from people in grocery stores and legal offices during an inquiry into search strategies.

Group Post-its to reveal new insights into the work

Note 110 could be paired with either 214 or 360. The thinking behind pairing 110 and 214 would be that both notes are about legal cases and how they are found, so they should go together. The thinking behind 110 and 360 would be that the two notes are about using a similar search strategy to find things: the more recent the thing, the better. Given the focus on how people find things, pairing 110 and 214 doesn't lead to new insight—it's no surprise that legal cases are searched in law offices. The only aspect of work that the group reveals is details about the job of the paralegal staff, which is better represented on work models. Pairing 110 and 360 raises up a common search strategy. It's the more interesting pairing because it shows how this strategy pertains across work domains (searching for cases and searching for groceries). It might be combined with other data to make the strategy explicit, as in Figure 9.5.

```
110.                                                  U2
The more recent a legal case, the more persuasive it is
```

```
214.                                                  U2
Legal case precedents are searched by paralegal staff
```

```
360.                                                  U4
At milk case, buys 1 gallon or 2 quarts depending on
expiration date
```

Figure 9.4 Grouping notes to reveal design significance.

Recent stuff is best

```
110.                                                  U2
The more recent a legal case, the more persuasive it is
```

```
360.                                                  U4
At milk case, buys 1 gallon or 2 quarts depending on
expiration date
```

```
720.                                                  U8
The most recent house listings are the most desirable; good
houses sell quickly
```

Figure 9.5 Revealing a common theme.

When notes are collected together, they are given a name to represent the group. A good group name states the work issue that holds all the individual notes together. It is a succinct phrase that summarizes the content of the group. “Different ways of searching” would not summarize the content in the above example; it would just say what you could learn by reading the content. “Recent stuff is best” states the issue; then the individual notes give examples of this

Labels are the customer’s voice speaking from the wall general statement. A good group name is written as though the user was talking to the designer; direct, immediate language has more impact than third-person language. When the notes use the user's language, the whole wall speaks the user's issues to the design team—they become a central communication device.

First-level groupings like the above are themselves collected into a group of groups, which are grouped into higher-order groups. The result is a hierarchical structure that breaks the data about the user into manageable chunks. We use green Post-its at the highest level, which describe a whole area of concern within the work practice. Under this, pink labels describe the specific issues that define that area of concern. Blue labels describe each aspect of the issue. And the individual notes under the blue labels describe the instances illustrating the blue label. When well written, the labels tell a story about the user, structuring the problem, identifying specific issues, and organizing everything known about that issue. The labels represent the new information in an affinity. We limit each first-level group to four notes to force the team to look deeply and make more distinctions than they would otherwise be inclined to. It pushes more of the knowledge up into the group labels.

Labels become the meaning we design from

For example, Figure 9.6 is a section of an affinity describing delegation. It's part of a larger story about why people communicate in doing their job—one reason is to delegate (individual notes have been skipped for brevity).

This section of the affinity brings together data from many customers and many work situations to tell the story of delegating work. When sharing the data or reviewing the wall yourself, you might read it like a story: "People delegate work either because they don't have time to do the work themselves or because they choose not to deal with it. They pick someone else to do it either by who has time, who reports to them, or is otherwise appropriate given the organization. Different ways of delegating have different styles: people can delegate doing the work but remain responsible for it, they can delegate a task by assigning it during a meeting, or they may pass it on informally." Each pink label names an issue that is described by the blue labels underneath it so that each section of the affinity tells a coherent story about part of the work,

The affinity tells a story of the customer that matters for design

- We delegate our work (green)
  - Why we delegate (pink)
    - I don't want to deal with it (blue)
    - I have too much work (blue)
  - How I choose who to delegate to (pink)
    - Whoever is available does it (blue)
    - The person with the job does it (blue)
  - How I go about delegating (pink)
    - I gave it away, but I'm still responsible (blue)
    - I gave it away at a meeting (blue)
    - I pass it on informally (blue)

Figure 9.6 A section of an affinity diagram.

and the whole wall brings together all issues and observations to tell a single story about the customer population.

Steps

- Print the notes captured during interpretation sessions in a 3 × 5-inch grid and cut apart so each is on its own label-sized slip of paper.
- Put notes up on the wall one at a time. After each note goes up, add notes that go with it.
- When there are too many groups to keep track of, start labeling them with blue Post-its.
- As groups accumulate individual notes, break them down so there are no more than four notes in a group.
- Add pink- and green-level notes to collect groups.

Others who use the affinity process forbid talking while building the affinity; we encourage it. We view this process as an opportunity to gain team consensus, which is best supported by discussion. All work is done in pairs so people can discuss their insights with each other and get someone else to check their thinking. Writing the labels reveals what you're thinking; if anyone disagrees they can object. All the data instances are there to

The affinity captures the insight of all the brains on the team support one interpretation or another. Each person's different perspective is shared, and a common perspective built through discussion. Discussion also helps move people from thinking in buckets (all notes with "legal case" on them get tossed in one group) to thinking in work practice—people police each other's notes. When people can't agree on where a note should go, they talk about what underlying work issues they see. When people don't understand a note, they go back to the list of notes from that interpretation session or to the interviewer to ask what happened in the interview. We've seen no problems resulting from letting people talk, and doing the inquiry together requires talk. It lets all the brains work together.

Building the affinity in a day creates a team event that binds the team together and encourages creating new perspectives. Building smaller affinities more quickly, or building up one affinity over time, would allow team members to incorporate each piece of data before having to deal with the next; as we discussed above, this leads to assimilation instead of promoting a paradigm shift. Instead, in a single day the team has to face a whole new way of looking at things. As a team process, the affinity forces the team to learn each other's points of view and discuss their differences. But like the interpretation session, it puts strict bounds on disagreement; team members talk about the different meaning they draw from one note at a time. When they are done they have a single structure representing all their customer data, which organizes their knowledge and insight and gives them a basis for design.

Building a 1500-note affinity is exhausting. It's an entire day of reading and conceptualizing hundreds of little bits of data and matching them with other little bits of data. It's like a combination of "Concentration" and translating Shakespeare into Latin: the words on a note have to be interpreted to translate them into the underlying work practice issue; then the note has to be matched with the note you saw five minutes ago and you know is on the wall somewhere. Everyone's working at once, moving back and forth along the wall, discussing notes with each other, yelling general questions to the group at large ("Who interviewed U4?"). Some team members will be more comfortable with the apparent disorganization than others. But the result is exciting for everyone: a single, sweeping reorganization of the customer data arranged like a

The affinity organizes hundreds of Post-its into a story in a single day story. You can read a good affinity from beginning to end to see every issue in the work and everything about it the team has learned so far, all tied to real instances. There's no better way to see the broad scope of the problem quickly.

Consolidating flow models

Consolidating the flow models reveals the communication patterns that underlie the way the customers do business. It's a basic marketing tool—it shows who the customers are, what they do, and how they interact with each other. It shows what part of the work practice of a customer population you currently address and how you might expand existing systems to support more of the job, more of the whole business process, or more people in the workplace. The flow model shows the scope of the work domain a project intends to address and shows how the work the project is focused on fits into the customers' larger work practice. Flow model consolidation reveals the common structure that underlies all the different ways organizations define jobs. It does this by using roles as the essential element of work practice on which to base consolidation.

The flow model reveals the common roles in different job definitions

Roles are collections of responsibilities that accomplish a coherent part of the work (Wirfs-Brock [1993] uses roles in a similar way). Roles have a primary intent, the reason why the role was created in the first place. When people organize themselves to get a job done, they naturally break the job up into roles: "You write the paper," they say. "I'll review it." The roles people create are not random or idiosyncratic; they are driven by the needs of the work (Fisher 1980; Wirfs-Brock [1993] applies these ideas to software). Writers are too familiar with their own work to review it well, so splitting the reviewer and writer roles makes sense. Reviewing for technical accuracy and reviewing for grammar and spelling could go together, but they use very different skill sets, so technical review and editorial review are often separated into different roles. But checking for appropriate references and checking that the content is technically valid both depend on knowledge of the field, so it doesn't make sense to break these responsibilities into different roles.

Because they are driven by the needs of the work, roles tend to be consistent across organizations. The mapping from roles to individuals—the selection of particular roles an individual takes on—is much more idiosyncratic. A person will take on roles they find congenial or have skills for; organizations will create jobs that combine different sets of roles. The roles don't change; the mapping to people does. We do care to track whether a particular set of roles commonly is assigned together and who tends to take them on in a segment of the market—that a particular role tends to be taken by women or that banks tend to merge these two roles. This will affect how a system helps people switch roles and may influence how we package or sell the system.

Roles are very consistent across any work domain

The primary job of consolidating the flow model is to identify the roles played by individuals and combine similar roles across individuals. The roles that a person plays are suggested by their responsibilities and tasks and by their interaction with other people. But we aren't just grouping similar responsibilities. The responsibilities of a role hang together in the work practice, and responsibilities may be repeated in different roles. It should be possible to conceive of hiring a person to play a role—if that doesn't make sense, the role is probably not real.

Individuals play multiple roles

The first step in consolidating flow models is to generate a complete list of responsibilities for each individual.

Analytical Scientist

—run experimental tests on substances
—interpret test results
—document and report results of tests
—schedule test requests from multiple people and departments
—clean glassware
—research appropriate test equipment for group
—report results and trade-offs to group
—order basic supplies
—**help other scientists run tests**

It is common, when flow models are created by real teams, to discover overlooked responsibilities by examining the interaction between

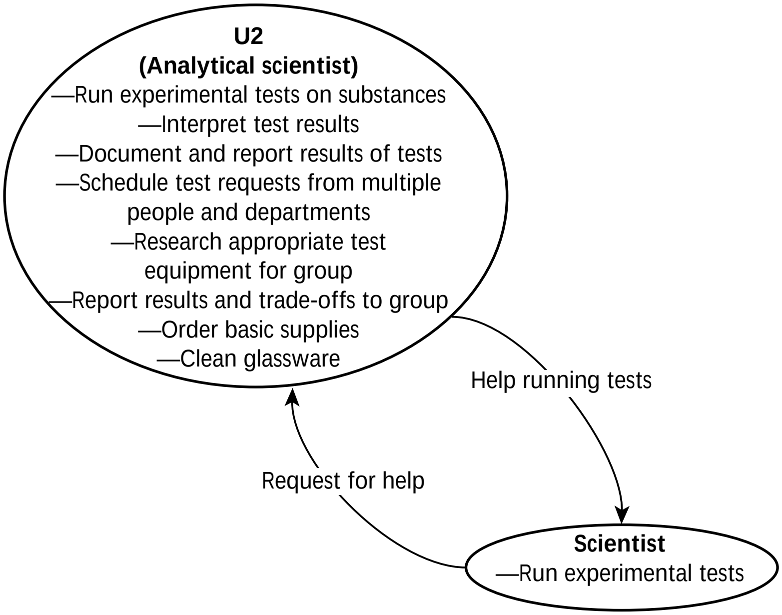


Figure 9.7 Identifying responsibilities.

people. In Figure 9.7, the flow to "Scientist" indicates an additional responsibility: to give other scientists help on running tests. Informal responsibilities such as this are as important to how work really gets done as the formal responsibilities assigned by the organization. So we add it to the list before considering roles.

A role is a collection of responsibilities, organized to accomplish one primary intent. For a role to be coherent, it must include all the responsibilities that are critical to that intent. These responsibilities cannot be separated into different roles. So the first responsibility we identify is "Experimenter"—the person who runs an experiment. It's a good starting point because it's the primary job function of this individual. Then we look at the rest of the responsibilities and ask if they go with this role (much as we asked whether two notes should go with each other in the affinity): "Interpret test results" is critical to the Experimenter. An experiment is run by the strict rules of experimental science; scientists need to know that the results are reported by the same strict rules or their hard work is wasted and their reputation jeopardized. It belongs with the Experimenter role.

A role collects responsibilities, which accomplish an intent

"Document and report results of tests" is more of a judgment call. It's possible to conceive of a head scientist who oversaw the experiment but left it to others to report results. But we aren't building logical structures here; we are deriving structure from the data. If every scientist interviewed in this work domain reports his own results, then it's not real *for this work domain* to separate documenting and reporting into a distinct role. It's just another responsibility of the Experimenter. We keep "help other scientists run tests" with the role for the same reason—all scientists advise and assist each other. This is a claim about what it means to be a scientist in today's laboratories. It can be supported with the data, by checking with the interviewers, and by checking back with the customers if necessary.

Experimenter

—run experimental tests on substances
—interpret test results
—document and report results of tests
—help other scientists run tests

We then go on to the next responsibility, "schedule test requests from multiple people and departments," and ask the same questions: What's the primary contribution of the responsibility to the work? What other responsibilities go with it? It doesn't seem that scheduling test results has to be part of the Experimenter. It would be reasonable, in a high-throughput lab, to hire someone to schedule experiments for maximum efficiency. So we define a new role. The result of looking at the rest of the responsibilities is a tentative list of roles and their responsibilities for this individual:

Lab Scheduler

—schedule test requests from multiple people and departments

Lab Maintainer

—clean glassware
—order basic supplies

Equipment Researcher

—research appropriate test equipment for group

—report results and trade-offs to group

In each case, these roles can reasonably be separated out into a different job function from the Experimenter. The group's manager might act as Scheduler, tracking requests and handing them out so that equipment and people are busy but not overbooked. Lab assistants might play the Maintainer role, keeping the lab running smoothly. And an outside agent might be assigned to research equipment and provide options.

The assignment of roles to individuals or job functions varies from one organization to the next and from one individual to the next. Roles that are separate in one case may be combined in another. In our example, the team identified a different set of roles for another analytical scientist:

Tester

—run a test on samples

—convert raw data into meaning

—report results of tests to requester

—describe what's needed of new equipment

Method Developer

—develop a new test procedure through experimentation

—document the new test procedure in standard form

—assist other scientists in using the new procedure

Multiple people play the same role

Consolidation in a flow model happens by recognizing when different people are playing the same roles. Here, Method Developer is a new role, but Tester is clearly the Experimenter role with a different name. (If the same people analyze both individuals and recognize the similarity they would use the same name. But when the team breaks into subteams, different people may do the analysis. Even when it's the same people they don't always recognize the role until they have a chance to step back and compare.) Experimenter and Tester each have a responsibility the other doesn't have—"help other scientists run tests" and "describe what's needed of new equipment," respectively. But both responsibilities fit right into the role, so we can combine the two roles into a single consolidated role definition. Here again we use affinity-style thinking to look at the meaning of two different items and combine them when they go together.

We choose the name "Experimenter" as a better description of the primary intent of the role. Just as we use plain language on the affinity, we try to keep role names plain and everyday. This makes the real work of the role more immediate. However, we try to capture the essential work of the role in the name. Even though some experiments are tests, "Experimenter" better captures the flavor of the work and mind-set of the people.

Experimenter

—run experimental tests on substances
—interpret test results
—document and report results of tests
—help other scientists run tests
—describe what's needed of new equipment

Design the system for the role variation that actually occurs

It is normal to build up responsibilities of the consolidated role this way. We expect that not every responsibility will be discovered in every interview, and in fact, our second scientist may never have had the occasion to help another. But the consolidation shows what's natural to the role. It tells us to expect that the first scientist may be called to give opinions on equipment and the second may be asked for help. It tells us that any new design for the system or organization must allow for these events. For example, if the organization were to measure scientists strictly on the number of experiments they perform, they would lose the synergy of interaction between scientists. This is how we build up an understanding of a whole job out of multiple interviews. This is how consolidation reduces requirements skew—it identifies needs that the customers haven't stumbled over yet. And this is how to deliver systems based on actual customer data without sacrificing flexibility; the flexibility built into the system accounts for the actual variation in work practice, not hypothetical situations that never actually occur.

In consolidation, we keep track of how roles map to individuals. It will matter for design to know that one person played a dozen roles, or that a single role was played both by low-level technicians and Ph.D. scientists. (Why is a Ph.D. doing work a technician could do? Couldn't we sell them a system or change the process to make better use of their time?) So we assign a color to each job function, department, or demographic group we wish to track, and color the role to show where it came from. If scientists are yellow, the Tester role will be yellow. If lab technicians (who also run tests) are pink, the Tester role will be yellow and pink. With this coding, designers can scan the model and see immediately how it maps to people's job functions.

The final step of consolidating a flow model is to consolidate the artifacts and communications between people. Each artifact and each communication represent an interaction not just between people, but between roles. When the second scientist tells the first what she needs from a new spectrometer, she is the Experimenter talking to the Equipment Researcher. When the first gives help to another, he is an Experimenter helping another Experimenter. The consolidated flow model carries these individual flows over, showing them between roles rather than between individuals. The artifacts or communications themselves may be consolidated and given a single abstract name: "help on device use," "assistance reading a method," and "suggestions on getting around device limitations" might all be represented on the consolidated flow as "help with devices and procedures." The flow can be simplified by showing only the flows relevant to the project focus.

Link the roles with real communications

Steps

- Select six to nine individual flow models that are complex, interesting, and cover the key variants of the work domain.
- List responsibilities and identify roles of each person, group, and place on the individual flows. Name the roles.
- Collect similar roles from all models and lay them out on a consolidated model.
- Rewrite responsibilities and name each role.
- Collect artifacts and communications from the actual models. Draw them between roles on the consolidated model.
- Transfer any breakdowns from the actual models onto the consolidated model.
- Compare the remaining individual models against the consolidated flow. Add any roles, responsibilities, or important flows that are not represented by the consolidated model.

Flow model consolidation leads a team from knowledge of individual users to understanding the structure of work across a customer population. It's a fairly efficient method for doing this; after consolidating about nine good and diverse models in a work domain, additional models will offer few surprises (teams that have gathered much more data—from up to 40 customers—quickly discovered that they were duplicating what they already knew). Between 15 and 20 customers from a typical work domain is enough to see the pattern of the flow of work between people as they do their jobs.

Between 15 and 20 customers is enough to see the pattern of role and communication

The flow model is nearly always a useful model to build and consolidate. Nearly any job requires working with others, receiving information and handing results to others, or cooperating with others in some way. Only when the project focus is narrowed to the interaction with the tool only—usability of an interface or interaction with a device—can the flow be omitted. Even then, there's a potential for overlooking important interactions. It's better to build it anyway.

The consolidated flow model is the designer's tool for seeing the roles that underlie idiosyncratic organizational structures and interpersonal communication patterns. It shows the central roles and key responsibilities of the work practice being studied and how they coordinate and pass artifacts around to make work happen. The consolidated flow model is the best map to how work is done, showing the breadth of work and the details of how people interact. The flow model shows what roles people play, so that if you have systems already in place, you can see what roles you support. It can show how the systems taken together support the whole of the work (or don't). It shows what other roles the same people are likely to play, which are natural roles to support next—the potential customers would already be sold on your system or your company. It shows who else a role has to interact with to get a job done; supporting these other roles or the interaction between them is also a natural growth path. The consolidated flow model is your map to your customer population. It shows you where you are and where you are going.

The consolidated flow maps the players in the customer population

Consolidating sequence models

A consolidated sequence model reveals the structure of a task, showing the strategies common across a customer population. Individual sequence models describe one real instance of work, showing how a person accomplished a task in that case. Consolidated sequence models bring together many instances of many individuals accomplishing the same task, revealing what is important to doing the work: what needs to be done, the order and strategy for doing it, and all the different motivations driving specific actions. A consolidated sequence model shows a designer the detailed structure of the work they need to support or replace. It shows all the different intents that must continue to be accomplished in the presence of the new system or rendered unnecessary. It shows the overall structure of the task, which may be mirrored in the system to make it more useful and intuitive. And it shows where the task is needlessly complex and could be simplified by a new system.

Consolidated sequences show task structure and work strategies

Tasks to be supported by a new system may be performed by a customer population that spans organizations and industries. Even within a single company, different departments will implement different procedures, and people with different cognitive styles will approach the work differently. Nonetheless, over and over again, we find common structure within any domain of work. People only develop a few different strategies for accomplishing similar tasks. The key is to learn to see the common structure in the detailed actions people take: the common activities, intent, and strategies for accomplishing a task.

People use only a few strategies to do a task

The sequences in Figure 9.8 show how two system managers diagnosed problems. Skimming U5's sequence, we see that he is notified by an automated process that something is wrong; he pokes around looking for problems; then he calls for help. These immediately become potential activities: notify, diagnose, get help. Shifting our attention to U4, she is notified by a person, pokes around on the hardware until she recognizes that the problem is something AT&T has to fix, and she puts in a call to them. Again we see the

Identify the activities across all sequences

U5	U4
Fix All-In-1	**Fix router problem**
• Trigger: Watcher sends mail that the All-In-1 (A1) mail system isn't working	• Trigger: Person walks into office to report problem—can't access files on another machine
• Log onto failing system to search for problem	• Go into lab to look at equipment
• Discover the A1 mail process has crashed (ACCVIO)	• Flick switches to do loop-back tests, isolating wire, MUX, router
• Look at the time of the crash: only this morning	• Determine problem—bad router
• Try to restart the process by hand	• Call AT&T people in second organization
• Process won't restart	• Do something else while waiting for AT&T to show up
• Look at the process log; can't tell why it won't start	• AT&T comes to look at problem
• Call expert backup group for help	• Look in book to tell which wire is which; show which nodes are on which wires and which wire goes to which router; paper for easy access
• Ask them to log into the system and look at problem	• Tell AT&T people which router is at fault and which wire it's on
• Keep looking for problem in parallel	• AT&T people fix problem
• Search for problem	• Log problem and fix
• Discover that process can't create a needed directory	• Done
• Try to create needed directory by hand	
• [Look to see if directory created]	
• Can't create directory; disk is too fragmented	
• Call expert backup to explain problem; type and talk on speaker phone at the same time	
• Discuss problem; agree on the exact procedure to follow	
• Implement fix	
• Write mail to users describing changes that affect them	
• Done	

Figure 9.8 Two ways to diagnose a problem.

U5	U4
Notify	
• Trigger: Watcher sends mail that the All-In-1 (A1) mail system isn't working	• Trigger: Person walks into office to report problem; can't access files on another machine
Diagnose	
• Log onto failing system to search for problem	• Go into lab to look at equipment
• Discover the A1 mail process has crashed (ACCVIO)	• Flick switches to do loop-back tests, isolating wire, MUX, router
• Look at the time of the crash: only this morning	• Determine problem—bad router
• Try to restart the process by hand	
• Process won't restart	
• Look at the process log; can't tell why it won't start	
Get help	
• Call expert backup group for help	• Call AT&T people in second organization

Figure 9.9 Identifying activities.

basic structure of activities: notify, diagnose, get help. (We'll save the rest of the sequences for later.) For now, we'll match up the steps in the sequence that initiate a new activity in Figure 9.9.

The first step of a sequence is the trigger that initiates it. Triggers may consolidate, as when several individual sequences start with someone reporting a problem in person. More often, as in this case, the trigger steps identify alternatives. Either way, we define an *abstract step* to represent both triggers. An abstract step states the work done in each of the instances independently of the specifics of that instance. In this case, we just list the two different triggers we have discovered (Figure 9.10). In other cases, triggers might introduce different strategies—a system manager who is notified of a problem by a help desk may go right into hypothesis testing, but a problem report that comes from an automated process may always start by researching the problem. When this happens we keep the triggers separate, to show how they initiate different branches of the sequence. It also happens that triggers are not at the same point

Identify and name abstract steps across all sequences

Abstract Step	U5	U4
• Trigger: Find out about problem —Automated procedure —Someone reports problem	• Trigger: Watcher sends mail that the All-In-1 (A1) mail system isn't working	• Trigger: Person walks into office to report problem; can't access files on another machine

Figure 9.10 Alternative triggers.

in the sequence at all. Email from a user may in fact not be the first report of a problem, but the response to a query as part of the research activity. Such a trigger needs to be moved down in the sequence.

The next steps all contribute to diagnosing the problem. Our task is to match up steps accomplishing the same thing in each instance and define abstract steps for them. We don't yet know exactly how the steps match up; we only know that they all have to be sorted out before getting to the steps in which U4 and U5 call for help.

Match up steps doing the same thing

The first step in each case positions the user logically (in the case of logging in) or physically (in the case of going to the computer lab) to start diagnosing the problem. Logging in or going to the lab are details unique to the instance; the work being done is for the users to go where they can deal with the problem: our next abstract step (Figure 9.11).

Both U4 and U5 next try different things on the system until the problem is identified. "Discover the A1 mail process has crashed" and "Determine problem—bad router" both seem to mark the point at which the user identifies the problem. U4's sequence has a step in which U4 flicks switches and runs tests to determine the problem. The team who wrote U5's sequence didn't write down such a step, but it's implied by "Discover the A1 mail process has crashed"—U5 must have done something (looked at running processes or looked at process logs) to discover the process is down. But as U5's sequence indicates, all that's happened so far is to discover why the symptom is happening; the underlying problem (a full disk in U5's case) may not have been determined yet. So the consolidation looks like Figure 9.12.

At this point, we're consolidating the different kinds of problems that the system managers discover to see the common structure of diagnosis across all problems. If we wanted to design for each kind of

Abstract Step	**U5**	**U4**
• Go to the place where the problem can be solved (physically or logically)	• Log onto failing system to search for problem	• Go into lab to look at equipment

Figure 9.11 Going to deal with a problem.

Abstract Step	**U5**	**U4**
• Execute commands and tests on suspect system to identify anomalous behavior	• (Do something to discover the A1 process isn't running)	• Flick switches to do loop-back tests, isolating wire, MUX, router
• Determine cause of symptoms	• Discover the A1 mail process has crashed (ACCVIO)	• Determine problem—bad router

Figure 9.12 Identifying a problem.

problem uniquely, we wouldn't do this; we would keep the kinds of problems separate in the consolidated sequence. But for this problem, seeing diagnosis is a fine enough level of detail.

Next, the two users diverge in their strategies. U5 goes on to try to fix the problem. But U4 decides that she can't fix this problem and that she has to call on AT&T to do the fix. Neither U5's decision to go forward nor U4's decision that AT&T has authority to fix the problem are written explicitly, but both are implied by the user's actions. So the abstract steps branch to account for the two cases. Consolidating them, we get Figure 9.13.

Watch for different strategies to do the same thing

This process repeats until the whole sequence is consolidated. We identify the sections of the sequences that match, match up individual steps, and name abstract steps for them. Either after a whole activity or at the end of the sequence, we step back and ask the intent of each step. Why is the user doing this at this point? What are the obvious and the nonobvious reasons for doing the step? There may be more than one intent to any step, and there may be high-level intents that cover a whole set of steps. It's easy to identify and support the main intent of the sequence. It's harder to see all the additional, secondary intents that are also important to the customer.

Identify intents of the steps

Activity	Abstract Step	U5	U4
Diagnose problem	• Estimate impact of problem	• Look at the time of the crash: only this morning	
	• Decide whether I can fix the problem	• (Decide to fix)	• (Decide AT&T has to fix)
	• If I decide I can fix it:		
	• Attempt fix	• Try to restart the process by hand	
	• See if fix worked	• Process won't restart	
	• Try to figure out why it didn't work	• Look at the process log; can't tell why it won't start	
Get help	• Decide I can't fix it, call on help	• Call expert backup group for help	• Call AT&T people in second organization

Figure 9.13 Diagnosing a problem.

We decide what they are and write them down. We talk to the interviewer if we aren't sure of an interpretation or check back with the user. The result, for the sequences we've been doing, looks like Figure 9.14.

Of course, a team would consolidate three or four actual sequences at once, not just two. The first cut at abstract steps would be correspondingly more robust. Once the initial set of sequences has been consolidated, the rest of the sequences are compared with the consolidated sequence and used to extend it. Incorporating more sequences will add additional steps, show new strategies, and provide more alternatives for steps that are already represented.

Steps

- Select three or four sequences addressing the same task. Look for detailed sequences that, at a quick scan, seem like they will consolidate reasonably well.
- Scan the sequences to identify activities. Mark the point where the first activity ends in each sequence.
- Match the triggers across sequences. Be aware that the instances may start at different points in the story.
- Match steps of the sequence within the first activity. Write in omitted steps if necessary to make matching steps easier.
- Write abstract steps as you go. Write any breakdowns on the abstract steps as you come to them.
- At a convenient stopping point—the end of the activity or the end of the sequence—go back and write intents for each step.

Activity	Intent	Abstract Step
Find out about problem	• Learn about problems quickly • Discover problems before users do • Provide quick response	• Trigger: Find out about problem —Automated procedure —Someone reports problem
Go to problem location	• Make it possible to do diagnosis and take action	• Go to the place where the problem can be solved
Diagnose problem	• Find cause of problem	• Execute commands and tests on suspect system to identify anomalous behavior • Determine cause of symptoms
	• Decide who's been affected • Decide if any additional action should be taken to notify people of status	• Estimate impact of problem
	• Make sure I don't do things I'm not supposed to	• Decide whether I can fix the problem
Fix problem	• Fix the problem at once	• If I decide I can fix it: • Attempt fix • See if fix worked • Try to figure out why it didn't work
Call on help	• Get the people involved who have the authority or the knowledge to fix the problem • Ensure problem gets fixed, even if not my job	• Decide I can't fix it; call on help

Figure 9.14 A consolidated sequence model.

Consolidated sequence models show the common structure of a task across a customer population. Developing a consolidated sequence of a task shows strategies used to achieve it, the structure of the task in activities, and the intents achieved in doing the task. These define a backbone into which new variations can be incorporated and accounted for. In our example above, it's not hard to see how a new trigger or new step in diagnosing a problem could be accounted for within the structure we developed. Armed with this knowledge, designers can structure their systems to reflect the structure of the task. Where the structure is inherent to the task, it can be built into the system; where it is driven by constraints of the environment, the system can remove steps and streamline the work.

Only consolidate tasks that the system will support, that you will redesign, or that you need to understand in detail. Use the flow model to identify the important tasks—the ones that help the user accomplish their central responsibilities. If the task will not be supported by the system, there's no need for a consolidated sequence model for that task. It's sufficient to scan the individual models for intents or breakdowns that might need to be addressed or that might inform another model. If a task is to be obviated, the consolidated sequence may still be useful because it identifies the intents that the current work practice addresses. Getting rid of the task will cause problems unless all these intents are supported in other ways or rendered irrelevant.

Make sure your system accounts for all intents before automating a task

Consolidating artifact models

Consolidated artifact models show how people organize and structure their work from day to day. Individual models show the structure and usage of the things people create and use while doing their jobs. Consolidating artifact models shows common organizing themes and concepts that people use to pattern their work. They complement sequence models by describing the things manipulated while doing the task described by a sequence. They provide clues to the appropriate structure for a system in the concepts they represent. They reveal work intents that must be supported and that might otherwise be overlooked. And an inquiry into the details of their structure shows how to support specific tasks.

Consolidated artifacts make conceptual distinctions concrete

Just as people only use a few strategies to plan their work, and define consistent roles to break it up among people, they use a consistent set of conceptual distinctions to organize how they think about work. These conceptual distinctions become concrete in the structure that people impose on artifacts they create and use—either by building the artifact in a particular way or by making annotations on an artifact given to them. Because the tasks that people do have similar structure, the intent and usage of artifacts are also similar. An inquiry into the individual artifacts that support similar types of work reveals this common structure.

The first step when consolidating artifact models is to group artifacts of a similar type—all artifacts that have the same intent or usage in the work. Deciding what is similar enough to consolidate together is modulated by project focus. A project to develop a personal organizer tool might want to study different kinds of calendars: personal organizers, shared wall calendars, online calendar tools. Which of these should be consolidated together? Consolidating all types would highlight common aspects of scheduling and organization, but would tend to bury the unique usage and intent of the different tools. For example, the primary characteristic of a wall calendar is that it is shared and can coordinate multiple people; a personal organizer is private but easy to carry anywhere. Consolidating the different kinds of calendars separately would spotlight each kind, but would tend to hide common issues across all types. Since our project is to create a new organizer product, we decide to try consolidating all the tools together so we can identify and transcend common issues. If we were creating generic PC software, we might have chosen to consolidate online calendars separately to better understand the strengths and limitations of the competition.

Let project focus determine which artifact types to consolidate

Once similar artifacts are collected, we identify the common parts of the artifacts (Figure 9.15). These parts and their relationships are the first and primary distinctions created by the artifact. These initial distinctions are driven by physical and cultural limitations as well as by the nature of the work. So a personal calendar has a cover to protect it from spills and scuffs. The need for a cover is driven by the environment, not by the nature of scheduling. The cover creates pockets that are useful places to store things, but they are not inherent to scheduling either. The to-do lists and kids' pictures one finds in these pockets suggest how, when a personal organizer becomes part of daily life, it can play a larger role in keeping things organized than just scheduling. On the other hand, the rubber band and tape both identify the current day and seal off the past—this suggests a recurring intent that *is* inherent to scheduling. Both these

Identify common parts of the artifacts

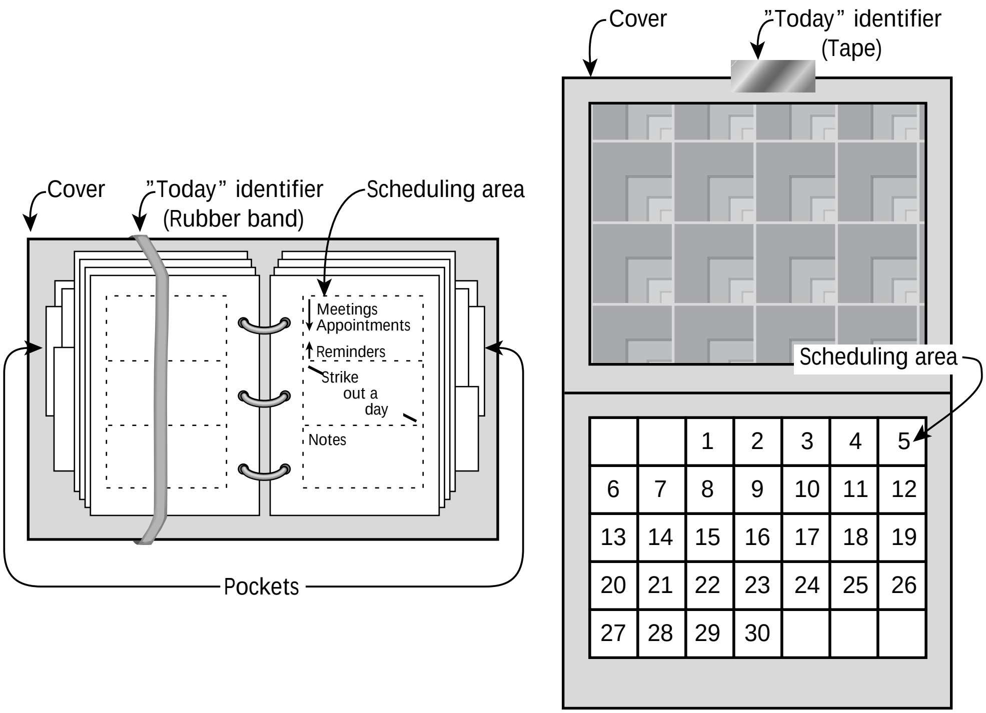
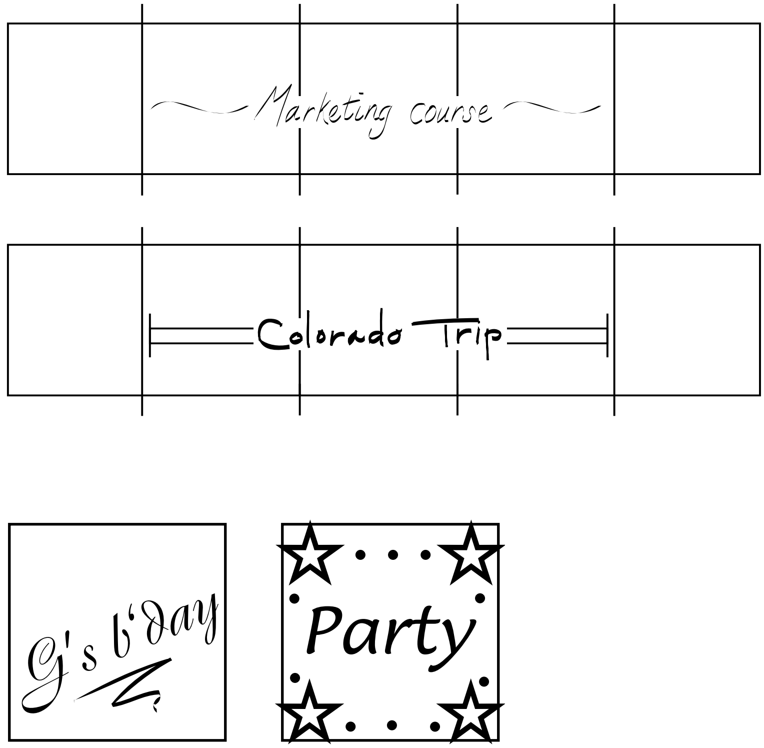


Figure 9.15 Two kinds of calendars.

mechanisms suggest that you schedule into the future and use the past only for reference.

Identify the usage and intent of each part

Having identified common parts, we can look within similar parts to identify structure, intent, and usage. While the primary parts of a calendar are pretty much determined by the manufacturer, the user has more scope for structuring the contents of a part in the way that makes sense to them. So a common part of a calendar is the scheduling area—the week or month view that everyone uses. Within that area we look for the different ways people organize time. So a multiday meeting is usually represented with a symbol that crosses days—people clearly think about one event spanning several days, not about a series of days, each of which is individually booked. Our tool had best provide for events that span days. So the inquiry into a part starts by observing one characteristic of one model, inquiring into its meaning for the work or the concept it embodies, and identifying that concept in the other models.



The contents of a part identify concepts and also presentations of those concepts. Looking at how events are written, we see that some are highlighted so they stand out from the rest. Clearly "important event" is a distinct concept. We also see a variety of ways that the event is highlighted. Depending on our focus, we may care to capture these different presentations. If we are developing an online calendar and if most people use double underlines to highlight important events, it makes sense to use double underlining in our calendar tool.

Look at how the parts are presented to grab your eye

The artifact will keep us honest if we let it. The artifact suggests that some events should be marked as important. It is natural for engineers, trained to worry about future extensibility and to hate special cases, to argue from this to something like a numerical priority scheme. Events could be given a priority from 1 to 10, views could be defined to show only events above a certain priority, functions could be defined to search for the next

Keep online artifacts simple by letting real data guide design priority 1 event, and so on. But we do not have the data to support any of this. Saying some events are important is a very much simpler concept. Not only would these extensions make the implementation more complicated, they would also make the tool harder to understand and deal with. The result is a loss for the user, not a gain at all. Or to take another example, people sometimes tell us they write in different pens for different reasons. But the artifact tells us that in reality people write in the pen that is handy at the time. Being true to the message that the artifact gives us will help keep us from overcomplicating the system.

When an artifact like a calendar is predefined for later use, the structure people use may not match the structure they are given. They may go beyond the given structure, as when they separate reminders from meetings on a day. Or they may simply ignore the given structure, as when they draw the line for a multiday meeting right across the lines separating the days. Whenever the users depart from the given structure of the artifact, it reveals concepts and strategies that are real in the work. They represent opportunities for you to support the work better.

Having identified the parts and their usage and looked at their structure, we are ready to draw the consolidated model (Figure 9.16). In this case, we decided to look at calendars of different types knowing they might not consolidate well. In the event, we've identified many common intents and structures, yet because personal calendars are so different from wall calendars, the usage and mechanisms differ. It often works well to put the common or typical case in the center of a consolidated artifact, with variants around the sides. So we choose to put personal calendars down one side and wall calendars down the other, highlighting common intents and showing how each kind of calendar achieves that intent in its own way. The actual schedule part, where we saw little difference in intent or usage, we put in the center. Finally we step back and scan the whole model, looking for additional intents revealed by putting all the information together. By putting everything about this kind of artifact together, the diagram helps designers consider all aspects of the artifact coherently: common intents and the different ways they are achieved, the structures people create to help them, and the concepts they use to organize their work.

Make the consolidated artifact a good communication tool

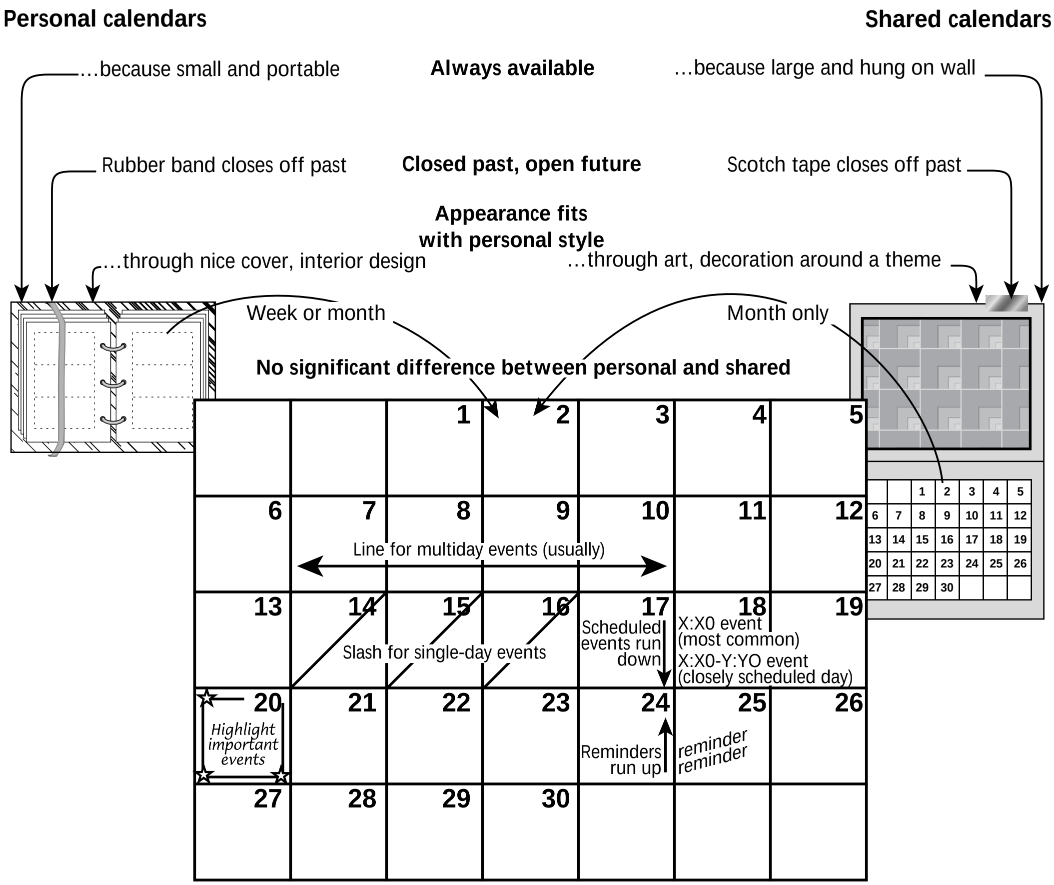


Figure 9.16 A consolidated artifact.

Steps

- Group the artifact models by the role they play in the work.
- Identify the common parts of each artifact. Identify the intent and usage of each part.
- Identify common structure and usage within each part. Identify breakdowns.
- Build a typical artifact, showing all the common parts, usage, and intent, and showing how they are presented where relevant. Show breakdowns.

Consolidated artifact models open a window into the mind of the users, showing how they think about the work they do. They are the most direct way to see how your users think. In addition, they help identify hidden intents that might otherwise go undetected and be unsupported in the system you build. They record the footprints left by multiple sequences, often more than you could ever observe in person. One team examined scores of tracking tickets, collecting from each one the different intents and events that it recorded. In this way they quickly learned about different issues in the work represented by many hours of actual experience.

Consolidated artifacts show the footprints left by tasks

The level of detail to follow in consolidating an artifact depends on your project focus. If you expect the artifact to be rendered obsolete by the new system, do a quick consolidation emphasizing usage and intent. Look for secondary intents that imply potential problems should the artifact be removed. If you expect to support the work that the artifact supports, do a full consolidation, looking at concepts and structure as well. This will inform the organization of your system. And if you expect to put the artifact or its equivalent online, or if your system will create instances of the artifact (e.g., if you print calendars), capture details of presentation as well.

Consolidating physical models

The physical model shows the structure of the physical environment as it affects the work. Individual physical models show the workplace and site for each user interviewed. Individual models show how the place is structured, how it is organized to support work, and how people and things move through the space in the course of getting work done. The consolidated physical models show the common physical structure across the customer population and the key variants that a system will have to deal with. It keeps the design team aware of the limitations and constraints imposed by the physical environment.

Physical models reveal how space and layout affect work

Just as with the other aspects of work practice, physical structure repeats over and over. At first glance, office buildings present many different shapes, materials, and architectural styles. Yet inside the door, there is invariably a lobby area, with a receptionist or security

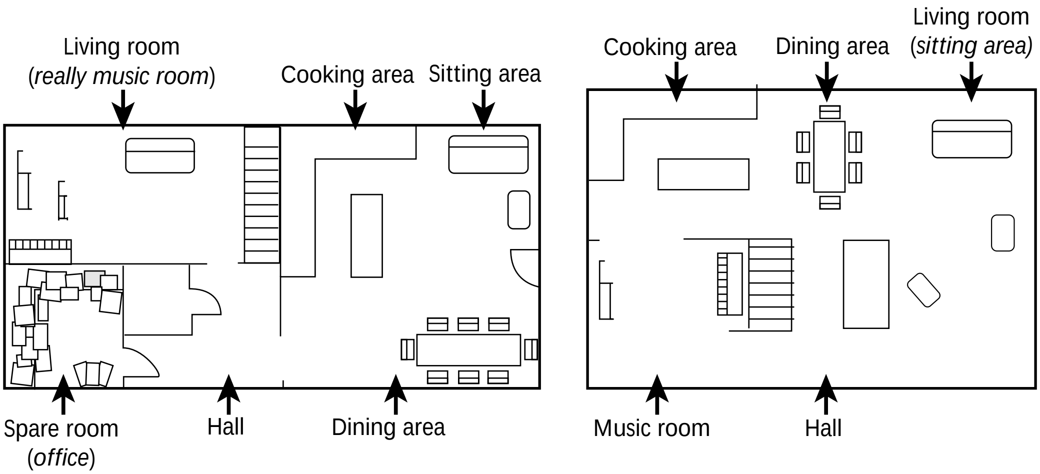


Figure 9.17 Determining the usage of space.

guard behind a desk who helps locate people. Beyond them are people's offices, labs, and shared work areas. Looking beyond a single building, as soon as a company grows, recurring issues crop up around travel between sites, communication between sites, support for meetings attended by people at several sites, and so forth. Consolidation identifies and highlights these common structures and issues.

Consolidation of the physical model begins by separating the models into types of spaces. Usually one set of models represents a whole site or multiple sites. It focuses on whole buildings and relationships between them. Then there's another set that represents individual work spaces. Individual work spaces may be separate rooms, cubicles in a large open room with partitions, or separate desks in a larger room. And sometimes there are specialized spaces that are useful to consolidate—labs, loading docks, meeting rooms, and so forth. Individual models belonging to each of these groups are collected together (Figure 9.17). Always depend on the usage of a space to determine where to sort it, not its formal name—an unassigned office with a round table where staff meetings are held is a meeting room, not an office. A salesman's car may be his workplace.

Identify unique usages of individual space

Within each set of models, we catalog the common large structures and organization. Buildings, rooms, walls, where people sit in relationship to each other and the hardware they use—these are all distinctions that can be identified on site models if relevant to the project focus. Within an office, the location of desks, chairs, the in-box, and the telephone relative to each other and the occupant all reveal the organization of the space to support the work. Identify types of hardware, software, and network connections. At this point, the relative position of spaces, objects, and people is what matters. Whether an object is on the left or right is irrelevant; whether the user can reach it without getting up is what matters. When deciding how to interpret placement always consider the actual usage of objects, not their formal role. An in-box with gum wrappers and empty soda cans in it is a trash can.

Look at how objects cluster and their proximity to people

Once the large structures have been identified and cataloged, the model is open to another layer of inquiry (Figure 9.18). Sites are large and hard for individual users to change much, so they suggest constraints a system must live with and problems it might overcome. Identify these and write them on the model. But workplaces are much more malleable and reveal how people think about their work. The way people lay things out represents their attempt to build a physical environment that mirrors the way they do their jobs. When people do similar work, in a similar culture, to accomplish similar jobs, they re-create the same structures to support it. When telephones, calendars, and address books are repeatedly collected in one corner of the desk, it suggests a place for communications and coordination as a common theme. It suggests that a tool supporting coordination had better include finding people, talking to people, and scheduling work with people, since the physical model revealed that these are all part of the same task. Write these insights directly on the model as well.

Identify the constraints that the environment imposes on work

Movement through a space is also driven by the needs of the work, and we identify movement on the physical models when it is relevant to the project focus (Figure 9.19). Movement of people through space and movement of documents around an office are both useful to represent. The movement of people through space shows what the system must allow for and suggests opportunities to reduce the need to walk around. Movement of things in the course of

Show movement patterns and breakdowns

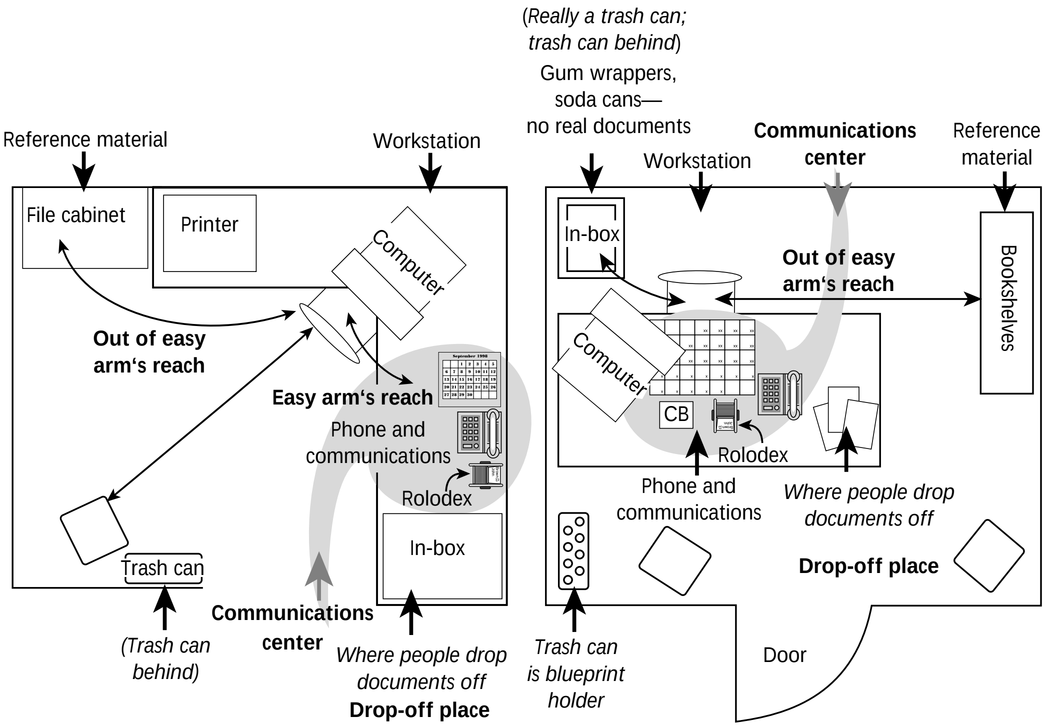


Figure 9.18 Inquiring into usage and structure of space.

doing work makes the sequence of work physical, highlighting transition points in the sequence when an artifact moves from one place to the next. Draw the movement on the models.

When all the spaces and artifacts are identified and examined, you are ready to create a consolidated model (Figure 9.20). Draw a single model, showing one instance of each common space. Where possible, use a single picture to show the structure of that space and things within it. For a system design focus, ignore aspects of the environment that do not matter to the work. Absolute distance from the worker doesn't matter; whether things are ready to hand does. Whether things are to the left or right doesn't matter. Potted plants don't matter. Where artifacts and tools really are in different places, we show them in all the places they might be—so we show a printer in the office and down the hall. The

Draw the model to reveal the issues the team should talk about

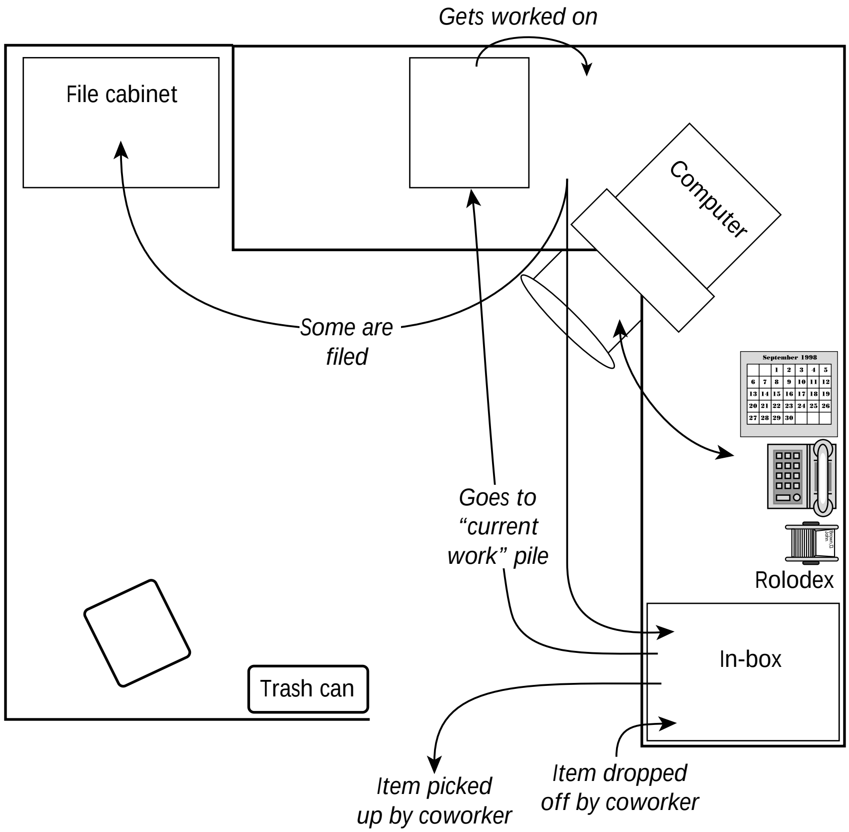


Figure 9.19 Movement through a space.

consolidated physical model shows the common structure and all the variations in that structure across users.

Steps

- Group the physical models by type of place.
- Then walk each model in turn, identifying the different places in the model. Label each place with name and intent.
- For each type of place, identify common structure. Show where the artifacts and tools appear in the place.
- Look at movement on each of the individual models.
- Build a consolidated model showing all the parts and their structure. Carry over intents, usage, and breakdowns from the individual models. Write any insights on the model.

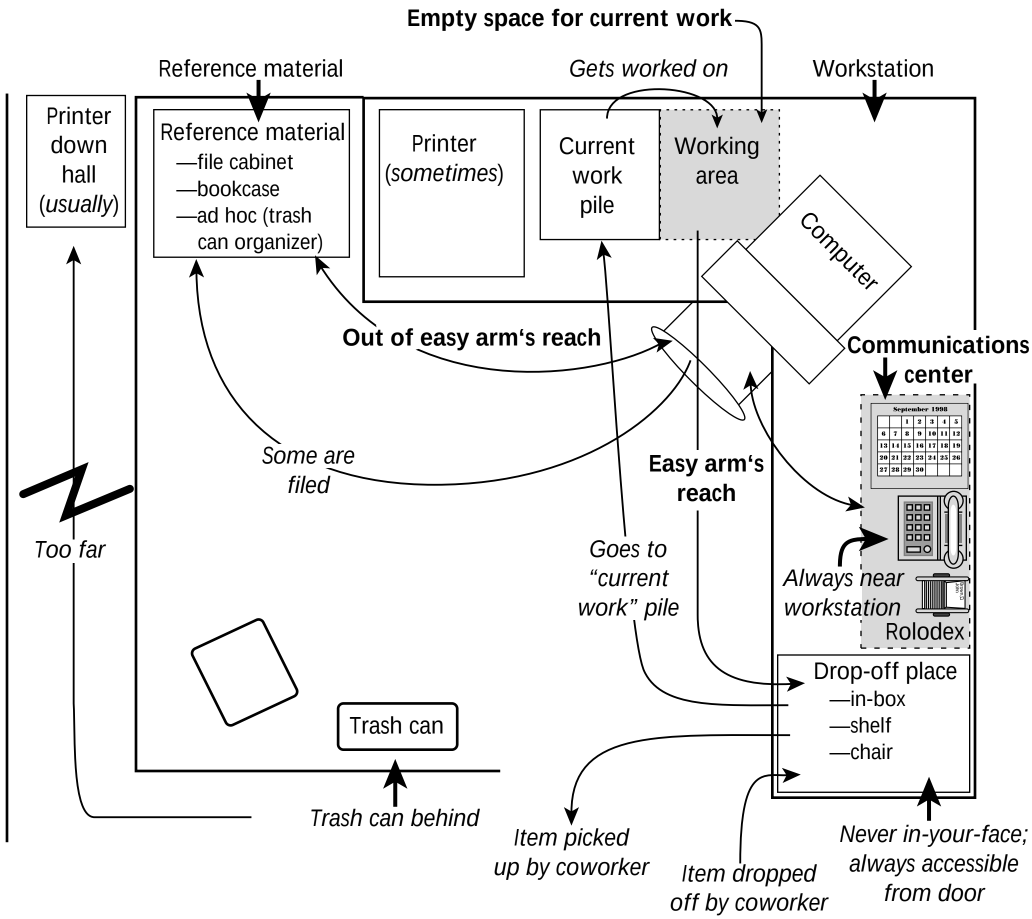


Figure 9.20 A consolidated physical model.

The consolidated physical model is a single model that shows the common issues imposed by the physical environment. It shows the hardware and software used by people in its context of use, the kind of access and movement allowed by the physical environment, and the constraints that affect people across the customer population. If a system does not live within these constraints or provide ways to overcome them, it will not be successful. Businesses studying their own work practice can make good use of the consolidated physical model not only to work around constraints of the current physical plant, but also to assist in designing new buildings and building layouts.

For these projects, system design can include redesign of the walls around people.

The consolidated physical models also show the common strategies in how people structure their environment to support work. This structuring provides clues to how people think about and organize their work. A system that incorporates this organization has a better chance of being acceptable to users and supporting the work well. And the consolidated physical model shows how people and things move through the workplace, indicating the stages of work process that a system may support or eliminate.

The physical model shows how the physical environment supports and constrains work

A physical model is particularly important whenever the work to be supported involves multiple places or movement between places. This is a broad set of problems: even writing is printed on a printer (usually in another room), using materials that had to be collected (usually from another place), for review by one or more other people (who usually sit somewhere else). So even if the primary job is stationary, the whole job taken together may interact with the physical environment in interesting ways. Anytime the job includes handing work off between groups, or coordinating between multiple people, the physical model will be interesting for seeing how the groups transcend or manage physical separation. It will force the design team to be real about the impact a design direction will have. When the job is stationary and doesn't interact with others in other places, how things are clustered and used in the workplace reveals thought patterns and distinctions relevant to the system. Building physical models of each space important to the work reveals this structure and gives important clues to how people think.

Consolidating cultural models

The consolidated cultural model shows the common aspects of culture that pertain across the customer population. It is an index of issues that matter to the people doing the work—what they care about, how they think about themselves and the jobs they do, and what constraints and policy they operate under. The consolidated cultural model can be crucial to choosing the direction a design should take. Do system managers like running around to do their job? Then don't try to tie them to their desks. Either make them a portable system, or make their application quick to get in and out of. Are salespeople closely monitored? Then either make it easier for them to report their actions so they spend less time on it, or redesign their organization so they have more independence. Are customers closely regulated by the government? Then make producing the required documentation simple. These are the kinds of issues addressed by the cultural model. It indicates a direction for the design, and it shows within that direction what constraints have to be accounted for.

The cultural model reveals common values, friction, and policy

Every organization has its own culture—its own ways of doing things and its own attitudes about the world and the work it does. Yet these differences exist within severely restricted limits. Any environmental testing lab will be strongly influenced by the Environmental Protection Agency in the United States. Any computer hardware maker is affected by the competitive and fast-paced nature of the business. Any service industry has to worry about reducing turnaround time on their service because turnaround time is money in such a business. The nature of the business itself creates many of the pressures on an organization.

Culture is not unique within populations doing common work practice

Within the organization, the same kind of repetitive patterns emerge. Any organization that combines watchdog and service responsibilities creates a web of influences and attitudes around them. Purchasing, for example, both helps you get what you need and makes sure you follow approved procedures. Internal PC support both keeps your machine running and tries to make you run standard configurations and standard tools. Whether the service or watchdog aspects of the organization predominate, a pattern of interpersonal friction, influence, and pushback appears.

Even between people and work groups, we find repeating patterns of influence. Networks in companies are typically global these days, which means it is the working day for some part of the network all the time. Often 24-hour maintenance is provided by handing off responsibility rather than working three shifts. This shows up

Even patterns of friction repeat across businesses as an interdependency on the cultural model. Asking a secretary to handle ongoing coordination of all aspects of an office is a common strategy for getting work done, but it creates a relationship of nagging and helping out in one direction, and requests and dependency in the other.

The first step of consolidating cultural models is to walk through each individual model, cataloging and grouping influencers (bubbles). We group influencers when they have the same kind of cultural influence, guided by our focus. So, for most purposes, regulatory agencies can be grouped together—but in the United States, a pharmaceutical company is so intertwined with the Food and Drug Administration that we might keep them separate from other regulatory agencies. If we are supporting system management, we might group all clients together—but if we notice that there's a special relationship to client management, we might keep them separate. If we are modeling an internal client, we generally keep the departments separate and use their real names so we can see the real interaction between them. We keep an eye on the influences—if we'd be prone to group an influencer with others, but notice that the actual influences are very different, we may choose to keep it separate so we can see the difference. After identifying and grouping the influencers across all models, we lay them out on the consolidated model, adjusting them to show relationships and overall direction of influence cleanly (Figure 9.21).

First, find all the influencers

Next we consolidate influences. We walk through the instances again, collecting all the influences between each pair of influencers. When we've collected them all, we do a quick sort to get rid of duplicates and near-duplicates. The remaining influences are written on the consolidated model (Figure 9.22). As we go, we settle on wording that reveals the emotional tone of the influence and get rid of information about communication flow that wandered onto this model (a common error).

Then, add unique influences between influencers

Every organization has its own culture and attitude about the "right way" to do business. This culture may be promoted directly by management or may be pervasive, with no clear source. We sometimes find it useful to represent both cultures on the consolidated model (Figure 9.23). The model will show both where the culture is

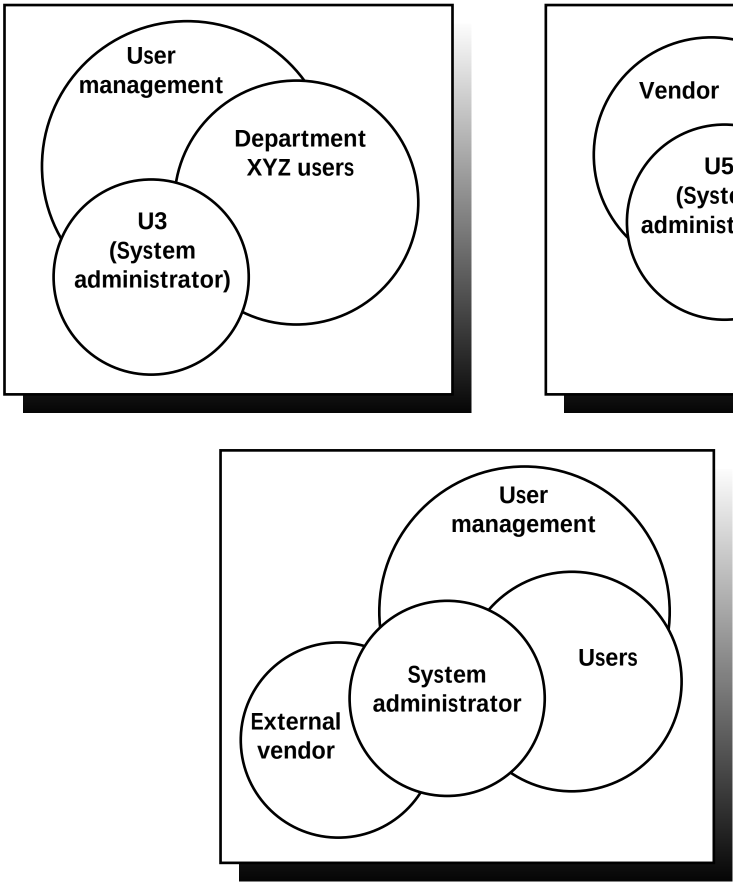


Figure 9.21 Identifying common influencers.

common across instances and where it differs. For example, some companies are totally customer-driven, while others appear not to know customers exist. The consolidated cultural model represents the issue and either shows the common attitude across the population or the variety of differing positions. Figure 9.24 shows a complete consolidated cultural model.

Keep variation across business or national cultures

The cultural model is one of the easiest to consolidate—it's usually fairly clear what goes together on the model. But the impact of the model is very great. The consolidated cultural model takes a bunch of

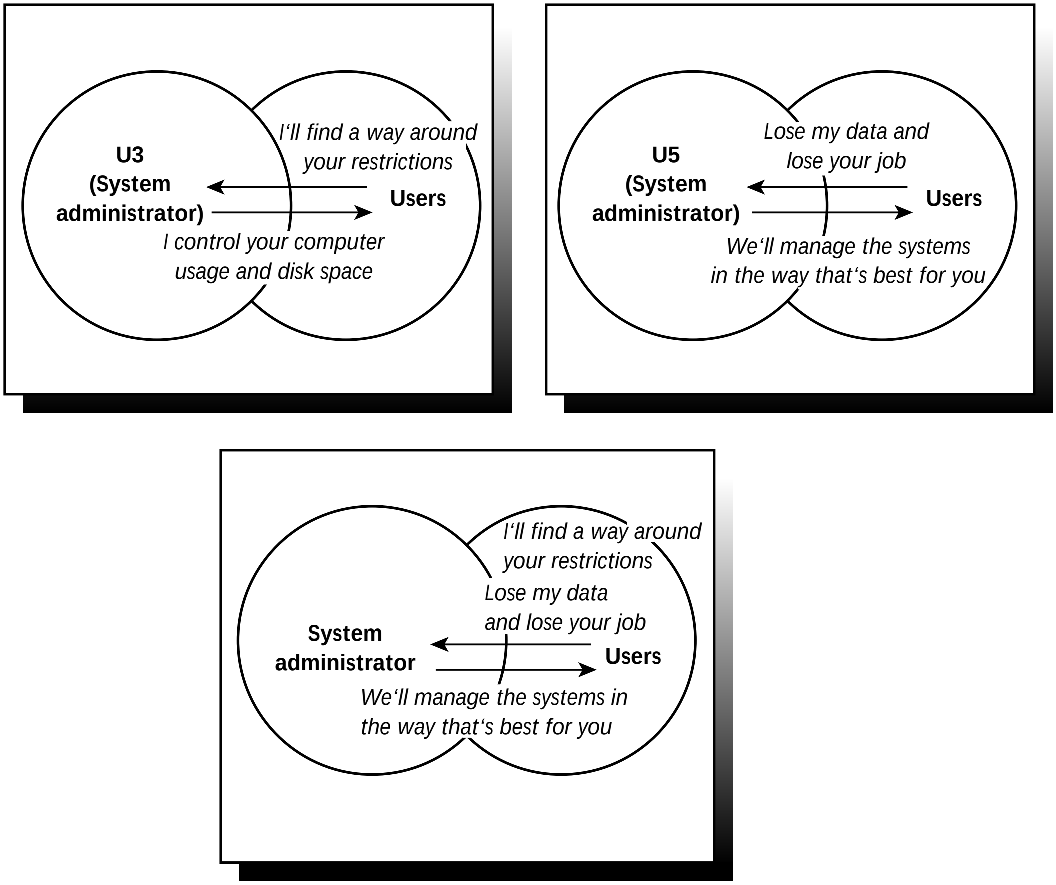


Figure 9.22 Consolidating influences.

Steps

- Catalog influencers from the individual models.
- Group influencers who constrain the work in the same way.
- Collect influences from the individual models. Group by the pair of influencers they go between.
- Sort each group of influences, eliminating duplicates.
- Draw the final model, showing all unique influencers and influences. Copy over any breakdowns.

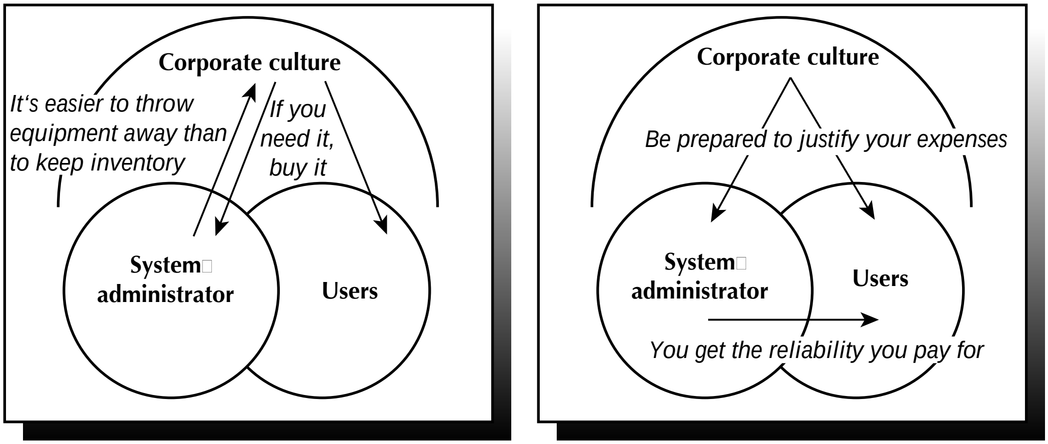


Figure 9.23 Two cultural attitudes toward money.

disconnected anecdotes and reveals the common themes and issues that a whole customer population cares about. By addressing these primary values, a system can distinguish itself from its competitors. The design team can address the issues, and the marketing team can use them to highlight benefits people really care about. Then the rest of the cultural model shows how to keep the system from trespassing on the customers' way of doing business either by violating a value or by failing to fit into the user's work style or environment.

The cultural model is always important when a system is designed for an internal organization or group. It's critical when characterizing a market—it shows what the market cares about and what pervasive influences they have to respond to. It's also important when the work being supported involves multiple groups of people interacting—the way people push back on each other shows up in the cultural model. The model is less important when the project is narrowly focused on the work of an individual; in this case, the few cultural issues of the user's values and self-image can be collected on the affinity.

The cultural model reveals the important values to address

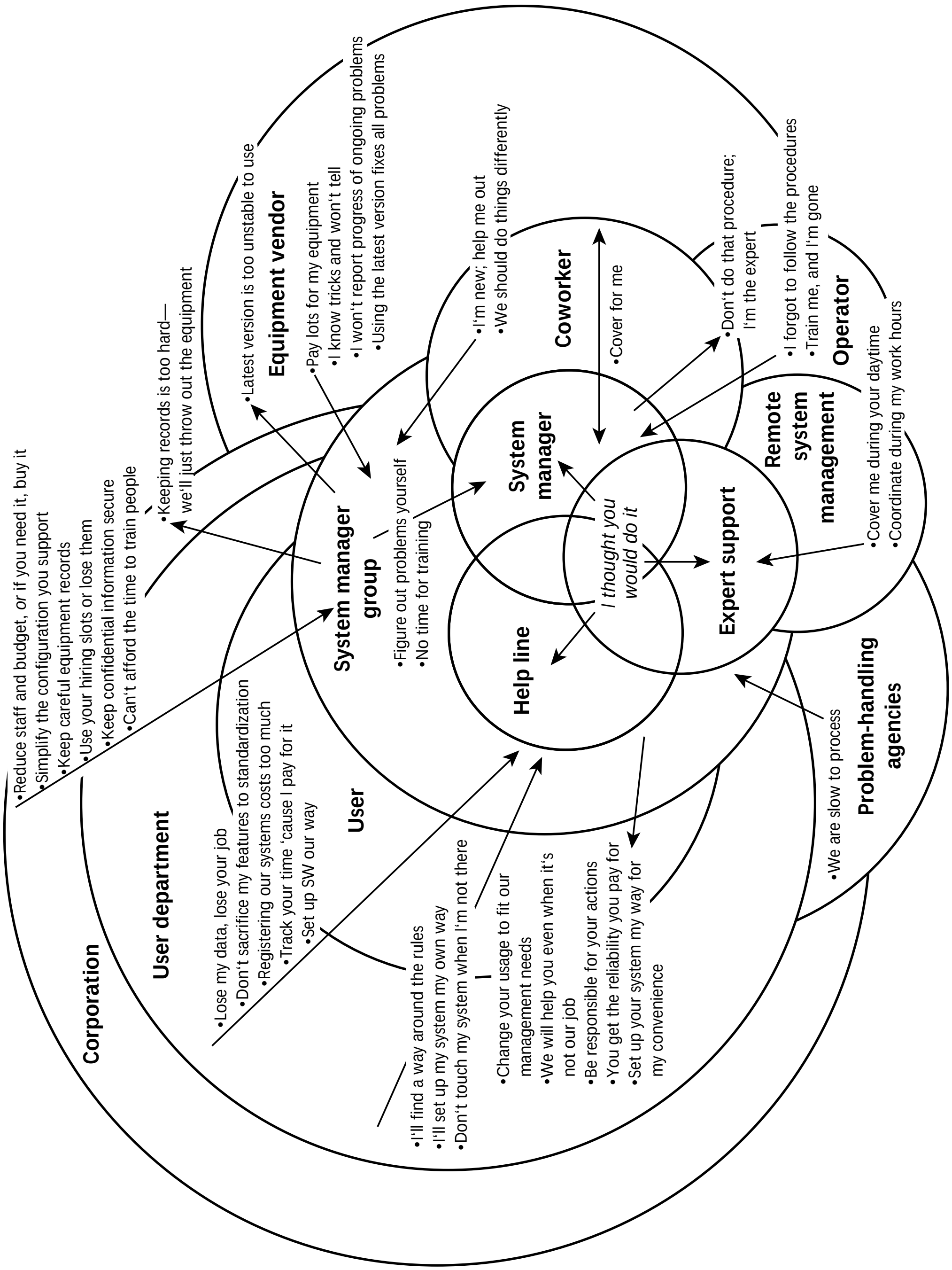


Figure 9.24 Consolidated cultural model.

The thought process of consolidation

Looking back over the different kinds of consolidation, it's apparent that the same kind of thinking process drives them all. We collect the data points of an affinity across users and build them up into groups. We organize responsibilities from different users into roles on the flow. We collect work steps and group them into abstract steps and intents. We collect and group parts of artifacts and places in the physical environment. And we collect influencers and influences in the cultural model. The detailed items say what to pay attention to; inquiry into each item reveals meaning for the project focus and how to group it with others. Out of that comes common structure and meaning.

Induction reveals the pattern and meaning hidden in work instances

Taken together, the consolidated models provide the detail about work needed to inform system design. Out of these models a design team can draw implications that guide design.

The customer's *intent* is the first and most critical implication to draw from the models. Sequence models show what the customer is trying to do and how they go about doing it. Artifact and physical models identify additional intents from the structures people create. The affinity shows intents directly. And the cultural model shows why people care—the constraints and values that are the reasons why an intent is important to customers. If designers can invent ways people can achieve their intent more directly, they streamline the work and reduce unnecessary steps.

People achieve their intents by putting *strategies* in place. The flow model reveals strategies for breaking up the work into organized units across people. The affinity collects strategies and shows how they relate to other work issues. The sequences reveal alternate strategies used to achieve the same intent. Designers can build these strategies into their systems or choose to improve on them.

Some strategies are made concrete in *structure*. Grouping tools into a cluster, separating work into piles, and organizing notes on a page are all different structures that make work strategies possible. These structures can be re-created in an online system when they are useful; when not, the system can provide better alternatives.

Structures also represent *concepts.* Concepts are created by people to help them manage and think about their work. When they create artifacts in the course of doing work, they naturally represent the concepts in the artifact. The affinity names and highlights additional concepts. With an understanding of the concepts that organize work, designers can structure systems to implement and communicate in terms of those concepts. Building the user's concepts into the system makes it easier to learn and use.

Useful design data reveals the intent, strategy, structure, concepts, and mind-set of the user

Finally, all these implications are affected by the customer's *mind-set.* The cultural model shows mind-set explicitly, but it can also be inferred from the physical environment and the detailed steps that people take in accomplishing a task. Understanding the customer's mind-set points designers at the important issues to solve and ensures that the final system will fit with the customer's work and culture.

Understanding intent, strategy, structure, concepts, and mind-set are key to effective process and system design. The work models make these aspects of work visible to designers. Each model captures a unique perspective, and each shows the common pattern of work and the variation across a customer population. They make the customer real to the engineer—so real that when, at two in the morning, he or she must make a design decision one way or another, the consolidated customer work has sufficient weight that there's a chance that the decision will be made in favor of the customer.