

The Cost Structure of Sensemaking: Analysis of Three Case Studies

Daniel M. Russell* & Mark J. Stefik
Information Sciences and Technology Laboratory
Xerox Palo Alto Research Center

DRAFT
October 21, 1993

The main problem of information isn't retrieving it, but making sense of it. The work of sensemaking includes not only retrieval but also the work of extracting, recombining, reorganizing and re-representing information. When on-line information systems focus solely on retrieval, most of the information work takes place on other media such as paper. This yields media clash and inefficiencies in performance. We present three case studies of computer information systems used for sensemaking tasks and a framework of analysis in which the work of sensemaking is classified in recurring behavior patterns we call *learning loops*. We consider the real costs of different approaches to sensemaking by comparing the inherent costs for the operations and representations they employ.

* The first author has moved to the Advanced Technology Group at Apple Computer, Inc.
(DMRussell@Taurus.Apple.COM)

Sensemaking versus Retrieval	1
Media Clash and the Affordances of Paper	1
Terms of Analysis.....	2
Learning Loops.....	2
The Cost Structure of Information Work.....	4
Case studies of Sensemaking	5
CASE 1: Making Sense of Laser Printers.....	5
Background.....	5
The Sensemaking Task	5
The Learning Loops.....	7
Operations and Cost Structure.....	8
Operations for Phase 1: Creating and instantiating a representation .	8
Operations for Phase 2: Clustering the encodons	9
Operations for Phase 3: Organizing the clustered data.....	10
How Intermediate Representations Reduce Costs	10
The Value of Encoding.....	11
The Value of Clustering	12
The Value of Concept Identification	12
Analysis of Representational Shifts	13
Case Summary and Conclusions	15
CASE 2: Making Sense of Document Recognition.....	16
Background.....	16
The Sensemaking Task	16
Learning Loops.....	17
Operations and Cost Structure.....	18
Operations for Phase 1: Creating an initial representation.....	18
Operations for Phase 2: Extending and instantiating the initial representation.....	19
The Value and Combinatorics of Collaborative Sensemaking	21
Problem-Solving by Committee.....	22
Indirect Measurement of Committee Work	22
Counting Interactions	23
The Costs for Centralized and Distributed Sensemaking.....	24
Case Summary and Conclusions	26
CASE 3: Making Sense of High School Algebra	26
Background.....	26
The Sensemaking Task	26
The Learning Loops.....	27
Operations and Cost Structure.....	28
Operations for Phase 1	28
Operations for Phase 2	29
The Value of Heavyweight Schemas.....	29
Reminding	30
Collecting Course Data	30
Case Summary and Conclusions	31
Conclusions	32
Reducing the Cost of Sensemaking.....	33
REFERENCES	33

Sensemaking versus Retrieval

When I visit a library, I sometimes know exactly what I am looking for: the date the light bulb was invented, the name of the capital of Spain, the frequency of sound emitted by bats, or perhaps the latest book by Spider Robinson. In such cases, the problem is one of finding the information, using appropriate catalogs, reference books and their tables of contents and indices.

These examples fit within the classical formulation of information retrieval: there is a corpus of data. You get items from the corpus by describing them with a query. For example, I can ask for the abstracts of "all journal articles published in the last ten years that mention 'bats' and 'ultrasound' in their title." The chief metrics used for measuring the goodness of information retrieval systems are *recall* and *precision*. Recall is the percentage of relevant documents that are retrieved. Precision is the percentage of those documents retrieved that are relevant.

However, information retrieval is a narrow prescription for information systems. It is useful when the main problem can be simply solved by looking up an answer, but it fails to characterize much of what we do when we seek information from documents. Beyond information retrieval, there is still the "other 90%" of the information work: organizing the information, breaking it apart and recombining it in new ways, and generally re-presenting it for some use. Even the prototypical "going to the library" task is more than just finding information. Usually when visiting the library I don't know exactly what I am looking for. Often there is much more information available than is tractable to deal with, and relevant information may be filed in non-obvious places. Part of my process is to sample information, understand parts of it, refine my goals, and then repeat.

We use the term *sensemaking* to connote a view of information use larger than that normally associated with "information retrieval" and yet common to many tasks. In this view, information is not just retrieved but also reorganized to be used. We are interested in the performance of the larger sensemaking task, not just the performance of retrieval.

Sensemaking is central to many kinds of jobs, such as market analysis, technology analysis, intelligence analysis, policy analysis -- almost any kind of a job whose title includes the term "analyst." It is a major part of what people do who read a lot, retrieving and correlating information in documents -- even if their main job is not "analysis."

Media Clash and the Affordances of Paper

A retrieval-centered view is much too narrow for designing information systems. It divides the total information work into two parts: retrieval work and everything else. Computer systems designed using this point of view provide tools for making and refining queries, but few facilities for using or manipulating the information once it is retrieved. The lack of appropriate facilities on the computer forces a sensemaker to use paper. Consider a library research task where the set of retrieved documents is delivered on paper or in a separate capture file. The user may organize it by stacking the documents in piles or placing them in folders; the user may annotate documents by attaching notes, writing in the margins, or highlighting interesting sections. The user may clip out particularly relevant sections from long documents. However, to then prepare a report based on the reorganized information, he needs to transform the retrieved information into something that can be used in the task.

The retrieval-centered approach divides retrieval work and organizational work along the lines of computer work and paper work, requiring additional steps every time we cross media. Retrieving

data in one form but using it in another is an example of a *media clash*. Media clashes are rampant in sensemaking because organizing or using data interacts with further retrieval.

There is an opportunity here to significantly augment our use of computing to support sensemaking tasks. Recent advances in networking and mass storage technology have brought access to gigabytes or even terabytes of information within easy reach of many people. The 1990s promise to be the first decade of the computer age where the emphasis is on extracting and re-using information rather than producing it; *where the average computer users get more information out than they put in*. This is a shift in the patterns of information work on computers, characterized by a reversal in the dominant direction of information flow for the average information worker.

This paper presents three case studies of successful sensemaking systems. In each case, the organizational part of the work is deftly supported by a computer and integrated with retrieval. Our goals in this paper are to offer concepts for understanding sensemaking tasks and computational support for them. In each case we offer a first-order analysis of the costs of all the sensemaking operations, not just those inherent in a retrieval-centered point of view.

Terms of Analysis

We now introduce concepts that we will refer to throughout: learning loops and cost-structure analyses.

Learning Loops

Sensemaking is the process of iteratively constructing a representation that will support an analysis task. Creating such a representation is not a one-shot deduction, but is characterized by iteration as the sensemaker creates a representation and fits information to support its use.

Figure 1 shows the central pattern of iterative activity we find in each of the sensemaking cases that follow. A sensemaking task often contains several learning loops woven together in complex patterns. Regardless, each of these stages, activities and feedback loops occur in all of the sensemaking cases we have studied.

Although learning loops occur in sensemaking tasks even without computers, we are interested in cases of sensemaking where an information system is central to the approach. The information systems we consider provide capabilities to process information and a medium for storing and manipulating representations. For us, a representation is any symbol structure used to refer to something else in the world. Representations can be stored, retrieved, and manipulated in computers, usually to accomplish some specific set of tasks. Every representation directly encodes some set of the available information while omitting others. In this way, the design of representations is influenced by a judgment of what is important for the analysis and relevant to the task at hand.

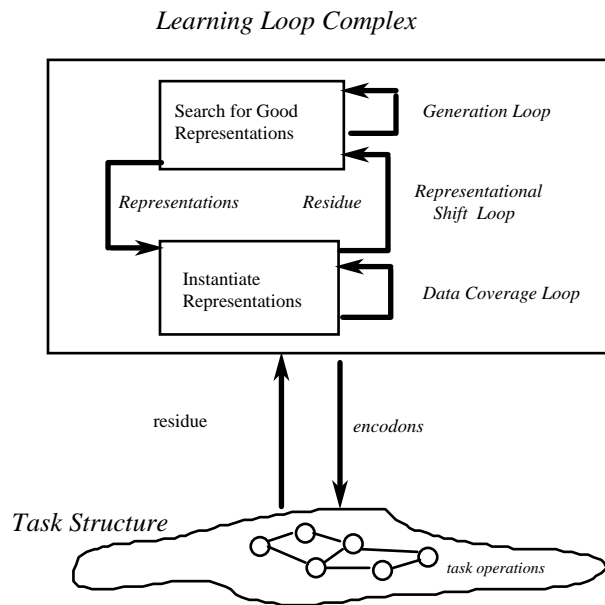


Figure 1. Sensemaking is finding a representation that successfully organizes information from collected documents to satisfy a set of objectives. The learning loop complex describes the pattern of actions typically followed to create and instantiate the representation.

The learning loop complex has the following steps:

1. *Search for representations.* The sensemaker creates representations called *schemas* that capture the important regularities in a way that is useful for his task. A schema expresses relevant data about a class of objects. The *generation loop* searches for a schema to represent a class of objects.
2. *Instantiate representations.* During the early parts of sensemaking, schemas provide top-down or goal-directed guidance. They prescribe what to look for in the data, what questions to ask, and how the answers are to be organized. In the *data coverage loop*, a sensemaker retrieves data and encodes it in instantiated schemas called *encodons*.
3. *Shift representations.* If there were no surprises in filling in encodons, sensemaking would be trivial. We could just define the schemas and then instantiate them. However, sensemaking seldom works this way. When there are surprises in filling in the encodons, the sensemaker can go back and revise the corresponding schemas. Representation shifts during sensemaking are intended to reduce the cost of task operations. Forcing a change to the representation in this way is a bottom-up or data-driven process. Ill-fitting or missing data are called *residue*. The *representational shift loop* is guided by the discovery of residue. When there are relevant data without a place in the representation, the schemas can be expanded. When data do not fit the established categories, the original schema categories may need to be merged or split. New categories may be added. Thus, sensemaking iterates between top-down representation instantiation and bottom-up search for representations.
4. *Consume encodons.* The sensemaker uses the encodons in a task-specific information processing step. In sensemaking tasks, these are the final steps that produce the results of the analysis. Consuming encodons can be complex (as when they are fed into another step of the

analysis process) or very simple (as when the solution to a problem can be read directly from the encoded representations).

Collectively, we refer to the constellation of loops as the learning loops of sensemaking. This process is a more general form of processes of gathering and interpreting numeric data, as in Exploratory Data Analysis (EDA) [Tukey,77]. The EDA "residual" is very similar to our notion of "residue." The primary difference is that EDA representations and accounts are statistical descriptions of numeric data, rather than the less-formalized relational representations used here.

Figure 1 portrays the learning loops in a canonical form. Many analysis tasks contain multiple learning loops intertwining in various ways. The encodons created by one learning loop can be the documents consumed by another. In our case analyses, it has proven useful to identify the individual loops for generation, data coverage, and representational shifts as a way of determining what sensemaking operation is being performed and what tradeoffs are made.

The Cost Structure of Information Work

We characterize information work in terms of component steps, where each step performs an operation on data. Any step can be shown at different levels of granularity. For example, inside the large steps of the generic loops of sensemaking task there can be steps for collecting documents from a corpus, instantiating representations, or viewing representations and residues. Even more detailed steps include creating categories that group encodons together or order them.

Each step of a learning loop has a cost, that is, requires various resources. The easiest resource to characterize is usually time. We characterize steps according to how long it takes to carry out the operations. The time of an operation usually depends on the amount of information elements being processed. For example, for a sorting operation we can ask whether the operation takes twice as long or more when the length of the list of data is doubled. This amounts to determining how cost scales with size.

Time is not the only resource used by operations. There are also physical, cognitive, and computational resources. For example, in sorting papers on a dining room table one resource is the amount of room on the table. Another limited resource is the sensemaker's memory. Thus, an operation that would require a sensemaker to quickly remember twenty 10-digit random numbers would present excessive demands.

A *cost structure analysis* accounts for the resources and trade-offs in action required to carry out a task [Russell, et. al., 1993]. A comparative cost-structure analysis compares alternative ways of carrying out a task -- using different approaches with their own operations. For example, we could compare the operations and time needed to draw a chart with paper-and-pencil versus performing the same task with a particular graphics or spreadsheet program.

Sensemakers act to minimize their perceived costs. This process of deciding what to do is itself an imperfect task, limited by available cognitive resources. Faced with this, sensemakers use imperfect and incomplete information, often relying on simple behavioral strategies to select a low cost solution.

In the following cases we are concerned with three elements of cost structure: accounting for the costs of information work using particular operations and representations, accounting for differences in cost when representations are shifted, and accounting for the cost-changing effects of using a

sequence of intermediate representations. In general, when a sensemaker changes the representation of information, it is in an attempt to reduce the cost of a later step that uses it.

The goal in these analyses is to illuminate issues and opportunities in the design of effective information systems to support sensemaking tasks.

Case studies of Sensemaking

This section presents three cases of using information systems in real tasks that required more than information retrieval. Each case requires making sense of information contained in a large corpus of documents. Each case is presented in a way that illustrates particular issues both in the methodology of cost structure analysis and in the cost structure of sensemaking tasks.

The presentation of the first case is longest because we use it to introduce our analysis approach and because it is the most detailed. The last two cases build on this and go more quickly.

CASE 1: Making Sense of Laser Printers

Background

Like all large manufacturers of technical equipment, the Xerox Corporation has an organization of field service technicians (known as "tech reps") who maintain and service equipment for its customers. Training courses are developed for teaching about new products and also for providing a baseline of experience to new employees. Every year a few thousand employees take these training courses.

In the summer of 1989, a team of experts was organized within the Xerox education division to design a new course on laser printing for training Xerox technicians. Their charge was to create a "generic service training course" to cover a wide range of laser printers including new ones manufactured by Xerox and other companies. Other short courses were intended for field service on particular copiers. The emphasis of the course was to teach basic principles of operation widely applicable across all laser printers.

The group chose IDE (the Instructional Design Environment) [Russell, 89] a hypermedia knowledge-structuring tool, to help in this analysis task.

The Sensemaking Task

The required product of the sensemaking was an outline for a course on laser printing. A key part of the task was to understand the commonalities and differences among printers. This involved answering questions such as: what are the basic similarities and differences between laser printers, operation, ways of failing and normal functions? What systems should the course cover?

Figure 2 presents a flowchart of the process used by the group. During the initial two-month period, the lead group identified 21 different kinds of laser printers and several different kinds of scanners to be considered. Members of the lead analyst group searched for representations and a vocabulary for describing the devices. Their goal was to provide a common language of description to make it easy to find similarities between devices.

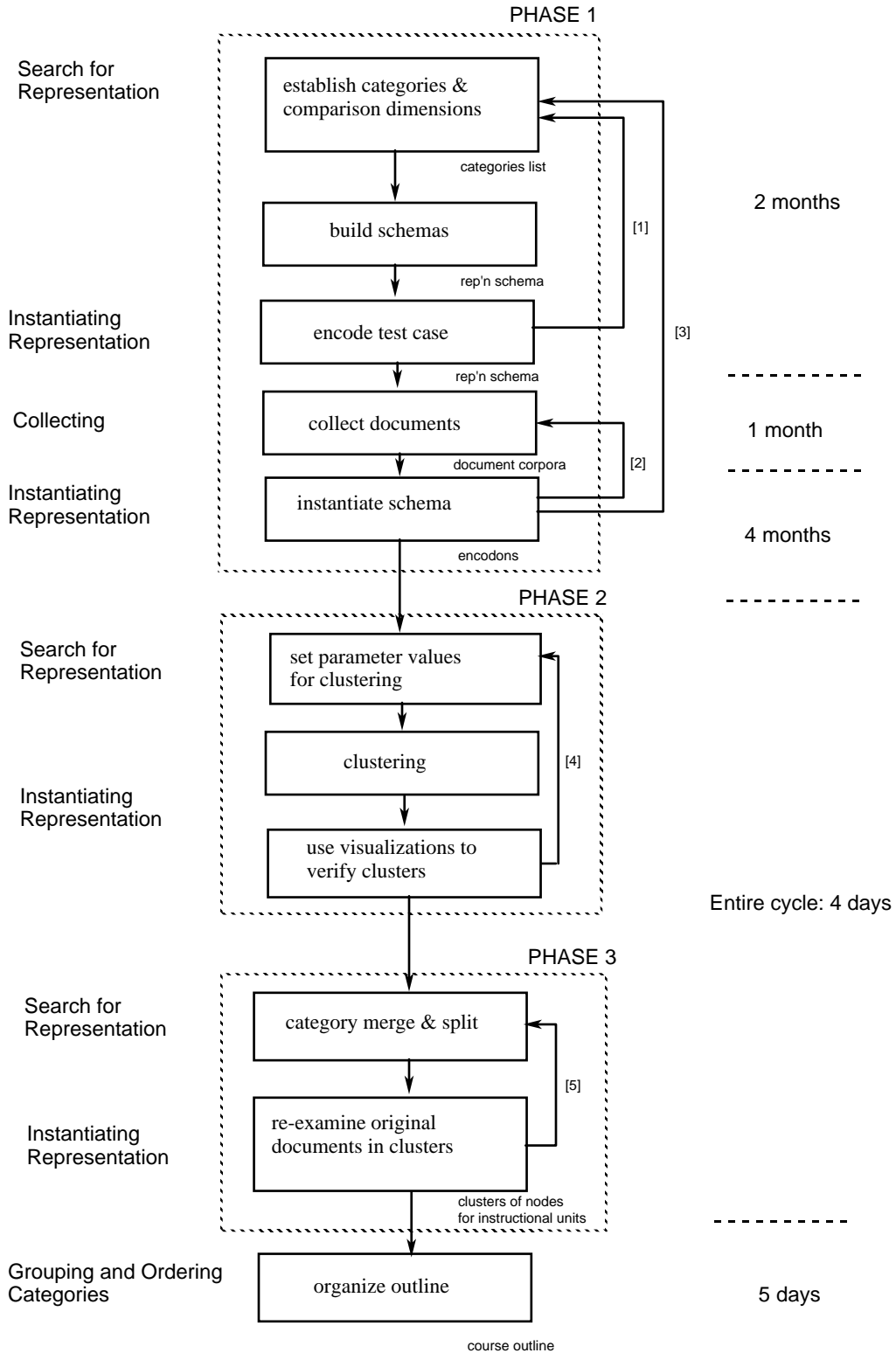


Figure 2: How could a team of instructional authors understand the field of laser printers? This flowchart shows the steps they followed to discover clusters of common terms, ideas, subsystems and functions among twenty-one different laser printers. The left column shows what kind of work was being done, while the right hand column shows how long each subtask required. Dashed boxes mark off a phase of work corresponding to a single learning loop.

It was not feasible for an individual to do all of the work. Work was divided among three full-time work groups. During the next five months, the work groups collected and entered information about their assigned printers. To make their results comparable, they encoded their information about printers using the representation schemas provided by the lead group. The results of this encoding effort were a set of IDE databases, one per printer, containing between 300-1000 encodons each.

The next phase used computers to compare the different printers, identifying important similarities and differences. Because the printers were designed and documented by different companies, there were substantial variations in the way printers were described. Each printer had subsystems for moving paper, scanning images, scaling images, and so on. Different printers used different designs and consequently had different capabilities, modes of failure, and so on.

The output of the second phase was a set of common categories from each of the databases. These clusters were then used by the sensemakers in the third phase to identify the topics and concepts for the course. By examining the sources on which the clusters were based, the groups identified the educational concepts behind the clusters.

The last phase was to create an outline by organizing the concepts found in the analysis. An outline was represented by relational links that sequenced the encodons representing the educational concepts. The group considered several alternative course outlines. These were compared by a viewing program.

The Learning Loops

Each phase of this case analysis corresponds to one learning loop complex.

The loop labeled (1) in Figure 1 is a representation shift loop. Here the lead group used test cases to develop the schemas. Over the two month period, this group read the documentation about a few test printers, built schemas, and encoded the information using the schemas. Loop (1) shows the path taken when residue forced a change to the schemas. Later we will see that this loop took the lead group through four major representational shifts. The generation and data coverage loops for this cycle are not shown.

In the data coverage loop (2) the work groups used the schemas from the lead group to represent the 21 printers. To ensure uniformity of encoding, the work groups were discouraged from changing the schemas. Their recourse in the event of encoding difficulties was to contact the lead group and to request a representational change that would then be used by all of the work groups. This representational shift loop is labeled (3) in the figure.

The second phase of the process was to identify similarities across printers, faults, components, and subsystems. To do this, the computer system performed a cluster analysis using the schemas created by the work groups. The output of this phase was a set of identified clusters in each of these spaces, that is, identified sets of similar functions, identified sets of similar faults, identified sets of similar subsystems, and so on. The groups set values for the clustering parameters and also synonym lists for the linguistic analysis. They would then run the program and examine the results. Loop (4) is the representation shift loop for this process, where they would adjust the parameters for the next pass in order to yield clusters that they found meaningful and useful. Again, the data coverage loop used by the program and generation loop used by the group are not shown.

The third phase of the process took the comparison results or clusters and abstracted out educational concepts for teaching. In some cases, these concepts corresponded directly to clusters, such as common fault modes and functions for printer subsystems. In addition, other pedagogical concepts about operation and repair were developed. Loop (5) represents the representational shift loop for this process, where the group modified and extended the set of pedagogical concepts in order to adequately cover the examples from the corpus.

The last part of the process was to develop a course outline, trying different orderings for presenting the material for the course. The learning loops for this phase are not shown. Later examples in our collection of sensemaking cases are very similar and illuminate the kinds of representations that were used here. We elide this part of the process so as to focus on the elements of this case that make it unique.

Operations and Cost Structure

In the following, we first review the operations at each stage. The cost of sensemaking is then considered in terms of the costs of these operations. We then consider variations on the plan that leave out some of the steps in order to illustrate how the composed operations were efficient in producing the final course outline.

Operations for Phase 1: Creating and instantiating a representation

Much of the work of sensemaking is classification. Classification groups large sets of objects in terms of their common properties. Schemas are used to represent classes. Notationally, schemas correspond to frames or classes in knowledge representation languages, or to nodes in a hypertext system. Figure 3 shows an IDE schema, including slots for the part name, part number, and tasks. Some of the slots can point to other schemas -- so adjustment tasks and repair tasks are considered part-of a description of the component they affect.

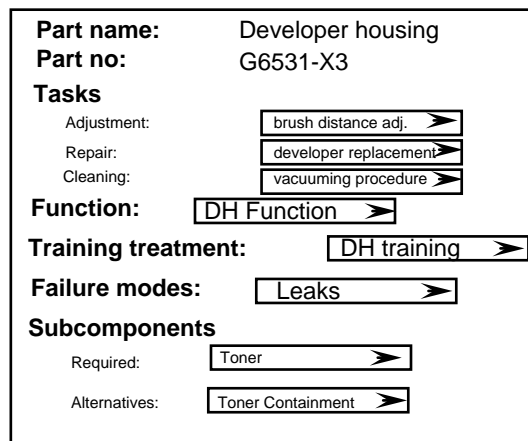


Figure 3: A typical encodon from the laser printer data showing an instantiated fragment of the larger representation. Boxed values are pointers to other instantiated schemas.

In the first phase, the lead group used test cases to develop representations to be used by the three work groups. In developing these representations, they asked and answered several questions: What are the objects being modeled? Into what classes do they fall? What are their parts? In what ways are they similar? How do the parts differ in different objects? In the latter part of the first

phase, the work groups then used these schemas in describing all of the printers to be covered in the course.

We first consider operations for building schemas. Schemas are not just defined and then instantiated to produce encodons, but are repeatedly edited and changed to reflect new information or new structure as discovered. In the manner described by learning loops, tentative schemas are defined, residues are discovered during instantiation, and the schemas are modified to accommodate the changes. Operations include defining a class, merging two classes, splitting a class, specializing a class, adding a slot, defining sets of legal slot values, promoting a slot, demoting a slot, and splitting a slot to form multiple slots.

Operations for encoding information included finding relevant documents, browsing them, indexing them, and encoding slot properties in the schemas.

Formal representations are ones that a computer can interpret. Typically, such representations have very constraining descriptions. For example, a formal characterization of perceived color might be a description in terms of red-green-blue intensities or an explicit list of color names. An *informal* representation is one that is open to human interpretation, but which has no machine interpretation. For example, arbitrary text descriptions and figures usually have this character. *Semi-formal* descriptions are intermediate in this dimension: some structure is imposed and a computer is able to make limited interpretations of the descriptions. Semi-formal representations allow some form of escape to provide descriptions that are human-interpretable.

Recognizing that work groups would be looking at cases with some features missing in the test cases and that these cases would lead to residue, the lead group decided to use semi-formal descriptions. Thus the schemas provided a structure which described some of the object properties, as well as provisions for some open-ended text descriptions. This approach enabled the groups to complete their descriptions without agreeing on all of the details of formal representation before starting to encode, and without requiring much iteration and representation shift and negotiating during the encoding phase when the groups were located separately.

Operations for Phase 2: Clustering the encodons

The second phase of the printer sensemaking task was to identify groups of related objects in the printer information, that is, groups of related printers, related subsystems, related fault modes, and so on. This process of inferring groups from patterns in their data is called clustering. Common clustering methods always define a distance metric like the following:

$$d_{ij} = \sum_{k=1}^n w_k \Delta_{ijk}$$

In other words, for each kind of object being compared, there is a set of properties, P_i . Each property is treated as an independent dimension and has a value which is its coordinate in that dimension. For example, the properties for document handlers would include the set of acceptable paper sizes, the capacity, a category such as "duplex printing capable" and a description of the mechanism. The Δ term in the equation refers to the difference between two property values for objects i and j within dimension k . For properties with non-metric values, a difference of 0 is usually assigned if the property values are equivalent and a difference of 1 is assigned if they are not. The output of clustering is a set of node clusters all within a small distance of each other. Each node is placed in a cluster with other similar nodes if its fit is good enough (i.e., above a cutoff value).

The distance equation indicates that the overall distance between two objects is a weighted sum of the differences between their property values, where the weights reflect the relative importances of

the considered properties. To account for interactions among properties, more sophisticated distance metrics are often employed. However, the phenomena of interest to sensemaking arise even in the use of this simple distance metric.

There are different ways to compute clusters, but most methods include the following steps: (1) assign values to the weights, (2) compute distances between the objects and the cluster centers, extracting property data from the documents as needed, (3) add objects to a cluster if they are near enough and form new clusters otherwise, and (4) evaluate the cluster pattern. If the cluster pattern is unsatisfactory, such as if most of the objects fall in one cluster, then the weights are adjusted and the process repeats. In practice, there are several parameters to explore in the clustering process: the between-node similarity distance, cutoff points for link matches (i.e., role similarity measures), and the choice of synonym dictionaries

Finding the distinct concepts in a large corpus is a non-trivial task. In this case, it was possible to carry out much of the process automatically. The collected work groups examined the cluster results produced by the computer, discussed whether the clusters were meaningful, and then adjusted the parameters. The distance metric used structure both from the schema (the names of slots, type information and relations to other elements) and from the natural language descriptions in some of the slots. The similarity of two natural language descriptions was based on the frequency of words in the text descriptions. High frequency words (such as "the", "and" and so on) were omitted using a stop list. A set of synonyms identified words of nearly equivalent meaning, and all words were reduced to stems so that words like "buffering", "buffered," and "buffers" would match. Two components have a small distance if they have a large shared synonym set, and also play similar functional roles within their subsystem.

A crucial enabler for the use of computers in finding clusters was that the information about the printers, subsystems, fault modes and so on was in a form where the property values could be automatically retrieved. Not only was the information on-line, but it was already parsed into comparable properties and indexed by object and property name.

Operations for Phase 3: Organizing the clustered data

During the third phase of printer sensemaking, the work groups identified the topics and concepts that would be presented in the course. The set of concepts was anticipated by the design of the schemas. Restated, because the concepts for the course were ones that could be derived simply from the concepts in the schemas -- descriptions of printers, functions, fault modes, and so on.

The operations of this phase are roughly the same as the operations of the first phase. The difference in practice is that the primary source material for this phase was the set of clustered data rather than the original printer documents. Pedagogical concepts were tested for coverage according to how they matched the clustered and encoded information.

The clustering process helped the group to identify several concepts that had been overlooked in the earlier courses. For example, the program showed that the concept of "buffering image data" was common to many printers that handled paper or images with different rates or latencies. The concept had not been made explicit in earlier courses, perhaps because different printers implemented buffers in different ways, obscuring their common function.

How Intermediate Representations Reduce Costs

The key to understanding the cost structure of this sensemaking task is to focus on how each of the operations yield a sequence of intermediate forms of the information that efficiently support the next operation. Without the intermediate forms, alternatives for later operations would be prohibitively expensive. By analogy with an industrial manufacturing process, the source information is akin to a raw material which is refined through several stages. Following the analogy, the source material is the ore, which is melted and refined to create ingots, processed again later and remixed to create specialized metals, cast in molds to create parts, which are then finished and assembled into the final manufactured goods.

Working backwards, the final training concepts needed to be created from the clusters of functions, fault modes, and subsystems. To determine the clusters, the information needed to be made indexable so that the properties of two printers could be efficiently compared. To efficiently compare two printers or subsystems, the text and diagrammatic information had to be reduced to a common encoded schematic form. To create the encodons, the indices, categories, and vocabulary needed to be determined for the schemas. This brings us to the beginning of the process. The phases of this project are shown in Figure 4.

To illustrate the underlying efficiencies of the approach used, we now offer a thought experiment to contrast the costs of alternative approaches. Some of these suggestions go against common sense. As we explore the alternatives, we quickly realize how many of these contrasting approaches are untenable.

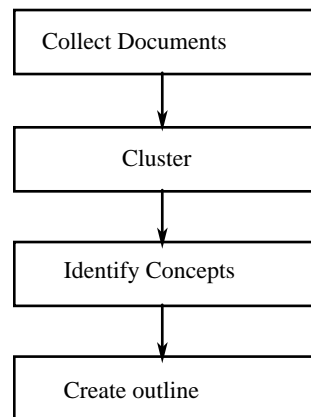


Figure 4. The four phases of the laser-printer sensemaking case.

The Value of Encoding

We begin by considering the first two phases. Suppose that the encoding part of the first phase was skipped: that is, the documents were collected and the objects and properties were identified, but that the encodons were not created on-line. In phase 2, the group needs to cluster together related printers, faults, subsystems, and so on. How would they do it?

In a naive way, we can imagine the group leader going to a whiteboard and calling out to group members for the properties of their printers, faults, and so on. In visualizing this scenario, we need to keep in mind the amount of material in the original source form. Each printer was documented in from 300 to 6000 pages.

Assuming that the group needed to access each property only once (which is unlikely), the time required to cluster the objects, accessing all of the properties of all of the objects assuming 1 second per page of source documentation would be something like:

Time = 21 printers * 500 entries/printer * 3000 pages/printer * 1 sec/page = 31,500,000 seconds

which is 8750 hours. With 40 hour work-weeks, this amounts to 218 weeks or about 4 years .

There are several possible objections about these assumptions giving rise to the large time estimate. For example, people could work in parallel looking things up. That gives perhaps a factor of 10 if perfect parallelism were achieved. They might be able to skip whole sections of the documents if the analysts were familiar enough with them. But then, additional time would be needed to account for inconsistencies of coding in the original documents and even more time is required for conversations as the leader calls out for data.

This illustrates the economic rationale for the sensemakers' encoding in the first phase. With the data on-line, they were able to perform several iterations of the clustering algorithm economically, since the data comparison and accounting was performed at the speeds of a scientific workstation rather than at the conversational speeds of humans. We don't know exactly how the group would have performed the clustering operations without computers. Using a conservative ratio of about 10,000:1 for comparing the raw computing speeds of people versus computers, we get

Time = 4 clustering passes * 4 hours * 10,000 = 160,000 hours

which is about 76 years, assuming 40 hour work weeks. The estimate is so far beyond the resources that would be devoted to this task that it is clear that the group would have needed to find a very different way of doing the work.

Of course, nobody would follow so naive an approach. An alternative would be to adapt structures from earlier courses. The problem with that suggestion in this context is that the whole point of the sensemaking task was to make a new kind of course that was not limited to a single family of printers.

In summary, a major benefit of the encoding is that it *enabled* automatic cluster analysis, providing the group with analytic possibilities -- with increased thoroughness and accuracy -- that would otherwise have been entirely beyond their means.

The Value of Clustering

The work in the third phase is to identify the educational concepts to be covered in the course. In this they had develop and answer questions: Which printers are similar? What are the common functions of these printers? Does this fault mode arise in other printers? Does this fault mode arise for all printers having the same function, or does it depend on the design of the subsystems?

As before, we ask how the work of this phase would be different if the preceding information work had not been done. The effect of omitting the encoding work of the first phase would be pretty much the same as in the preceding analysis, so we will not repeat that here. Suppose instead that the sensemakers created the encodons, but they didn't use the automatic clustering process of the second phase.

The sensemakers would still be faced with the prospect of grouping and comparing printers in order to answer their questions. To determine whether a given fault mode appears across different printers, they would want to scan the data to see where the fault mode appeared. If two printers had the same functions but different fault modes, they would want to verify whether the difference was real or was a discrepancy in encoding. In short, if the clustering were not done ahead of time, they would still have to perform clustering -- perhaps repeatedly -- in the work of the next phase.

The Value of Concept Identification

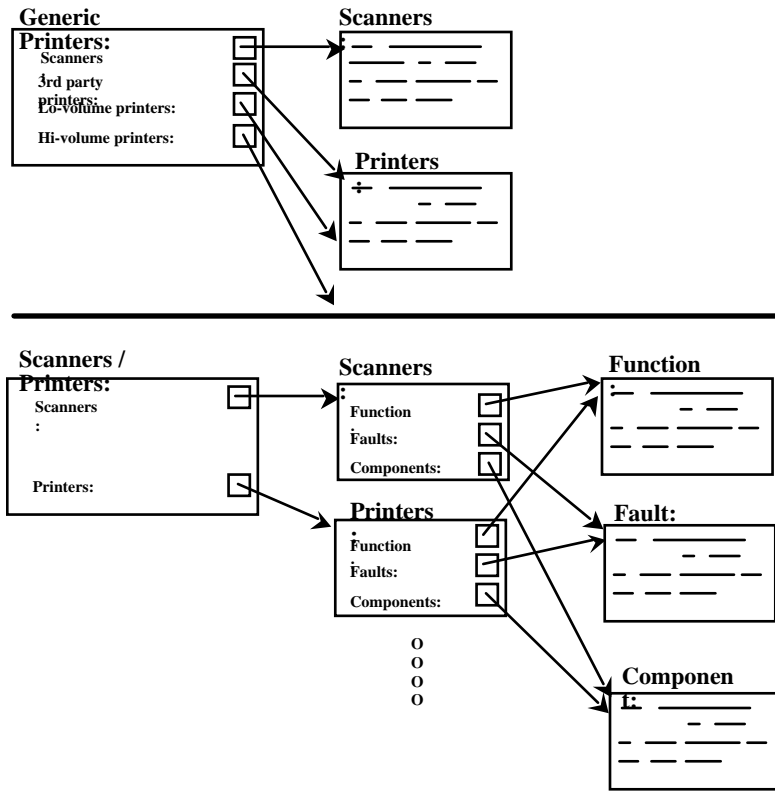
In the outlining phase, the task is to assign an ordering to the concepts to be taught in a course, based on pedagogical criteria such as presenting prerequisites for a topic before presenting a topic.

We imagine that the work group attempts to create a course outline without first identifying the concepts to be included in the course. If the concepts were not identified prior to starting the outlining process, then the work of identifying concepts would still need to be done -- perhaps repeatedly -- in the process of making the course outline.

This concludes the first part of our cost-structure analysis of laser printing case. The information work involved much more than information retrieval. It is dominated by processes for extracting data from a reluctant media (i.e., paper) and encoding the information in a semi-formal representation. Each phase of the sensemaking task refined the data to make it easier to process for the next stage. This process is efficient in the sense that if a step is omitted, the work of the next step is increased by at least that amount.

Analysis of Representational Shifts

As the lead group developed the representation to be used for the laser printer analysis, they made four major representation changes. These representation shifts occurred early in the process (Phase 1 of Figure 2). The shifts they made are shown in Figures 5.1 and 5.2. Some shifts happened when they began to understand the nature of the clustering method, others occurred as they grappled with what needed to actually be in the course under construction.



Version 1: The course is organized as a tour through a set of scanners and printers. Each device is described by a card of text.

Changes:

- * focus on printers, move scanners to secondary role (objective change)
- * add detail on functions, faults, components (couldn't answer questions with current representation)
- * remove "volume-band" category (not a useful distinction for answer procedure)

Version 2: Course now centers on descriptions of printers and scanners, with functions, faults and components for each. Function, fault and components slots are free text.

Changes:

- * drop scanners category (change objectives to printers only)
- * add substructure to fn/ fault / component (adopt new answer procedure)
- * change system class to allow hierarchical system defs (increase representation quality)

(continued -- Figure 5.2)

Figure 8.1: Representation schemas change over time to more accurately reflect what the user wants to represent, and to decrease the overall cost of use.

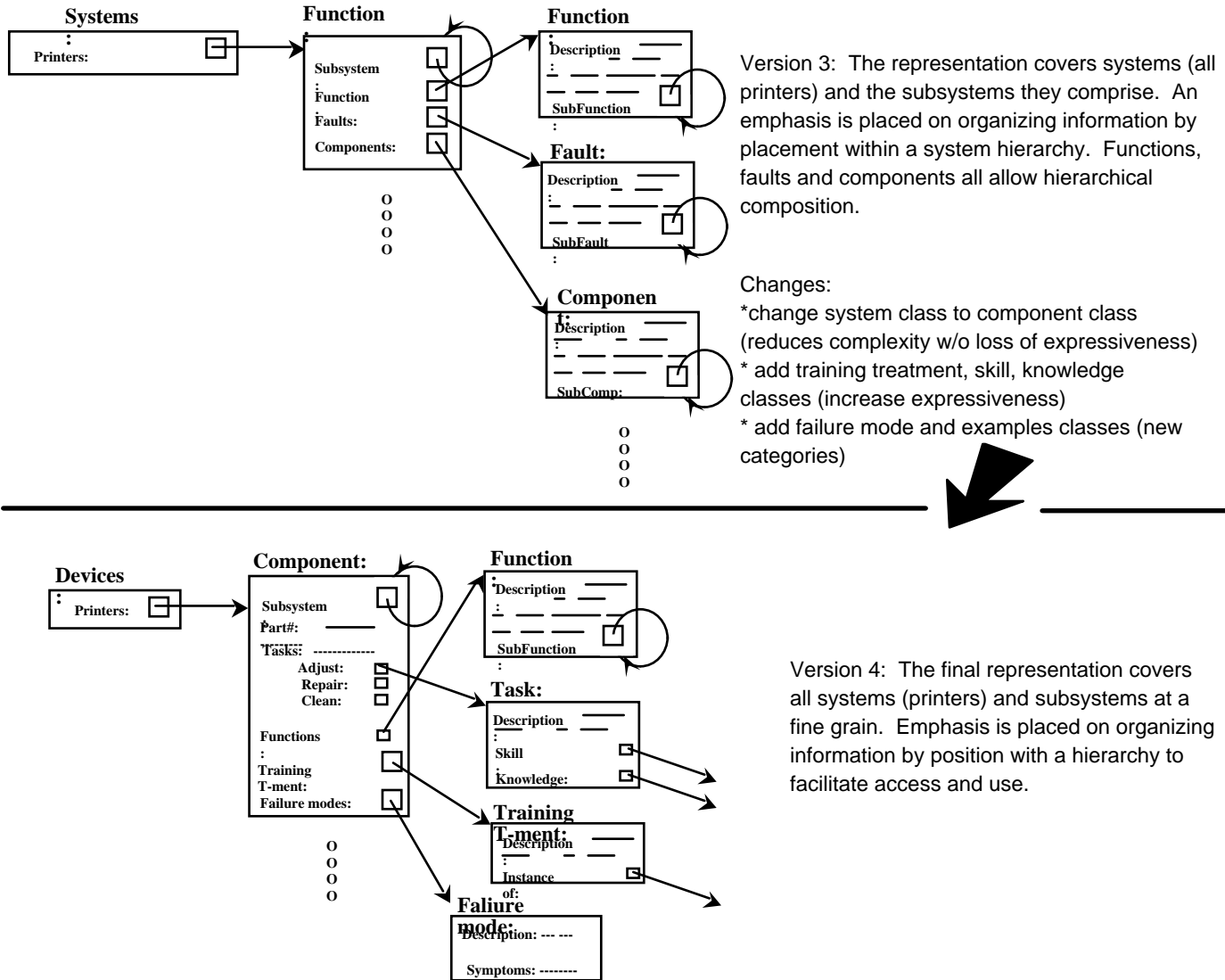


Figure 5.2 In the final form of the representation, automatic concept clustering tools could find groups of devices, components, function (etc.) that spanned machine and implementation boundaries.

The first version of the representation followed marketing groupings that had been used in previous courses. It categorized devices according to four categories: low-volume printers, high-volume printers, third party printers, and scanners. A course organized around these lines is a tour of products as presented in the marketplace. The schemas for the old course did not adequately characterize the information needed for the new one. The new training for technicians was intended to provide a framework for understanding printers according to how they worked. Marketing categories are not particularly useful because printers in different categories may work in essentially the same way, while two printers in the same category might work in completely different ways.

In this version of the representation -- Version 1 in Figure 5.1-- differences in principles of operation did not show up as discrete differences in properties. Without substructure and specification of

domain concepts such as components, mechanisms or failure modes, it was not likely that the properties would be encoded uniformly by analysts.

In the second version of the representation, the structures are expanded to include data about functions, faults, and components used in different printers. The lead group decided to drop scanners altogether as being too different from the printers. The earlier versions of the course had grouped scanners with printers along marketing lines rather than functional ones. Although version 2 made more of the relevant properties explicit, it still lacked a recursive mechanism for representing components.

Version 3 had provisions for recursive structure, but it permitted representational ambiguities between subsystems and components because it placed Function above Component in the representation. Two different encoders could reasonably choose quite different representations for a printer in terms of its functions. Such ambiguities would plague later analysis -- when printers and functions were to be clustered according to their commonalities.

Version 4 remedied this problem and provided a robust and computationally tractable representation for human and computational use during encoding and algorithmic clustering.

In summary, the representational shifts made by the lead group anticipate the rest of the work process. They prepared for the large encoding investment to be made by the work groups. When sensemakers make an investment of this magnitude, they are betting that there will be a payoff either as a reduction in the overall task time or in the improved quality of their results. Returning again to the automatic clustering of the second phase, we can see that the ultimate schemas developed by the lead group defined the terms by which the objects -- printers, fault modes, and so on -- could be compared. They provided the basis for the later extraction of course concepts.

Case Summary and Conclusions

IDE provided leverage for sensemaking in the following ways:

- provided an ability to handle large amounts of data that would otherwise prove intractable (reducing cognitive loads; discovering abstractions to improve organization)
- discovered categories that were not evident to experts. It identified structures and functional relationships not immediately apparent in the data
- allowed exploration of a larger design space by direct manipulation of structures through the browser displays
- made it simple to analyze data in context, checking the validation of automatic clustering.

In previous training courses about printers and copiers, a leader or central committee designed an outline of educational concepts ahead of time. They defined a set of forms, had analysts collect data, and organized data according to the outline. For the new course, the designers recognized that determining an outline without looking at the printer data would have discouraged or precluded the discovery of new concepts. Recognizing that they could not determine all of the educational concepts beforehand, the group leader decided that a more open-ended analysis was needed.

This recognition fundamentally changed the nature of the process. It moved the locus of responsibility for finding common terms from a preliminary process carried out by leaders to an iterative process carried out by the groups. It shifted the responsibility so that those who analyzed the data had a part in determining the framework. It admitted more open-ended possibilities for sensemaking by coupling the analysis more directly to the gathering and perception of the data. This

also led to the sequence of representational shifts over time in a search for one that would satisfy their task requirements.

CASE 2: Making Sense of Document Recognition

Background

In the summer of 1987, the administration of the Xerox Palo Alto Research Center (PARC) formed several study groups to make recommendations for changes to its research program. This activity was motivated by changes in markets, technology, and competition that affect Xerox's business and reflected possible scientific opportunities.

This case is about the sensemaking activity of one of the study groups, charged with making recommendations for a possible new research program on document recognition. The term "document recognition" was not precisely defined, but was understood vaguely to encompass computational means for extracting information and editable representations from document images.

The Sensemaking Task

The study group needed to formulate and answer some questions, such as:

- What is document recognition?
- What is the nature of a document recognition business?
- What Xerox business capabilities are relevant?
- How would ongoing Xerox business activities be affected?
- What are Xerox competitors doing in this area?
- What are Xerox's strengths in research in relevant research and technology?
- What research should Xerox do in this area, and at what scale?

The corpus used by the study group was open-ended. Information about the structure of Xerox, including its main divisions, subsidiary companies, and overseas divisions, was already available and augmented by personal contacts who could supply the latest particulars. The librarian in the study group was skilled at retrieving information about patents, companies and markets. Finally, the relevant scientific literature was accessible to the group because they had personal and corporate libraries. A major challenge of the sensemaking task was in selecting and organizing data from these very diverse and plentiful sources.

Figure 6 gives a flowchart of the activity of the study group. The group met twice a week in sessions that lasted 2-3 hours. There were three major phases in the activity. In the first phase, the group spent a week determining the goals of the report and its rough organization. At the end of this period they had a rough top-level outline. During the second and longest period, the group developed a detailed outline of the report. They spent seven weeks collecting documents, organizing the information, developing detailed questions, and creating answers to them. At the end of this period they had a 30 page outline. Finally, during the last two weeks, the group assigned sections of the outline to different members and created a 90-page report.

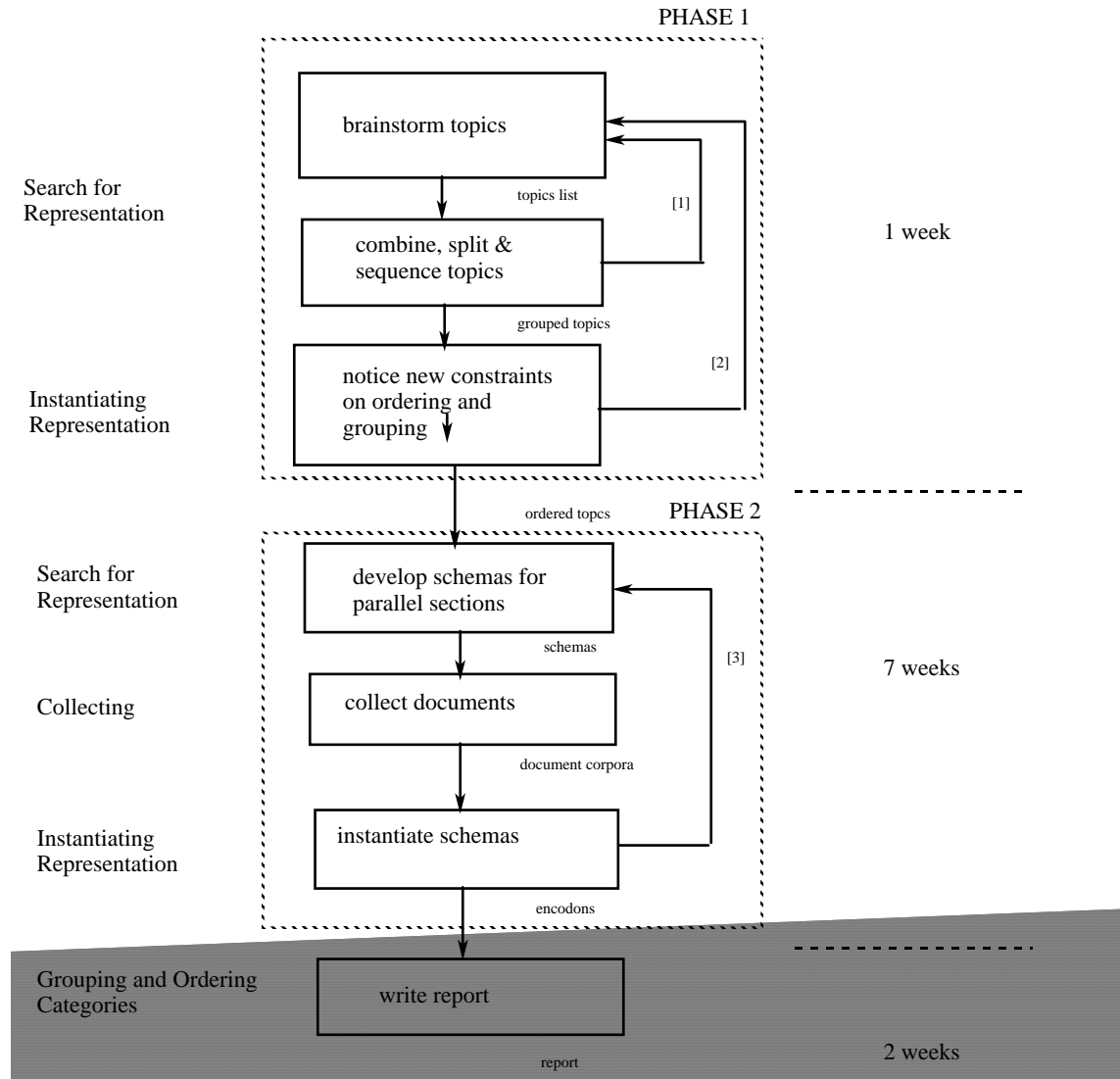


Figure 6. The activities of the study group as they used Cognoter to make sense of "document recognition" and write a report on the subject.

The study group worked in a computer-augmented meeting room. They used Cognoter, [Stefik et al., 1987] -- basically a multi-user outlining tool. The room included a blackboard-sized screen -- a *liveboard*, -- which allowed everyone to see the ongoing information work. During a session, contributions could be made by any participant and viewed and edited by everyone from their individual computer monitors. After each session, an outline of the group's work was printed out and distributed. Between sessions, individuals could create additional text that could be merged in the group's workspace at the next session.

Learning Loops

We divide the sensemaking task for the document recognition case into three phases: developing a rough outline, developing a detailed outline, and writing the report. As in the first case, each phase contains a learning loop.

Compared with the laser printer case, this case was much more top-down in its orientation. The members of the group were specialists in the different areas to be covered by the report. Thus the group included people who had firsthand knowledge about Xerox business practices and organization, products in the marketplace that used recognition technology, and the state of the art in relevant scientific and engineering areas. Consequently, the input data for the rough version of the outline that was largely developed from the memories of the participants, and served to work out the boundaries and ordering of major topics for the report. The rough outline for the report was about four pages long.

In the brainstorming step, the study group just wrote down the topics with no particular attention to ordering. During the next two steps in the figure, the sensemakers tried to impose an organization on the topics. We will discuss the actual representations and operations they used later. For now it is enough to know that in the process of trying to group and order the topics, they often found topics that were difficult to sequence or which did not fit neatly into one group or another. Loop (1) is a representation shift loop that deals with this residue by combining or splitting topics.

Because the output of the first phase was a rough outline that was then filled out by the second phase, another way to look at the first phase is as the “search for representation” of a larger process. This shifting of view is commonplace in the analysis of the learning loops of a task. It arises from looking at work at multiple levels of granularity. Generation, test, residue, and representational shift arise at all levels.

During the second phase of the process, the group systematically collected documents in order to find answers to the questions implied by the rough outline. For parallel sections of the outline, they developed schemas, where each slot in a schema corresponded to a topic to be covered. For example, the outline developed several scenarios of the use of document recognition. In Figure 9, Loop [2] is the data coverage loop in which documents for covering the topics were collected. Loop [3] is the representation shift loop responding to residue encountered in the instantiation process, leading to changes to the schemas for the sections. The generation loop for this learning loop is not shown.

Operations and Cost Structure

We now consider the representations and operations used by the study group in this sensemaking case.

Operations for Phase 1: Creating an initial representation

The group began by brainstorming topics to be covered by the report. The sensemakers used a computer tool [Stefik, et al., 87] to enter their proposed topics and to discuss the ordering. Although Cognoter was a free-form brainstorming application, and never planned as a schema development tool, the group quickly fell into a style of developing *lightweight schemas* as templates to guide their analysis.

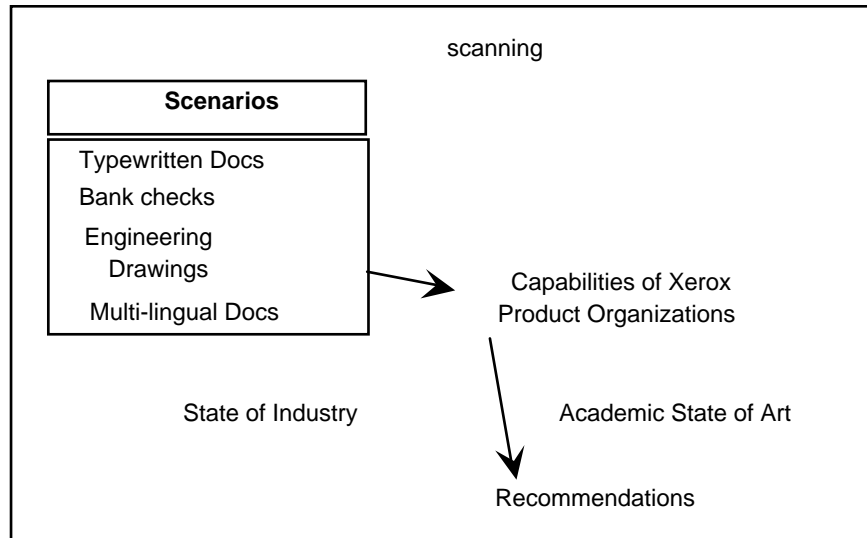


Figure 7. Arrows indicate constraints on the order of presentation. Groups of items are kept in a box. Such groupings (both boxing and spatial layouts) formed the basis for lightweight schemas that developed to help organize work.

Figure 7 shows a simplified version of the workspace as it appeared during brainstorming. The actual workspace was much larger, being projected onto the liveboard and extending onto the 21-inch monitors in front of each participant. Operations included creating an item, entering an item description, creating groups of items, and linking items.

These operations afford efficiencies for incrementally creating and ordering a set of topics. The efficiencies arise from the properties of transitivity and composition in the task. Consider the two operations for imposing organization on the items: putting items into groups and linking a pair of items into presentation order. The grouping operation creates a new level in the workspace -- replacing all of the selected items at one level of the work space by an item representing and containing a group. The linking operation represents a constraint on ordering. If item A is linked to B, this is taken to mean that A comes before B in the outline.

The efficiency of using the operations comes from two sources. First, grouping provides an abstraction in the visual display, making it possible to perform operations on many items at once. Thus, operations applied to a group act on all items in the group. A link from one group to another is interpreted as meaning that all of the elements of the first group came before all of the elements of the second group. Secondly, ordering operations are transitive. If A comes before B and B comes before C, then A comes before C.

Similar operations are now available in all outlining tools and in many commercial word processors that provide an "outline view" of the text. Very few steps are required for a sensemaker to visualize and create substantially different outlines.

Operations for Phase 2: Extending and instantiating the initial representation

Most of the time on this project was spent developing the 30-page detailed outline during the second phase. The sensemakers used the same program for this phase as for the first phase, so in a sense the

available low-level operations were the same. What distinguished this phase was a shift in the patterns of use of the tool -- so that larger aggregate operations became apparent.

The main behavioral addition in Phase 2 was the explicit adoption of repeating patterns in the lists as lightweight schemas to guide the retrieval processes and information organizing steps. For example, Figure 8 shows a list that served as a schema for sections of the report that described documentation recognition scenarios. We call such lists "schemas" because they were used as templates or patterns used to guide the development of parallel sections. They are "lightweight" to indicate they lack the type declarations on slots and computer support to enforce them. Each element in a schema is a topic or question that needs to be covered in the scenario. Schemas representing such parallelism in document structure were used for several sections in the paper. The schema structures used seemed to emerge partly because of the structuring conventions of typographical layout and partly as a spatial grouping device, as noted in [Marshall & Shipman, 93].

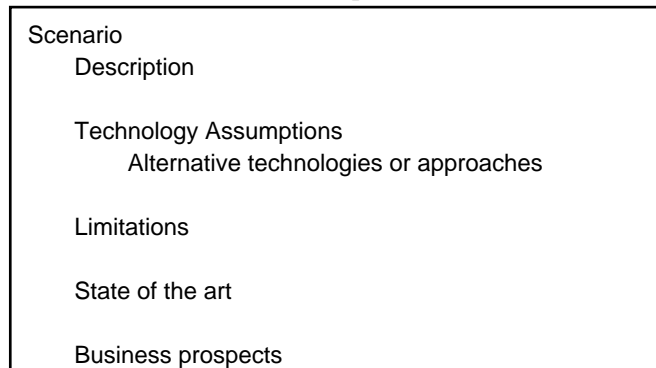


Figure 8: A list used as a schema for developing parallel sections. Indentation serves to denote hierarchically organized sections.

When the sensemakers recognized that several sections should have parallel structures, an initial schema was developed and then instantiated for each scenario, research area, or whatever. In the course of filling in instances, new topics came from working with the data and were judged important enough to include in all of the parallel sections. Residue was discovered and led to changes.

<i>Research Topics & Technologies</i>						
<i>Scenarios</i>	Marking	Scanning	Paper Handling	Theories of Document Use	Document Description Languages	Recognition Processing
Typewritten			o			
Transaction			o			
Standard forms			o			
Office Documents	o	o	Use of paper handling tech. in Office Doc. processing	o	o	o
Engineering Drawings			o			
Hand Annotations			o			
Multilingual			o			
Computer-based			o			
Multi-media			o			

Figure 9. Using a cross-product of two schemas to generate new schemas for sections of the report. This simple combination trick is the primary organizing representation for work in Phase 2 of this case. (Only one of the entries is filled in here. In the actual case, all of the slots were expanded appropriately.)

The later parts of the report needed to analyze both business and research plans. A recurring pattern of behavior in this activity was the taking of cross-products of schemas. For example, Figure 9 shows a cross-product of scenarios with technologies. This cross-product led to the creation of a table of how different technologies played a role in each of the document recognition scenarios. Similar cross-products were used to assess the strengths of various competitors, academic research organizations, and Xerox product divisions in different technologies. Cross-products were also used to catalog the interests of product divisions in different market segments. Given such tables, the sensemakers could choose to organize the information in the report in sections either by row or by column.

This table-making behavior could be characterized as “just being systematic” in working through the information. What was striking to us was the extent to which the availability of the lightweight schemas in the information management task lent itself to supporting such systematic search and analysis.

The Value and Combinatorics of Collaborative Sensemaking

In our analysis of the laser printing sensemaking case, we identified the operations and representations used by sensemakers, identified variations in their approach, and then compared the costs of the variations. In the laser printing case, there was a large but circumscribed and previously collected corpus of documents. The key in understanding the cost structure of the task is summarized by the information refinery metaphor: in their multi-step process, the data is

transformed from one form to another in a way that minimizes the costs of using it at each stage. Earlier, in their attempt to make the problem tractable in the first place, the lead analyst group searched for effective representations. Their solution was the final representation shown in Figure 8.2 and the process in which it was used, as shown in Figure 2.

For the current case, making sense of document recognition, we have already identified the operations and representations. The next step is to identify variations of the approach and determine their costs. In this case, however, the key to the analysis is not so much the refining and reduction of information as the synthesizing of it by gathering and recombination. The document recognition case differs from the laser printer case in that there was no circumscribed and available corpus. Instead, the sensemakers were faced with an open-ended corpus. The key to their work was to impose an organization on their search process, where the heuristics to guide the search were held by different people. Our analysis focuses on the main cost of their collaborative work, which was the time required for communication and coordination.

Problem-Solving by Committee

Experimental and theoretical results have shed light on the dynamics and economics of problem solving by committee [Clearwater, Huberman, & Hogg, 92]. The results were obtained by analyzing simulating large numbers of computational "agents" working on search problems. When agents collaborate, they perform better than the same group of agents solving the problems independently. In the paradigm used for the experiments, the basic search method was "generate and test." For non-cooperating agents, the solution time is governed by the density of solutions in the problem space. For cooperating agents, once an agent finds a partial solution, the solution is posted on a computational "blackboard" visible to the other agents. If agents start off at different locations in the search space, there is a much higher likelihood that some of them will find hints (in the form of partial solutions or constraints) worth communicating to other agents. Those hints can be used to reduce subsequent search. In the experimental results, the speedup was greatest when all of the agents had access to all of the hints and when the agents had the most diversity in initial starting conditions and methods of solving the problem.

Sensemakers in the document recognition workgroup had several properties in common with the agents of the theoretical setting. They were diverse: each had different areas of expertise; they represented different disciplines bearing on the problem; they had different connections into Xerox, different familiarities with document recognition products, and other relevant areas. They had to combine information from multiple sources, and to combine their expertise in synthesizing and representing the information. They worked through the outline jointly as a group.

Indirect Measurement of Committee Work

It would be satisfying to analyze the performance of the sensemakers directly within the terms of the cooperating agents model, but there are practical difficulties. For the human sensemakers, it is difficult to quantify their characteristics in terms of diversity. The task that they worked on did not have a single identifiable answer. Furthermore, much of sensemaking can be characterized as an "anytime algorithm," a procedure that trades time for quality in order to complete by specified deadline. In spite of these limitations, we can still use a post-hoc analysis based on indirect measurements to gain insight into first-order effects.

We first need to determine what needs to be explained. In the laser printing case, what needed to be explained was the value of the major cost of the case -- the 5+2 months spent encoding the laser printing data plus developing schemas for it. In the document recognition case, the major cost was the 7 weeks spent developing the detailed outline. The major payoff in the laser printing case was

the feasibility of doing a thorough job of searching for course concepts using the computer-based clustering algorithm. The major payoff in the document recognition case was the complete writing and editing of the 90 page final report in two weeks. That is an exceptional performance for a topic so technical and uncharted. Understanding the issues surrounding that, then, is the main goal of our analysis.

Two factors help to explain the speed of writing of the final report. The first is that the sections of the final report were written in parallel by separate individuals. Most of the writing was done by five people (2 members of the group did no writing), each of whom had to write only 18 pages on average. The second factor is that the detailed outline that they wrote from was 30 pages long, with all of the arguments laid out in detail at essentially a paragraph level. All that remained was the final crafting of words and formatting. Indeed, there was very little final editing or revision required of the document. Given these factors, the writing does not seem so remarkable at all. Each writer was required only to complete slightly more than one page per day from very detailed notes. But this just shifts the mystery. How did the group develop the outline so quickly?

Counting Interactions

In our analysis of the laser printing case, we sought to determine the value of intermediate steps and representations by assessing the consequences of leaving them out. In this case, this would correspond to leaving out the 7 weeks of preparation of the detailed outline, or alternatively, to consider how the detailed outline might have been prepared by individuals acting separately rather than together.

What the centralized arrangement trades against is the possibility of distributing the sensemaking task among group members. The distributed approach for doing the task has group members write drafts of their sections in parallel, exchange them for comments, and repeat through multiple drafts until completed.

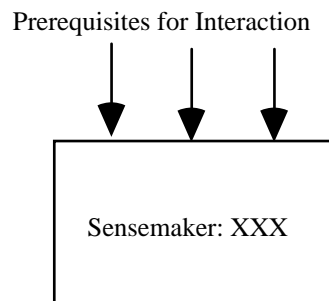


Figure 10 An interaction element. Interaction elements are those elements in the final document that required the interaction of at least one sensemaker besides the writer. Interaction elements have prerequisites for determining when the interaction can take place. These prerequisites correspond to other elements -- called triggering elements -- that must be available in the document before the second sensemaker can react and thereby engage in the interaction.

Figure 10 shows the basic unit of our comparative analysis of the distributed task: the *interaction element*. Roughly, this unit corresponds to a sentence in the final document written by one sensemaker and then modified or critiqued by another. Not all sentences in the final document had explicit input from more than one sensemaker. A content analysis of the sentences in the document based on the specialties of the participants suggests that about 20% of the sentences required reviews

and comments by more than one sensemaker. Some interactions involved one sensemaker advising another that a new element should be added to the document. These interactions are *reactive*, meaning that the critiquing sensemaker makes the suggestion *in response* to reading something in the document. If the prerequisite element is not yet in the document, then the sensemaker will not interact. These prerequisites impose a sequencing or partial ordering on the generation of elements and other interactions.

How many interaction elements are there in the final document? We employ two estimation methods. The first extrapolates from the amount of time spent in meetings during the first two phases of the actual process. Assuming that there is one interaction for every five minutes of meeting time, this leads to the estimate:

$$\text{Est. \# Interaction Elements} = 8 \text{ weeks} * 3 \text{ days/week} * 3 \text{ hrs/meeting} * 22 \text{ elements/hr} = 1584$$

The second estimate is based on a content analysis of the final report and the assumption (from above) that about 20% of the sentences were of interest to multiple sensemakers.

$$\text{Est. \#Interaction Elements} = 90 \text{ pages} * 50 \text{ sentences/page} * 20\% = 1800$$

Although neither of these estimates can be taken as definitive, they are useful for suggesting some worst case bounds in our analysis. Combining the estimates, we will use a low figure of 1500 interaction elements in the following.

The Costs for Centralized and Distributed Sensemaking

The process for the distributed approach can be characterized by the following pseudocode.

```

Repeat until finished (that is, all interactions are done)
begin
    Sensemakers write or revise the next draft of their sections in parallel.
    Sensemakers pass out copies of their document to the other sensemakers, read
    each others sections, and make written comments.
end

```

The time required for the distributed approach is determined primarily by the number of required iterations for the loop. At each iteration, assume that all interactions that could take place will take place. As each sensemaker reads the gathered documents from the others, if the triggering elements for an interaction are present in the document, then the reader responds to them by writing the comments for the author. These interactions are specific to a sensemaker. Suppose that Sensemaker-1 is the writer for a section. On one iteration, Sensemaker-2 may react to something in his section. Sensemaker-1 then incorporates that addition. On the next iteration, Sensemaker-3 may see that addition and react to it. Sensemaker-3 could not make this suggestion earlier because the “triggering element” was not previously in the document. To foster parallelism, each sensemaker works from a separate copy of the drafts. For this reason they do not see each others suggestions during the same round. Consequently, there are some chains of interaction that must go through several cycles before the interactions settle out.

The number of iterations depends on the length of these chains. Specifically, the number of iterations is the length of the longest chain of interaction elements in the final report. Variation in the length of the longest chain gives rise to variation in the time required for the distributed approach to converge.

In the best possible case, none of the chains of interaction elements is of length greater than one (not counting the first element). The group will then need only two drafts before completion. Using the times from the actual case, this suggests that the best possible time for writing the report using the distributed approach would be about four weeks, assuming three weeks for the first draft and one week for the revision cycle. This is about twice as fast as the centralized process.

Unfortunately, the expected time to task completion is quite sensitive to variations in these assumptions and the assumptions underlying the optimistic analysis are not realistic.

When concepts are developed sequentially, then interactions between them are exposed incrementally. In the absence of a detailed outline, person A may be working from an immature version of a schema. He may not have the required information to complete part of the discussion. He may not recognize some residue that arises in the discussion, or may make an inadequate adjustment to the text to account for it. In other words, the chain length of interaction elements is not likely to be so short.

Worst-case estimates are quite pessimistic. One of the estimates of a bad case would divide the 1500 total interactions among the five writers to yield 300 each. There is no reason to assume that elements and changes in one section cannot trigger changes in another, so the worst possible case could be even worse than this. Since there are now so many interactions, we assume that the time required to change a draft will be much less since it will involve just working through a subset of specific changes. Even assuming that the group meets once per day, this approach requires up to 300 days to complete or a year minus weekends or time off. This is seven times more elapsed time than the centralized approach.

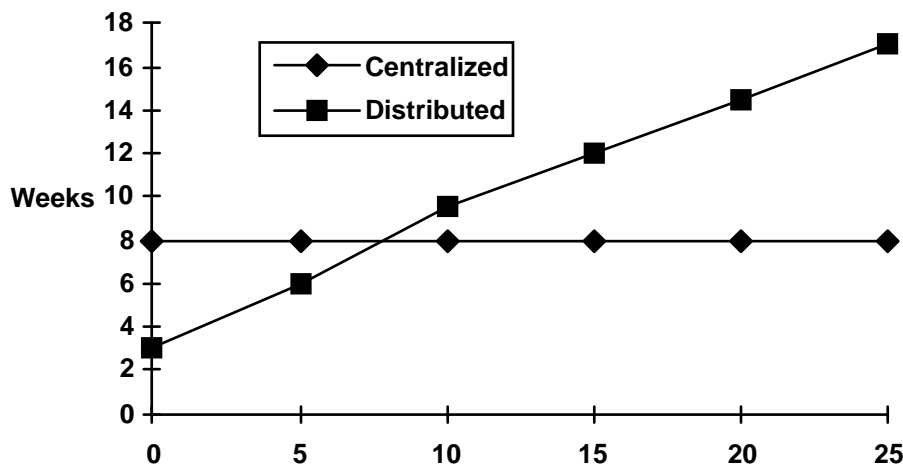


Figure 11. Time requirements for the distributed and centralized sensemaking methods. The horizontal axis gives the maximum length of an interaction chain for the task. As the length increases, the time for the centralized method stays constant but the time for the distributed version of the task increases.

Figure 11 shows the time requirements in weeks for the distributed and centralized approaches as a function of the longest chain length of interaction elements. The centralized approach depends on the number of interaction elements rather than the longest chain length so its time is constant. The time is dominated by delays for writing and meeting again to exchange documents. When there are many interactions, the participants get less done between meetings, so we assume that they will schedule them at closer intervals. In Figure 11, we assume that the first interval between meetings is

two weeks to allow for writing the first drafts, the second interval is one week, the intervals through the tenth meeting are 1/2 week, and all subsequent intervals are one day.

For the total elapsed time of the distributed approach to equal that of the centralized approach we need only have a maximum chain length of eight, allowing one week for writing each draft. However, even this case consumes more man hours than the centralized approach. In the centralized approach, the time required of the participants during the eight weeks of the first two phases was little more than the time that they spent in the group meetings. In contrast, during the writing phase their engagement was much closer to full time.

This result may seem counterintuitive, since parallelism is usually associated with more efficient ways of doing things. However, the more closely the collaborators *need* to coordinate the less potential there is for gains by parallelism. In summary, the potential gains of parallelism in the distributed approach are swamped by the overhead of coordination: delays in waiting for interactions and the time used in detailed rewriting.

Case Summary and Conclusions

This case illustrates the dynamics of collaborative sensemaking and the opportunities for supporting it with computation. The main benefit of the computational support for this was that it provided a very fast communication channel for the group -- the liveboard and monitors -- viewable and accessible by all. It required only seconds to call up a schema or to compare two parts of the outline side by side. The computational workspace made the learning loop operations fast and visible. When the group met, they were thoroughly engaged in interacting with each other on the project. By analogy with the optimization of computer programs, the fast medium for coordination sped up the "basic cycle time" of the sensemaking process when it mattered most -- when the entire group was present. For highly coordinated tasks like this, the centralized approach is more efficient because it is less effected by the accumulation of communication delays.

CASE 3: Making Sense of High School Algebra

Background

During the summer of 1990, a summer course was offered in the School of Education at Stanford University to teach students the basics of instructional design [Russell & Kelley, 91]. The course had an ambitious goal. With twelve students in the design class, each was assigned to create one section of a high school algebra course, roughly one chapter's worth. The objective of the entire class was to write a year's worth of instruction by the end of the summer course. The information system used for this task was IDE 2.0, a later version of the system used for the laser printer analysis of Case 1.

The Sensemaking Task

Figure 12 presents a flowchart of the process used by the class.

The sensemaking task has three phases. During the first phase, the instructor prepared schemas relevant to course design, based on her experience with IDE for this purpose on several previous courses. During the second phase, the students used IDE to create a design for a course in high school algebra. They used the schemas provided by the instructor and an open-ended corpus of material about algebra. The output of this phase was a set of encodons instantiating the schemas, which described the course in detail. The third phase of the task was to generate a draft of a textbook

for the course in high school algebra. This portion of the task was entirely automatic. A computer program created the draft from the detailed information in the encodons.

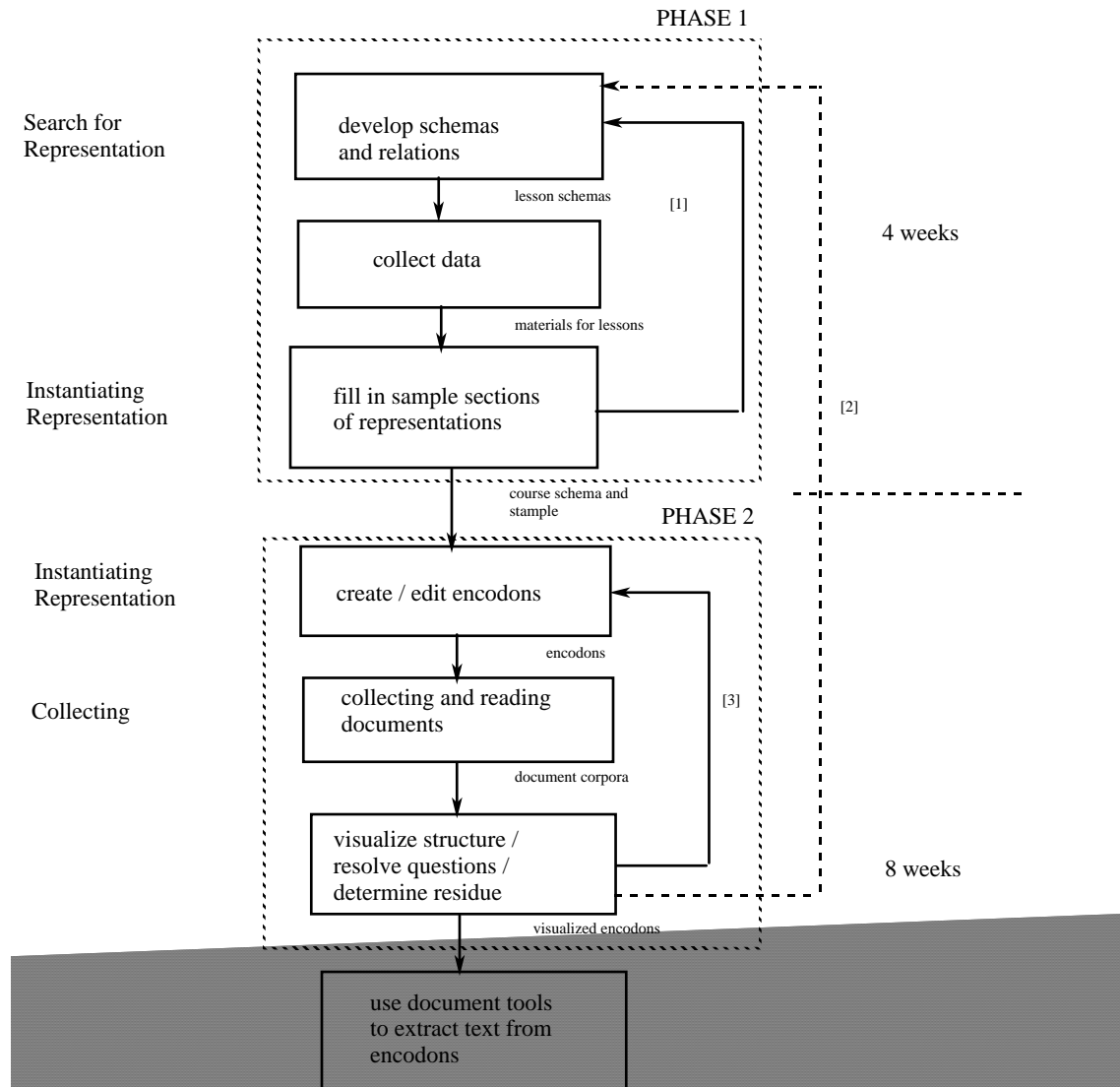


Figure 12. Overview of the process used for the Algebra sensemaking task.

The Learning Loops

As before, each phase of the course corresponds to a learning loop.

Loop [1] is a representation shift loop used by instructor in creating the initial schemas. The data coverage and schema generation loops for this phase are not shown. Loop [3] is the data coverage loop as carried out by the students in the class, as they gathered materials and instantiated the schemas. This loop was carried out by all of the students separately and in parallel. Loop [2] is the representational shift loop that would have occurred had the students been active in changing the schemas themselves. (But due to time pressures of the class, they did not take part in schema

definitions.) It is indicated as a dashed loop because little feedback was done in this case. For the most part, the representations developed by the teacher proved usable to the students in this class. The only notable exception was that the students suggested adding a slot to represent algebra “test questions.”

The main work of sensemaking in the second case involves interactions and communication among the students who are both coordinating their work and learning from each other. Understanding that work is a major part of the following analysis.

Operations and Cost Structure

We now consider the operations and representations used in this sensemaking case.

Operations for Phase 1

During the first 8-week phase of the process, the course leader developed schemas and relations for describing the course elements. The leader had recently completed a similar set of schemas using IDE 2.0 for her own use in designing another course. IDE 2.0 was similar to IDE 1.0 except that it provided automatic provisions for updating instances. Consequently, when a change was made to a schema, all of the related encodons were updated immediately.

The schema operations were essentially the same as those described for the first phase of the laser printing case. The representations were “heavyweight schemas” and the operations were appropriate for creating and changing them. There were schemas for topics, subtopics, lessons, and student assessments.

Each student was assigned a section and expected to use the information system to develop its topic as a complete set of lesson plans (approximately 20 lesson plans) including possible assessment questions, anticipated learner difficulties with the particular content, and a rationale for curriculum decisions. The class's objective was to collaborate in creating a design for a complete algebra course.

-
1. Expressions, variables, and intro to equations
 2. Negative numbers
 3. Equations and inequalities
 4. Polynomials and Factoring
 5. Exponents and Radicals
 6. Rational Expressions, Operations, and Equations
 7. First Degree Equations
 8. Relations, Functions, and Variations
 9. Complex Numbers
 10. Quadratic Equations and Functions
 11. Conic Sections
 12. Linear Systems of Equations
-

Figure 13. The initial Algebra course outline

The instructor developed the top level outline for the algebra course in Figure 13. This outline was somewhat artificial, for teaching purposes it was constrained to provide each student with a section to construct. The instructor also completed encodons for one section of the course to use as an example.

Operations for Phase 2

During the next eight weeks each student collected information about high school algebra and filled in the encodons for the assigned section. Some high-level behavior patterns included collecting documents, reminding or remembering where they were working, identifying global course requirements, updating course data, instantiating schemas, detecting residue, and negotiating changes with other students..

Students organized their lesson plans not for just a single lesson, but also with respect to a larger, year-long structure. Although each student was responsible for a single section, they were collectively responsible for the entire course. They needed to make sense of the subject matter (algebra), how the material divided into segments (lesson contents), and how all of the lessons tied together into a whole (course organization).

The operations reflect the students' shared responsibility for the overall course. The "input" to the second phase was not just the schemas from the first phase and the open-ended corpus about high school algebra. Each student also had to make sense of the encodons created by his fellow students. The databases created by the students were merged on a weekly basis so that they could be examined using browsers that were built into IDE 2.0.

During the weekly labs, the students filled in the encodons with information about their assigned topics. The schemas guided their investigations. The slots in the schemas reminded them of things to prepare in the lesson plans, and provided a structure for organizing the topics and subtopics coherently. The schemas also reminded them of relations to other sections. They discussed the sections with each other, and coordinated their work between course sections. An overview of the evolving course outline -- showing the work of all of the students -- was always available for inspection. This was an aid in discovering overlaps and gaps in the material, and a guide for establishing prerequisites. In specifying the connections between their materials, the students also had to check for uniformity of terminology.

The third phase -- printing out a rough draft of a course textbook -- was done automatically by IDE, based on the detailed encodons created in the second stage.

The Value of Heavyweight Schemas

As in the previous two cases, we have identified the operations and representations used in this sensemaking task. Next we consider the major costs and benefits of their approach, and compare it with alternatives.

The major cost in this task was the 8 week period when the students worked in parallel to create the encodons describing their parts of the course. The major benefits were that the students learned about course design and that their encodons could be used directly to print out a draft of an algebra textbook.

The "automatic" production of a 12-chapter textbook, even a rough draft, would seem to be a remarkable feat. In fact, the program did little more than traverse the information structures recorded in the encodons and print out the information in a pleasing format. The regularity of structure in the information was the key to the simplicity of the automatic processing.

As an alternative to the computer-based approach using IDE, we suppose that the instructor provided the students with printed materials -- a course outline and a completed example of a

textbook section. We now want to compare the sensemaking operations on this alternative medium. For some of the operation, such as identifying course requirements, the computer medium would have little advantage over a paper one. To understand the advantages to first-order, it is enough to consider two operations: reminding, and collecting course data.

Reminding

Ultimately, the course consisted of about 300 elements, or about 25 elements per student. Elements themselves represented from 1 to 4 pages in the text. Each element corresponds to a completed lesson plan: the lesson text, exercises, homework, possible misconceptions, and so forth.

During the period of developing the course, the students needed to keep track of the course requirements, to identify unfinished business, and to schedule their work on finishing it.

IDE supported graphical tree-structured browsers to give different views of the developing course material. For example, a student could call up a topic/subtopic browser. Such a browser would show the topic/subtopic hierarchy from any selected starting point in the course. A student could call up a browser that would show lessons and the corresponding assessments. IDE supports annotation facilities that allow a student to leave notes about unfinished business. In effect, a student could ask: Where was I? What still needs to be done in this part? What subtopics have unfinished business?

In a paper medium, much of the work of keeping track of work to be done is transferred to the student. For example, any topic/subtopic view of the encodings -- revealing both what is finished and where the gaps were -- would need to be created and maintained by hand. The effect of this transfer of work includes not only the time needed to keep records by hand, but also less time to develop the algebra materials themselves, interference with other work as the student needs to remember to update records when he makes changes, and more work to create the overview displays.

Collecting Course Data

An approximation fostered by the organization of this sensemaking task is that the students can work mostly independently. Each student had his own section to write. Although there were interactions between sections -- such as when they might need to discover and negotiate about which sections would cover some material that is a prerequisite to other material -- the course was designed to keep these interactions relatively small by the way that the initial partitioning of the algebra subject matter was divided into topics.

In this regard, the algebra sensemaking case is different from the document recognition sensemaking case. In that case, the interdisciplinary nature of the report meant that the required interactions between sensemakers on each section of the report were far greater. If the sections in the document recognition report were characterized as "highly coupled," the ones for the high school algebra case would be characterized as "loosely coupled."

There were two ways of automatically gathering information about the work of other students: queries and browsers. Queries were searches used to determine what other sections mentioned or did not mention some topic. The query procedure was general in its function in that it could seek out arbitrary combinations of words. Students invoked only about two queries each week. Browsers were more specialized, in that they were capable only of showing a very limited number of relations. For example, a browser could show the topic/subtopic relations across the integrated course database in the form of a tree. However, by selecting nodes in the browser, students could gain

access to the corresponding lesson plans, examples, and text. The students used the browsers quite frequently, typically several times an hour. The instructor explained the student use of the browsers as satisfying three needs: They gave a student an ability to see what work they were doing and how it fit into the overall work of the entire class. They enabled the students to model their work by examining other materials in the complete course. They gave the students a common vehicle for coordinating their joint work.

In a paper medium, these queries and browsings would have had a much higher cost. Each week, the students could have been given paper copies of the course materials developed by the others. A major difference is that independently-written paper documents would have no common index. To recreate the browsing facilities would have required transferring more work to the students. For example, the topic/subtopic browsers each student could create a topic/subtopic tree for his own part. Even so, further indexing work would be needed before another student could use such a tree to go directly to the pages containing the lessons for any given node. The query information facility would have been much more difficult to reproduce in paper, since the amount of indexing of "all word phrases used" would have been too labor-intensive to carry out. If the paper medium were used without indexing, the costs would have shown up as increased costs of retrieval. The material for the course ultimately required a paper document of 400-500 pages. With so much information, the time to find something without indexing would be changed from a few seconds to several minutes.

Case Summary and Conclusions

IDE had three main functions in this case: (1) the system provided formal schemas that guided the students in identifying and organizing course information; (2) IDE reduced the overhead of processing by reminding the students about unfinished business; and (3) it reduced the costs of interactions between students working on separate but related parts of the course.

The algebra course sensemaking task was loosely coupled in that the students could work mostly independently. Even so, the number of interactions greatly effected completion time. IDE greatly reduced the costs of handling these interactions. If the course had used a paper medium, either the students would have had to absorb the cost of providing graphs and indices for each other, or they would have incurred larger costs in searching for information in each other's parts.

The results of the algebra sense-making task were judged by the instructor as very successful in comparison with previous classes in course design. Traditionally, instructional design is taught by having students develop a lesson or two, perhaps teaching the lesson, and then listening to lectures about course design. Developing such large instructional units (e.g., an entire year's class plan, or a textbook) is extremely rare in university courses and usually very difficult to manage.

The computational aid to sensemaking experience had other positive effects. Students became reflective about the effects of their design decisions and grew to understand some very subtle issues. The instructor judged that not only had the students learned about the overt topic of the class (instructional design). But in addition, they also discovered a great deal about algebra. Although this is a common enough outcome in instruction design classes, in this case, the results were both broader (more different areas of algebra) and deeper than the instructor had found in earlier classes.

Conclusions

Our work seeks to develop theories for generating, comparing, and evaluating alternative designs for information systems used for sensemaking tasks. In this paper we presented learning loops as a framework for understanding the process of sensemaking, and cost structure analyses as an approach for comparing different strategies and systems for sensemaking. We now summarize our conclusions from the three cases.

Case 1: In the laser printer case, the sensemakers organized the operations and representations in their work to yield a succession of intermediate representations such that the costs for the succeeding processing stages were reduced. The main investment was in the encoding of the information. The main benefit was the opportunity for a thorough computer-supported cluster analysis, which enabled them to reliably and quickly identify course concepts. Any other approach to processing the information that made use of all of it would have taken at least as long.

Case 2: In the document recognition case, the sensemakers collected, structured, and encoded information from an open-ended corpus. A major resource of the group was its breadth of expertise for the task at hand. This was a tightly-coupled process which integrated the contributions of the different sensemakers at a very fine-grained level. The major cost was the coordination of the tightly-coupled interactions. The major benefit to their sensemaking process was that the expansion of that outline into the final report was very efficient. The computer system gave the group a shared and interactive working memory. The liveboard / Cognoter combination speeded the cycle time where it mattered the most -- during the tight and labor-intensive work of combining the sensemaker's points of view. The time required for a distributed approach to the problem was very sensitive to the length of the longest chain of serialized interactions. For so tightly-coupled a task, the time required for a distributed approach would have been several times longer than that required for the computer-supported centralized approach.

Case 3: In the algebra course design case, the sensemaking process was collaborative, but loosely coupled rather than tightly coupled. The major cost was the use by the students of the extensive heavyweight schemas to represent course information. The major benefits were greatly reduced overhead in reminding about unfinished work, and greatly reduced overhead in collecting and acting on information created by other students for related sections of the algebra course. A paper-based approach would have incurred much greater costs, either in developing indices by hand or in retrieving information.

The effective support of sensemaking activities requires another layer of support beyond that offered by traditional information retrieval mechanisms. We offer the following recommendations for creating sensemaking systems.

Provide a Seamless Workspace: The concept of an information workspace is well-established in information systems as a place where users can read, write and manipulate representations easily. Examples of workspaces in our cases here include the shared schema databases, the central electronic whiteboards of collaborating workgroups, and the user-defined cards in a hypertext database. Sensemaking tasks involve multiple learning loops so that the results of one learning loop become the corpus for the next. The information retrieval and information access mechanisms must be integrated so results can be shared. Making the workspace seamless avoids media clash.

Permit Schema Creation and Modification: To support representational work, an information system should provide operations for editing, comparing, and clustering schemas (or their functional equivalent). It should provide support for instantiating schemas and for propagating changes made in schemas to their instances. To reduce the overhead of brainstorming operations in schema generation, information systems can also provide very lightweight processes for creating and editing schemas.

Provide Visualizations and Data Manipulation Tools: In many sensemaking tasks, the search for representations can be subtle. We say that a sensemaker generates or proposes representations hypothetically, in order to see how they work out. This "seeing" metaphor is apt, in that it connects sensemaking with visualization skills. Examples of displays from the cases include the IDE browsers and cluster viewers and the Cognoter diagrams. Examples of data manipulation tools include the clustering and search tools.

Reducing the Cost of Sensemaking

Making sense of a large amount of information is critically important in today's world. In the analyses presented here, it has become clear that the sensemaking process is characterized by a cost structure imposed by the available information, its structure, its properties and the tools the sensemaker has at hand to manage that information. Instead of using computers only to produce information, people can use them to extract and re-use information from what is increasingly out there -- making sense for their work and other parts of their lives. Reducing the cost of sensemaking steps is the prerequisite to increasing our ability to handle massive amounts of complex data.

REFERENCES

- Clearwater, S.H., Huberman, B.A., Hogg, T. Cooperative Solution of Constraint Satisfaction Problems. *Science* Vol 254 , 1181, 1991.
- Hogg, T. & Huberman, B.A. "Better than the Best: The Power of Cooperation." *Lectures in Complexity*, vol 5, L. Nadel & D. L. Stein, eds. Addison-Wesley (to appear)
- Marshall, C., Shipman, F. "Searching for the missing link: Discovering implicit structure in spatial hypertext" *Hypertext '93 Conference Proc.*, Seattle, WA. (Furuta & Stotts, eds.) (to appear)
- Russell, D.M., Stefik, M.J., Pirolli, P., & Card, S.K. The cost structure of sensemaking, *Proceedings of INTERCHI*, Amsterdam, Netherlands (ACM Press) April 1993.
- Russell, D. M., Burton, R. R., Jordan, D. S., Jensen, A-M., Rogers, R., Cohen, J. R. Creating instruction with IDE: tools for instructional designers; *Intelligent Tutoring Media*, v 1, n 1, pg 3-16, 1990.
- Russell, D. M. (1988) "The IDE-Interpreter" in *Intelligent Tutoring Systems: Lessons Learned*, J. Psotka, D. Massey Jr., S. Mutter (eds), Lawrence Erlbaum Associates Inc., Hillsdale, NJ.
- Russell, D. M., Kelley, L. K. (April, 1991) "Encouraging reflection through an automated design tool: Using IDE for teacher education" *American Educational Research Association (AERA) Conference Proceedings*, Chicago, IL.
- Salton, G. Developments in automatic text retrieval. *Science*, vol 253. pp. 974-980, 30 August 1991.

Stefik, M. J., Foster, G., Bobrow, D.G., Kahn, K., Lanning, S., and Suchman, L. Beyond the chalkboard: Computer Support for Collaboration and Problem Solving in Meetings. *Communications of the ACM*, 30:1, pp. 32-47, January 1987. (Reprinted in Greif, I. (Ed.), *Computer-supported cooperative work: a book of readings*. San Mateo, California: Morgan Kaufmann Publishers, Inc., 1988, pp. 335-366.)

Tukey, J. *Exploratory Data Analysis*, Addison-Wesley, Reading, MA (1977).