The Cost Structure of Sensemaking

Daniel M. Russell, Mark J. Stefik, Peter Pirolli, Stuart K. Card
User Interface Research Area, Xerox Palo Alto Research Center
3333 Coyote Hill Rd., Palo Alto, California 94304
Phone: 415-812-4308 Email: DMRussell.parc@Xerox.com

ABSTRACT

Making sense of a body of data is a common activity in any kind of analysis. ***Sensemaking*** is the process of searching for a representation and encoding data in that representation to answer task-specific questions. Different operations during sensemaking require different cognitive and external resources. Representations are chosen and changed to reduce the cost of operations in an information processing task. The power of these representational shifts is generally under-appreciated as is the relation between sensemaking and information retrieval.

We analyze sensemaking tasks and develop a model of the cost structure of sensemaking. We discuss implications for the integrated design of user interfaces, representational tools, and information retrieval systems.

KEYWORDS: sensemaking, cost structure, representation search, representation shift, learning loop, information access.

INTRODUCTION

When confronted with problems that have large amounts of information, an often proposed solution is to improve information retrieval (IR). However, even in tasks that require much retrieval of information, speeding retrieval by itself may help very little. IR subtasks are best understood in their embedding in a larger overall task structure. The larger task often involves sensemaking, the process of encoding retrieved information to answer task-specific questions.

A person performing an information-rich task has an array of resources -- both internal cognitive resources and external resources for information storage and computation. Methods for carrying out a task can be described in terms of operations, where each operation requires particular resources. The benefit of each approach depends on how it changes the relative costs and utilities of various operations, and thereby, the time versus quality tradeoffs in the method. Collectively, these factors and tradeoffs form a cost structure guiding choices made during sensemaking. In this paper we consider the cost structures of sensemaking tasks, quantifying some of the effects of external representations and automation.

CASE STUDY: Making sense of laser printers

We present a case study to lay the groundwork both for our description of sensemaking tasks and our approach to determining their cost structures. For these purposes, this example case is representative of many others we have studied the last year.

In the summer of 1989, a team from the Xerox education division designed a new generic training course on laser printing for Xerox technicians. Prior versions of the laser printer course were pedagogical tours of kinds of printers organized around market categories such as high, low and medium copy volumes. A major goal of the new course was to decrease overall training time by unifying terminology and shifting common material from courses on specific printers to the general introductory course. The course needed to cover a wide range of laser printers including new ones manufactured by Xerox and other companies. The group decided to use IDE (the Instructional Design Environment) [6] a hypermedia knowledge-structuring tool to capture and organize information for the course. To use IDE for analysis, users create and link hypermedia nodes that are instances of a larger representation schema, as shown in Figure 1.

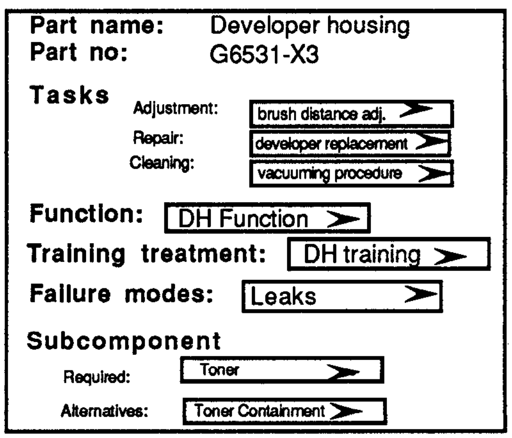

Figure 1. An instantiated IDE schema describing a printer (an encodon). Boxed values indicate links to other encodons.

In the first part of the process, the lead group identified 21 different kinds of laser printers and several different kinds of scanners to be used as source material for the course.

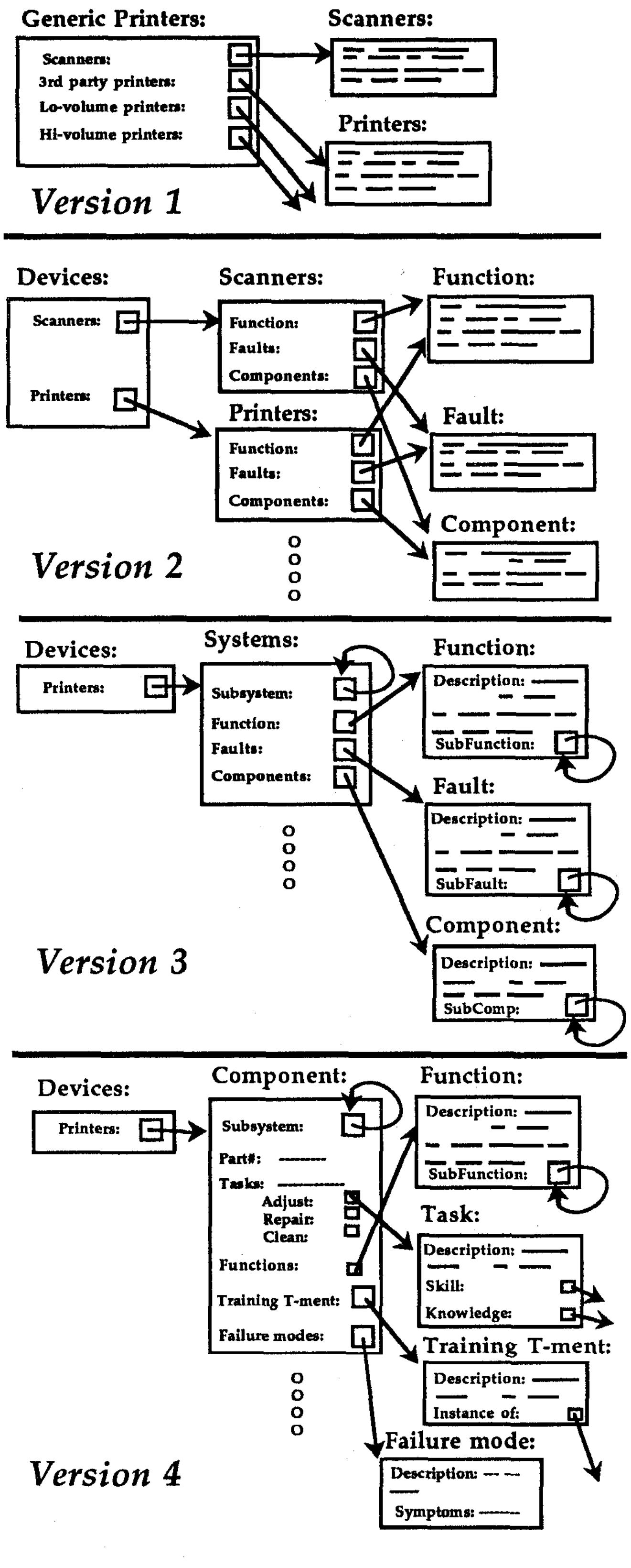

Figure 2. Representations and their associated use procedures evolve as new requirements are uncovered and properties of a representation are understood. In this case, the representation went through 3 major shifts, each time improving usability or decreasing cost-of-use. Note that each box stands for an IDE node type with a set of specified slots.

They quickly discovered that the schemas for their old course did not adequately characterize the information needed for the new one and modified their domain analyses for the new course. Ultimately, this led to a decomposition of printer descriptions in terms of subcomponents, functions, failure modes, and even techniques for training. The evolution of the IDE schemas they developed is charted in Figure 2. Figure 3 shows the process they used to produce the schemas in Figure 2 and ultimately their course design.

The final version of the schemas set the stage for the two main subtasks: understanding the similarities and differences between the various printers or their subsystems, and developing a course outline of the identified concepts. The work was divided among three work groups. During the next five months, the groups collected and entered information about their assigned printers, using the schemas from the lead group.

The next phase used computers to compare the different printers and to identify recurring concepts. An automatic clustering program took as input the schema descriptions, synonym dictionaries, and clustering parameters. It created as output clusters of closely related elements, such as clusters of related subsystems, clusters of related functions, and so on. The clustering algorithm selected an element in the database and measured a "distance" between it and other objects. The distance metric used information from the schemas such as slot names, type information, and relations, and from the text descriptions in the slots. These clusters were the basis of the concepts for the training course.

The last phase was to create an outline by organizing the concepts found in the analysis. An outline was represented by education concepts sequenced by relational links.

Learning Loops in Sensemaking

Figure 3 reveals that the team's attempt to make sense of information about laser printers consisted of cyclic processes of searching for representations and then encoding information in them to reduce the cost of operations. We call this recurring pattern in sensemaking a "learning loop complex" as illustrated in Figure 4.

The learning loop complex has three main processes:

1. *Search for representations.* The sensemaker creates representations to capture salient features of the data in a way that supports the use of the instantiated representation. This search cycle is the ***generation*** loop. Both representations and procedures for using them are created.

2. *Instantiate representations.* The sensemaker repeatedly identifies information of interest and encodes it in a representation that emerged from the generation loop. Instantiated schemas are called ***encodons*** and are created in the ***data coverage*** loop.

3. *Shift representations.* 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. ***Residue*** is ill-fitting or missing data and unused representations. 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, split, or new categories may be added. Thus, sensemaking iterates between the top-down representation instantiation and bottom-up representation search processes.

4. *Consume encodons.* The sensemaker then uses the encodons in some task-specific information processing step. In 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.

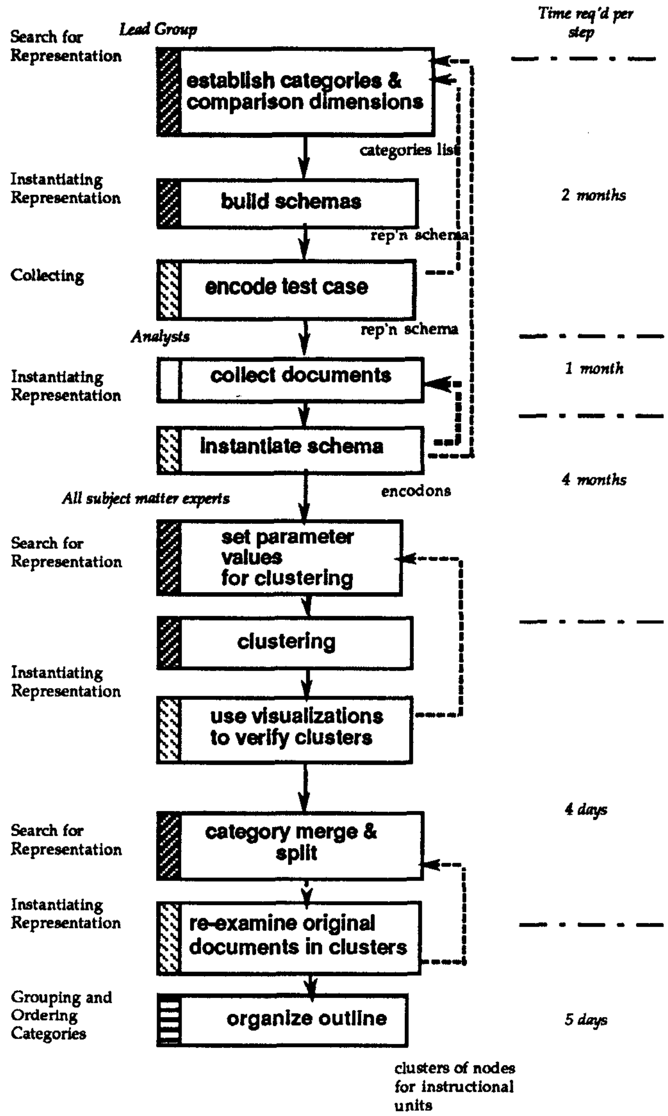

Figure 3. This diagram shows the steps the analysts went through to make sense of laser printers They collected and organized information, then found clusters of common terms, ideas, subsystems and functions in data describing twenty-one different laser printers.

But representation search is not simply top-down. If there were no surprises in creating encodons, sensemaking would be trivial; merely define the schemas and then instantiate them. Sensemaking seldom works this way. Schemas must be revised when there are surprises creating encodons, or as new task requirements come to light.

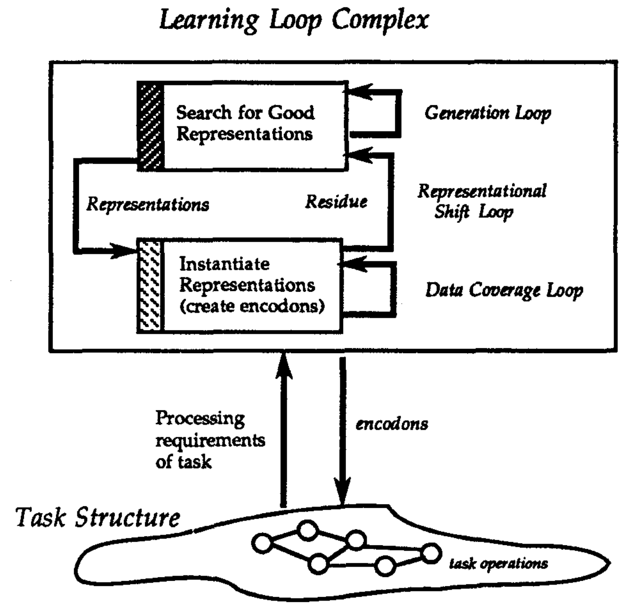

Figure 4. Sensemaking is finding a representation that organizes information to reduce the cost of an operation in an information task. The product of the learning loop is the representation and encodon set

Ubiquity of Learning Loops

Is this learning loop pattern peculiar to the case at hand or is it widespread in information-rich tasks? Over the past year we have carried out several studies including four retrospective studies of information systems for sensemaking [6] and several field studies of information workers at Xerox and other companies in the Bay Area.

Figure 5 shows process sensemaking process flow diagrams for four of these studies. The first of these is the laser printer case from Figure 3. The second flow diagram maps the activities of a study group over a ten week period creating a report about the research potential for a new technology. The third diagram maps the activity of a group of students over eight weeks in an instructional design class developing detailed plans for teaching high school algebra. And the fourth flow diagram describes the activity of a business analyst who writes monthly newsletters analyzing high technology areas.

In each case, the same basic operations and cyclic processes describe the activities used to make sense of the information. For these four cases (and for all other studies we have made) we find that each is an expansion of the learning loop complex. Despite differences in domain, approaches or individual styles, making sense of a complex body of information always appears to follow a common pattern.

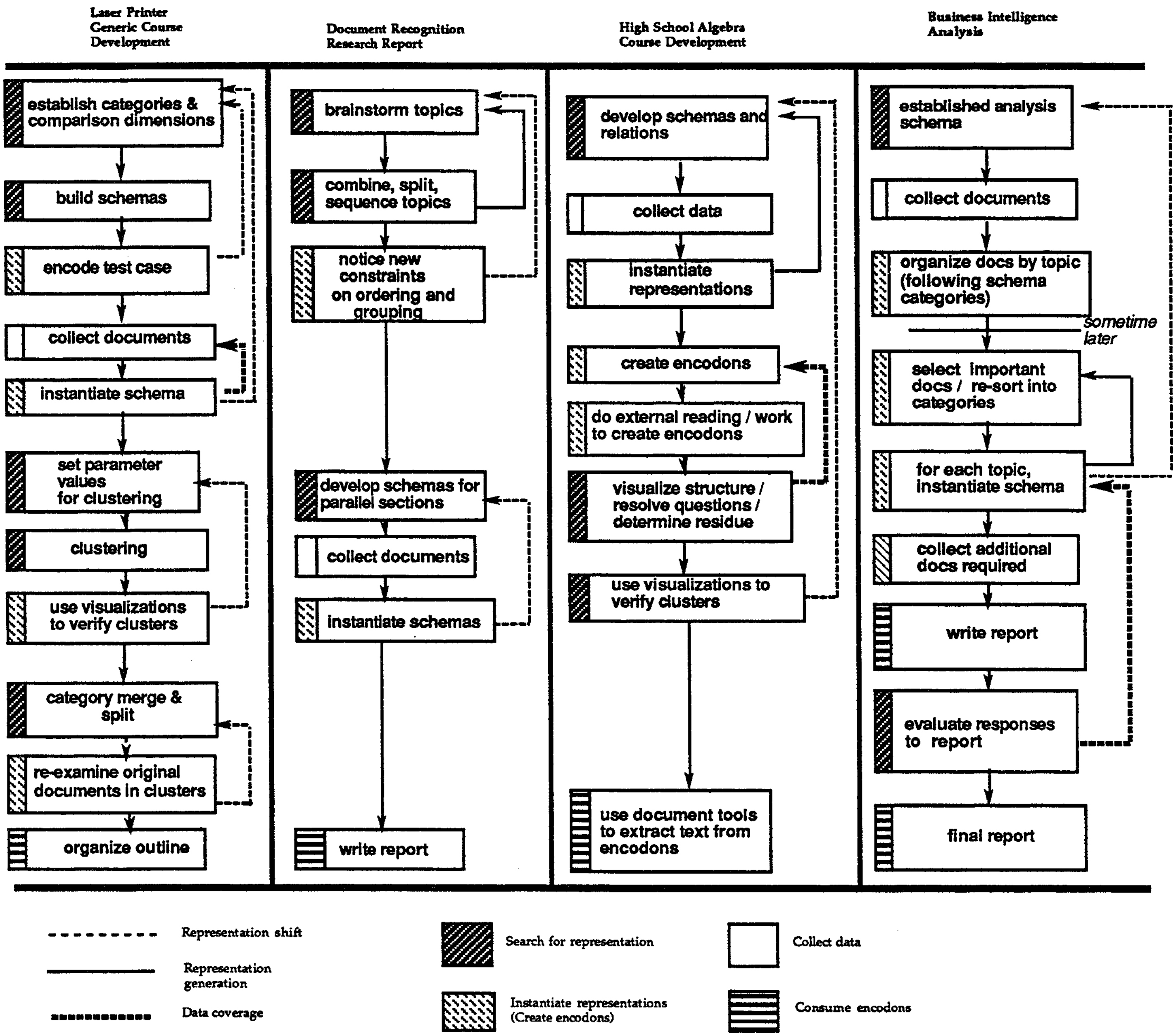

Figure 5. Sensemaking in four different information-rich tasks..

ANALYZING PRINTER SENSEMAKING

Even when different representations are equivalent from an information-content point of view, they can differ radically in the cost of carrying out particular operations. It is well known that the form of a representation profoundly affects the effectiveness of search during problem-solving. [4, 9]

It is not surprising then, that in a sensemaking task of any significant complexity, external representations are used to support the cognitive demands made on the user. However, creating a representation is a significant problem. The first representation created is rarely the best or correct one. In the following analysis, we show that sensemakers change representations either to reduce the time of the overall task, or to improve a cost versus quality tradeoff. In this analysis of the cost-effects of different representations, we consider two subtasks from our printer case -- (1) finding the central course concepts through cluster analysis (the middle boxes labeled "4 days" in Figure 3) and (2) ordering them under pedagogical sequencing constraints (the organize outline box at the bottom of Figure 3).

Example 1: Lowering the Cost of Clustering

The analysts in the printer case used an IDE operator to cluster data about printers in order to discover a central set of key concepts in a large body of data. However, finding the central concepts in a large corpus is a non-trivial task. Clustering methods always define a distance computation 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 has a value. (For example, the properties for document handlers would include the set of acceptable paper sizes, the capacity, a category such as "recirculating," and a description of the mechanism.) Given two objects we assume that it is possible to compare their property values. This 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 typical educational analyses, this process is informally carried out, primarily because the cost of careful computation (and re-computation!) is exorbitant.

The Main Cost: Data Extraction

Extracting data from documents is a key subtask in clustering and encoding. Extraction requires finding the relevant documents containing the information, selecting the document parts containing the information, and then transforming the information into a canonical form. The document parts may be particular paragraphs, table entries, or graphical elements from figures. In the cases we have examined, data extraction is often the most time-consuming task in sensemaking.

In this case study, the analysts extracted data to perform their encodings from a variety of source documents including training manuals, service documentation, engineering documents and customer support manuals. Each printer was documented by 10 - 20 documents, ranging in size from 30 - 300 pages. Virtually none of them were indexed for such access. Thus, each printer was documented with 300 - 6,000 pages. The lead group spent two months designing the IDE schemas, instantiating them to evaluate their usefulness. Then the three analyst groups spent five months extracting data and encoding it using the schemas. They created 21 computer databases, one per printer, each containing between 300-1000 entries each. In approximate numbers, each group processed about 190 pages per day, encoding on the average of 30 individual schemas each working day.

For this task, data extraction and encoding required over 75% of the total time from beginning the project to final course outline.

The Main Gain: Automated Clustering

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.

Consider the sequence of representation shifts of Figure 2 in light of this representational investment. These shifts took place in the representation shift loop labeled 1 in Figure 3. In version 1 of the representation in Figure 2, differences in principles of operation did not show up as discrete differences in properties. Since the schemas had no substructure and no specification of domain concepts such as components, mechanisms or failure modes, it was not likely that the properties would be encoded uniformly even if they were encoded at all. For this version the extraction processes for each property would require sophistication beyond the reach of automation. Version 2 made more of the relevant properties explicit, but still lacked a recursive mechanism for representing component substructure. Version 3 had provisions for recursive structure, but permitted representational ambiguities between subsystems and components. Two different encoders could reasonably choose quite different representations for a printer -- thus complicating the canonicalization phase of property extraction. Version 4 remedied this problem and provided a robust and computationally tractable representation for both human use during encoding and algorithmic clustering.

In summary, this illustrates how shifts in representation made large changes in the cost structure of the clustering task.

Anytime Algorithms

An important observation about many sensemaking methods is that they are ***anytime*** algorithms. Anytime methods provide the best solution that they can find in given limited time. Given more resources, they continue to search for better solutions. We call the common use of such tradeoffs in information processing tasks the ***anytime principle***.

A key parameter governing the complexity of clustering is the number of iterations for altering property weights and searching for appropriate cluster patterns. In this case study, automating the clustering process drastically reduced the cost per iteration, enabling the course designers to fine tune the parameters over many iterations and to search more thoroughly and systematically for the printer concepts to be included in the general course.

Another key parameter is the number of properties considered per object. The more properties considered, the more costly the encoding and comparison processes, but also, the more accurate the characterization of concepts. For the new course, the designers were able to consider many more (2 - 3 times) properties than before, using time gains from automation to extend the comprehensiveness of search.

Example 2: Lowering the Cost of Ordering

Another important task in course design is organizing the concepts into a teachable sequence -- the last box in Figure 3. This task is governed by pedagogical rules. For example, presentation of the prerequisites for a topic should precede presentation of the topic and the structure of the course should help students identify the main points through progressive disclosure.

Figure 6 shows a boxes-and-arrows representation of a partial ordering of topics for a course. This is a variation on the representation observed in similar cases. A box around a set of topics (shown simply as letters in the figure) indicates a set, and an arrow starting from a topic or set to another represents an order constraint. In the figure, topics *a* and *d* are candidate starting nodes. Since topic *a* comes before the set *t*, it comes before the elements of that set: topics *c*, *r*, and *u*. By transitivity, it also comes before the elements of the *v* set and the *y* set.

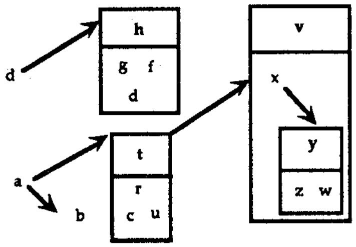

Figure 6. A graph showing partial ordering of constraints for presentation. In this example, 5 explicit ordering constraints and 4 grouping constraints suffice to order 16 nodes.

A direct manipulation interface provides an active representation for rapidly exploring alternative outlines. With computational support for the constraint reasoning, many alternative orderings of topics can be explored in a few minutes [7]. In the final phase of course development, a sequence exploration tool of this type was used, allowing a much larger space of possible sequences to be explored. The analysts estimated that they explored approximately 10 times more topic sequences than in earlier courses without such a tool, and a perceptible gain in the quality of the final course design. Again, a shift in external representation of the information resulted in a shift in the cost structure of the task.

THE COST STRUCTURE OF SENSEMAKING

From the standpoint of systems designed for sensemaking, we are concerned with evaluating different approaches to sensemaking, particularly when the introduction of alternative technologies can dramatically change the efficiency or effectiveness of the overall process. That is, we are concerned with an analysis of the steps as defined in the learning loop complex. As a way to analyze these costs, we begin by defining the following cost terms:

- *FR*: finding a representation schema to support the required operators in the target task,
- *IE*: instantiating the encodons,
- *FD*: finding data to create the encodons, including both finding the documents and selecting the information,
- *TT*: the target task.

The costs of sensemaking are the combined costs of the steps in the learning loop complex. The total cost, C_T, is the cost of sense making, C_{SM}, plus the cost of the target task, C_{TT},

$$C_T = C_{SM} + C_{TT} \qquad (1)$$

where

$$C_{SM} = C_{FR} + C_{IE} + C_{FD} \qquad (2)$$

In the following, we define ***gain*** as the increase in quality or quantity of work performed by using a particular method. Optimizing a sensemaking system is searching through the space of combinations of methods to maximize the expected gain to cost ratio.

The task of sensemaking or its corresponding target task may be either recurring or happen once-only (one-off tasks). The laser printer case is one-off sense-making with a one-off target task of producing a course outline. The business intelligence example has one-off sensemaking (the production of schemas) with the continuous target task of encoding new intelligence items and writing newsletters. Each type of sensemaking task has different optimization characteristics.

One-off Tasks

In one-off tasks the sensemaker chooses methods to maximize the expected gain within a cost or time limit.

Typically, the one-off optimization problem can be greatly simplified. In the laser printer case, the gain is the number of common instructional elements placed into the general course. We compare two methods by associating a gain function g_m with the manual method of producing the course and g_c with the computer-based method for finding concept clusters and ordering concepts. For small investments in sensemaking (especially t_{FR}, t_{IE} and t_{FD}) the manual method is more cost-effective than the computer-based one, but for large investments in sensemaking the computer-based method is more profitable. Figure 7 shows that the computer-based strategy should only be chosen if the deadline is after t_c, the crossover point.

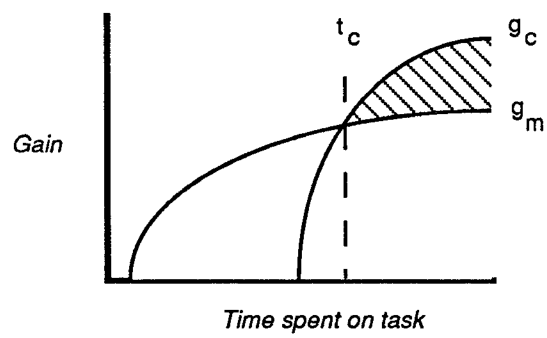

Figure 7. Tradeoffs in the gain achieved by a manual method (gm) or computer-based method (gc). The best solution is determined by maximizing gain while minimizing total cost within the deadline time.

Recurring Tasks

For recurring sensemaking subtasks the optimization problem is to maximize long-term rate of gain over many task cycles. This occurs in the jobs of business and financial analysts who scan, select, and integrate information from new reports to produce their own highly standardized periodic newsletters.

We can think of knowledge workers as organisms in an information ecology, roaming among information sources. Their problem is to choose their information sources to obtain the best gain for time spent. This metaphor leads to modeling recurring sensemaking with concepts from foraging theory. The following assumptions are often used in such analyses:

- The sensemaking subtasks are encountered and processed exclusive of one another. For instance, tasks are often structured such that one cannot search for data at the same time one is instantiating encodons, and one usually cannot create multiple encodons simultaneously.
- The subtasks come in k subtypes. Such task typing can often be performed on the basis of the kinds of data being processed. For instance, a business intelligence analyst may find certain periodicals or sections of periodicals (e.g., research news) more profitable than others. Often it is assumed that each subtype is encountered over a given unit of time as a Poisson process with parameter l_i, $0 < i \leq k$. That is, different types of periodicals arrive at the sensemaker's desk at varying rates. As the periodicals arrive, they demand different subtasks for processing.
- Associated with each subtype i of the sensemaking subtasks is a gain function $g_i(t_i)$, which indicates the expected gain as a function of the time t_i spent on that subtask. The gain function eventually show diminishing returns over time.

When these conditions hold, then can we maximize the long-term rate of gain, R, by determining the amount of time t_i to spend in each subtask. To do this, one ranks the subtask types by their expected profitabilities. The rate of gain for subtask i is:

$$\frac{g_i(t_i^*)}{t_i^*} = \max_{t_i}\left[\frac{g_i(t_i)}{t_i}\right] \tag{3}$$

The sensemaker keeps a list of the subtasks in order of descending profitability by computing each subtask's long-term rate of gain.

$$R = \frac{\sum_{i=1}^{j} \lambda_i g_i(\hat{t}_{ij})}{1 + \sum_{i=i}^{j} \lambda_i \hat{t}_{ij}} > \frac{g_{j+i}(t_{j+1}^*)}{t_{j+1}^*} \tag{4}$$

The $\hat{t}_{ij}$ are the rate-maximizing times for the i_{th} task when the subtasks to perform (pursuit list) includes items through rank j. Choosing which subtasks to discard from this list can be calculated according to a marginal value theorem formulated by Charnov [3]. Stephens and Krebs [8] discuss the implications of Equations 3 and 4, which model an agent foraging through an open ecology, and discuss on the underlying constraints and assumptions.

If variations the model of Equations 3 and 4 hold, then uniform improvements in access costs to the types of data on the pursuit list mean that selection should become even more restrictive (assuming no additional constraints on which data to process). For instance, suppose that a new information access technology increases the rate of delivery of both high- and low-quality periodicals arriving at an analyst's desktop. The optimal strategy in the face of uniform increases in the rate of encounter with data is to exclude more low-quality periodicals, rather than processing all newly available periodicals.

The assumptions of the foraging model do not hold for all cases we have considered. For example, the first assumption that only one task can be processed at a time breaks down when there is more than one sensemaker. This is especially important in the case where some of the sensemaking agents are computational and therefore operate on a different cost basis.

In addition, there are interaction effects in sensemaking not covered by this model. For example, in sensemaking tasks requiring several kinds of data, the evaluation of gain for different subtasks can vary as some data requirements become satisfied. This is a simple example of a more general phenomena in which part of the job of sensemaking is to establish the goals of the task.

TECHNOLOGY AND SENSEMAKING

There are many possible ways that technology can affect sensemaking, and we can only consider a few here. By the anytime principle, a reduction of cost (or increase in gain) associated with a step frees time to invest in other steps. For instance, if a representation is supplied at the beginning of a task, then C_{FR} is zero, and more effort may be placed into other areas (e.g., increasing the amount of information).

In the laser printer case and several others that we have observed, most of the time in sensemaking is in data extraction. This focuses attention on reducing costs of the three steps of data extraction -- finding the relevant documents, selecting the information, and transforming the information into canonical form. In one of our case studies, the sensemaker looks up data about laptop computers in a collection of magazines and product sheets. His goal is to make a purchasing recommendation meeting given constraints. The data representation created by sensemakers carrying out this task invariably includes tables giving properties of competing laptops. Representation shifts are changes to the table structure as the sensemaker decides which properties are most relevant and retrievable and ultimately are able to help solve the problem of determining the best choice.

The laptop case lends itself both to proposing technology for the sensemaker and to estimating the benefits of using it. Consider the moment when the sensemaker has filled in the name of the laptop for which he is seeking information, and needs data for the memory size column. It is clear that the next data sought is the memory size of that laptop and that the sensemaker's next activity will be to extract that from the document.

For online documents, automatic specialized methods could use encodons to generate queries for retrieving document fragments likely to contain the needed information, displaying the relevant parts, and canonicalizing the data to the form needed. For paper documents, a handheld scanner might be coupled with specialized extraction methods to reduce encoding time. In both cases, the savings make time for other activities, such as using visualization tools to survey the effects of different choice criteria.

CONCLUSIONS

This paper presents preliminary steps in understanding the cost structure of sensemaking and the role of sensemaking in various information processing tasks. The goal is to provide widely applicable predictive models of task performance and prescriptive models for recommending system features.

The ideas that information retrieval is part of a larger process of information use, that computer systems should amplify information-based work processes, and that information has costs associated with search and access have been reported earlier [e.g., 1,2]. What is new in our current analysis is the emphasis on representation use and shifts in analyzing task performance, the identification of the learning loop, the anytime principle and the characterization of sensemaking.

Representation design is central to the sensemaking enterprise. The learning loop complex crystallizes the pattern of activity in which representations are changed to reduce the cost of task operations, changing the sensemaking cost and gain structures.

By characterizing sensemaking as an interlocking set of different types of subtasks, we show how tradeoffs can be made in one place for gains in another (the anytime principle). The relative sensitivity of different parameters in the sensemaking process suggests places where changing methods or representations can produce payoffs. Our studies in information visualization methods indicate the extreme effects -- both positive and negative -- that different visualizations can have on users. In order to design information systems, we need to understand the most important factors that determine a user's ability to interact-with and perceive information. The studies presented here show that focussing on a single aspect of the problem while ignoring the entire process may be shortsighted.

This picture is still incomplete with many interaction effects still unaccounted. Nevertheless, this model provides a beginning of a comprehensive approach to the integrated design of user interfaces, information retrieval and representation using systems.

REFERENCES

1. Card, S.K., Robertson, G.G., Mackinlay, J.D., The Information Visualizer (1991), *CHI '91 Conference Proceedings*, New Orleans, 181-188.
2. Card, S.K. (1989) Information Workspaces, *Friend21 International Symposium on Next Generation Human Interface Technologies*, Tokyo, 5-A:1-9.
3. Charnov, E. L. (1976). Optimal foraging: The marginal value theorem. *Theoretical Population Biology, 9*, 129-136.
4. Larkin, J., Simon, H. A. (1987) "Why a picture is (sometimes) worth ten thousand words" *Cognitive Science*, 11, 65-99.
5. Hancock, C., Kaput, J.J. (1990) Computerized tools and the process of data modeling. 14th International Conference on the Psychology of Mathematics Education.
6. Russell, D. M., Stefik, M. J. (1992) "Beyond information retrieval: Sensemaking:" Xerox PARC ISTL Technical Report.
7. Stefik, M., Foster, G., Bobrow, D.G., Kahn, K., Lanning, S., & Suchman, L. (1987) Beyond the Chalkboard: Computer Support for Collaboration and Problem Solving in Meetings, *Communications of the ACM*, 30:1, 32-47,.
8. Stephens, D. W., & Krebs, J. R. (1986). *Foraging theory* Princeton University Press, Princeton, NJ.
9. van Baalen, J. (1992) Automated design of specialized representations, *Artificial Intelligence*, 54, 121-198.