

# Two Years before the Mist: Experiences with Aquanet

Catherine C. Marshall  
Xerox Palo Alto Research Center  
3333 Coyote Hill Rd., Palo Alto, CA 94304, USA  
marshall@parc.xerox.com

Russell A. Rogers  
DownEast Technology, 15 Lower Main Street, Belfast, ME 04915, USA

## Abstract

Aquanet is a collaborative hypertext tool that combines elements of frame-based knowledge representation and graphical presentation. In this paper, we examine the first major application of the tool in an analysis task, a two year long technology assessment that resulted in almost 2000 nodes and more than 20 representational types. First, we cover the implications of the representational resources provided and representational decisions that were made. Then we discuss how spatial layout was used in lieu of the complex relations Aquanet's data model supports. Finally, we show how distinct regions emerged to reflect particular activities and how they were subsequently used as the basis for a later collaboration on a similar task.

## 1 Overview: The tool and the task

Aquanet is a collaborative hypertext tool designed to meet the needs of knowledge structuring tasks like analysis and argumentation. The tool combines elements of frame-based knowledge representation and graphical presentation; it has its roots in gIBIS [4], Germ [3], NoteCards [5], and IDE [15]. This paper examines how Aquanet was used in a large application, a multi-year assessment of a specific area of research and technology development. This type of analysis is common in business intelligence and other kinds of information-intensive interpretive processes.

### 1.1 Hypertext to keep your knowledge in place

Aquanet enables people to describe a domain of discourse in terms of *basic objects* and how they are interrelated (*relations*). For example, in an evaluation of a computer technology, one type of basic object might be a "system," and another type might be a "developer." These two elements might be connected by an "implemented" relation. Instances of these types can then be manipulated in a shared information space. The structures that may be built within any particular information space are specified by a *schema*, a collection of Aquanet basic object and relation types. [10] presents a more extensive description of the tool and its data model.

Figure 1 shows a limited view of a much larger Aquanet information space. A user can zoom or scroll this view to see more of the space, which can extend in all directions. Objects shown in the primary view are drawn according to a user-specified *graphic appearance* that is associated with each type of basic object and relation. For example, the selection - an instance of a System object - is portrayed on the computer screen as purple rectangle that displays the object's name, which in this case is "Atlas." As is apparent in the figure, objects may overlap; users can manipulate the stacking order to see obscured objects.

In Figure 1, the view onto the information space is partially obscured by a separate scrollable window that shows the selected System's internal structure, its *slots*. In this case, the slots "Name," "Commercial?," "Language-pairs," and "Performance Data" and their values are visible in the smaller window.

Like many other spatially-oriented systems, the Aquanet information space can contain multiple references to the same object. We call these references *virtual copies*. All of the copies share the same graphic appearance, and when one is selected, all visible copies are highlighted. This mechanism becomes important in our later discussion of informal relations.

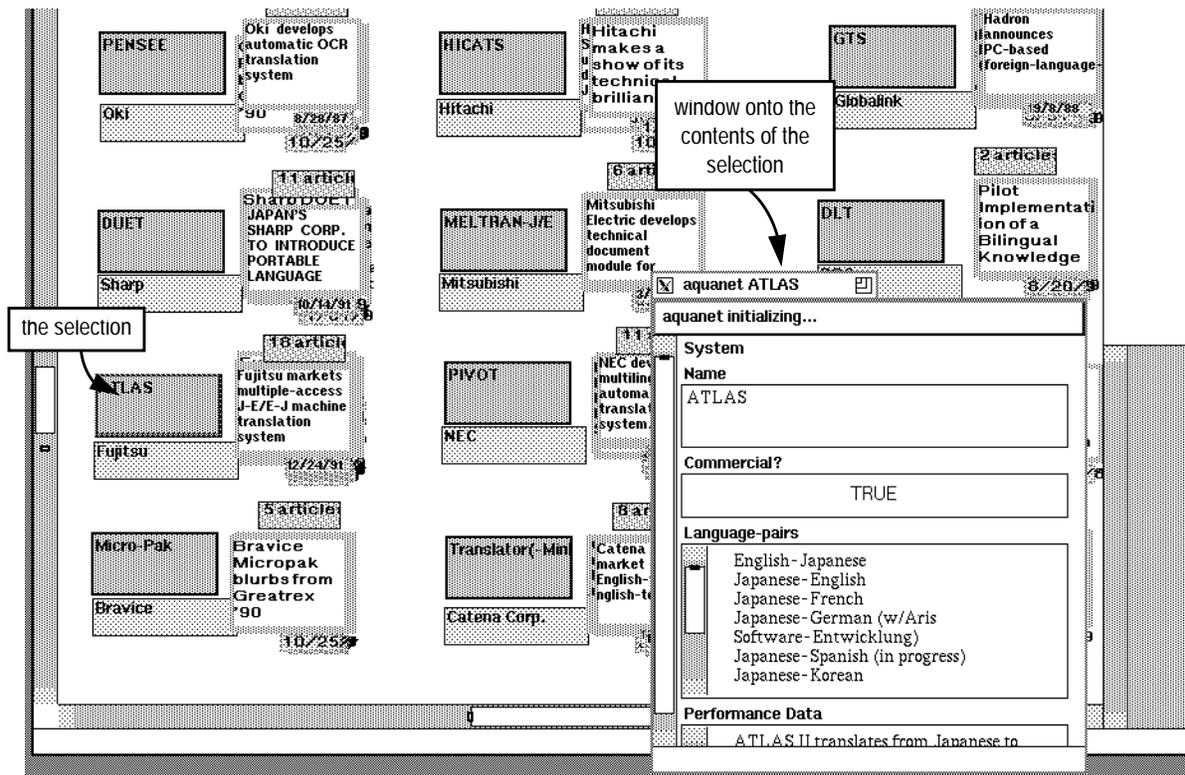


Figure 1. A portion of an Aquanet main view with a separate window onto the contents of a node.

## 1.2 The analysis

So far, Aquanet has been used for a number of diverse tasks by a small user community at Xerox PARC; early use has given us some additional insight about knowledge structuring tasks, and how well the tool addresses the requirements of people engaged in such tasks. In particular, we are interested in how the tool can be used to organize information to help people form multiple interpretations of complex domains.

To highlight important aspects of how work practice and the Aquanet tool have interacted, we will focus on the most extensive analytic use so far, a two year long technology assessment of current machine translation efforts; this task began in August, 1990. Like many long-term analytic tasks, the goal was very general and vague. Early notes characterize the assignment as "machine translation - what should we know about it?" And like many analytic tasks, we worked on collecting and interpreting information off and on; often we had to pick up where we had left off several months earlier.

To initiate the task, we began collecting every kind of information we could: books on the topic, research papers from conference proceedings, news-

wires from on-line databases, and informal commentary from experts in the field. Over the course of two years, we amassed a large, amorphous collection of on-line and hardcopy information, generated by researchers, marketers, journalists, and other people interested in the field. Gradually, we constructed an extensive structure based on the information we collected (although only a small fraction of the hardcopy information was scanned and incorporated in the on-line structure; we referred to much of the rest of it in its hardcopy form). We created almost 2000 hypertext objects of twenty visually distinct Aquanet types in the process, and organized and reorganized the objects in a complicated spatial layout.

In this paper, we will examine three important issues arising from our machine translation technology assessment. First, we will look at Aquanet's representational resources, and the decisions they required us to make; this will enable us to reflect on the task of schematization and articulating structure that is necessary for defining and using representations. Second, we will examine how the structure of the domain was realized in an information space; this will allow us to question how well a specific aspect of

Aquanet’s data model, complex relations, worked out in practice. Finally, we will examine the spatial layout that resulted from the analysis to illustrate how distinct, activity-related regions emerged, and the impact they had on a later collaboration.

## 2 Starting out: representational resources and representational decisions

Aquanet requires that every information space have an underlying collection of types that describes what kinds of basic objects and relations can be created (a schema), what kinds of internal structure each type has (in the form of primitive and entity-valued slots), and how larger structure can be built up (by type restrictions on the contents of entity-valued slots). Although this set of types may be developed over time, at the outset of a task, a user must either invent or choose some kind of simple knowledge structuring scheme before she can proceed. Thus a meta-task, schema development, is introduced into the analysis.

At the outset of our task, we felt confident we could build up a “machine translation” schema from scratch in the manner supported by the tool; we were acquainted with the field, and quite familiar with the tool itself. In this section, we will look at several aspects of schema development - finding a representation that highlights the phenomena of interest, discovering the necessary abstractions, and working from paradigmatic instances - by tracing the progress of several of the types we invented for this task.

One of the first types we created was a System. The first paper design for a System basic object (recorded in October of 1990, after an initial period of reading up on the topic) had over fifteen slots. By the time Aquanet was ready for use in January of 1991, the basic object we created had been reduced to six primitive valued slots: the name of the system, the source and target languages it handled, whether it was commercial or not, its performance characteristics, an abstract describing methodological concerns, and what type of interaction the system required (e.g. post-editing). We also had an initial list of systems we thought were important or beared looking into. By early 1991, we had acquired some on-line information about some of the systems; we brought it into Aquanet by inventing an Article basic object that had roughly the same slots as the fields in the on-line database the information was retrieved from. We use Systems and Articles, two of our original set of types to illustrate the following discussions of representational salience and the discovery of abstractions.

### 2.1 Perspicuity and the single user

In general, coming up with a collection of types to describe a domain of interest proved to be a hard task for both ourselves and other Aquanet users. It is necessary to guess “what’s important” and to choose an appropriate way of representing it. Aquanet offers a variety of representational resources beyond nodes and links, and our prior work with NoteCards users suggests that even something as simple as a node-link model poses representational difficulties.

An example of this representational salience problem is apparent in the design and evolution of the System object. As we described earlier, our initial design for a System object had a boolean-valued slot to characterize if the software was commercially available or not; we started our work in the main display space by creating an informal list that mixed both commercial and research systems we were familiar with. As the task proceeded, the list grew, and we began separating it into a list of commercial systems and a list of non-commercial systems. Thus the same information had both a formal representation as a slot value, and an informal representation as a position in a list displayed in the information space. This representational transformation is shown in Figure 2A.

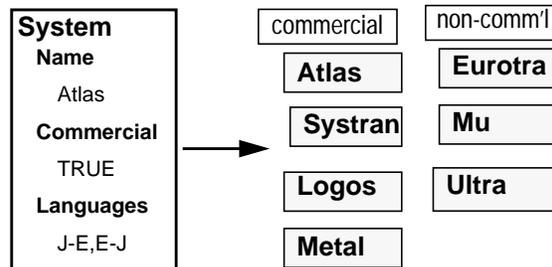


Figure 2A. Object content - commercial availability - transitions to a new representation as a list.

As the task continued, the amount of on-line information about the commercial systems proved to be far more abundant than the information about research systems, so we moved the two lists farther apart, and began forming messy piles of articles next to each of the commercial systems (see Figure 2B.) Eventually, these messy piles were straightened out, and a Player (a corporate developer) was created beneath each system. After the messy piles were made into neat stacks, each stack was labelled using a visually distinctive green object that displayed the number of items in the stack since individuals were no longer visible (see Figure 2C).

Finally, relationships between Systems, Players, and - to a minor extent - Articles and their Labels

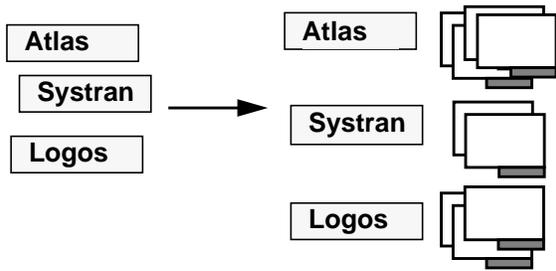


Figure 2B. List representation of commercial systems becomes the basis for notetaking.

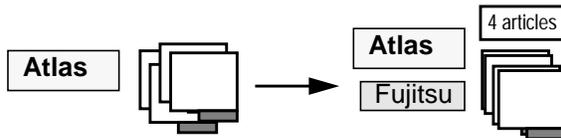


Figure 2C. Commercial systems grouped with developer and label.

were institutionalized as either Commercial Systems or Research Systems; the transition of unlinked basic objects to a Commercial System relation is shown in Figure 2D. A Commercial System had two entity-valued slots, one containing a System, the other, a Player, and one integer-valued slot to show the number of on-line sources (Articles) that had been collected about it; this number was shown as part of the relation's graphical appearance.

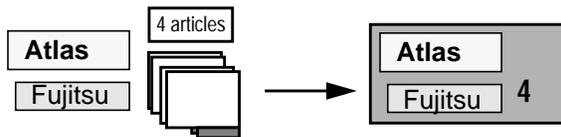


Figure 2D. Informal commercial system representation is formalized as a relation.

This metamorphosis illustrates a related problem in inventing an appropriate representation: as a characteristic grows in importance (like our difference between commercial and research systems), it requires us to add layers of redundant encoding to increase its visual salience in the information space. As Figure 2 illustrates, whether a system was commercial or not was represented in at least three ways - first as a slot, then as a position in a list or region, and finally as a relation - and not always consistently.

As time went on, another representational difficulty emerged: the information available about commercial systems tended to be very different than what was available for research systems. Methodological information - for example, whether a machine translation system used an interlingua-based, transfer, or direct approach - was readily available for research

systems, but not for commercial systems. On the other hand, filling in the slot for performance data (e.g. lexicon size, speed, and accuracy) was impossible for many of the research efforts, but was not particularly hard for most commercial ventures. Further formalization of the internal structure of Systems was discouraged because of these discrepancies, coupled with our early commitment to the form of the System representation. We had already integrated a significant amount of source material in the existing System type, and did not relish the thought of moving it to a clearer notion of two distinct system subtypes. This leads us to our second source of tension in the design of an appropriate representation - identifying the general characteristics of a type to use the power of an inheritance mechanism.

## 2.2 Missing the abstractions that make inheritance work

A second problem with early schematization involves the tacit assumption that users will engage in top-down design of their representations, that they will identify the general characteristics essential to an object type instead of encoding idiosyncratic properties common to a small subset of individuals. If a user can find this set of abstractions, she can use subtyping and inheritance to her advantage. But if a user begins working from a misleadingly homogenous subset of instances, she can easily commit to an awkward representation.

In Aquanet design scenarios, we envisioned the inheritance mechanism would take care of diverging internal structures. Under these scenarios, we would have realized that our on-line sources might have a variety of forms, and we would have come up with a general representation for them (a generic Article), then subsequently specialized the object type for each new source (a Comline article, a Dialog article, a Newsbytes article, and so on), matching subtype slots to source characteristics. But in practice we did not anticipate this disparity, and instead ended up shoe-horning new sources into a structure based on the first set of sources we found.

We designed the initial representation of our Article type so that its internal structure matched the fields of the first data source we used (Comline). Subsequent use of Dialog and other on-line sources provided us with articles that had different internal structures, but we had little motivation to match the new internal structures with corresponding type definitions. It was more important to preserve the unique graphic appearance of Articles, and ignore the less

useful internal differences. Thus, multiple Dialog fields were stuffed into the “DESC:” slot of our Articles, a slot taken directly from one of the fields used in Comline. In effect, opportunistic design and use of the representation won out over top-down planning.

### 2.3 Working from instances to develop types

From our own experiences with the machine translation application and the experiences of our small user community, we noted a common desire to work from concrete examples; users wanted to manipulate instances *in situ* in order to define or alter the generic structure of their types. This style of interaction can also be contrasted with systems designed to handle *in situ* modification of generic types as in the MacWeb system’s structured type mechanism [12]. The ability to define and manipulate types using a paradigmatic instance circumvents many of the frustrations caused by premature commitment to a schematic structure. Nielsen and Richards noted a similar phenomenon with people learning the Smalltalk object-oriented programming environment [13].

The Aquanet users we worked with wanted to begin creating new object instances without the overhead and liability of creating even a rough characterization of their task domain, or trying to match the characteristics of their task domain with an existing schema; they wanted to create new instances with little commitment to what type of object the instances would eventually end up being. These problems with premature commitment echo the experiences we had with NoteCards users (also documented in [11]). NoteCards users had difficulties chunking information into cards, naming cards (characterizing instance content), filing cards (categorizing content), and creating and maintaining consistent link types (forming a schematic structure of their domain).

In Aquanet’s case, many aspects of schematic structure are readily accessible from instances (for example, the definition of primitive-valued slots, and the graphic appearance of Aquanet objects). Thus, in later versions of the tool, we support some amount of *in situ* schema modification and the creation and manipulation of generic, untyped objects and their gradual migration to structure. In cases where the mapping from one type into another is unambiguous (i.e. their slot types correspond in a one-to-one manner), type migration does not require any additional user interaction. If the mapping is not straightforward (slots are added, lost, or the mapping is ambiguous), human intervention will be required.

More extensive use of facilities to support grad-

ual definition of structure will help us understand whether we can sufficiently compensate for the inherent difficulties of schematization to realize the interpretive benefits of this special form of electronic writing. For while people working with large amounts of information engage in activities that we would readily characterize as knowledge structuring - activities like filing, organizing piles of annotated documents on their desk tops, and producing outlines, lists, and tables - they are not accustomed to articulating how their knowledge domain is organized, even if they have a strong implicit sense of what kind of information is available, and how to find it.

### 3 Don’t link me in: relations of an informal sort

Aquanet’s concept of relations represents a more general, more expressive basis for knowledge structuring than the hypertext notion of links. But in our machine translation analysis, as well as in other early applications, we observed far less use of relations than we had anticipated based on our previous experiences with Toulmin structures and other analytic tasks [9]. Instead of articulating how different types of objects were related, users conveyed implicit relational structures through the use of spatial layout. So not only were we helping to end the tyranny of the link as Halasz suggested in [6], we were also beginning to realize the potential for the anarchy of the node.

Relationless hypertext use is not too surprising given common practice: people readily categorize materials by filing them or putting them in piles (see [7] or [8]), but they rarely express more complex relationships between individual documents or notes in an equivalently formal way. Instead they rely on spatial cues in their own environment, for example, how far away two stacks of paper are from each other, or the relative position of items in a list or chart. The Colab project noted a similar reliance on spatial cues in its meeting support tool, Cognoter [18].

In our application, informal relations were defined by providing the appropriate spatial setting for basic objects; in effect, Aquanet’s virtual copy mechanism was used to reference the same objects from different spatial contexts. In the simplified diagrammatic example shown in Figure 3, the same (highlighted) System basic object, Atlas, is used in three informal relations. In the area marked (1), the System basic object is being used in a notetaking setting; it is next to the pile of source Articles that provide the information integrated and reformulated in

the System’s slots. In the area marked (2), a reference to the same Atlas object is seen in a setting of other System objects; in this setting Systems can be collected, compared, and ordered, possibly by their apparent maturity or level of use. In the area marked (3), the System object Atlas is associated with a list of individual researchers, developers, and sponsors responsible for producing the system. If formal relations were to be defined, (1) might be expressed as a one-to-many “based-on” relation; (2) might be a “set” relation (as described in [14]); and (3) might be a complex “influenced” relation between researchers, developers, sponsors and a System.

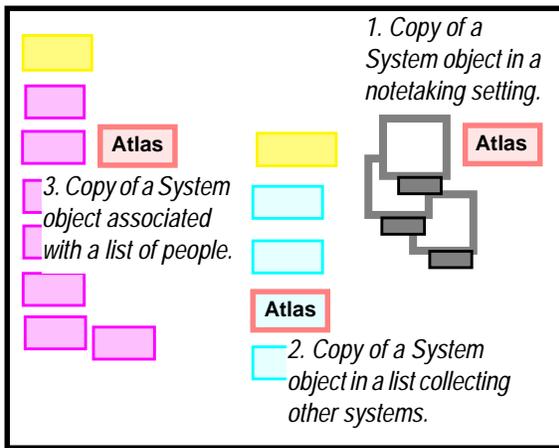


Figure 3. Using an object in multiple contexts to represent relationships.

Thus instead of using the mechanism that Aquanet provides for constructing a coherent, visually interlinked structure, we and other early users depended on our own interpretation of spatial context to convey most of the relational information. In the original conception of the tool, navigation relied on the graphic portrayal of links in the main display area; the virtual copy mechanism was implemented so the tool itself could avoid layout conflicts. But in practice, we used virtual copies much like links - as a means of accessing the same content from multiple contexts.

Informal relations like this are easy and natural for users to construct, but they pose difficult problems for collaborative use - how do collaborators promote mutual intelligibility of the space [20] - and for information management - how can these implicit, layout-based interpretive structures be maintained.

The problem of mutual intelligibility, which we experienced in this and other Aquanet information spaces, has been addressed by users in a variety of ways, including “label objects” to title a list or parti-

tion a space and face-to-face explanations of the informal structures being shared. In the future, we will experiment with other ways of attaching explanations to rationalize spatially-defined structures.

One method of solving the information management problem is for the tool to recognize and support certain widely used relational primitives. We are currently working on algorithms to induce schematic structure from spatial layout. This approach can be further facilitated by providing users with simple functionality for alignment and spacing, along the lines of most drawing editors.

Our observations of Aquanet use have led us to reconsider the role of relations in the tool. In designing the tool, we combined a richer, more general form of connection than the link with gIBIS’s browser-based mode of navigation and access. Later, we found Aquanet’s relations to be less commonly used than the tool’s design anticipated, and the structures users built to be more volatile than expected (in the sense described in [1]). Thus we find it necessary to both support the interpretation and reinterpretation that takes place in a spatial context without user-articulated relational structures and to assist users in identifying structural regularities that they may choose to formalize.

#### 4 Space: the final frontier.

Systems like Streitz et al.’s Sepia [19], Smith et al.’s WE [17], and Schuler and Smith’s AAA [16] anticipate the cognitive need to separate a complex task like writing into multiple activity spaces. We encountered a similar kind of *de facto* partitioning in our information space: as the analysis continued, and the different types of informal structure discussed in the previous section got built up, activity-related regions emerged.

To a user, the Aquanet main view is a large, unpartitioned display space. The view’s plane extends infinitely in both directions, and objects can be stacked up on it to an arbitrary depth. It is rather like an information prairie - where you are standing at any given time seems to be the center, and yet not more important, nor in any way distinguishable from where you were standing an hour ago. So it is up to the user to both manage and cultivate this space.

##### 4.1 The emergence of workspaces

In our machine translation task, we found that large areas of the main view were partitioned in a *de facto* way by tasks. Figure 4 shows a central portion of the main view (zoomed to about one-quarter of its origi-

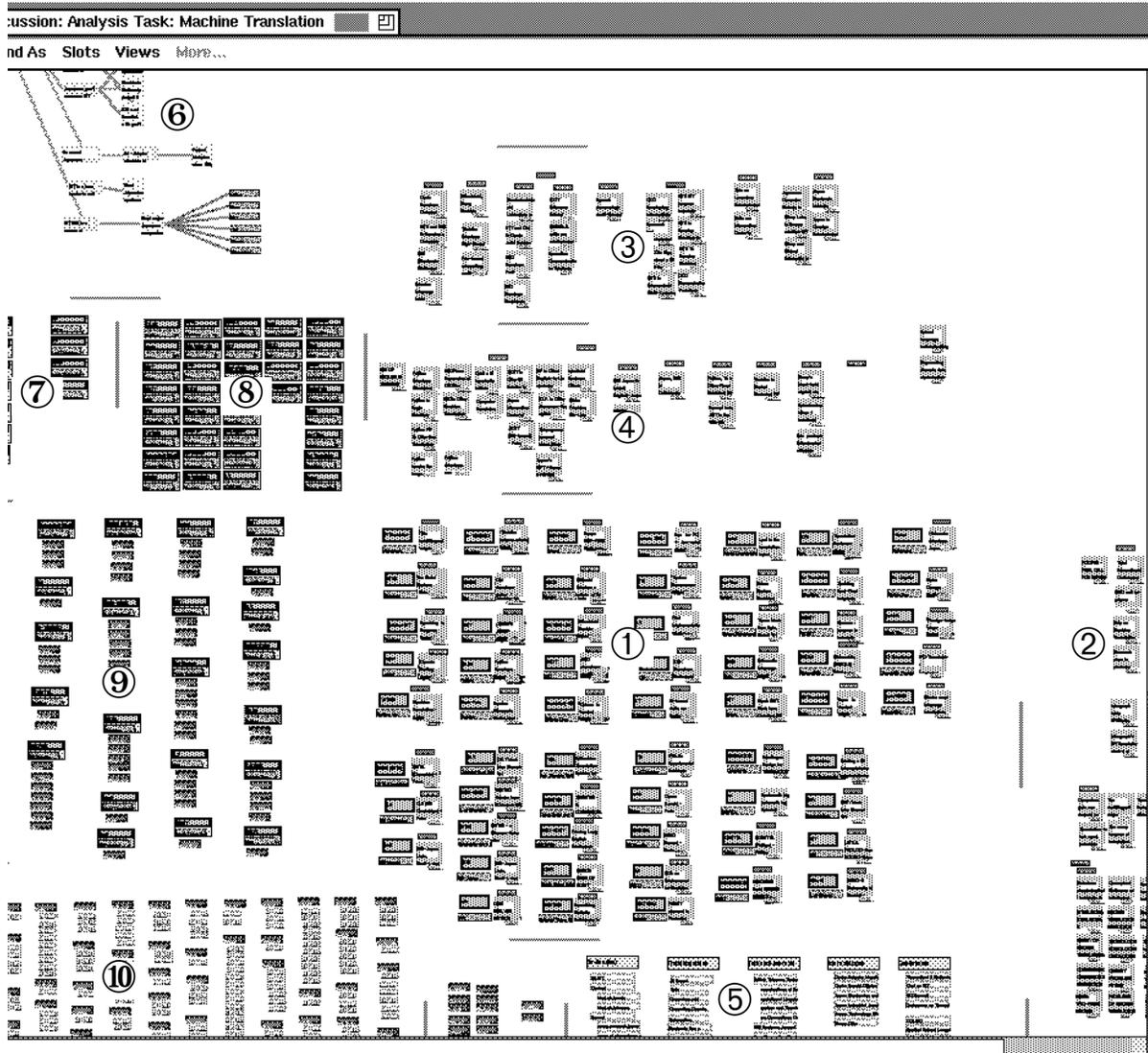


Figure 4. Activity related regions segment the main display area.

nal size); the outer reaches of many areas are omitted, since enough of their structure is visible to explain their intent. Numbers have been added to identify most of the distinct areas that emerged during the course of the analysis; the numbering suggests the order in which the areas grew (although many of the later areas were formed more or less concurrently).

As we noted in Section 2, the first objects created, Systems, were placed in lists in Area 1; this became our notetaking area. This area was used continuously throughout the task to sort incoming information and to associate it with systems (and later with industrial and academic Players in the field who was responsible for their development). As source Articles were piled

up by Systems (both to save space, and to aggregate them), it became necessary to create small labels to document how many Articles were in the pile. Thus the notetaking area ended up consisting of two sets of lists, one of commercial systems, one of research systems, and each list item consisted of four types of objects, a System, a stack of Articles, a Player, and a Small Label, laid out in a somewhat regular pattern (see the example in Figure 2C).

In Figure 4, the area labelled 2 had two functions - it was both where source materials were brought into the information space, and it was where miscellaneous general source Articles were "kept." In the current state of the analysis, there are several sloppy lists

of sources we didn't know what to do with. During earlier phases of the analysis, the number of objects in this area fluctuated wildly as we found new data sources, and spent time processing them. Articles imported into this area were originally tiled, so we could easily determine how much was there, and what progress we had made categorizing them.

The areas labeled 3 and 4 bear some similarity to 2; areas 3 and 4 consist of lists made up of Articles covering enabling technologies (like electronic dictionaries), and technologies that incorporate machine translation (for example, on-line multilingual database services). Because this was not the main focus for our analysis, Articles are just sorted and arranged loosely into lists, and no further notetaking or organization has been done. But since the analytic assignment was vague, the distinction between relevant and irrelevant information was vague too.

The labelled outlines in Area 5 were maintained sporadically to reflect what sources we were using - our on-line live sources like Dialog, conferences whose proceedings we had collected, magazines and journals that typically covered the field, and so on. This area was maintained mainly for documentary purposes - for a future consumer of our analytic results, and wasn't primary to the analysis itself.

The upper left portion of the main view, Area 6, provided the main interpretive structure for the analysis. It is an issue structure that follows a rough IBIS form with Issues, Positions, and Arguments that were supported by evidence, objects drawn from other parts of the information space using the virtual copy mechanism. Much of this network is not visible in the figure, because we tried to separate it from the central space so it could expand. Instances of IBIS types were particular to this subspace; they did not appear in other functions elsewhere in the analysis.

Most of the remaining work areas served as a means of grouping like items together in a particular context. Areas 8 and 9 are compact lists of what we felt to be the major research projects, minor research projects, major commercial ventures, and minor commercial ventures. These lists are easily distinguishable by color on the display (although not in Figure 4). Area 10 brought together systems with the people who were central to their development, and Area 7 was used to infer who central researchers were (from their collected publications). Together these areas were used to trace the flow of research ideas, possibly into commercial products.

Area 11, which is mostly truncated in Figure 4, was very much an "odds and ends" space. Among

other things, it contains instances of types that we created that turned out to serve no useful purpose. Since effort went into creating them, these objects were not deleted outright, but rather put out to pasture. One example of this strategy involved "Concepts," a type created early on in our analysis, then abandoned as we became more familiar with the terms and ideas of the field. From our informal observations of people using paper organizing systems, we know that such undeletable, unusable, uncategorizable leftovers are common to most organizing tasks.

## 4.2 Workspaces and collaboration.

Activity-related regions like the ones we have discussed are not only useful for a single person performing a task; they are also a useful basis for communicating about the space in a collaborative task, and for structuring subsequent collaboration. Communicating about the information space requires that the person generating the space rationalize its structure and design (what each region of the space is used for and how it is organized), as well as articulating any implicit constraints on the space (what type of objects are allowed there, or any ordering specifications). For example, to enable a collaborator to take part in the notetaking activity that resulted in region 1 of Figure 4, she would have been aware of the implicit purpose of that area, that it was constrained to contain four types of objects (Systems, Players, Articles, and Labels), that Players were placed below the System they developed and associated Articles to the right, and that more important Systems were placed in the left part of the region, and less important ones in the right. Recording this type of rationale both supports potential collaborations and helps capture important elements of the process by which they are created.

The material collected for this analytic task was re-used for a smaller scale, shorter term, more focused analysis performed by a linguist. The linguist was interested in Spanish-to-English machine translation efforts yielding a commercial product, currently available as an off-the-shelf software package. This is a somewhat different task, yet the material we collected and organized should have provided substantial benefit. But before the material could be re-used, three enabling activities had to take place.

First, the material had to be filtered. Much of the material we had amassed, integrated, and interpreted was of little use for the new task. Second, we had to separate the relevant (and possibly relevant) materials into a new information space. Since the linguist's purpose was different, and our organization was closely

tied to our interpretation of our own task, we did not want to permit any reorganization of our space (although we welcomed any supplemental content the linguist would add to particular objects). Finally, we had to rationalize the organization of the new information space and the semantics of our types to our new collaborator. This proved to be somewhat difficult, since we had never articulated these implicit aspects of our activities before.

Besides content- and activity-based filtering, creating a new information space, and providing rationale for its organization, we also found it necessary to retrieve supplemental information; systems that we had regarded as minor, and ancillary to our analysis became important in view of the new objectives. Later we also transferred this new information back to our original information space.

## 5 Evaluation: Did Aquanet help keep our ideas in place?

We have examined some of the difficulties we discovered when we used Aquanet for knowledge structuring task: First, the tool coerced us into prematurely schematizing our domain and committing ill-formed information and ideas to structure; second, Aquanet's concept of relations were hard to articulate and spatial layout often took the place of more formal relational definitions; and finally, we needed additional layers of structure in the form of rationalized workspaces to make our activities coherent, both to ourselves and to a later collaborator. These problems can all be addressed by a series of modifications and enhancements to the existing tool. But we are still left with a more pressing question of evaluation: are knowledge structuring tools worthwhile considering the changes they make to the overall activity and the demands they impose on their users?

An extreme point of view is to claim that the only reason for people to use knowledge structuring tools like Aquanet is if they provide computational facilities to perform automated interpretation of the represented materials or manipulate the information in ways human find difficult (for example, animate it, analyze structural dependencies, or subject it to sophisticated layout algorithms). Of course, this view also requires a level of formalization and consistency incompatible with the way we have seen tools like Aquanet most effectively used. It is also difficult to maintain representational generality while incorporating specialized kinds of interpretive algorithms.

On the other hand, tools like Aquanet can be seen

as providing a good front end for users building collaborative knowledge bases, where content and structure are negotiated and incrementally formalized as warranted for other uses. From this viewpoint, a user or applications programmer writes a separate program to manipulate or use the knowledge structures constructed with Aquanet. This outlook is not incompatible with the current implementation and some of the early applications of the system; a programmer's interface can provide such functionality.

But our approach to future development sees the creation and manipulation of representational structures (both explicit and implicit) as a crucial part of the interpretive process, as an alternative to other forms of electronic writing (see [2]). We intend the tool as a means for users to produce multiple interpretations - one that lays out the analytic framework and rationale for more critical examination by collaborators and other readers of the analytic results. Trigg and Irish encountered this in their work with NoteCards users [21]; a writer's subsequent activities may rely on the interpretation she has come to during the structuring of gathered materials, even if she never exports or reuses the actual structure or its contents. This has been shown to be true for our application assessing current machine translation research and technology for analytic work as well.

## Acknowledgments

Frank Halasz provided crucial guidance and support throughout the Aquanet project. Bill Janssen has also contributed a great deal to the development of the tool. We would like to thank Marikka Rypa for participating in the collaborative use of our machine translation analysis and Tom Moran and Frank Shipman for many helpful discussions about spatial structures. We would also like to thank Tom Moran, Dan Russell, Mark Stefik, Ramana Rao, and the ECHT '92 reviewers for their comments on an earlier version of this paper.

## References

- [1] M. Bernstein, J.D. Bolter, M. Joyce, and E. Mylonas, Architectures for Volatile Hypertext, *Proceedings of Hypertext '91*, San Antonio, TX, December 16-18, 1991, pp. 243-260.
- [2] J.D. Bolter, *Writing Space: The Computer, Hypertext, and the History of Writing*, Lawrence Erlbaum Associates, Hillsdale, New Jersey, 1991.

- [3] G. Bruns, Germ: A Metasystem for Browsing and Editing, MCC Technical Report STP-122-88, Austin, Texas, 1988.
- [4] J. Conklin and M.L. Begeman, gIBIS: A Hypertext Tool for Exploratory Policy Discussion, MCC Technical Report Number STP-082-88, Austin, Texas, 1988.
- [5] F.G. Halasz, T.P. Moran, and R.H. Trigg, NoteCards in a Nutshell, *Proceedings of the ACM CHI + GI Conference*, Toronto, Ontario, April 5-9, 1987, pp. 45-52.
- [6] F.G. Halasz, 'Seven Issues': Revisited, Hypertext '91 Keynote Talk, San Antonio, TX, December 18, 1991.
- [7] T.W. Malone, How Do People Organize Their Desks? Implications for the Design of Office Information Systems, *ACM Transactions on Office Information Systems 1(1)*, January, 1983, pp. 99-112.
- [8] R. Mander, G. Salomon, and Y. Wong, A 'Pile' Metaphor for Supporting Casual Organization of Information. *Proceedings of CHI '92*, Monterey, CA, May 3-7, 1992, pp. 627-634.
- [9] C.C. Marshall, Exploring Representation Problems using Hypertext, *Hypertext '87 Proceedings*, Chapel Hill, North Carolina, November 13-15, 1987, pp. 253-268.
- [10] C.C. Marshall, F.G. Halasz, R.A. Rogers, W.C. Janssen Jr., Aquanet: a hypertext tool to hold your knowledge in place, *Proceedings of Hypertext '91*, San Antonio, TX, December 16-18, 1991, pp. 261-275.
- [11] M.L. Monty, Issues for Supporting Notetaking and Note Using in the Computer Environment, Dissertation, Department of Psychology, University of California, San Diego, 1990.
- [12] J. Nanard and M. Nanard, Using Structured Types to Incorporate Knowledge in Hypertext, *Proceedings of Hypertext '91*, San Antonio, TX, December 16-18, 1991, pp. 329-342.
- [13] J. Nielsen and J.T. Richards, The Experience of Learning and Using Smalltalk, *IEEE Software*, May, 1989, pp. 73-77.
- [14] H.V. Parunak, Don't Link Me In: Set Based Hypermedia for Taxonomic Reasoning, *Proceedings of Hypertext '91*, San Antonio, TX, December 16-18, 1991, pp. 233-242.
- [15] D.M. Russell, T.P. Moran, and D.S. Jordan, The Instructional Design Environment, in J. Psocka, L. D. Massey, & S. A. Mutter, eds., *Intelligent Tutoring Systems: Lessons Learned*, Lawrence Erlbaum Associates, Hillsdale, N.J., 1987.
- [16] W. Schuler and J.B. Smith, Author's Argumentation Assistant (AAA): A Hypertext-Based Authoring Tool for Argumentative Texts, in A. Rizk, N. Streitz, J. Andre, eds., *Hypertext: Concepts, Systems and Applications - Proceedings of the First European Hypertext Conference*, Cambridge University Press, 1990, pp. 137-149.
- [17] J.B. Smith, S.F. Weiss, and G.J. Ferguson, A Hypertext Writing Environment and its Cognitive Basis, *Hypertext '87 Proceedings*, Chapel Hill, North Carolina, November 13-15, 1987, pp. 195-214.
- [18] M. Stefik, G. Foster, D. Bobrow, K. Kahn, S. Lanning, and L. Suchman, Beyond the Chalkboard: Computer Support for Collaboration and Problem Solving in Meetings, *Communications of the ACM 30(1)*, January, 1987, pp. 32-47.
- [19] N.A. Streitz, J. Hannemann, and M. Thuring, From Ideas and Arguments to Hyperdocuments: Travelling through Activity Spaces, *Hypertext '89 Proceedings*, Pittsburgh, Pennsylvania, November 5-8, 1989, pp. 343-364.
- [20] R.H. Trigg, L.A. Suchman, and F.G. Halasz, Supporting Collaboration in NoteCards, *CSCW '86 Proceedings*, Austin, Texas, December 3-5, 1986, pp. 152-162.
- [21] R.H. Trigg and P.M. Irish, Hypertext Habitats: Experiences of Writers in NoteCards, *Hypertext '87 Proceedings*, Chapel Hill, North Carolina, November 13-15, 1987, pp. 89-108.