Information Awareness and Representation

Matthew Chalmers

Computing Science, University of Glasgow, Glasgow G12 8QQ, United Kingdom

matthew@dcs.gla.ac.uk

Abstract

This paper discusses how information representation affects the degree and character of awareness afforded by computer systems: awareness of people and of information artifacts. Our discussion ranges from system design to theoretical concepts, and we aim to link and show consistencies across this spectrum. We begin by describing a prototype collaborative filtering system, Recer. This system tracks ongoing activity in the web browsers and text editors of a group of people, and offers recommendations of URLs and local program files that are specific to and adaptive with that activity, and that reflect patterns of earlier activity within the community of use. We then take a more general look at collaborative filtering, and compare it with two other approaches to engendering awareness of useful information, information retrieval and software patterns. We discuss how each implicitly or explicitly involves collaboration, formalisation and subjectivity in its core representations. We then explore the artifact–centred approach to awareness that Recer represents, and relate it to the activity–centred approach more familiar within CSCW. We use this comparison in discussing in more theoretical terms how representation and formalisation of information affects awareness, interpretation and use. Our intention is to explore and understand the choices that designers have for the core representations of information systems, and the consequences for awareness that follow for users. We wish to relate such practical design issues to the more theoretical discussion in CSCW around concepts such as common information spaces, the space–place distinction, and the status of formal constructs.

1. Introduction

One might characterise CSCW as being differentiated from traditional approaches to work with computers and information by its stance against reductionism. Where there was a narrow focus on raw data as abstract artifact, our wide and multidisciplinary field interprets the artifacts of work as contextual and subjective, and shifts attention from raw data to contextualised, significant information. It has also been said that in CSCW "the work involved in both putting information in common, and in interpreting it, has not been sufficiently recognised" [2]. This paper looks at the choices and consequences we have when designing information systems, with regard to the representation and interpretation of shared information, and computer support for awareness of both the people and the information involved in use.

The nature of programming languages and databases means that our anti–reductionist field cannot avoid reduction and formalisation. While we gain from understanding the infinite detail of context and situation, we must take a pragmatic stance if we are to design the finite and formal representations of computer systems to support awareness. The need to reduce, select and filter grows with the number of people and volume of information we work with. We rely on support for not just awareness of people and information, but for selection, interpretation and action. We therefore have to make a choice, in representing people and artifacts, as to what we exclude and include, what we represent in detail and what is abstract, what we actively interpret and what we leave to human interpretation.

As this paper continues, we will address progressively 'higher level' issues related to these choices but, in order to ground discussion in design practice, we will first look at one 'path model' system, Recer. We will outline the system architecture and give brief examples of its use. Rather than fully exploring the system's internal content and structure, we wish to give enough detail to allow comparison with other contrasting approaches to the access of computer–based information, namely information retrieval and software patterns. By focusing on the scope and formality of their characteristic information representation schemes, we discuss how each approach affords awareness of information and of people. Later sections take a broader view of how systems and theory in CSCW have generally approached representation and awareness, and we explore how formal representation issues relate to discussion of awareness couched in

more spatial and linguistic terms. The paper concludes with a summary and a look at prospects for future developments.

2. Recer and the Path Model

Recer is an experimental system that recommends URLs and filenames that appear to be relevant or useful to a person's activity. An earlier prototype was described in [3]. It is based on tracking an individual's activity in desktop tools, and building up a *path*, a time–ordered history of that person's use of symbols. In our current system we log URLs and filenames displayed in the Netscape web browser and the *xemacs* editor, and store each path as a table in an Oracle database. Each time a person loads a new web page or switches between editor buffers, the time and the URL or filename involved is entered in that person's table as a new (timestamp, string) pair. We can use standard database facilities to keep this table ordered and indexed. This lets us quickly find, for example, the symbols used in a given time period or the times when a given symbol was used. A path serves as the representation of a person's activity, and the most recent path entries serve as the representation of that person's current activity. Note that URLs and filenames are entered into the same table, and we do not split up the path into any fixed episodes or categories but instead rely on more dynamic analysis as part of the recommendation process, as we now describe.

The set of paths representing a group of people is treated a shared, reciprocally available resource. A slider in the recommender's interface allows a person to select the time period that delimits their 'recent' symbols. Recer periodically takes these recent symbols and searches for past occurrences of each one. By default, this search is through all paths, including one's own, but alternatively one can specify who one draws from. The system collects the context of each past occurrence of each recent symbol i.e. each past occurrence's 'window' of path entries that subsequently occurred within a chosen number of minutes. The system tallies the symbols collected from these windows, discards any symbols that were recently used and then presents the remaining symbols as a ranked recommendation list. Clicking on a URL in this list triggers a web browser to display it. Clicking on a local filename triggers xemacs to load and display it. We are also optionally experimenting with 2D visualisations of the windows around recommended symbols. After display of a recommendation list, by default the system sleeps for a time then wakes, collects a new set of recent symbols and begins again.

We are experimenting with two modes of activity logging. In the first mode, only filenames and URLs are recorded. In the second, contained words and URLs are additionally recorded. In the web browser, recording starts each time a user accesses a new web page. The new page's URL, and possibly its contained words and URLs, are logged. Logging all contained symbols assumes that all that is loaded is perceived by the user: a weak assumption for long documents at least until we can be more specific about where the user scrolls and jumps. In the case of the editor, the cursor is tracked. In the first logging mode, only a switch to a new file buffer triggers a path entry. In the second mode, the word or token nearest to the cursor is found every second and, if different to the previously found word, logged. This assumes that only what is near the cursor is significant, which is a more restrictive assumption than we would prefer.

By default Recer interprets the most recent sequence of path entries as an implicit request for recommendations. However, the 'recent' set is displayed in the interface, and one can explicitly add to and delete from it. For example, one could discard all the passively obtained symbols and simply type in new URLs. One can thus make the expression of information need as explicit and actively directed as required.

We briefly describe two example recommendation lists, leaving fuller details of the implementation to a forthcoming paper. The author, resident in Switzerland at the time, had just browsed web pages giving Alpine weather forecasts i.e. the system had these pages' URLs as its 'recent symbols'. The recommendations made on this basis were mostly for ski resorts' web pages, including JPEG resort maps and other information from resorts (and web sites) the author had never visited. These were good recommendations because the author, and his colleagues whose paths he drew from, usually checked if the weather was suitable for a ski trip before getting details of cable car times and pistes. Since this first 'weather' stage was done, recommending yet more weather pages (i.e. pages topically similar to recent activity) would have been of little help. This is an example of recommendations' focus not on topical similarity, but on consistent patterns of use. When past occurrences are near to each other, their windows overlap. Symbols within overlaps get higher tallies. Highly ranked symbols tend to be, therefore, those

symbols most consistently used in the contexts most similar to recent activity. We assume that consistency of use in similar contexts is the best indicator of appropriate (literally, utile) recommendations.

The second example involves recommendations of software components. The recent recorded activity was from editing the Java class that implemented the core recommendation algorithm and the interface for Recer. This led to the display of URLs for Java documentation used during a period working on the same class some weeks earlier, including pages for the Swing UI components used in the interface. Also recommended were names of files containing more peripheral components of the system, including Lisp programs and shell scripts.

In each of these examples, the recommendations were made because of consistent patterns of use within a particular group of people, rather than overlap of user–independent object content. Also, note that recommendations involved interweaving heterogeneous data types: HTML, JPEG images, Lisp and Java code, and so forth.

Recommendations for code files can come from the recent use of other, related code files, but also from reading related documentation or from using related program keywords and names. Similarly, working on code can lead to recommendations of all these types of information. The particular type of information such as web page or code file is not as important here as consistent use in similar contexts. We see the mixture of the types of information usually treated separately.

As pointed out in [18], the interpretation of human activity by the computer involves a vocabulary, the symbols identifying the artifacts and actions of computer–based activity. Path–based recommendations involve a vocabulary of symbols more amenable and accessible to the computer, such as URLs, filenames and content words. Artifacts such as web pages, local files and programs are among the 'vocabulary' of symbols that Recer can more easily obtain and interpret. These symbols and the actions upon them are the predominant human 'utterances' the computer is presented with and, even though we expect to expand the range of logged types, Recer must make its model of the world with what it has. The limits of its 'language' are the limits of its world, or at least the limits of its model of our world.

The context of each such use of a symbol, and hence the activity of a person at the time of that use, can be represented as the temporally neighbouring symbols. If we accept that it is common that a person interleaves activity in various tools, moving between mail tool, web browser, editor and so on, then the symbols gained from all of those tools are significant and their significance is mutually interdependent. Since human activity involves a heterogeneous mix of data types or media, our work exemplifies a view that the computer should be able to represent and make use of that mix.

Our intention is that path systems' representation and interpretation of symbols should show some accord with the model of language of [19], as exemplified by "the meaning of a word is its use in the language" and "the limits of my language are the limits of my world." By building up a record of that activity over time, the consistent temporal patterns of symbols that make up the language of use can be represented. The current activity of the user can be represented by the most recent utterances, as future activity is of course yet to be perceived. An attempt to predict the latter requires assuming a degree of consistency in the language i.e. consistency in the patterns of symbol use. Whether such a prediction is correct or not, the recording of whatever the user actually does will adapt the representation. Although still experimental and prototypical, path systems' history of activity of a community of use is intended to be a representation of those utterances, contexts, and that language.

3. Recommendation, Retrieval and Reuse

In this section we move on from looking at one particular system that affords access and awareness of computer-based information, to look at three approaches to such systems: collaborative filtering (including the path model), information retrieval and software patterns. These have been chosen to show the variety of approaches to information representation and formalisation one has at hand when considering how to support the awareness of information and of people.

First we consider collaborative filtering (CF) and its extension, the path model. Given its name, it is somewhat ironic that CF has not often been discussed within the literature of CSCW since its initial development in the Tapestry system almost a decade ago [9]. A survey of CF systems appeared some time

ago in the Comm. ACM [13], and recently a paper appeared in Proc. CSCW 98 discussing how to initialise and evaluate a set of CF programs [15].

Also known as recommender systems, CF forms an important approach to the sharing and awareness of information amongst a group of people. It is rumoured that CF's first commercial use was in a dating agency, which is most obviously a way to provide awareness of people, but more commonly in CF we use people's activity as a means to afford awareness of information artifacts. In CSCW we more often use artifacts to find out about people and their activity, as we will discuss further in a later section.

In a stereotypical CF system, each participating person records their set of choices from a collection of information objects. The person may do this explicitly, for example by filling in a rating form. Passive recording of selective activity is increasingly used, however. An example is Amazon.com's use of the set of books bought by each of its customers. Recommendations for interesting or useful objects are made by finding the most similar purchase records to the individual's record. (Note that 'similar' here refers only to the overlap of people's records i.e. the size of set intersection.) It is here that one could recommend the owners of these similar records, as 'similar people', but usually the records are collated to find and recommend which objects were chosen frequently. Objects that already occur in the individual's record are usually filtered out at this stage. In loose terms, therefore, CF recommends new objects that similar people liked. A great strength of CF is that each item recorded in a set can be of an arbitrary type. URLs, books, music, films and so forth can be easily mixed. On the other hand, recent or repeated choices in a record are indiscernible from old ones. Recommendations may be tailored to the individual, but are not specific to his or her recent or current activity.

The path model, exemplified by Recer, extends CF by introducing contextual specificity of representation and recommendation. As was described earlier, it involves recording not just which information objects were used, but when each was used. A path is made for each individual, and the symbols recorded as used by a person during a period of time represent the context or activity of the person during that time. Unlike CF, only the most recent activity recorded in the path is taken as the current context of the person. Recommendations for interesting or useful objects are made by taking this current context, and then looking back through the shared pool of paths to find objects consistently used in similar past contexts. Similarity here refers to the degree of symbol overlap between the user's current context and a past context. The path model echoes the paths and trails of Vannevar Bush's *As We May Think* but, as discussed in [4], this approach was primarily based on an analogy with Hillier's Space Syntax theory, that interrelates patterns in pedestrians' movement with the physical layout of the city.

In paths, an information object or symbol is represented by its patterns of co–occurrence, with regard to other symbols, in use. Recommendations are made on the assumption that what was previously and consistently used in similar past contexts may be useful in the current context. The path model's explicit dependence on person and context stands in contrast to the anonymous and non–contextual analysis characteristic of the traditional approach to finding useful or relevant documents and files: information retrieval (IR).

In IR, an information object is represented by the patterns of symbols it contains. For example, in a system such as AltaVista, a textual document is represented by the words it contains. Use and activity are generally not persistently represented. A query to a search engine and the list of 'hits' the engine returns do not adapt system behaviour deeply and permanently. In general, IR may offer awareness of authors of documents but can do little to support the mutual awareness of readers. A few systems are beginning to show a move away from this, as in DirectHit's use of statistics of the popularity of hits (www.directhit.com). Traditional IR retains its great strength, offering access to large volumes of information, because it avoids any CF-like bootstrapping period and creates its primary information representations prior to use.

Overall, IR takes an objective standpoint with regard to awareness because, in its information representation, its focus on the artifacts of work excludes subjective and contextual features. Awareness of people and their activity is not strongly afforded, as they are unrepresented. An IR system is therefore designed for the 'average person in the average context,' but it may never quite suit any particular person in any particular context. The basis for making a person aware of an information object is static, whereas paths and collaborative filtering are inherently subjective and dynamic. (For a more extended comparison of IR, CF, paths and workflow, the reader is referred to [5].)

Software reuse can be considered as an information access problem, and carries with it an assumption of collaborative work amongst programmers. It involves the sharing of software components in order to increase quality and decrease the unnecessary and costly replication of work. Ideally, a developer of a program would be able to find components that others have used effectively in similar situations to their own, and then weave them into their own design. This would not only speed their own design work, but enhance the community of design practice. Such reuse happens so rarely, however, that the consequent inefficiences have led to what is commonly called a 'software crisis' facing the computer industry.

Information retrieval has not made a large impact on the problem of enhancing software reuse, because of the difficulty of automatically analysing the content of a program or program component so as to identify key features or indices, to match with queries. Compared to textual documents, it is more obvious in the case of software components that deeper meaning or utility goes beyond content. It stems from contextual features such the ways in which it can be linked to and integrated with other components, and real world situations in which it has been and can be usefully applied. Experience of use—of what works well in practice and what doesn't—is important to represent and communicate to potential reusers but this is, at most, only implicitly expressed in the component's code or content.

These factors have influenced the design of software patterns, an influential current approach to support reuse of good designs [8]. Like paths, software patterns draw from architectural and urban design theory, in this case the pattern languages of Christopher Alexander [1]. A software pattern is an attempt to fix or encapsulate expert programmers' design knowledge in an informal representation, in fact in natural language. Alexander is quoted in [8] as saying that a pattern "describes a problem that occurs over and over again in our environment, and then describes the core of the solution to that problem, in such a way that you can use this solution a million times over, without ever doing it the same way twice."

A pattern has four essential elements: a brief, descriptive name, the problem i.e. a description of when to apply the pattern, the solution i.e. the elements that make up the design, and the consequences, results and trade–offs of applying the pattern. The description of the software component's internal design is only a part of the pattern, complemented by contextual information such as the preceding problem and later consequences. Note that code for a component is not usually given. In order to support greater generality of application, the component is reduced from actual program code to an outline of its design, a simplified stereotype expressed in natural language. There has been pressure to formally represent patterns, and to provide automatic access and application, evinced for example in the to and fro in the Pattern Languages of Programming series of conferences. Nevertheless, such formalisation has not happened. Gamma et al. state that the desire to more generally and flexibly describe the applicability and consequences of pattern use meant that they deliberately avoided such formalisation.

Gamma et al. offer a simple diagram showing how all their patterns relate to each other, and related patterns are usually mentioned in each pattern description. They say that they offer a catalogue of patterns, not a language of patterns as Alexander claims to have developed for buildings. "Given the variety of software systems that people build, it's hard to see how we could provide a 'complete' set of patterns, one that offers step–by–step instructions for designing an application."

The generalised pattern serves as an abstraction that requires a degree of expertise on the part of a programmer in order to create and integrate a corresponding software component. A programmer interprets the pattern him– or herself, in the light of their current activity, and decides what action to take. This reflects the common opinion that everyday experience of software design, built up over the years, is good enough to afford use of patterns without strong interpretive support from the computer. In terms of awareness, the focus is almost entirely on this abstract but informal representation—and to some extent of its author. Other uses and users of the pattern are not represented, and so one must trust the author without any direct presentation of relevant past activity. The programmer trusts that the pattern author's experience and skill makes up for this.

Of course, there are systems that do formalise and interpret software components. Most object–oriented programming environments, for example, describe each component in terms of a small fixed set of easily obtainable categories of objective features, such the classes and methods declared inside it, the classes it imports, its creation date and so forth. In doing so they apply a retrieval-like model. They are good for finding simple matches in objective terms, but weak in terms of offering awareness of what is 'good' or more subtly and contextually utile. While they can be very useful for working one's own code and standard

library code, they are not so obviously ‘reuse’ tools for access to others’ code. We suggest that extension to larger volumes of information, beyond that which one has worked with, is difficult because the names and structures used are less familiar. Such tools are further along a spectrum of awareness approaches for software components, a spectrum that begins with those such as patterns that have a policy of leaving interpretation and action to the human, and very little formalised representation of artifacts and activity.

Path systems aim to afford awareness and reuse by going further along this spectrum, with the computer doing a good deal more representation, interpretation and action. Patterns to some extent involve awareness of a trusted author, while paths involve awareness of colleagues to draw from. Showing the people one’s recommendations come from, which we do not do at present, would offer awareness of whose work in particular was most related to one’s own. Both paths and patterns allow flexible interpretation of applicability, even though paths are formal and refer to particular components and documents, while patterns are relatively informal and involve abstract designs. The similarity we would highlight most is that they both support awareness of software designs because they avoid over-formalisation and objectification, and so offer flexible reinterpretation.

In making such comparisons of approaches to information representation, we do not wish to say that any particular approach is right or wrong. Instead we wish to show how each has characteristic strengths and weaknesses as to what it can make one aware of, and that these affordances or potentials for use are determined by its approach to information representation. Such affordances therefore stem from how artifacts, human activity involving them, and particular people are represented using the formal vocabularies of computers—and what is left to human memory. This in turn determines how awareness and interpretation are supported by the computer, and what is left to human interpretation and language. In this way we begin to see a spectrum or design space in which we can place particular approaches to and systems for awareness. We see such awareness approaches and systems not as isolated entities, but as parts of a relative system, related and differentiated by variations in representation and interpretation— what is done, and who (or what) does it. The next section aims to explore where work in CSCW fits within this view, and hence how our discussion so far relates to other work on awareness in CSCW.

4. Artifacts, Activity and Awareness

Path systems such as Recer stand in contrast to the kind of system one would typically associate with the term ‘awareness tool’ as it is used inside CSCW. A stereotypical CSCW awareness tool, we suggest, focuses on presenting the ongoing appearance and activity of people but de–emphasises the artifacts they work on and how those artifacts relate to past activity. Earlier sections showed examples of how activity might alternatively be used to maintain awareness more of artifacts than of people. In our field, we have tended to focus on, for example, media spaces that show the ongoing activity in a number of remote workplaces, chat systems that show the most recently typed messages within a group of communicating people, notification systems that keep us up to date with changes to a database, and virtual environments that show current position and gaze within a modelled world. Such tools do not take on the work of interpreting others’ activity automatically. They focus on the present but do not draw heavily on the past and or support future action so actively.

To look more closely at a well–known example, the Portholes system [7] presented regularly updated still video images of a number of selected offices, allowing the system’s users to maintain awareness of each other by means of ongoing observation of the activity apparent from the images. A user interprets the images him– or herself, in the light of their current activity, and decides what action to take. No log or database of images is built up, and relatively little image manipulation or interpretation is done by the computer. With such media, human memory and perception are taken to suffice instead, so the system need only show transient images. Portholes and similar media spaces reflect a common stance that everyday experience of human interaction, built up over the years, is good enough to afford useful awareness without strong interpretive support from the computer. When part of a person’s current activity or context is the observation of the activity of others, he or she can decide what action best suits that context.

Of course, some media space systems do go a little further in analysing and interpreting information such as images, describing the person associated with an office as, for example, ‘away,’ ‘alone’ or ‘in a meeting.’ In general, however, this automatic interpretation is in terms of a usually small fixed set of pre–categorised states of activity, and consequent automatic action is similarly restricted. Such systems are

presented with an apparently simple vocabulary of symbols, for example how many people are in an office and the openness of the office door, and interpreting the openness to interruption of the office owner. We find yet another example of how the interpretation of a 'simple' artifact can be complex and subject to contextual features such as who wishes to interrupt, why they wish to interrupt, and what is being discussed in the office [11].

Such systems are, again, somewhat further along the spectrum of tools and approaches ranging from those with a policy of leaving interpretation and action to the human, and very little computer–based representation of artifacts and activity, to those that put heavy emphasis on representation, interpretation and action, such as path systems. As we look along this spectrum, it is not only the interpretation of others' activity that shifts from the human to the computer. The human activity at both ends of this communication needs to be represented, with the viewer or reader as well as the viewed or author represented. As Recer showed, each person's activity can then be represented not only as a dynamically changing contextual phenomenon, such as their recently used set of symbols, but also as a basis for interpretation of past experience, as an utterance or behaviour framed by a language. We represent this past experience by the set of paths shared by a group. The artifacts involved in such a representation are objects whose meaning is primarily defined and articulated via this shared history of use.

Each specific occasion of use of a shared object is to some degree shaped by the meanings previously established for others, as well as each actor's past experiences, perceived current situation and future plans. It can also be taken as a public act, a stage in the ongoing debate and resolution of meaning. As Bannon and Bødker [2] put it:

> Cooperative work is not facilitated simply by the provision of a shared database, but requires the active construction by the participants of a common information space where the meanings of the shared objects are debated and resolved, at least locally and temporarily. Objects must thus be interpreted and assigned meaning, meanings that are achieved by specific actors on specific occasions of use.

Crucial to the concept of the common information space (CIS) is the potential for representation of an author's meaning in a way that allows later access and reinterpretation by a later reader. The CIS is differentiated from a 'raw' database by the greater degree to which the author and the reader work to understand the each other's contexts of use, and the communicative significance is dependent on this interpretation. In paths, symbols used within a window of time represent the context of author and reader, and the recommendation process and associated visualisations are intended to support the work of understanding.

The raw database, typified by the IR approach to information access discussed earlier, is non–indexical in that it does not employ contextually specific references, but is akin to "definite noun phrases whose meaning is claimed to be specifiable in objective, or context–independent terms" [17]. We again see the tension between reducing contextual representation, so as to offer the power of automatic and predictable treatment of information, and increasing it so as to enrich communication of the author's meaning. As was mentioned in the introduction, one problem of the latter is that it can require an indefinite or infinite amount of detail. Continuing the quotation of Suchman, and retaining her emphasis:

> But the *communicative* significance of a linguistic expression is always dependent upon the circumstances of its use. A formal statement not of what the language means in relation to any context, but of what the language–user means in relation to some particular context, requires a description of the context or situation of the utterance itself. And every utterance's situation comprises an indefinite range of possibly relevant features.

Our approach tries to steer a middle course between both extremes, with a path system's interpretation of each symbol or utterance involving an intermediate amount of contextual detail. We cannot communicate all of what the language-user or system user means in relation to that context, but we do aim to offer more than more traditional information access approaches that rely on such indexical or objective terms.

Schmidt (1997) recently discussed the tension between the detailed contingency of situated action and the process of generalisation over the specificities of situation necessary to create formal constructs. He points out their crucial coordinative and communicative significance, and the way that formal constructs "would be of only marginal utility if they were not inscribed upon artifacts." Such an artifact is for him a shared space with a role:

> [To] give permanence to the protocol for which it stands proxy in the sense that it conveys the stipulations of the protocol in a situation–independent manner. […] Written artifacts can at any time be mobilized as a referential for clarifying ambiguities and settling disputes: 'while interpretations vary, the word itself remains as it always was.' They are, for all practical purposes, unceasingly publicly accessible.

Schmidt puts forward a strong case as to the variability of the normative effect of formal constructs, based on their underspecification of situation. This effect may range from script–like sequentiality of actions to flexibly interpreted, map–like resources for situated action.

Our earlier discussion of the flexibility of interpretation of formal constructs echoes Schmidt's point. However, although the artifact 'remains as it always was,' its interpretation by subsequent readers may vary, as each interpretation is itself situated in a context of activity and history. It can not be guaranteed that the communicative intentions of the original author will be met. Interpretation of a formal construct may thus potentially give rise either to ambiguities or to clarifications, and may create disputes or settle them.

We find ourselves following the argument of Ricoeur, who saw written artifacts as semiological texts or text analogues whose design necessarily involves distanciation [14]. Distanciation, as he puts it, shows four distinctive traits in comparison with speech. The author's meaning is inscribed in the text, a fixed and external but finite representation. It can not display ostensive references i.e. references to the author's context beyond what was inscribed. Instead of a known audience directly apparent to a speaker, the range of potential readers is unlimited. Finally, the text's meaning is free to vary from the mental intention of the author. These four traits taken together constitute the text's 'objectivity' (p. 210) or 'semantic autonomy' (p. 37). Formalised information representations, as with all texts or text analogues, show distanciation that reveals a decrease in emphasis on situational detail, and an increase in objectivity and autonomy. In a way that Schmidt's 'underspecification' echoes, these attributes bring openness to interpretation i.e. varied semiological use that is free to vary from that intended by the author or designer, and which may or may not be useful to the reader or user.

The path approach therefore can be seen as designed to show an intermediate level of distanciation. The path representation affords awareness of activity and artifacts that are to some degree specific to the current person and context. We obtain a new perspective on the way that a path system, when presented with a quite novel symbol, has no meaning on which to base active interpretation. It has no past uses or 'language' to draw upon, and so it can 'say' nothing. ("What we cannot speak about we must pass over in silence.") With strong distanciation, as in IR, objectivity and contextual independence increase. Interpretation and use can therefore occur in any context and for anyone, and gain the same results—meaningful or not.

We can also take a semiological view of another influential approach to understanding awareness' relationship to representation, the distinction of *space* and *place* [10]. Harrison and Dourish suggest that the significance and utility of a space increases when the people people build up a past that involves it, a history of experiences that allows the space to obtain the richer quality of 'place.' This change involves supporting the development of 'appropriate behavioural framing,' the emergent patterns of human behaviour and interaction that offer understandings of the space. It is important to note that their discussion refers not only to physical environments but also media spaces, information spaces such as Usenet groups, and hybrids of the physical and the virtual. In developing technical mechanisms to support awareness, achieving this quality of 'place' is an important goal.

They focus on information spaces that employ aspects of physical spaces in order to support cooperative work, and suggest that naïve mimicry of the physical world does not significantly help in making a useful place out of a sterile information space. One critical factor, they say, is "support for adaptation and appropriation of the technology by user communities." This leads to the development of a "communally-held sense of appropriate behaviour, and a context for engaging in and interpreting action." We see this as essentially the same as the "debate and resolution" of Bannon and Bødker, and we suggest that the common information space is equivalent to, or at least in deep accord with, 'place.'

Harrison and Dourish go on to build up a complex set of terms and concepts, such as 'spaceless places,' in interpreting issues of technology design and information representation in terms of space and place. We take a slightly different approach to understanding the development and support of awareness. We treat a space or place as a medium for significant action, and as an addition to the other media that are interwoven in our activity and language i.e. we consider spatial structure and action as being part of language. We consider 'space' to be one of the physical phenomena which has the potential to be used as a symbol, and

so consider ‘place’ to be a space interpreted as a symbol in a language, given meaning by its patterns of recurrence in human use. The meaning of a space is its use in the language, as understood in and through the activity of those who use it.

Their principle “Space is the opportunity; place is the understood reality” now can be seen in the same way that a pattern of sound waves can be a word, a curve of ink can form a letter, a move of the hand a subtle gesture. In each case, the former is a perceivable pattern in one or more physical phenomena, which has the potential to be used symbolically, while the latter is the symbol in language. We continually mix phenomena in our everyday communication, and spatial media are an essential part of that mix. While space has its unique characteristics that differentiate it from other media, it has no privileged position above or apart from them.

The differences between media are usually very obvious. The varying physical phenomena involved, and the approaches to relatedness and differentiation that are particular to each are to some extent what distinguish media and let us treat each medium as if it were an isolated individuated entity. On the other hand, we tend to overlook their similarities. Their common potential for use in language makes each medium of information representation potentially semiological, and therefore symbols in different media can become part of the same language. Hence we see ‘appropriate behavioural framing’ as the understanding of language, as semiological activity with social and structural norms.

This standpoint, interweaving phenomena and media such as space, data and text, also offers an alternative view of the ‘complex forms’ of Harrison and Dourish. The first such form is the ‘spaceless place,’ such as Usenet news groups. Spaceless places are navigated and used by means of relationships that are non-spatial but that nevertheless support “the tension between connectedness and distinction which leads to placefulness.” As they put it, different social norms make for different places without an underlying notion of space. The concept of spaceless places seems complex when media are conceived of as disjoint, but we see language as the underlying notion of both spatial and non–spatial media. Language unites them and makes them ‘placeful’ or understood.

In the second class of complex forms, ‘hybrid spaces,’ physical and virtual spaces merge or overlap. Actions in physical space may be ‘projected’ into a media space by means of cameras. While it may be difficult for the people involved to handle symbols that have been projected into a new context, one can consider these actions as involving a common set of people, objects and symbols, perceived through various media. Difficulty stems not so much from the fact that a combination of media is involved, but that the particular representations and interactions that combine those media, i.e. the design, are a poor fit with the people and their activity.

Rather than seeing hybrid spaces as a special case, we find it difficult to identify spaces that are *not* hybrid. Actions, words, gestures, concepts and goals that are not particular to one medium, workplace or information space are inevitably involved, and span such spaces because the same people are involved in them all. Media spaces, virtual worlds and all information representations would be useless if they did not overlap with and share references to our everyday verbal, written and gestural language. As Harrison and Dourish themselves put it, “after all, a virtual world filled with virtual offices and virtual desks isn’t populated by virtual people, but by real ones.”

Inspired by the notion of space and place, the Placeless Documents project at Xerox PARC has tried to bridge media usually treated separately, and to blur the boundary between subjective and objective information representations [12]. Placeless’ focus is on representing textual documents, work processes, people, system services and devices as flexibly interpretable sets of properties, in the form of (name, value) pairs. Properties may be objective, as in the length of a file in bytes, or subjective as in a tag to say that a document is important for my meeting tomorrow. Normally disparate types are treated uniformly and interwoven in Placeless, as it allows document manipulation tools to be used to search, classify and modify represented objects via the single underlying medium of properties. In a text editor, a letter may be presented and edited in the normal way, although Placeless represents the content, date of modification, author and so forth not using the traditional file system but as a set of properties in its database. A workflow process for a travel approval may also be represented as properties in the database. These properties may be selected to suit the reader, translated into text form, and passed to the same everyday editor as was used to write the letter. Editing and saving the document will change the underlying properties and trigger the appropriate follow-on actions, as in a specialised workflow interface.

Looking back at the earlier discussion of space and place, we suggest that 'placeless' here means flexibly interpretable because of a moderate level of distanciation. Artifacts may exist in one traditional medium but be translated and reinterpreted as another, Placeless' properties. This formal representation is designed to allow reinterpretation and reuse in ways that are not tightly constrained by the traditional tools, norms of use and 'appropriate behavioural framing' associated with an artifact's traditional medium.

Placeless takes many media and reduces it to one, the property. We see the property as analogous to the semiological symbol. By and large Placeless offers reinterpretation and reuse of properties in only one medium, the textual document. Document tools are familiar and powerful to many users, and the corporate setting of Placeless makes a document-centred focus almost inevitable, but we see the potential for reinterpretation of Placeless properties in other media. Properties could be translated into patterns of sound suitable for the visually impaired, or into patterns of tangible physical objects, as in Durrell Bishop's Marble Answering Machine [6].

We can now take a similar look back over our work with Recer, and how we mix and fix symbols from different media. Like Placeless, we reduce a number of media and types to one, path entries. The way we do this is to some extent determined by the type system and functionality of each tool. For example, in logging activity in the xemacs editor, we are constrained by what xemacs represents as manipulable entities and which Lisp functions it has to handle them. In display and recommendation, we must make choices as to which tools to use. We decided on a simple approach, splitting display between editor, browser and Recer's own interface in a rather primitive way that served our experimental purposes at the time. When a user clicks on a recommended URL to trigger display, it is sent to the web browser. Clicked–on filenames are sent to xemacs. We could potentially send a file to the web browser, or print it out, or send it to a speech synthesiser for audio output, or roll a particular marble out onto a rack. However, irrespective of the media involved in logging and display, our algorithms for searching, matching and recommending do not rely on distinctions between these original media. They rely on their common semiological use in human activity, and hence on time, person and context. It is this change in emphasis that lets us bridge across media usually held as separate in information systems, and lets us work with this everyday mix of types rather than against it.

Recer focuses on the medium of path entries, and relies on other tools to handle logging, analysis and presentation specific to other media. It concentrates on a form of analysis of symbols and contexts particular to path entries, and leaves more objective and context-independent analysis to other tools. We may plan to increase the variety of tools that feed Recer with path entries, and that can be used for display and awareness, but it is only as part of this system of interwoven tools that path systems gain meaning and utility. Similarly, it is the pattern of differences in types handled and phenomena employed that defines where Recer and the path model fit in our space of approaches to awareness. Our work is close to Placeless, space/place and the CIS, for example, but still related to and dependent on IR and objective content analysis. We see all of these as tools we can use as we deem appropriate, seeing spatial and linguistic concepts, formal and informal representations, contextual and non-contextual tools, and subjective and objective analyses as different parts of the same language. Once again, Wittgenstein has been here before:

> Ask yourself whether our language is complete;—whether it was so before the symbolism of chemistry and the notation of the infinitesimal calculus were incorporated in it; for these are, so to speak, the suburbs of our own language. [...] Our language can be seen as an ancient city: a maze of little streets and squares, of old and new houses, and of houses with additions from various periods; and this surrounded by a multitude of new boroughs with straight regular streets and uniform houses.

5. Conclusion

In discussing how representation affects awareness of information and of people, we have taken a broad view spanning systems and theory. Throughout the paper we have used a semiological perspective, emphasising the subjectivity and contextuality of interpretation. We began by looking at one particular system, Recer, and how its approach to representation, formalisation and interpretation let it interweave types of information usually analysed separately, tools generally treated in isolation and individuals often treated as working alone. We compared the path model with other approaches to accessing and reusing computer-based information, looking at one very formal and objectifying approach, information retrieval, and one more informal approach, software patterns. We explored how awareness is affected by the design and interpretation of the shared artifacts, formal or informal, at the core of any such approach. We looked at

how CSCW has applied and developed concepts relating awareness and representation, attempting to equate or blur boundaries between concepts such as the CIS, space/place, and language.

In our current work, we are working on tools to improve selection and display of the context of recommendations and the people involved, and to allow the period of the past used for recommendations to begin and end as desired. One special case of the latter would be to make recommendations come only from the most recent activity of one's colleagues, so that Recer would act as a service for notification of near-current activity. Also, we will soon begin to record activity from a wider range of sources such as mail tools and spatial location sensors. We wish to find a larger group of people willing to be tracked in their activity, so as to experiment with longer-term and larger scale use. For example, we are exploring the possibility of supporting and enriching the information offered to museum visitors as they walk and browse amongst rooms and artifacts. And in theoretical areas, we wish to deepen our understanding of writers such as Wittgenstein and Heidegger, and of what philosophy of language can offer CSCW.

Overall, our aim is to let our system design practice be in accord with our theoretical standpoint, seeing them as mutually informative and mutually dependent. We wish to explore system designs where, in a way that matches contemporary semiology's and philosophical hermeneutics' treatment of human language, distanciation of deep system structure—of what serves as our computers' 'language'—can be relaxed to a greater extent than we in CSCW have allowed before. In doing this we will see what happens when we, as programmers and designers, give up progressively more control of how users classify, interpret and work with their own information.

References

1. Alexander, C. *The Timeless Way of Building*, Oxford University Press, 1979.
2. Bannon, L. & Bødker, S. *Constructing Common Information Spaces*, Proc. ECSCW 97, Lancaster, Sept. 1997, 81–96.
3. Chalmers, M., Rodden, K. & Brodbeck, D. *The Order of Things: Activity–Centred Information Access*, Proc. WWW7, Brisbane, April 1998.
4. Chalmers, M. *Informatics, Architecture and Language*, in Social Navigation in Information Space, Munro, A., Höök, K. & Benyon, D. (eds.), Springer, 1999.
5. Chalmers, M. *Comparing Information Access Approaches*, J. American Society for Information Science 50th Anniversary Issue, 50(12):1108-1118, September 1999.
6. Crampton Smith, G. *The Hand That Rocks the Cradle*. I.D., May/June 1995, pp. 60-65.
7. Dourish, P. and Bly, S. *Portholes: Supporting Awareness in a Distributed Work Group*, Proc. ACM CHI'92, Monterey, 1992.
8. Gamma, E. at al., *Design Patterns: Elements of Reusable Object–Oriented Software*, Addison Wesley, 1995.
9. Goldberg, D., et al. *Using Collaborative Filtering to Weave an Information Tapestry*, Comm. ACM 35(12), Dec. 1992, 61–70.
10. Harrison, S., & Dourish, P. *Re–Place–ing Space: The Roles of Place and Space in Collaborative Systems*, Proc. ACM CSCW'96, 1996.
11. Kristoffersen, S. & F. Ljungberg, *An Empirical Study of How People Establish Interaction: Implications for CSCW Session Management Models*, Proc. ACM CHI 99, 1-8, 1999.
12. LaMarca, A., et al., *Taking the Work out of Workflow: Mechanisms for Document-Centred Collaboration*, Proc. ECSCW 99, Copenhagen, 1999.
13. Resnick, P. & Varian, H. (eds.) Comm. ACM special issue on Recommender Systems, 40(3), March 1997.
14. Ricoeur, P. *Hermeneutics and Human Sciences: Essays on Language, Action, and Interpretation*, trans. J. Thompson, Cambridge University Press, 1981.
15. Sarwar, B. et al. *Using Filtering Agents to Improve Prediction Quality in the GroupLens Research Collaborative Filtering System*, Proc. ACM CSCW 98, 345-354, 1998.
16. Schmidt, K., *Of Maps and Scripts: The Status of Formal Constructs in Cooperative Work*, Proc. ACM Group 97, Phoenix, 138–147, 1997.
17. Suchman, L. (1987) *Plans and Situated Actions: The Problem of Human Machine Communication*, Cambridge University Press.
18. Winograd T. & Flores, F. *Understanding Computers and Cognition*, Addison Wesley, 1986.

19. Wittgenstein L. Philosophical Investigations (3rd ed.). Trans. G.E.M. Anscombe, Oxford University Press, 1958