



Finding and understanding reusable designs from large hierarchical repositories

Peter Demian¹
Renate Fruchter²

¹Department of Civil and Building Engineering, Loughborough University, Loughborough, Leicestershire, UK; ²Civil and Environmental Engineering Department, Stanford University, Stanford, CA, USA.

Correspondence:

Peter Demian, Department of Civil and Building Engineering, Loughborough University, Loughborough, Leicestershire LE11 3TU, UK.

E-mail: P.Demian@lboro.ac.uk

Abstract

This paper describes a prototype called CoMem (Corporate Memory) that supports the finding and understanding of useful items in large hierarchical repositories. The particular domain is civil engineering design, and the prototype is designed specifically to support design reuse in building construction projects. However, the underlying visualization and interaction principals behind CoMem are generalizable to the ubiquitous task of finding and understanding useful information in large hierarchical repositories. To support the finding, the entire hierarchy is visualized using a squarified treemap. Once an item from the treemap is selected, CoMem supports the understanding of that item by identifying related items in the hierarchy and visualizing the selected item in the context of these related items in a node-link diagram. The paper concludes with a brief discussion of a usability evaluation of CoMem that supports the claim that *finding* and *understanding* improve the process of reuse, and that the described visualizations assist with finding and understanding.

Information Visualization (2006) 5, 28–46. doi:10.1057/palgrave.ivs.9500114

Keywords: Information visualization; tree layout; hierarchical visualization; navigation; focus + context; fisheye view; interactive visualization

Introduction

The task of finding an inconspicuous piece of information in a large repository is becoming increasingly ubiquitous. With the exponential growth of the World Wide Web (WWW), and more and more of the world's information being collected, stored, and made accessible, we increasingly find ourselves in the position of needing to find the proverbial needle in a haystack.

This research investigates how the process of design reuse can be supported by a computer system that visualizes a repository of past designs from previous engineering design projects. Why is it that design reuse is very effective when one is reusing knowledge from one's own personal experiences or internal memory, but is often unsuccessful when one is reusing designs from an external computer repository?

This research distinguishes between two types of reuse:

1. *Internal knowledge reuse*: a designer reusing knowledge from his/her own personal experiences (internal memory).
2. *External knowledge reuse*: a designer reusing knowledge from an external knowledge repository (external memory).

Our ethnographic observations of designers at work show that internal knowledge reuse is effective since:

- The designer can quickly *find* (mentally) reusable items.

Received: 26 July 2005

Revised: 19 October 2005

Accepted: 10 January 2006

Online publication date: 16 March 2006

- The designer can remember the context of each item, and can therefore *understand* it and reuse it more effectively.

These observations of *internal knowledge reuse* are used as the basis for improving *external knowledge reuse*. External knowledge reuse can be supported by enabling designers to *find and understand* reusable designs. Our objective is to improve and support the process of external knowledge reuse in engineering design industries such as building design.

In our case, as in most engineering design situations, the data being managed is hierarchical in nature. As the design team members collaboratively develop the CAD model for the building project, they communicate and collaborate by creating discipline and component objects, and linking those objects to geometrical entities from the CAD model. A *discipline* object encapsulates a portion of the design from a particular point of view such as a discipline (for example, architecture), a subsystem (for example, Heating Ventilation and Air Conditioning), or a general issue (for example, cost). A *component* object is a design feature over which the design team collaborates. Component objects can also be linked to notes exchanged by the designers or to external files or documents. A company using CoMem would accumulate a hierarchical corporate memory consisting of multiple project objects, each project object containing multiple discipline objects, and each discipline object containing multiple component objects, with individual component objects being linked to objects in blinding CAD models.

How can *finding and understanding* be supported in large hierarchical repositories? This research deals with a three-level hierarchy with between 1000 and 10,000 nodes. This study argues that *finding* can be supported by providing an *overview* of the corporate memory which displays *all items* at a glance, and providing the user with filtering and navigation tools. This argument is based on

ethnographic evidence and the related research summarized below. *Understanding* can be supported using a *context explorer* that displays the found item in context, and allows the user to explore related contextual items from the hierarchy.

The need for an overview of the repository to facilitate finding

An overview of large information spaces reduces search, allows the detection of overall patterns, and aids the user in choosing the next move.¹ It is not obvious that an overview is the best approach for helping the user to find items from a large repository, especially if only a tiny fraction of the items shown on the overview are of interest to the user. For example, when submitting a query to a web search engine, it would be of little use to the user to see an overview of the entire WWW.

Two characteristics of a corporate memory make it well suited to an overview. Firstly, it is a much smaller repository than the entire WWW, and so an overview is a realistic approach. Secondly, the corporate memory does not consist of a flat list of documents as most document repositories do. It is composed of a hierarchically structured collection of projects, building subsystems or disciplines, and individual components. It has been observed from the ethnographic study that the designer will need to make comparisons at all three levels of granularity simultaneously (Figure 1). This suggests the use of some visual overview that allows such comparisons to be made, rather than returning a flat list of 'hits' that satisfy a query specified by the user.

Related work on overviews

The idea of *overview and detail*, that is, providing an overview for orientation and one or more detailed views for further work, has been used for many years in information visualization systems. It is formally articulated by Card *et al.*¹ (section 3.3).

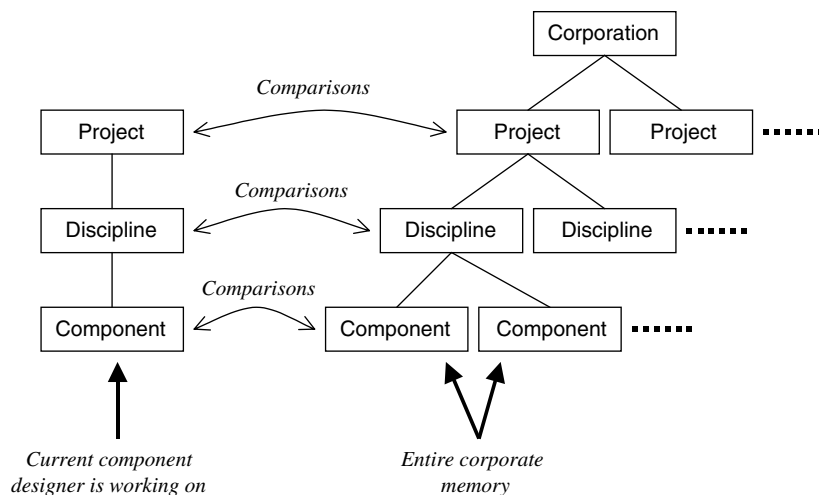


Figure 1 The designer makes comparisons at all three levels of granularity when finding reusable items.

The need for an overview is implied by *information foraging theory*.² According to this theory, users of information will modify their information seeking strategies or the structure of the environment to maximize their rate of gaining valuable information. These modifications are largely based on *information scent*.³ Information scent is the imperfect perception of the value, cost, or access path of information sources obtained from proximal cues, such as bibliographic citations, WWW links, or icons representing the sources. For an overview and detail interface, each item of detailed information must leave a sufficiently strong information scent in the overview in order to enable the information forager to make an informed decision about which piece of information to pursue.

Overviews have also been studied in the context of web site navigation. The objective of web navigation design is to enable the user to answer the following questions:⁴

1. Where am I?
2. Where have I been?
3. Where can I go?

For the purposes of CoMem, the third question is the most important. The symbols on the Overview must serve as sources of information scent, that is, proximal cues to the chunks of information that they represent. The user's perception of this information scent enables him/her to make an informed decision about where to go next.

Constantly visible site maps (comparable to overviews) were found to improve performance in information seeking tasks.⁵ They were found to provide a bird's eye view, reminding the user that there is a whole world out there waiting to be explored. Users' web navigation habits are often characterized using a 'hub and spoke' model.⁶ The overview can serve as a hub that provides links to all the items of information that are reachable.

Finally, some researchers have recognized the problems associated with showing everything on the overview, particularly for large information spaces. In addressing the question of whether the overview should show everything, Darken and Sibert⁷ write: 'Putting more and more on a map is like jumping on a hot air balloon to get a good view of the city. The higher you go, the more you can see, but the increased altitude also decreases the strength of the stimuli.'

The map metaphor

CoMem uses a map metaphor for the Overview. This metaphor emerged from the scenario-based design process of CoMem as a useful way for thinking about the Overview. A map is traditionally defined as 'a representation of things in space'. Recent definitions have shifted the emphasis from strictly objective representations of physical space to more subjective representations that facilitate the spatial understanding of things, concepts, conditions, processes, or events in the human world.⁸ This more contemporary definition brings maps within

the domain of *visualization*: the use of visual representations to amplify human cognition in support of particular tasks¹ (Chapter 1). Visualizations exploit human visual perception to amplify cognition to an extent that would be impossible using non-visual forms such as purely symbolic or textual representations. In particular, CoMem endeavors to exploit two properties of maps:

- The *spatial* property. A map is usually a smaller scale representation of an actual physical space. This small-scale representation effectively communicates the properties of *containment* and *proximity* between the entities represented on the map: San Francisco is in California; San Francisco is near San Jose.
- The *semantic* property. By overlaying certain marks (points, lines, areas, all represented according to some visual vocabulary) on the mapped space, a map is able to convey additional information (beyond containment and proximity) efficiently and rapidly. For example, political maps, topographic maps, natural resources, etc.

A useful example is that of a weather map. The weather map, as a map of the local geography, is useful in its own right for someone unfamiliar with the area. As the average newspaper reader is familiar with his/her local geography, simply by glancing at the weather map each morning, he/she can tell what the weather will be like in his/her area and the surrounding areas.

The Overview should express the 'geography' of the corporate memory: which projects contain which disciplines and components, and which items are 'close' to each other. The CoMem user is expected to develop a familiarity with the geography of the corporate memory. Given a problem that he/she is working on, the map will appear with different areas highlighted to indicate that they are potentially reusable, and the user can tell at a glance which parts of the corporate memory to explore further. For novice users, areas of the map can be highlighted according to CoMem's measure of relevance to the users' current design problem. Expert users who do not wish to depend on CoMem's relevance measure can input their own queries, and the results from these queries are highlighted on the map. This is comparable to different information being superimposed on the map: weather, topography, political boundaries, resources, population density and so on.

Modern writers on the history of cartography emphasize that it is impossible to study a map without considering its social context and the tasks for which it was intended. For example, Harley⁹ rejects 'cartographic positivism', the notion that cartography is objective, detached, neutral, and transparent. He denies that maps can be true or false, 'except in the narrowest Euclidean sense'. In this sense, an accurate roadmap is not one that accurately depicts the roads, but one that will help a traveler to reach his/her destination. It is in this sense that this paper talks about 'designing' the Corporate Map.

Treemap visualization

The corporate memory is hierarchical, where a corporation contains multiple projects, a project consists of multiple disciplines, and a discipline contributes multiple components. This hierarchy can become very large (up to 10^5 items in large corporate repositories containing engineering designs). The Overview needs to show an abstraction of the entire corporate memory in a single display, which would enable the user to decide which items to view in more detail.

In a treemap, projects, disciplines and individual components are represented as nested rectangles. The size of each rectangle is mapped to a measure of how much ‘knowledge’ this node encapsulates. For example, an object that has a rich version history and is linked to many external documents and annotations will be assigned a larger area. The color of each rectangle is mapped to a measure of how relevant this object is to the designer’s current design task.

Treemaps are particularly effective for large fixed-depth hierarchies such as our data structure. They make full use of the available display space. If properly designed, they can support comparisons and assessment of relevance at all three levels of granularity simultaneously.

They arguably complement the map metaphor. In light of the above discussion of maps, the treemap maps the parent-child link relationship to enclosure in 2D space. It therefore conveys containment. It does not, as yet, convey proximity between similar siblings. The idea of mapping similarity between siblings to proximity on the treemap was touched upon by the developers of the ordered treemap algorithm.¹⁰ On a purely visual level, Fiore and Smith¹¹ compare a treemap to a land-use map. They note that it is tempting to compare heavily subdivided rectangles to busy urban areas and large rectangles to calmer rural areas. However, such a reading is flawed. The large rectangles represent not empty plains but vast leaf-nodes with huge amounts of data.

The classic treemap algorithm¹² uses a slice-and-dice approach, subdividing each rectangle either vertically or horizontally amongst a node’s children. The most important disadvantage of the classic treemap is that its rectangles can have high aspect ratios, which makes it difficult to select or label the rectangles, compare sizes, or perceive structural relationships between nodes.

The squarified treemap algorithm¹³ and the clustered treemap algorithm¹⁴ attempt to minimize the aspect ratios of the rectangles. Both algorithms produce very similar treemaps, although the squarified algorithm was found to give slightly lower aspect ratios.¹⁵

The most important disadvantage of squarified and clustered treemaps is that changes in the sizes of the nodes produce dramatic changes in the layouts produced, which can be disorientating for the user. For example, the user might be used to seeing the architecture discipline of the Bay Saint Louis hotel always at the top left corner of the map. If this discipline increases slightly in size (for example because a new component is added to it) then it

can suddenly jump to the lower right corner because this layout produces the most squarified rectangles. Such drastic layout changes prevent the user from becoming familiar with the ‘geography’ of the corporate memory.

Another algorithm, the ordered treemap,¹⁰ addresses this issue. It attempts to maintain proximity relationships between nodes, which discourages large layout changes in dynamic data. The cost of this is slightly higher aspect ratios than those produced by the squarified or clustered algorithms. Ordering is discussed in more detail below.

Treemap design issues

Emphasizing structural relationships

In most of the treemap applications presented in the literature, only the leaf nodes are of interest. Non-leaf nodes are important mainly for emphasizing structural relationships (i.e. grouping siblings together). Three techniques have been proposed for emphasizing structural relationships: framing,¹² padding^{11,16} and cushions.¹³

In CoMem, non-leaf nodes (projects and disciplines) must appear distinctly in the treemap so that they can be selected and explored independently of the components they contain. This is important for two reasons:

- A designer might want to reuse an item at the project or discipline levels of granularity.
- Even if the designer is only interested in reusing components, he/she will need to assess whether a potentially reusable component comes from a similar discipline and project to the current design task.

CoMem uses *padding*, where gaps are left between a node’s children, leaving the parent node visible behind the children. This enhances the map metaphor, giving the treemap the appearance of a contoured topographic map. Padding space is left around the perimeter of a set of siblings (Figure 2, top row), as well as between siblings (Figure 2, bottom row).

Two further modifications are made to help emphasize structural relationships. Firstly, the amount of padding is increased with increasing level of granularity, so that sibling projects have more padding space between them than sibling components. Secondly, the rectangle outline thickness is increased with increasing level of granularity, so that project rectangles are drawn with thicker lines than component rectangles.

Both of these measures eliminate the maze-like appearance of large treemaps, and help the designer to tell instantly whether the highlighted node is a project, discipline or component, and how relevant its ancestors and descendants are. However, these measures also reduce the density of the treemap (as a result of the padding, some smaller nodes are not drawn at all) and its accuracy (the padding distorts the relative sizes of the siblings). The resulting treemap is shown in Figure 3.

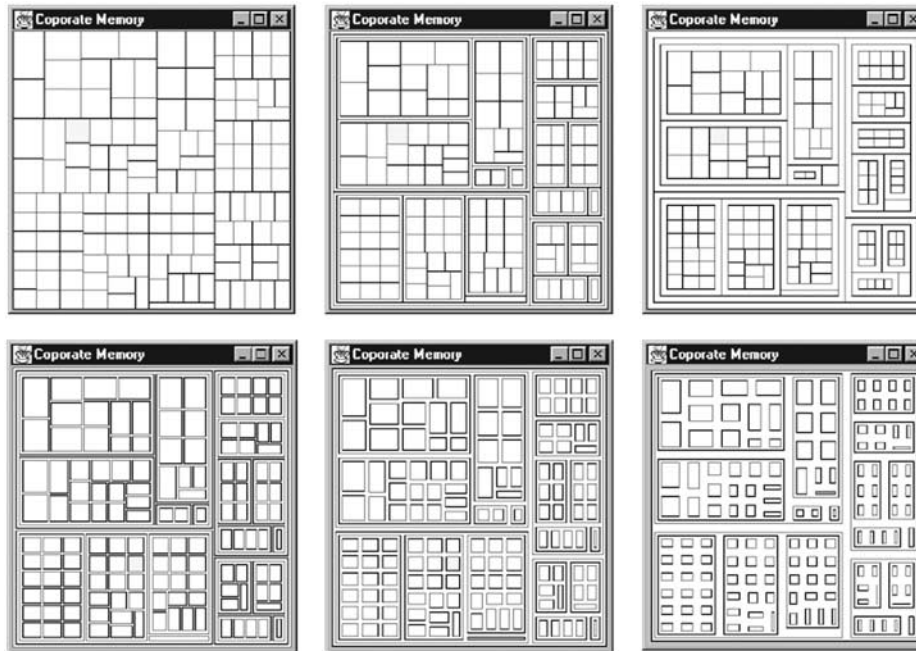


Figure 2 Increasing borders around sets of siblings (top row) and between siblings (bottom row).

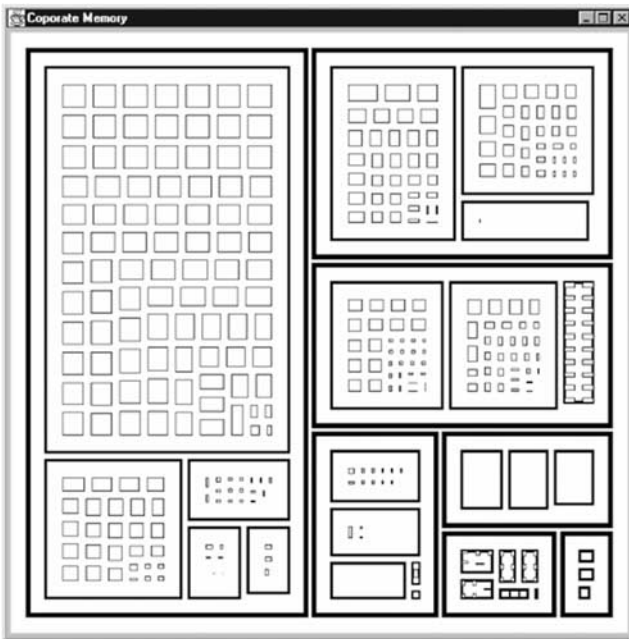


Figure 3 Treemap with varying padding and line thicknesses to help emphasize structural relationships.

Size function

The size of each rectangle is mapped to a measure of how much content this node contains. For project and discipline objects, this size will be the sum of the sizes of the constituent component objects. For a component object, the size is a function of the number of versions of

this component and the number of external documents linked to it.

The screenshots in Figure 4 were produced with a simple size function where the size of a component is a function only of the number of times this component was versioned. Functions that produce a wide distribution of sizes make structural relationships more apparent in squarified or clustered treemaps. The difference can be seen by comparing treemaps (a) and (b) in Figure 4.

With padding it is not necessary to exaggerate the size distribution. In treemap (d) at the lower right of Figure 4, where the number of versions is raised to the power 2.5 to give a large size distribution, large nodes are disproportionately favored while smaller nodes are not drawn at all. CoMem currently uses a simplified size function where the size of each node is directly proportional to the number of times it is versioned.

Color

The color of each rectangle is used to encode the relevance of this item to the designer's current design task, based on an automatically generated measure of relevance.¹⁷ This relevance measure is always in the range 0 to 1 and is used to generate a color that is a linear interpolation between pure red (for relevant items) and pure blue (for irrelevant items). Combined with the padding, which enables the designer to see the underlying rectangles for projects and disciplines, this coloring is extremely effective for making comparisons at all three levels of granularity simultaneously. The combination of padded treemaps with coloring according to relevance to

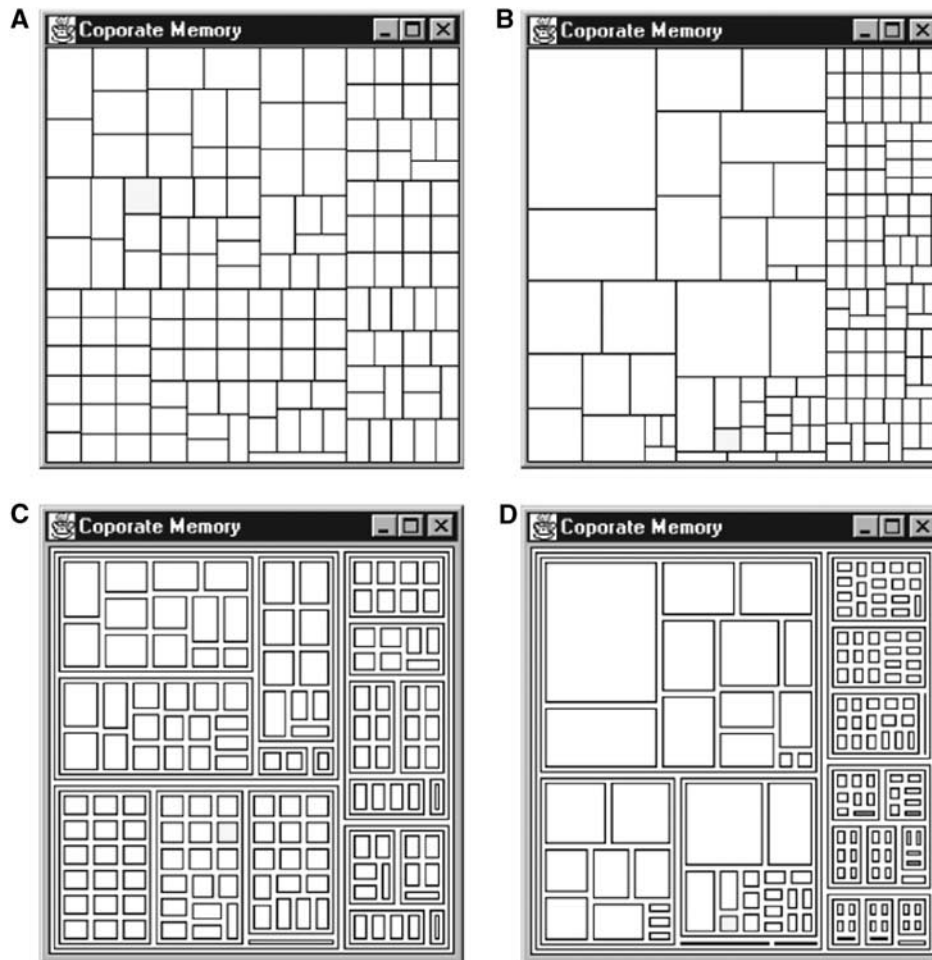


Figure 4 Experiments with the size function. Uniform size function (left column) vs exaggerated size function (right column); no padding (top row) vs padding (bottom row).

facilitate comparisons at all levels of granularity is an important contribution of this research.

Ordering

The squarified algorithm lays out siblings in order of decreasing size. This generally leads to rectangles with smaller aspect ratios. However, this results in an arbitrary placement with regards to similarity. In keeping with the map metaphor, it is desired that ‘similar’ siblings be laid out closer to each other. This would result in similar subsets of siblings forming meaningful ‘regions’ on the map which, when the relevance measure is indicated on the map, would appear as patches of high or low relevance. The user can therefore explore relevant regions more closely. As similar nodes are laid out closer together, the user is more likely to find relevant items serendipitously while exploring a nearby node.

One possibility is to use the ordered treemap algorithm, ordering the nodes by their relevance to the current design task. This would produce relevant regions as

desired. However, the generated layout will depend on the designer’s current design task (from which the relevance measures are calculated). This means that the designer will see wildly varying layouts depending on the current design task he/she is working on. The aim is for the colors only, and not the treemap layout, to depend on the current design task. A better approach would be to order siblings by some static attribute, so that similar components are laid out near each other.

Another possibility is to pre-compute an affinity matrix for each set of siblings that expresses how similar a node is to each of its siblings. This affinity matrix can then be used to lay out similar nodes closer to each other. There is currently no algorithm that lays out a treemap using an affinity matrix, and this is one possible direction for future research in treemaps.

Currently, CoMem uses the squarified treemap algorithm, and so variations in layout as the corporate memory grows and evolves remain a challenge to be addressed in future studies.

Labels

Each rectangle on the treemap serves as an ‘information scent’ leading to the item it represents. The size and color of this information scent (‘how much content will I find there, and how relevant is this content?’). Text labels can significantly increase this information scent by jogging the user’s memory, particularly for retrieval tasks where the user already has some idea what he/she is looking for.

Labeling treemaps is particularly challenging. CoMem centers each label horizontally over its rectangle. Vertically, the labels are positioned either one-third of the height from the top or one-third of the height from the bottom, alternating between the two for adjacent rectangles. This improves the distinctiveness of each label; with the labels simply centered vertically, the labels for a row of rectangles were found to resemble a continuous line of text.

The labels can either be scaled to fit the rectangle, or fixed sizes can be used, with project labels being the largest and component labels the smallest. CoMem currently provides both options. Figure 5 shows the labeling options available in CoMem. The advantages and disadvantages of each approach are listed in Table 1. Generally speaking, the scaled labels are more effective. If the label is scaled down below a threshold value, it is not drawn at all. In the case of the fixed-size labels, choosing the label font sizes is extremely difficult because the rectangles vary widely in size within each level of granularity.

With both scaled and fixed-size labels, occlusion of underlying labels is problematic. To address this, CoMem can draw partially transparent labels. The transparency is increased with increasing level of granularity (i.e. project labels are almost completely transparent and the component labels are completely opaque). The rationale behind this is that project object labels are more likely to be large and therefore occlude other labels. However, this is not always the case, particularly for disciplines or components with very short labels that become large when scaled up to fit the rectangle. In such cases, the label of a discipline can be larger than that of its project, and the fact that the discipline label is more opaque can be confusing. Another possibility is to assign the transparency of each label based on its size rather than level of granularity.

One simple refinement of scaled labels that addresses the problem of short labels appearing disproportionately large is to enforce the rule that no item is to have a larger label than its parent. A further refinement is to draw discipline labels in a different color so that the user can tell whether a label refers to a project or a discipline (Figure 6).

The improvement of labeling is one possible direction for future research in treemaps. As a supplement to the labels painted on the treemap, CoMem also displays the description of each rectangle in the form of a ‘tooltip’. If the user briefly lingers with the mouse over a rectangle,

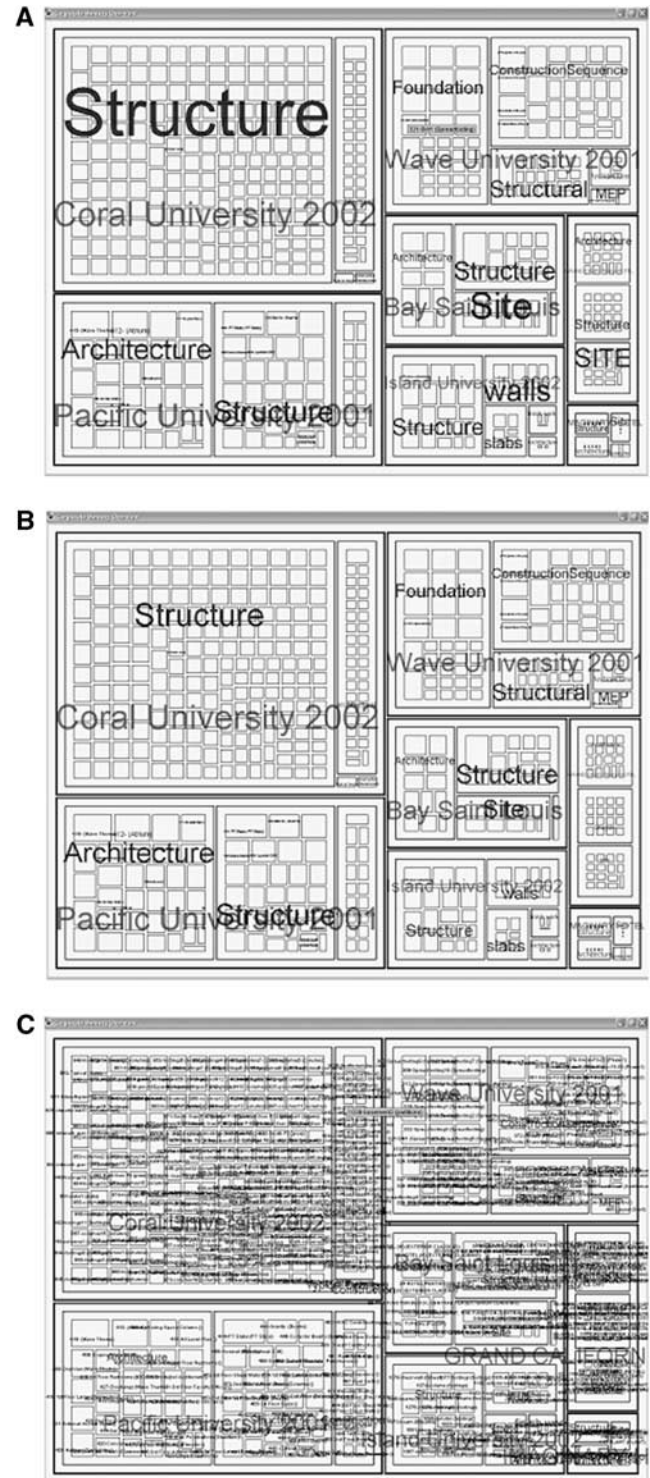


Figure 5 Labeling treemaps. (A) Labels are scaled to fit the rectangle; (B) labels are scaled to fit the rectangle but cannot exceed the size of the parent label; (C) fixed sizes of labels are used for project, discipline, and component labels, in which case all labels are drawn. In this last figure, the density of the text renders all labels illegible.

Table 1 Advantages and disadvantages of various labeling options

	Advantages	Disadvantages
Scaled labels	<p>The labels are as large as space permits, and so they are generally more legible.</p> <p>The user can easily associate the label with its rectangle (because the label extends from one end of the rectangle to the other).</p> <p>The size of the label functions as a convenient criterion for choosing which labels to draw: if after scaling the label will be too small to be legible, it is not drawn at all.</p>	<p>Short labels, when scaled up, are disproportionately prominent. For example, a discipline label can appear more prominently than the project to which it belongs because the discipline has a short name.</p>
Fixed-size labels	<p>It is easy to identify the level of granularity of an item from the font size of its label.</p>	<p>It is almost impossible to select suitable font sizes to use for the whole treemap, because the rectangle sizes vary widely within each level of granularity.</p> <p>If the rectangle is too small, the label can overflow outside the two sides of the rectangle. If the rectangle is too large, the label is drowned by empty space. In both cases, it becomes hard to associate the label with its rectangle.</p> <p>There are no obvious criteria for choosing which labels to draw (given that with <i>all</i> labels drawn, occlusion renders the whole treemap almost illegible). One possibility is relevance.</p>

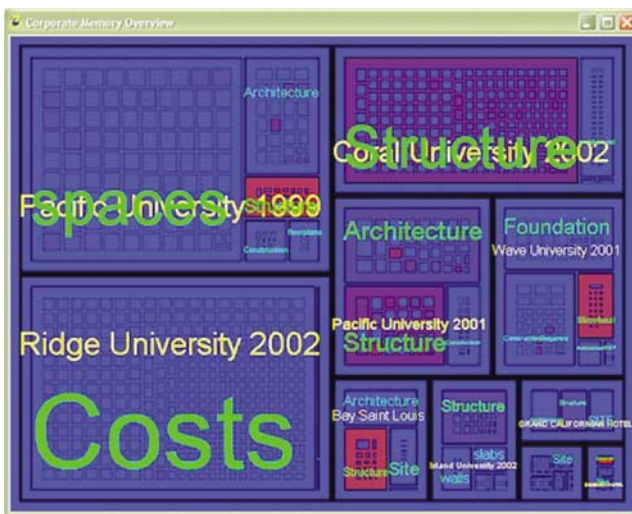


Figure 6 A labeled Corporate Map in which discipline labels are green, and project and component labels are yellow.

then the description appears in a small box. Therefore, the text description is available even if the rectangle is not labeled, or if the label is too small or is occluded.

Filtering

We argued that the Overview needs to show the *entire* corporate memory. However, even for a small corporate

memory, the Overview can be extremely dense to the extent that the user is unable to distinguish or click on individual rectangles. It will be necessary to allow the designer to add emphasis to certain parts of the corporate memory that are more relevant. There are two possible interaction mechanisms for adding emphasis to items on the treemap:

- Filtering out undesired items using dynamic querying
- Zooming in on potentially reusable regions of the map

CoMem currently allows the user to filter out items using *dynamic querying*. In a dynamic querying environment, search results are instantly updated as the user adjusts sliders or selects buttons to query a database.¹⁸ A designer can filter out items based on relevance, date, keywords or ownership.

Filtered items can be grayed out, allowing the user to focus on the remaining brightly colored items. Alternatively, filtered out items can be omitted, leaving more space for the remaining items. For a large corporate memory, it will probably be necessary to filter out some items in this way in order to make the remainder of the items on the map discernable. Figure 7 illustrates filtering in CoMem.

Understanding found items by seeing them in context

Once the designer has identified a potentially reusable item from the CoMem Overview, he/she needs to explore

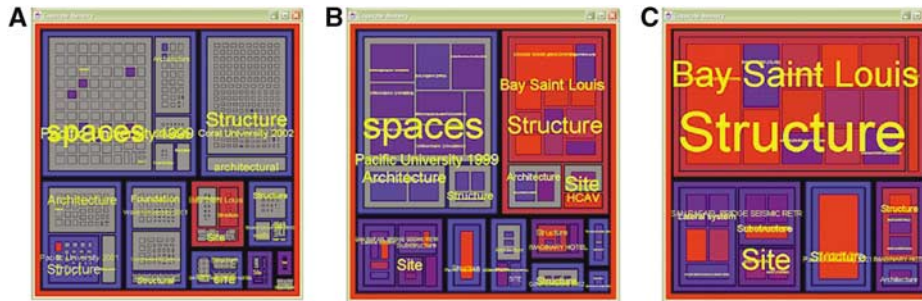


Figure 7 Filtering in CoMem. (A) Filtered items are grayed out. (B) Filtered components are not drawn at all. Filtered projects or disciplines are drawn grayed out if they have unfiltered components, otherwise they are not drawn at all. (C) Filtered components are not drawn. Filtered projects and disciplines are ‘pruned’, that is, they are not drawn, regardless of whether or not they have unfiltered children.

the project context of this item. This *focal item*, that is, the building subsystem or component being reused, was not designed in isolation but as part of a larger project. The focal item needs to be considered in its project context if it is to be understood and successfully reused.

The first constituent of the focal item’s project context consists of its *ancestors and descendants* in the hierarchy. For example, for a building structure, a braced frame is part of a larger structural system, which in turn, is part of a whole building. The braced frame consists of subparts: beams, columns, and connections.

The second constituent of the project context consists of *related items* in the parts hierarchy that lie outside of the path to the focal item and its sub-tree. A building consists of many intricately interrelated subsystems and components. The braced frame may be embedded in an architectural partition wall, or may be designed to be extra strong because it supports a library on the floor above.

From the perspective of understanding through exploring the context, two problems need to be addressed:

- Firstly, how can *related items* in the hierarchy be identified? Whereas the ancestors and descendents emerge naturally from the structure of the data, the *relatedness* between the focal node and its related items needs to be inferred.
- Secondly, how can the focal item be visualized with its ancestors, descendents, and related items so as to support exploration and understanding? This is both a visualization problem and an interaction design problem.

The fisheye lens metaphor and the fisheye view

CoMem uses a fisheye lens metaphor for the Project Context Explorer. This metaphor was suggested by Furnas¹⁹ as part of his *fisheye view*. In contrast to a zoom lens, which provides local detail at the expense of the global view, a fisheye lens simultaneously combines local detail with global context.

The *fisheye view*¹⁹ gives a methodology for generating a small display of a large information structure by controlling the field of vision, in analogy to a fisheye lens. Given a focal point, Furnas defines a *degree of interest function* over the remaining data items. Given a focal point, the user will not be equally interested in all items. Furnas decomposes a degree of interest into a *a priori* and a *posteriori* components.

The *a priori* component is a contribution to an item’s degree of interest, which transcends the given interaction, but depends on the global importance of that item. The *a posteriori* component is the contribution to an item’s degree of interest that depends on the current focal point, and is derived from some measure of distance between that item and the focal point.

Furnas goes on to describe the special case of fisheye views of hierarchical tree structures. In a hierarchical tree structure, the *a priori* component can be taken as the *level of detail* of an item (i.e. how high up the hierarchy it is). For a corporate memory, this maps to the level of granularity of each item. A project object is intrinsically more important than a discipline object, which is intrinsically more important than a component object. The *a posteriori* component can be taken as the *distance* to the focal node (i.e. the number of links in the shortest path between the focal node and the node in question).

Therefore, for a hierarchy one possible formulation is as follows:

1. Focal point: ‘.’
This is the focal item selected from the CoMem Overview, whose project context the user is exploring.
2. The *a posteriori* component of the degree of interest of node x is the distance between the focal point and x :
 $d(x,.)$
This is the number of links on the path between node x and the focal point.
3. The *a priori* component of the degree of interest is the Level Of Granularity:
 $\text{LOG}(x)$

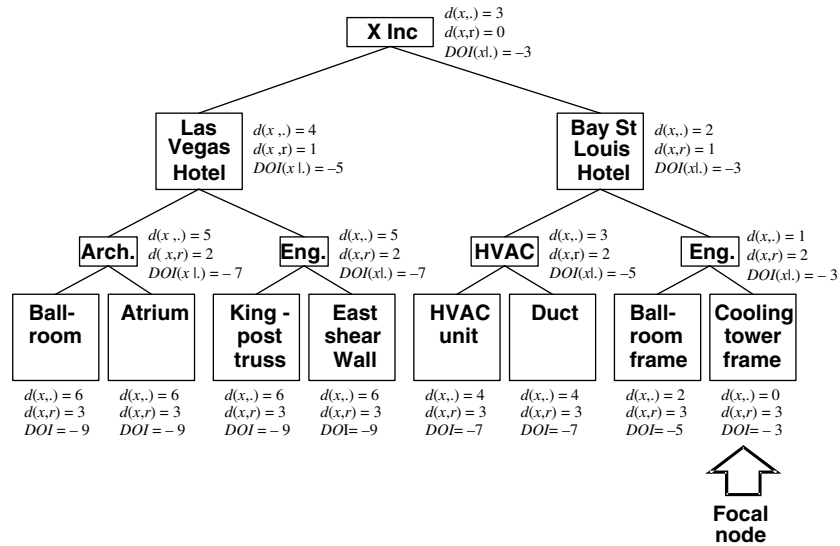


Figure 8 Degree of interest values for a small hierarchical corporate memory given the specified focal node.

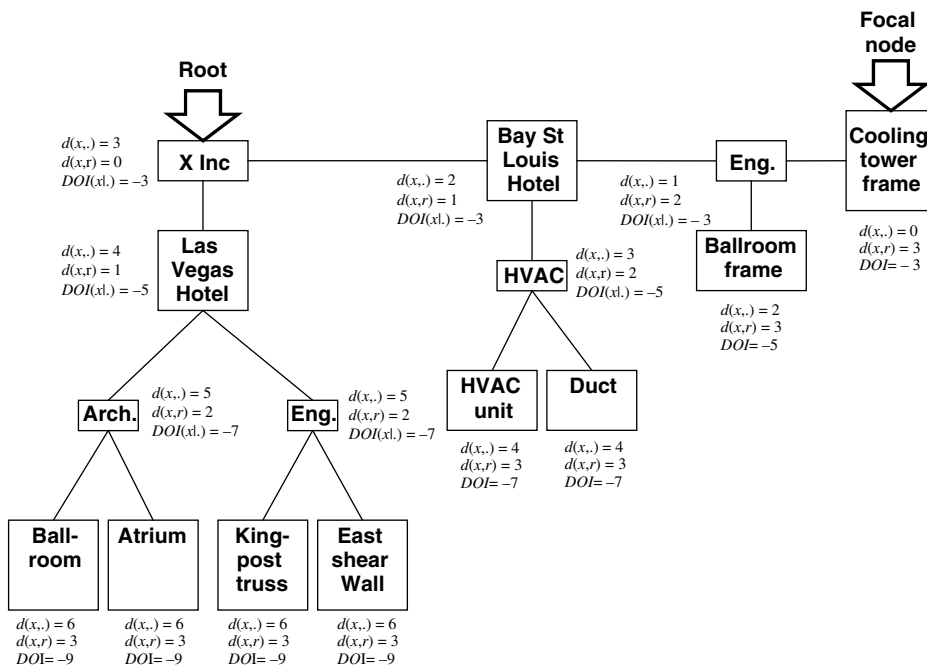


Figure 9 Iso-interest contours.

For a tree structure, this is defined as:

$$-d(x,r)$$

This is the distance between node x and the root r of the tree.

Therefore:

If x is the corporation, $LOG(x) = 0$

If x is a project object, $LOG(x) = -1$

If x is a discipline object, $LOG(x) = -2$

If x is a component object, $LOG(x) = -3$

4. The degree of interest (DOI) of node x is:

$$DOI(x|.) = LOG(x) - d(x,.)$$

$$DOI(x|.) = -d(x,r) - d(x,.)$$

Applying these equations to a small sample corporate memory (two projects, four disciplines objects, eight component objects) gives the values shown in Figure 8.

An interesting property of the formulation above is the emergence of what Furnas calls *iso-interest contours*, those points on the tree with the same degree of interest. This can be visualized by ‘picking up’ the tree from the root and the focal node, and letting the remaining nodes dangle below (Figure 9).

The fisheye view as formulated above for tree structures is useful because it addresses the first problem noted

above: how to identify related items given a focal item or node. It can be argued that items with a higher degree of interest are more closely related to the focal node, and are more likely to help the user understand the focal node. However, this formulation by itself is not sufficient to effectively identify related items because it is based only on structural relationships within the tree and does not take into account the contents of each node. For example, in Figure 9 above, both the *Las Vegas Hotel Architectural* subsystem and the *Las Vegas Hotel Engineering* subsystem are assigned the same degree of interest when compared to the *cooling tower frame* component. By common sense, the *Engineering* subsystem is more closely related to the *cooling tower frame* component, because the *cooling tower frame* is itself part of an *Engineering* subsystem from the *Bay St Louis Hotel* project.

Node-link diagrams of fisheye views

The fisheye view can be used to define a degree of interest function over a set of nodes in a hierarchy given a focal node. The issue of how to visualize such a hierarchy with varying degree of interest still remains to be addressed. Ideally, items with a higher degree of interest should be displayed more prominently.

There are two categories of techniques for visualizing hierarchies:

- Techniques using *connection* (i.e. node-link diagrams)
- Techniques using *enclosure* (i.e. treemaps)

CoMem uses a traditional node-link diagram to visualize the project context of a given item. Items are laid out in a 2D space, where the horizontal axis is the degree of interest (exploiting the iso-interest contours)

and the vertical axis is the level of granularity. Figure 10 shows this diagrammatically, and Figure 11 shows an actual screenshot from CoMem.

As with the Corporate Map, the color of each item is used to denote this item’s relevance and the size is used to denote the volume of content attached to this item. The relevance measure is calculated between the focal item and every other item in the corporate memory. This relevance measure is calculated in the same way as that for the Overview module. Whereas in the Overview each item was compared to the designer’s ‘problem item’, here each item is compared to the focal item.

Essentially, each rectangular area (quadrant) in the Project Context Explorer contains a category of relatives. In Figure 11, the focal item is a component, *36-Cooling Tower (Brace Frame)*, and it is drawn in red at the bottom left gray quadrant. The next quadrant to the right contains the focal item’s siblings. The subsequent quadrant to the right contains the focal item’s first cousins, followed by the second cousins.

Within each category of relatives, the items are sorted by relevance to the focal item, so that more relevant items appear closer to the top of the list. The display is resizable, and once space has run out, the remaining items on the list are not painted. If an item’s parent is painted, then a black line is painted between the parent and the child.

In order to support contextual exploration, if the mouse pointer is moved over any item, then that item and all of its ancestors are highlighted. If any of the ancestors have been pruned because of lack of space, they are temporarily painted in the white area at the top right of the display.

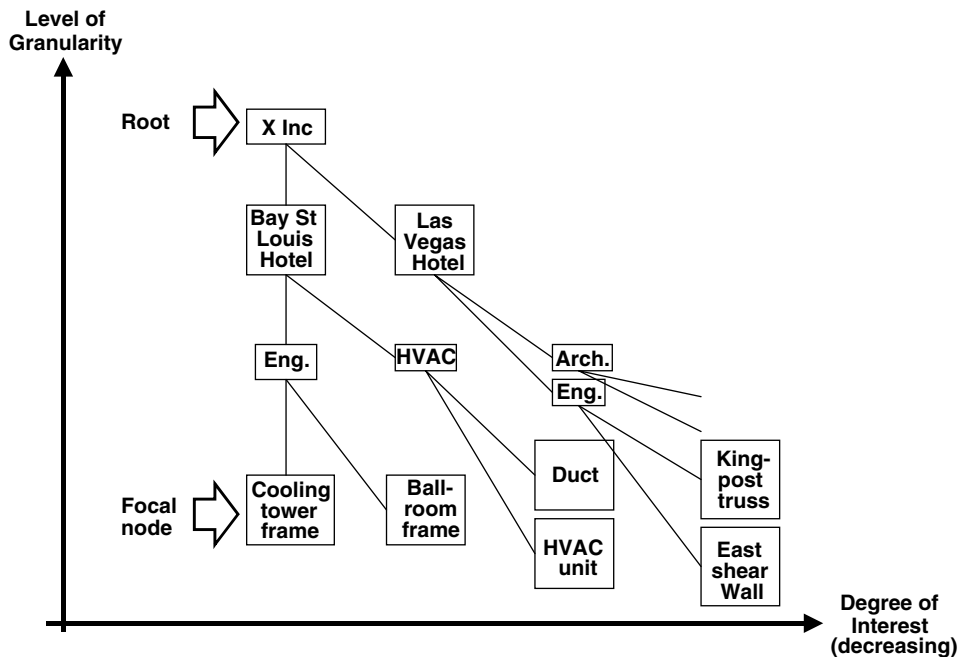


Figure 10 Diagrammatic representation of the CoMem project context.



Figure 11 Screenshots of the CoMem Project Context Explorer. The window is resizable. Each object is positioned in the vertical axis according to its level of granularity, and in the horizontal axis according to its degree of interest with respect to the focal point.

This addresses the concern noted above that the simple fisheye view formulation for trees by itself is not sufficient to identify related documents. In CoMem, the fisheye degree of interest is used as the *primary* measure of *relatedness*. However, between sets of items with the same degree of interest and level of granularity, the CoMem relevance measure is used as a *secondary* measure of relatedness. This measure of relevance is also used, in the absence of unlimited display space, to prune all but the most closely related items. This use of the CoMem relevance as a secondary measure of relatedness can be considered a refinement of the *a posteriori* component of interest within a set of siblings.

Focus + context visualizations

We will now turn to the question of how to visualize and interact with a focal item and its project context. Several alternatives that were considered (but ultimately discarded) when designing CoMem will be discussed.

Furnas' fisheye view, described above, places more emphasis on *how much* context will be displayed rather than *how* it will be displayed. In its simplest applications, the user specifies a cutoff degree of interest value, and items with interest below that value are not displayed at all. Other techniques place greater emphasis on *visually* combining local detail with global context. These are referred to as focus + context techniques. Focus + context

techniques address the second problem identified above: how to visualize and interact with a tree structure by combining a detailed view of a particular node with a view of its context. However, for the most part, these techniques do not address the first problem: how to identify related items. For most focus + context techniques, 'context' refers to the whole tree rather than a subset of related items.

Cone trees²⁰ visualize trees in 3D, allowing much bushier trees to be displayed. Selected nodes can be moved near the front of the 3D space and are considered to be the focal points, and surrounding nodes which remain visible in the cone tree (particularly, the path from the selected node to the root) can be considered the context. Furnas' fisheye formulation has been applied to cone trees to reduce the number of nodes on the screen.

In the hyperbolic tree,²¹ the nodes in a hierarchy are positioned in hyperbolic rather than Euclidean space. Any node can be dragged into the center of the hyperbolic plane thereby bringing it into focus, while keeping much more of the entire hierarchy visible. Figure 12 shows a small design repository visualized as a hyperbolic tree.

One possibility that was explored is the use of an outline tree, a Microsoft Windows Explorer-style interface, where hierarchies are visualized using indented lists of icons and labels that can be collapsed or expanded.

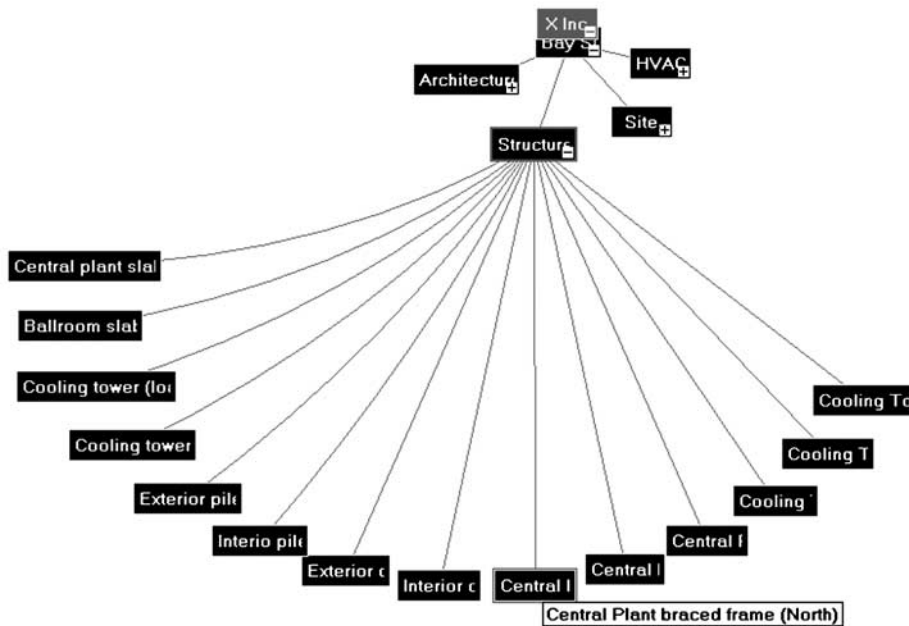


Figure 12 A hyperbolic tree – this screenshot was produced using Inlight Tree Studio.

The outline tree can be transformed into a focus+context view by mapping the degree of interest to the size of the icon and to the text used to label this item (Figure 13). This alleviates the problem of scrolling in an expanded tree. As the user changes the focal node, the size of each item is updated. The user is still able to expand and collapse trees and sub-trees in the usual way.

The focus+context techniques described above were explored but eventually rejected for use in CoMem. In informal preliminary studies, they were found to add little value beyond the simple node-link visualization used by CoMem.

Treemaps revisited

Treemaps were used to provide an overview of the entire corporate memory. Can treemaps also be used to explore the project context of a focal node? One possibility is to color each rectangle by its degree of interest value. Figure 14 shows a treemap colored by degree of interest. This visualization is not as effective as the fisheye view in Figure 11 because it does not emphasize the focal node and degree of interest distribution as much as the node-link diagram. In addition, it would be confusing to use the same interaction design for the two separate tasks of finding and understanding.

In Figure 14, the user gets a closer view of the focal node (the highlighted and solid yellow rectangle) by pruning out items with degree of interest lower than a cutoff value, and gradually increasing that value. This can be referred to as *fisheye zooming*.

A better approach might be to combine enclosure with *spatial* (rather than fisheye) zooming. *Zoomable user interfaces*²² have been shown to be more effective than

their non-zooming counterparts for many applications. These applications include image browsing²³ and web browsing.²⁴

Figure 15 shows a series of screenshots depicting a typical interaction where the user starts with a view of the entire corporate memory and progressively zooms in to a specific component (the focal node). All the time, the user can see the discipline and project to which the component belongs, as well as related components, disciplines, and projects in the corporate memory.

The advantages of this approach are:

- The ability to zoom means that screen real estate is almost unlimited, with lots of space to display CAD drawings, sketches, and documents linked to the objects in the repository.
- The hierarchy can be laid out in a self-similar manner, so that the interaction is the same at every level of the tree. A similar view is generated when a sub-tree is magnified. Components, disciplines, and projects are all displayed in a similar way. This self-similarity property has been called the *fractal tree layout*.²⁶

There are two major disadvantages to this approach. Firstly, the global view of the corporation as a whole and of the current project is lost as soon as the user zooms in. Secondly, it does not emphasize the degree of interest of each item. Recall that in the node-link diagrams of Figures 10 and 11, the items were positioned on the horizontal axis according to their degree of interest. In Figure 14, items are colored according to their degree of interest.

Table 2 contrasts the CoMem approach with the other approaches mentioned in this section.

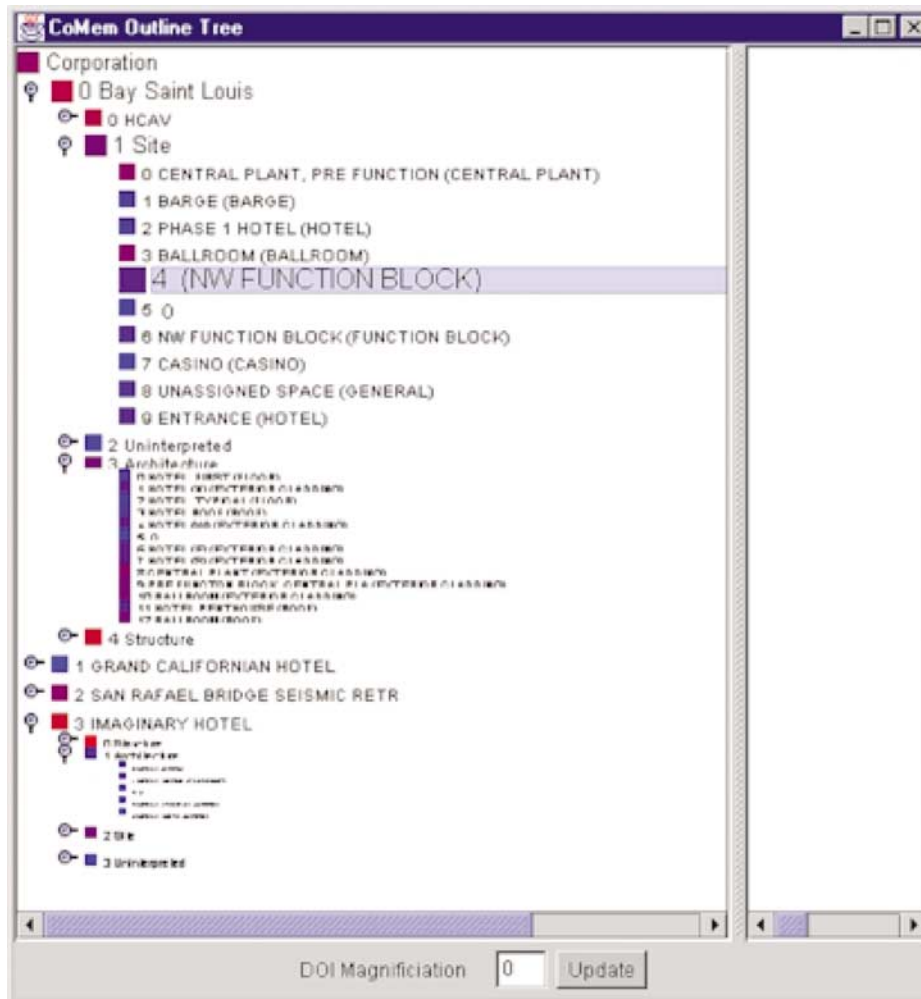


Figure 13 A fisheye outline tree view of the corporate memory. The NW Function Block is the focal node.

Closing remarks on the design of the overview and the context explorer

How can finding and understanding of reusable items be supported from a large corporate memory system? It was argued that the user needs to see an *overview* of the entire corporate memory. The user can identify potentially reusable items in the overview and then explore the context of this item in order to understand it.

CoMem uses the *Corporate Map* for the Overview, where the projects, disciplines, and components in the corporate memory are visualized using the *squarified treemap algorithm*. The Corporate Map provides a succinct overview at a glance of the ‘geography’ of the corporate memory. Over time, the user develops a familiarity with the Corporate Map.

The color of each rectangle is used to encode the relevance of that item to the designer’s current design task (i.e. the component or discipline on which the designer is currently working). This visual indication of relevance, combined with the user’s familiarity with the

geography of the corporate memory, should enable the user to quickly identify relevant regions to explore at greater depth.

Varying the treemap padding and line thickness are used as means of emphasizing structural relationships within the treemap. If the user notices a relevant item on the map, these measures should enable the user to tell instantly whether this item is a project, discipline, or component, and how relevant its ancestors and/or descendants are. The objective is to support reuse and comparison at all three levels of granularity simultaneously. Filtering is described as a mechanism for adding emphasis to items that are more likely to be reusable and averting information overload.

The Context Explorer enables the designer to explore the project context of a given item in the corporate memory. The fisheye view formulation is presented here as a formal mechanism for assigning a *degree of interest* to each item in the corporate memory given a focal node. The project context is then visualized by laying out the

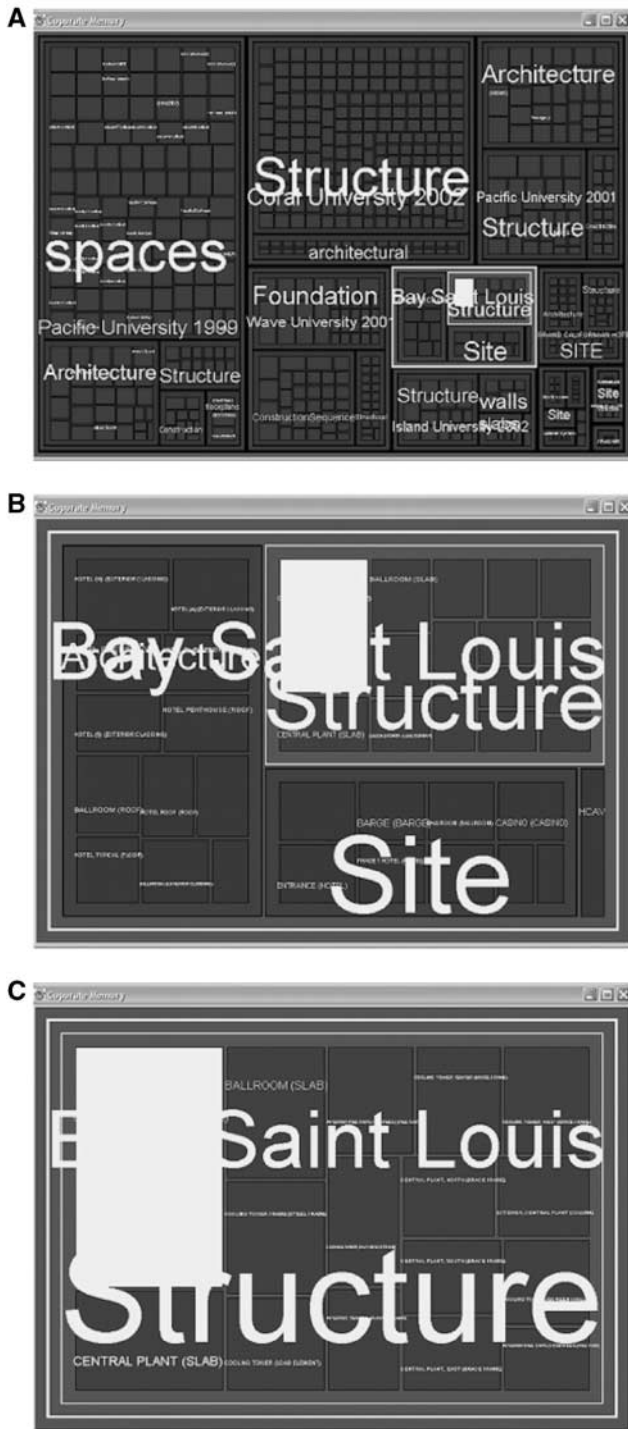


Figure 14 A treemap colored by degree of interest relative to the Bay Saint Louis cooling tower. The set of screenshots shows a series of interactions in which the user filters progressively more based on degree of interest.

hierarchy in a 2D space where the horizontal axis is the degree of interest and the vertical axis is the level of granularity. In addition, a relevance measure is generated

between each item and the focal item. This relevance is denoted using the color of each node and is used to prune less relevant nodes among nodes with the same degree of interest if space is limited.

More than any of the other approaches mentioned, CoMem emphasizes the degree of interest to help focus the user's exploration efforts. By exploiting the iso-interest contours, the resulting layout of the hierarchy highlights structural relationships surrounding the focal item. At the same time, using the relevance measure to color and prune nodes if necessary serves to highlight related items that are not necessarily structurally close to the focal item.

The stated objective of displaying high degree of interest nodes more prominently is achieved because a large number of low degree of interest nodes share the same amount of space as that shared by the relatively small number of high degree of interest nodes. Furthermore, relevant nodes that are buried deep in the tree, and would have been otherwise difficult to find, are always displayed prominently at the top of the list.

The main disadvantage of CoMem's Project Context Explorer is that it is not as interactive as other approaches. The subset of contextual nodes that are displayed is a function of the space available (i.e. the size of the window) and the user cannot interactively choose to show more, less, or different nodes.

Cone trees and hyperbolic trees address this by effectively visualizing the entire hierarchy in a limited space. Their major disadvantage is their implicit assumption that related items will be near the focal node (in terms of number of links). Related items that are not near the focal node are not prominently displayed.

The fisheye outline tree attempts to alleviate the problem of scrolling in outline trees by using the fisheye degree of interest to assign less space to nodes with smaller interest. In theory, a fully expanded tree can be displayed in a single screen. In practice, this would require an unreasonable amount of reduction in the size of items with less interest. If the reduction is limited to keep all labels legible, then the user will either have to scroll or collapse some sub-trees. The fisheye outline tree will still depend on the user exploring the project context by scrolling or expanding sub-trees to find related items deep in the hierarchy.

The treemap and zoomable treemap both abandon *connection* for visualizing hierarchies in favor of *enclosure*¹ (Section 2.4). However, treemaps tend to obscure structural relationships which, while less important in the CoMem Overview, are crucial when exploring the project context. The second problem with treemaps is that a choice must be made between mapping the color of each rectangle to the fisheye degree of interest or to the CoMem relevance measure. However, as noted above, it is the combination of the two that is quite powerful. If the treemap is colored by fisheye degree of interest, upward and downward exploration is supported (particularly by filtering) but sideways exploration of the project context

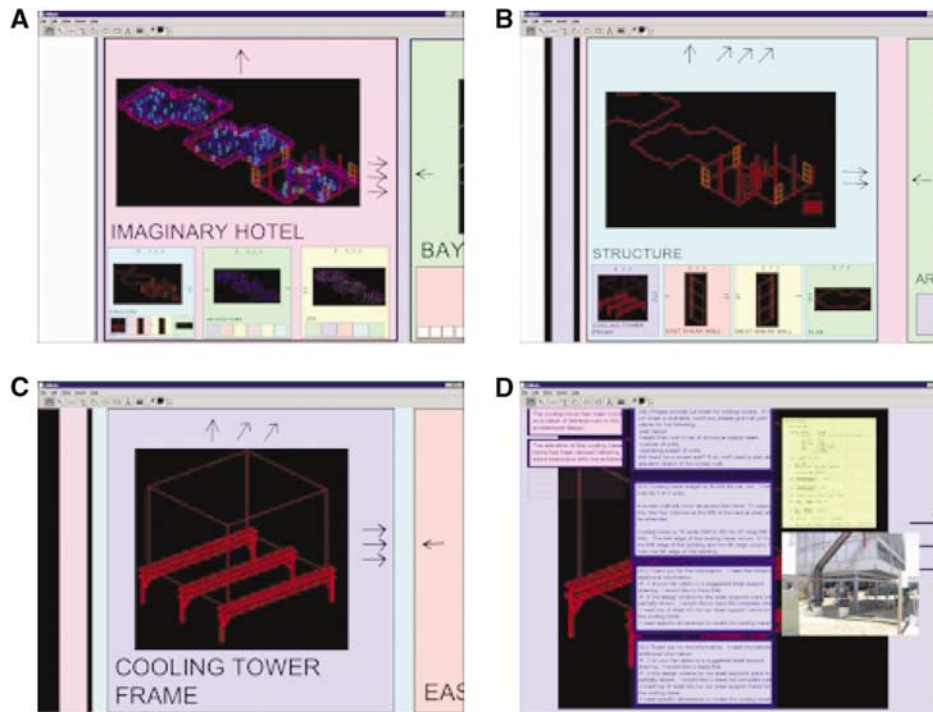


Figure 15 Exploring the project context using zooming. A series of screenshots depicting a typical interaction, where the user starts with a view of an entire project and progressively zooms in to a specific component. These images were prepared using the Jazz Java Toolkit.²⁵

becomes ineffective. During informal testing, the zoomable treemap was found to add little value. From users' feedback, its main advantage is its almost unlimited space, which allows the content (graphics, notes, images, documents) attached to each item to be displayed on the same zoomable canvas rather than in a separate display. Its main disadvantage is that it is not really a fisheye view: the user has to choose between a global or local view.

Usability evaluation

The purpose of the evaluation was to assess the extent to which CoMem enables the designer to *find* and *understand* reusable items from the corporate memory, and the extent to which this ability to find and understand improves the effectiveness of the reuse process. Since it is difficult to evaluate statements such as 'designer can find and understand' or 'external reuse is effective' in absolute terms, the strategy of the evaluation was to identify metrics for the validity of such statements and then to compare these metrics for CoMem vs 'traditional tools', as shown in Figure 16. Traditional tools are tools that reflect the current state of practice of design reuse in industry. A set of variables were introduced into the comparisons to identify specific circumstances under which CoMem leads to more effective external reuse.

The following tools were developed for the purpose of evaluating CoMem, and were used by the test partici-

pants as being representative of *traditional tools* used in current practice:

- **Outline Tree:** This is a prototype interface that uses indented lists of files and folders in the same way as Windows Explorer. The designer can use the Outline Tree to explore the corporate memory as if it were a set of files and folders on a computer, which reflects the nature of digital archives today, and the way current operating systems facilitate retrieval and exploration. It has an additional function to Windows Explorer: the generic icons for folders and files can be replaced by colored rectangles denoting the CoMem measure of relevance. (A fisheye version of this interface is shown in Figure 13.)
- **Hit List:** This is a prototype web interface that returns a list of hits in the same format as a web search engine, such as Google.²⁷ Given a problem the designer is working on, he/she can bring up the Hit List at any time, and it will display a list of items from the corporate memory ranked by their relevance to the designer's current task. The user can also search the corporate memory by keyword.

Tests were conducted for two kinds of finding tasks: *retrieval* and *exploration*. *Retrieval* occurs when the designer is looking for a specific item: 'I am looking for the cooling tower frame from the Bay Saint Louis Hotel that we worked on five years ago'. *Exploration* occurs when the designer has no idea what to look for, only that it should

Table 2 A comparison of the approaches considered in this section

<i>Approach</i>	<i>How are related items identified?</i>	<i>How is the project context visualized?</i>	<i>How does the user interact with the project context?</i>
CoMem	Fisheye formulation for tree structures. CoMem relevance measure is used to order and if necessary prune nodes among sets of items with the same degree of interest and level of granularity.	The hierarchy is laid out in 2D space, where the horizontal axis is the degree of interest and the vertical axis is the level of granularity. The color of each node denotes its relevance (compared to the focal node) and the size denotes the volume of content attached to it. Items that share graphics with the focal item can be visually highlighted.	The user highlights a node by moving the mouse pointer over that node. The node and its ancestors are highlighted using a thick yellow outline. If any of the ancestors had been pruned, they are temporarily painted at the top right of the display.
Cone trees	For both Cone trees and Hyperbolic trees, related items are assumed to be nearby in the tree, and so will be visible when the focal node is brought into focus. Fisheye formulation can be used to prune nodes.	The hierarchy is laid out in 3D space, with a set of children forming a cone below their parent.	The user brings the desired node into focus by clicking on it, which causes an animated transition that brings the node to the middle of the display.
Hyperbolic trees		The hierarchy is laid out in hyperbolic space. Nodes near the focal node are displayed more prominently near the middle of the space. Nodes further away from the focal node are displayed less prominently at the periphery.	The user brings the desired node into focus by dragging it to the middle of the display. Non-leaf nodes can be expanded or collapsed.
Fisheye Outline Tree	Fisheye formulation for tree structures.	The hierarchy is represented as a standard expandable/collapsible tree. The color of each node denotes its relevance (compared to the focal node) and the size denotes its fisheye degree of interest.	By expanding and collapsing subtrees.
Treemaps colored by degree of interest	Fisheye formulation for tree structures.	Treemap (nested rectangles). The color of each rectangle is mapped to the fisheye degree of interest.	The user can filter out items by degree of interest.
Zoomable treemap	Fisheye formulation for tree structures.	Nested rectangles are laid out on a zoomable canvas. The color of each rectangle is mapped to the fisheye degree of interest.	The user clicks on an item and the camera zooms in spatially onto this item.

be a relevant item: ‘I am stuck trying to design a hotel cooling tower, is there anything in the system that can help me get started?’

For retrieval, the time taken to find that item was the metric recorded to measure ability to find.

For exploration, the number of relevant items found was recorded to measure ability to find. For each task, an exhaustive list of useful items in the repository was prepared in advance by a human expert. This list was used to calculate a *recall score* for each test subject: the number of useful items found and listed by the prototype

user divided by the total number of useful items as judged by the human expert.

Each test subject was instructed to continue exploring the repository and list all useful items until he/she felt that all useful items had been found. The time taken to feel confident that the user has found everything to be found was measured as representing the ability to find in exploration tasks.

To measure the ability to understand, each subject was given a set of questions about the items he/she listed as reusable. Those questions were about the design

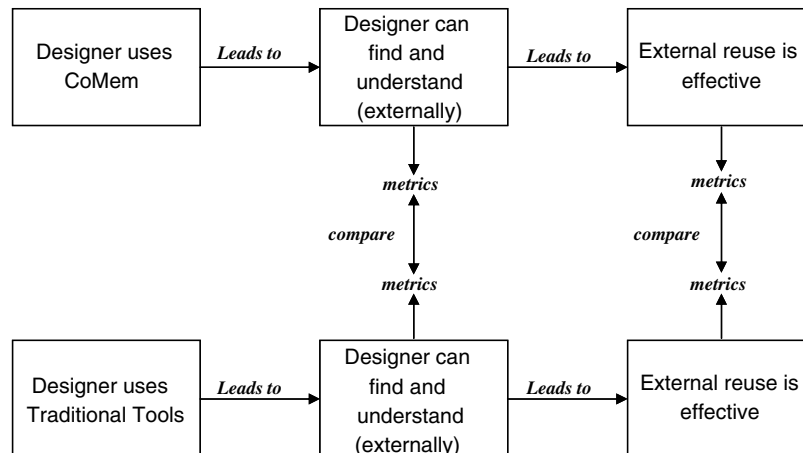


Figure 16 Evaluation of CoMem. CoMem is compared to traditional tools in current practice.

evolution or context of that item, such as: ‘Why did the design team choose that building material?’ A *context score* was calculated for each user by dividing the number of correctly answered questions by the total number of questions asked.

For effective external reuse, the extent to which the user agrees with the following statements was used as a metric that could be measured using questionnaires:

- If I had this system in my work, I would reuse content from previous projects *more frequently* than I do currently.
- If I had this system in my work, I would reuse content from previous projects *more appropriately* than I do currently.

A detailed discussion of the usability evaluation of CoMem is published in Demian and Fruchter.²⁸ Briefly, for time to complete a retrieval task, the Outline Tree allowed completion of such tasks in shorter times than both CoMem and the Hit List. For exploration tasks, the time taken to complete the tasks was comparable for all three prototypes. However, the recall scores and context scores attained by CoMem users were higher. Finally, from questionnaire feedback, CoMem was rated more highly for facilitating effective reuse.

References

- 1 Card SK, Mackinlay JD, Shneiderman B. *Readings in Information Visualization: Using Vision to Think*. Morgan Kaufmann Publishers: San Francisco, CA, 1999.
- 2 Pirolli P, Card S. Information foraging. *Psychological Review* 1999; **106**: 643–675.
- 3 Pirolli P. Computational models of information scent-following in a very large browsable text collection. *Proceedings of the ACM Special Interest Group on Computer-Human Interaction Conference on Human factors in computing systems* 1997; 3–10.
- 4 Nielsen J. *Designing Web Usability: The Practice of Simplicity*. New Riders: Indianapolis, IN, 2000.
- 5 Danielson DR. Transitional volatility in web navigation: usability metrics and user behavior. M.Sc. thesis, Stanford University Press, 2002.
- 6 Catledge LD, Pitkow JE. Characterizing browsing strategies in the World-Wide Web. *Computer Networks and ISDN Systems* 1995; **27**: 1065–1073.
- 7 Darken RP, Sibert JL. A toolkit for navigation in virtual environment. *Proceedings of the Sixth Annual ACM Symposium on User Interface Software and Technology (UIST)* 1993; 157–165.
- 8 Edson E. Bibliographic essay: history of cartography. *CHOICE: Current Reviews for Academic Libraries* 2001; **38**: 1899–1909.

Conclusions

At a global (macro) level, the results test the hypothesis of this research that finding and understanding improve reuse. Traditional tools such as search engines or Microsoft Windows Explorer-style interfaces do not support *find* and *understand* and traditional tools do not lead to effective reuse. CoMem supports *find* and *understand* and CoMem leads to effective reuse. This *supports* (albeit without conclusively *proving*) the claim that the steps of *find* and *understand* lead to effective reuse.

At a micro level, a comparison between the evaluation metrics from CoMem and those from traditional tools helps to identify the specific circumstances under which CoMem performs better than traditional tools. The variable presented here is the type of task: exploration *vs* retrieval. CoMem performs best in exploration scenarios.

To conclude, this research makes contributions by formalizing the design knowledge reuse process, and developing an innovative system, CoMem, to support this process. Usability evaluation results show that the CoMem Overview (a treemap) offers greater support for *finding* and the Context Explorer (a fisheye view) supports *understanding* more than traditional tools, and knowledge reuse using CoMem is rated to be more effective by test participants. This supports the hypothesis that *finding* and *understanding* lead to more effective reuse.



- 9 Harley JB In: Laxton P (Ed). *The New Nature of Maps: Essays in the History of Cartography*. The John Hopkins University Press: Baltimore, MD, 2001.
- 10 Shneiderman B, Wattenberg M. Ordered treemap layouts. *Proceedings of the IEEE Symposium on Information Visualization 2001*; 73–78.
- 11 Fiore A, Smith MA. *Treemap visualizations of newsgroups*. Technical Report, Microsoft Research, Microsoft Corporation: Redmond, WA, 2001.
- 12 Johnson B, Shneiderman B. Treemaps: A space-filling approach to the visualization of hierarchical information structures. In: Card SK, Mackinlay JD, Shneiderman B (Eds), *Readings in Information Visualization: Using Vision to Think, 1999*. Morgan Kaufmann Publishers: San Francisco, CA, 1991; 152–159.
- 13 Bruls DM, Huizing K, van Wijk JJ. Squarified treemaps. In: de Leeuw W, van Liere R (Eds), *Data Visualization 2000, Proceedings of the Second Joint Visualization Symposium organized by the Eurographics and the IEEE Computer Society Technical Committee on Visualization and Graphics (TCVG)*. Springer-Verlag: Vienna, Austria, 1999; 33–42.
- 14 Wattenberg M. Visualizing the stock market. *Proceedings of the ACM Computer Human Interaction (CHI) Conference, Extended Abstracts on Human Factors in Computing Systems 1999*; 188–189.
- 15 Shneiderman B, Wattenberg M. Ordered treemap layouts. *Proceedings of the IEEE Symposium on Information Visualization 2001*; 73–78.
- 16 Turo D, Johnson B. Improving the visualization of hierarchies with treemaps: design issues and experimentation. *Proceedings of the Third IEEE Conference on Visualization 1992*; 124–131.
- 17 Demian P, Fruchter R. Measuring relevance in support of design reuse from archives of building product models. *ASCE Journal of Computing in Civil Engineering 2005*; **29**: 119–136.
- 18 Shneiderman B. Dynamic queries for visual information seeking. In: Card SK, Mackinlay JD, Shneiderman B (Eds), *Readings in Information Visualization: Using Vision to Think, 1999*. Morgan Kaufmann Publishers: San Francisco, CA, 1994; 236–243.
- 19 Furnas GW. The FISHEYE view: a new look at structured files. In: Card SK, Mackinlay JD, Shneiderman B (Eds), *Readings in Information Visualization: Using Vision to Think, 1999*. Morgan Kaufmann Publishers: San Francisco, CA, 1981; 312–330.
- 20 Robertson GG, Mackinlay JD, Card SK. Cone trees: animated 3D visualizations of hierarchical information. *Proceedings of the ACM Computer Human Interaction (CHI) Conference, Human Factors in Computing Systems 1991*; 189–194.
- 21 Lamping J, Rao R. The hyperbolic browser: a Focus+Context technique for visualizing large hierarchies. In: Card SK, Mackinlay JD, Shneiderman B (Eds.), *Readings in Information Visualization: Using Vision to Think, 1999*. Morgan Kaufmann Publishers: San Francisco, CA, 1995; 382–408.
- 22 Perlin K, Fox D. Pad: an alternative approach to the computer interface. *Proceedings of the Twentieth ACM International Conference on Computer Graphics and Interactive Techniques (ACM SIGGRAPH) 1993*; 57–64.
- 23 Combs T, Bederson B. Does zooming improve image browsing? *Proceedings of the Fourth ACM International Conference on Digital Libraries 1999*; 130–137.
- 24 Bederson BB, Hollan JD, Stewart J, Rogers D, Druin A, Vick D. A zooming web browser. *SPIE Multimedia Computing and Networking 1996*; **2667**: 260–271.
- 25 Bederson B, Meyer J, Good L. Jazz: an extensible zoomable user interface graphics toolkit in Java. *Proceedings of the Thirteenth Annual ACM Symposium on User Interface and Software Technology (UIST) 2000*; 171–180.
- 26 Koike H, Yoshihara H. Fractal approaches for visualizing huge hierarchies. *Proceedings of the IEEE Symposium on Visual Languages 1993*; 55–60.
- 27 Brin S, Page L. The anatomy of a large-scale hypertextual web search engine. *Computer Networks and ISDN Systems 1998*; **30**: 107–117.
- 28 Demian P, Fruchter R. Usability evaluation of a corporate memory system. *Proceedings of the 2005 ASCE International Conference on Computing in Civil Engineering, 2005*.