CIFE CENTER FOR INTEGRATED FACILITY ENGINEERING

CoMem: Evaluating Interaction Metaphors for Knowledge Reuse from a Corporate Memory

By

Peter Demian & Renate Fruchter

CIFE Technical Report #158
JUNE 2004

STANFORD UNIVERSITY

If you would like to contact the authors, please write to:

c/o CIFE, Civil and Environmental Engineering Dept.,
Stanford University
Terman Engineering Center
Mail Code: 4020
Stanford, CA 94305-4020

COMEM: EVALUATING INTERACTION METAPHORS FOR KNOWLEDGE REUSE FROM A CORPORATE MEMORY

Peter Demian, PhD Candidate, Civil and Environmental Engineering, Stanford University,
Renate Fruchter, Director of PBL Lab, Stanford University

ABSTRACT

The objective of this research is to improve and support the process of design knowledge reuse in the architecture, engineering, and construction industry. Whereas internal knowledge reuse (reusing from one's personal memory or experiences) is very effective, external knowledge reuse (reusing from an external digital or paper archive) often fails. Ethnographic studies of designers at work, performed within the framework of this research, show that the three key activities in the internal knowledge reuse process are:

- *Finding* a reusable item
- Exploring this item's project context which leads to *understanding*
- Exploring this item's evolution history which leads to *understanding*

The approach of this research is to support the external reuse process so that it matches the internal reuse process. The hypothesis is that if the designer's interaction with the external repository enables him/her to:

- Rapidly *find* relevant items of design knowledge
- View each item in context in order to *understand* it, specifically:
    - Explore its project context
    - Explore its evolution history

then the process of reuse will be improved. This research addresses the following questions: (i) how do finding and understanding occur in internal knowledge reuse, and (ii) how can they be supported in external reuse?

This research presents a prototype system, CoMem (Corporate Memory), which consists of three modules to support each of these steps.

A usability evaluation of CoMem is performed using formal user testing. For this purpose, a usability testing framework and methodology is proposed. The key dimensions for the usability testing are the size of the repository, and the type of finding task: exploration versus retrieval. This research highlights the importance of exploration, which is normally overlooked by traditional tools.

The evaluation results show that CoMem offers greater support for finding and understanding than traditional tools, and reuse using CoMem is consistently rated to be more effective by test participants. This supports the hypothesis that finding and understanding lead to more effective reuse. This research makes important contributions by formalizing the reuse process, developing an innovative tool to support that process, and building a framework to study and assess such tools.

ACKNOWLEDGMENTS

Professor Terry Winograd and Professor Helmut Krawinkler provided much input from their respective domains of human-computer interaction and structural engineering.

Kushagra Saxena helped to conduct the user tests.

Dr. Greg Luth, founder and president of GPLA Inc, provided much needed real-world input into this research.

Protocol does not permit us to name them, but we are grateful to the CoMem user test participants, and those who allowed us to interview them or observe them during the ethnographic study.

CIFE member Intel provided a valuable test-bed for the evaluation of CoMem.

Funding for this research was provided by the UPS Endowment at Stanford, the Center for Integrated Facility Engineering, and the Project-Based Learning Lab at Stanford University.

TABLE OF CONTENTS

Chapter 1

INTRODUCTION

The average designer, whether consciously or subconsciously, draws from a vast well of previous design experience. "All design is redesign" (Leifer 1997). This can be experience acquired by the individual or by his/her mentors or professional community. This activity is referred to as *design knowledge reuse*. Specifically, design knowledge reuse is defined as the reuse of previously designed artifacts or components, as well as the knowledge and expertise ingrained in these previous designs. 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).

Internal knowledge reuse is an effective process, which some writers place at the very center of human intelligence:

> *We get reminded of what has happened to us previously for a very good reason. Reminding is the mind's method of coordinating past events with current events to enable generalization and prediction. Intelligence depends upon the ability to translate descriptions of new events into labels that help in the retrieval of prior events. One can't be said to know something if one can't find it in memory when it is needed. Finding a relevant past experience that will help make sense of a new experience is at the core of intelligent behavior. (Schank 1990, p. 1, 2)*

On the other hand external knowledge reuse often fails. This failure occurs for numerous reasons, including:

- To be available for reuse, knowledge needs to be *captured* and *stored* in an external repository. Designers do not appreciate the importance of knowledge capture because of the additional overhead required to document their process and rationale. They perceive that capture and reuse costs more than recreation from scratch. Consequently, knowledge is often not captured.
- Even when knowledge capture does take place, it is limited to formal knowledge (e.g. documents). Contextual or informal knowledge, such as the rationale behind design decisions, or the interaction between team members on a design team, is often lost, rendering the captured knowledge not reusable, as is often the case in current industry documentation practices.

Empirical observations of designers at work show that internal knowledge reuse is effective because:

- The designer can quickly *find* (mentally) reusable items.
- The designer can remember the context of each item, and can therefore *understand* it and reuse more effectively.

These observations of *internal knowledge reuse* are used as the basis for improving *external knowledge reuse*.

Knowledge reuse is a step in the knowledge life cycle (Figure 1). Knowledge is created as designers collaborate on design projects. It is captured, indexed, and stored in an archive. At a later time, it is retrieved from the archive and reused. Finally, as knowledge is reused, it is refined and becomes more valuable. In this sense, the archive system acts as a knowledge refinery. This research focuses on the knowledge reuse phase and builds on previous work that addresses knowledge creation, capture, indexing, and archiving (Fruchter 1996, Fruchter et al. 1998, Reiner and Fruchter 2000).

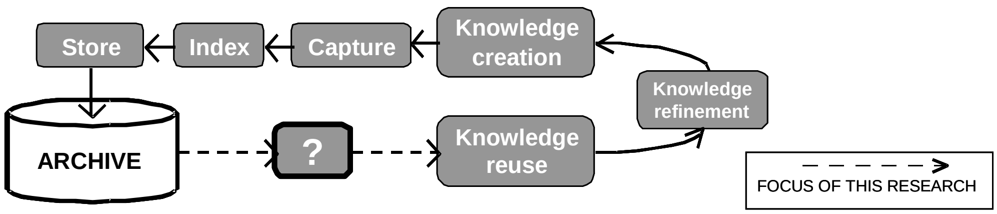


Figure 1: The knowledge life cycle. Knowledge is created, captured, indexed, and stored in an archive. At a later time, it is retrieved from the archive and reused. As it is reused, it becomes refined. This research focuses on the knowledge reuse phase.

Practical Motivation

The motivation behind the development of external knowledge reuse systems is that the capture and reuse of knowledge is less costly than its recreation. In many architecture, engineering, and construction (AEC) firms today, knowledge capture and reuse are limited to dealing with paper archives. Even when the archives are digital, they are usually in the form of electronic files (documents) arranged in folders which are difficult to explore and navigate. A typical query might be, "how did we design previous cooling tower support structures in hotel building projects?" In many cases, the user of such systems is overloaded with information, but with very little context to help assess and decide if, what, and how to reuse.

The Importance of the Knowledge Reuse Problem

Design Perspective

Why is design knowledge reuse an important issue? From a design perspective, the crucial concern is the tradeoff between productivity and creativity. At one extreme, the designer can choose not to reuse any knowledge at all from prior work. If successful, this approach can lead to a creative solution; if unsuccessful, it can waste a lot of time, with very little added value in the quality of the solution ("reinventing the wheel"). The second extreme is for the designer to reuse a lot of knowledge (or even an entire solution) from the well of previous design experience. If successful, this approach can save resources and lead to a better solution (for example, a novice learning from previous solutions created by experts); if unsuccessful, this approach can result in previous knowledge being reused inappropriately.

It is therefore important for the designer to take an approach which is somewhere in between the two extremes. In exploring this middle ground, the designer needs to ask questions such as:

- *Can I reuse anything from past experiences?* Are there similar situations captured in the external repository that might be useful?
- *How much can I reuse?* Small details or large portions of the design?
- *What should I reuse?* Actual physical components? Abstract concepts or ideas? Lessons learned from previous design processes? Design or analysis tools?

The underlying principle is that reuse should save resources (time and money), but not at the expense of the quality of the final solution.

Business Perspective

From a business perspective, an effective knowledge reuse strategy needs to enable a corporation to retain and reuse the knowledge accumulated from many years of experience. Specifically it should:

- Reduce the time wasted on recreating knowledge.
- Reduce the time wasted on searching for knowledge in obsolete archives.
- Retain knowledge in the corporation even after the retirement or departure of knowledgeable employees.

A knowledge reuse system can also be thought of as a learning resource:

- Novices can learn and benefit from the expertise of more experienced employees.
- Best practices are captured and reused by employees.

The underlying principle is that knowledge is a company's most important strategic resource, which, if properly managed, can drastically improve the company's productivity and lead to a greater competitive advantage.

Research Hypothesis and Questions

The objective of this research is to *improve* and *support* the process of design knowledge reuse in the AEC industry. Based on observations of internal knowledge reuse from an ethnographic study, the three key activities in the knowledge reuse process are:

- *Finding* a reusable item.
- Exploring this item's *project context* which leads to *understanding*.
- Exploring this item's *evolution history* which leads to *understanding*.

Hypothesis:

If the designer's interaction with the external repository enables him/her to:

- Rapidly *find* relevant items of design knowledge.
- View each item *in context* in order to *understand* its appropriateness, specifically:
    - Explore its project context.
    - Explore its evolution history.

⇨ Then the process of reuse will be improved.

This improved reuse will lead to higher quality design solutions, and save time and money.

This research addresses the following questions:

Question 1: How does *finding* occur in internal knowledge reuse? What retrieval mechanisms are needed to support the *finding* of reusable design knowledge in a large corporate repository of design content? What are suitable interaction metaphors and visualization techniques?

Question 2: What is the nature of the project context exploration in internal knowledge reuse? How can this exploration be supported in a large corporate repository of design content? What are suitable interaction metaphors and visualization techniques?

Question 3: What is the nature of the evolution history exploration in internal knowledge reuse? How can this exploration be supported in a large corporate repository of design content? What are suitable interaction metaphors and visualization techniques?

Points of Departure

Design as reflection-in-action. This research is the latest in a line of research projects on design knowledge management conducted at the Project-Based Learning Lab at Stanford University. These projects are based on Schön's *reflective practitioner* paradigm of design (Schön 1983). Schön argues that every design task is unique, and that the basic problem for designers is to determine how to approach such a single unique task. Schön places this tackling of unique tasks at the center of design practice, a notion he terms *knowing-in-action*:

> *Once we put aside the model of Technical Rationality which leads us to think of intelligent practice as an application of knowledge… there is nothing strange about the idea that a kind of knowing is inherent in intelligent action… it does not stretch common sense very much to say that the know-how is in the action – that a tight-rope walker's know-how, for example, lies in and is revealed by, the way he takes his trip across the wire… There is nothing in common sense to make us say that the know-how consists in rules or plans which we entertain in the mind prior to action. (Schön 1983, p. 50)*

To Schön, design, like tightrope walking, is an *action-oriented* activity. However, when knowing-in-action breaks down, the designer may consciously transition to acts of reflection. Schön calls this *reflection-in-action.* In a cycle which Schön refers to as a *reflective conversation with the situation*, designers reflect by *naming* the relevant factors, *framing* the problem in a certain way, making *moves* toward a solution and *evaluating* those moves. Schön argues that, whereas action-oriented knowledge is often tacit and difficult to express or convey, what *can* be captured is reflection-in-action.

Semantic Modeling Engine. This reflection-in-action cycle forms the conceptual basis of knowledge capture in the *Semantic Modeling Engine* (SME) (Fruchter 1996). SME is a framework that enables designers to map objects from a shared product model to multiple semantic representations and to other shared project knowledge. Figure 2 shows a simplified entity-relationship diagram of the SME schema (Figure 2 (a)), and an example of actual project knowledge (Figure 2(b)). In SME, a *project object* encapsulates multiple *discipline objects*, and a *discipline object* encapsulates multiple *component objects*. Each SME object can be linked to graphic objects from the shared 3D product model, or to other shared project documents or data (such as vendor information, analysis models, sketches, calculations).

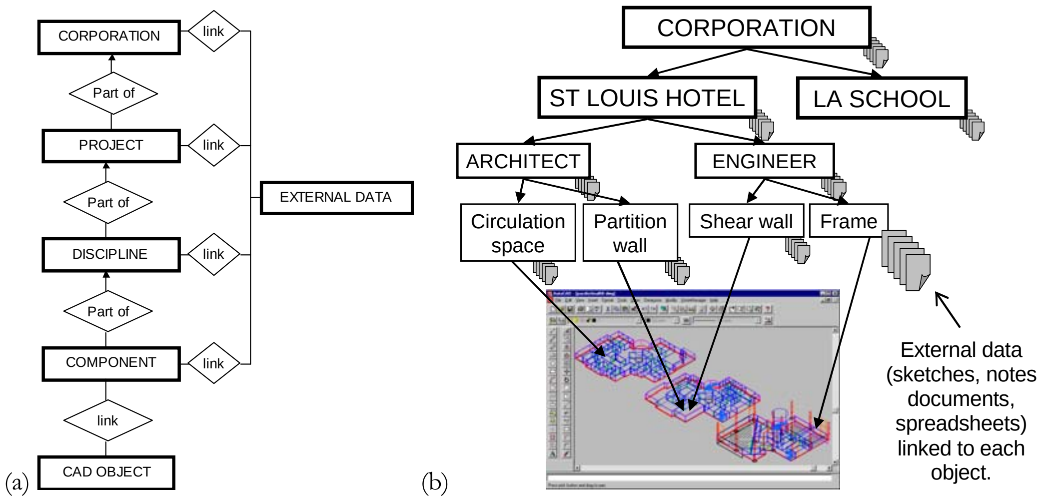


Figure 2: (a) A simplified entity-relationship diagram of the SME schema. A *project object* contains multiple *discipline objects*, and a *discipline object* contains multiple *component objects*. These semantic objects can be linked to graphic objects or to external data. (b) An example of actual project knowledge in SME.

SME supports Schön's reflection-in-action by enabling the designer to declare his/her particular perspective on the design (i.e. *framing* the problem) by creating a discipline object. Next he/she proceeds to *name* the individual components of the problem as he/she sees it by creating component objects. SME discipline objects are exported to external analysis tools to derive building behavior and evaluate it by comparing it to functional requirements. The designer uses these as the basis for making design decisions, i.e., making *moves* towards the solution and *evaluating* those moves.

Project Memory. The ProMem (Project Memory) system (Fruchter et al. 1998, Reiner and Fruchter 2000) takes SMEas its point of departure and adds to it the time dimension. ProMem captures the evolution of the project at the three levels of granularity identified by SME as emulating the structure of project knowledge: *project*, *discipline*, and *component*. ProMem automatically versions each SME object every time a change is made in the design or additional knowledge is created. This versioning is transparent to the designer. The designer is able to go back and flag any version to indicate its *level of importance* (*low*, *conflict*, or *milestone*) and its *level of sharing* (*private*, *public*, or *consensus*).

Corporate Memory. This research presents CoMem (Corporate Memory), a prototype system that extends ProMem firstly by grouping the accumulated set of project memories into a *corporate memory*, and secondly by supporting the designer in reusing design knowledge from this corporate memory in new design projects.

RELATED RESEARCH

The concept of *design knowledge reuse* is at the intersection of three other concepts: *knowledge*, *design*, and *reuse* (Figure 3).

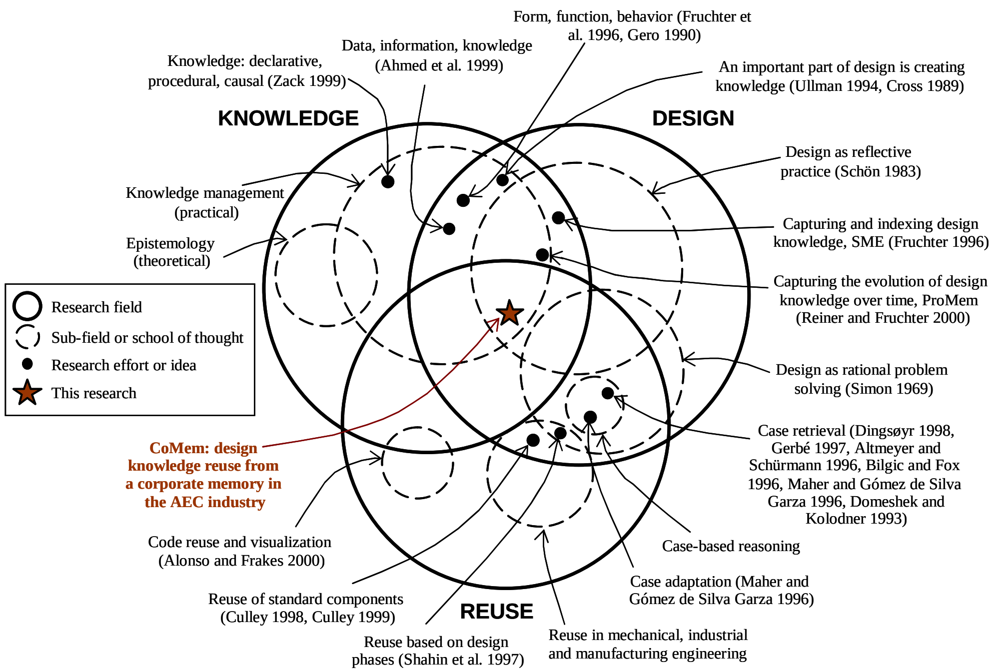


Figure 3: Related research.

Knowledge

The field of *knowledge management* is closely related to this research. In the knowledge management literature, knowledge is commonly distinguished from data and information. Broadly speaking, data are observations or facts out of context; information results from placing data within a meaningful context. Knowledge is "that which we come to believe and value based on the meaningfully organized accumulation of information through experience, communication or inference" (Zack 1999).

It has been rightfully noted that data, information, and knowledge are relative concepts (Ahmed et al. 1999). Although the precise distinctions between the three are not of immediate interest, there is clearly some dimension along which data would be ranked near the bottom and knowledge near the top. Intuitively, this dimension is closely related to *context*. Context is the framework within which information can be interpreted and understood. To clarify this notion of context, two commonly used knowledge classifications are presented below.

Declarative, procedural, causal. In this research, We take the term "*design* knowledge" to refer to knowledge *about* a certain artifact (*declarative* knowledge), for example the dimensions of a cooling tower frame. However, if a designer were to reuse this cooling tower frame in a new project, he/she would need to know *how* the original dimensions were calculated (*procedural* knowledge), and *why* they were given those values (*causal* knowledge).

Form, function, behavior. Within the field of design theory and methodology, knowledge related to an artifact is often categorized into *form* (or *structure*), *function*, and *behavior* (Gero 1990). An artifact's form is knowledge about its physical composition; its function is knowledge about what it should do; and its behavior is knowledge about what it actually does, or how well it performs.

Declarative knowledge is the principle output of the design process, but it is rendered more reusable if it is enriched with procedural and causal knowledge. Similarly, knowledge of the function and behavior is a useful supplement to knowledge of the form.

This research does not propose to make use of formal knowledge classifications. The important point to make is that the knowledge that is typically considered to be the output of the design process (i.e. the description of an artifact which enables someone to build it) is usually lacking the context which would enable this knowledge to be reused in the future. This is what is meant by *knowledge in context*; i.e. the additional knowledge that is generated or used during the design process, but which is not traditionally communicated as the output of the design process.

In order for knowledge to be reusable, it has to be as rich as possible, i.e. it has to be presented in the context in which it was created. This requirement may pose many challenges for knowledge capture because contextual knowledge is often *tacit* (Polanyi 1966), i.e. not encoded at all, or embedded in informal media, or impossible to detach from the people processing it. For example, Brown and Duguid (2000) write:

> *Knowledge entails a knower. Where people treat information as independent and more-or-less self-sufficient, they seem more inclined to associate knowledge with someone. In general, it seems right to ask, "Where is that information?" but odd to ask, "Where's that knowledge?"… It seems more reasonable to ask, "Who knows that?" Second, given this personal attachment, knowledge appears harder to detach than information. People treat information as self-contained substance. It is something that people pick up, possess, pass around, put in a database, lose, find, write down, accumulate, count, compare, and so forth. Knowledge, by contrast, doesn't take as kindly to ideas of shipping, receiving, and quantification. It is hard to pick up and hard to transfer… Third, one reason why knowledge may be so hard to give and receive is that knowledge seems to require more by way of assimilation. Knowledge is something we digest rather than merely hold. It entails the knower's understanding and some degree of commitment… while it seems quite reasonable to say, "I've got the information, but I don't understand it," it seems less reasonable to say, "I know, but I don't understand." (Brown and Duguid 2000, p. 119-120)*

Design

There exist several definitions of design, as well as design process models, design theories, and design methodologies (Dorst 1997, Cross 1989). In the context of design knowledge reuse, the following definition by Ullman seems fitting: "design is the process of developing information about an object that has not previously existed" (Ullman 1994). Cross (1989) makes a similar assertion: "the most essential design activity is the production of a final description of the artifact."

These statements about design are useful because they emphasize that design is an activity that generates knowledge, and implicitly this knowledge can be reused. However, they sidestep the crucial issue of how designs are generated, that is the "creative" part of design. It has been argued above that simply capturing the knowledge that is produced at the end of the design process is not enough. Supplementing descriptive knowledge about an artifact with contextual knowledge requires some understanding of the "inner workings" of the design process. The *process* by which the designed artifact evolves needs to be captured.

This "black box" of creative design has been the subject of much research. The earliest design researchers viewed design as a rational (or rationalizable) process made up of distinct phases. Later, attempts to incorporate more theoretical knowledge of designers and design problems into these rational phase models led to the view of design as *rational problem solving* (Simon 1969). Later still, perhaps as a reaction, fundamentally different views emerged, which took a phenomenological approach and regarded design as a subjective and deeply human experience (Schön 1983). Which paradigm best describes the design process as experienced by designers is an ongoing line of research (Dorst 1997).

The question of design paradigms is not central to this research. As noted above, this research continues along the path set by the SME (Fruchter 1996) and ProMem (Reiner and Fruchter 2000) research projects. These projects are based on the *Reflective Practitioner* paradigm (Schön 1983).

Design Reuse

Although much research is dedicated to design theory and design knowledge capture, considerably less focuses specifically on reuse. Research studies on design knowledge reuse focus either on the *cognitive* aspects or on the *computational* aspects.

Research into the cognitive aspects of reuse has helped to identify the information needed by designers. Kuffner and Ullman (1990) found that the majority of information requested by mechanical engineers was concerning the operation or purpose of a design object, information that is not typically captured in standard design documents (drawings and specifications). Finger (1998) observed that designers rarely use CAD tools to help them organize and retrieve design information. This research extends these findings by formalizing the requirements for contextual information when reusing items from previous projects. Ye and Fischer (2002) go further, noting that an important cognitive barrier to external reuse is the user's unfamiliarity with the contents of the repository. Users are not aware of what is in the repository and so do not know to look for it. They cannot anticipate the existence of a reusable component in the repository.

On the computational side, research into design knowledge reuse focuses on design knowledge *representation* and *reasoning*. Knowledge representation ranges from informal classification systems for standard components[1] (see for example Culley 1998) to more structured design rationale approaches (Regli et al. 2000 gives an overview). There is a tradeoff in design rationale systems between the overhead for recording design activities and the structure of the knowledge captured. History-based rationale approaches, such as electronic notebooks (Lakin et al. 1989), require a low overhead but result in a collection of disparate documents. Argumentation-based approaches (McCall 1987, Chung and Goodwin 1994) and device-based approaches (Baudin et al. 1993) provide a more uniform structure, but add a documentation overhead to the design process.

Highly structured representations of design knowledge can be used for *reasoning*. Two common approaches are case-based reasoning and model-based reasoning. However, these approaches usually require manual pre or post processing, structuring and indexing of design knowledge.

This research brings together the cognitive and computational approaches. We consider reuse to be a combined effort involving both the human and the computer. We therefore address the issue of design knowledge reuse as a human-computer interaction problem, and we take a user-centered approach to designing this interaction. We aim to provide a knowledge reuse experience that leverages natural idioms and metaphors in order to support the designer in doing his/her work, and we consider automatic reasoning approaches to constrain the user's knowledge reuse activities. In this approach, capture and indexing take place in real time, with the least possible intrusion on the design process. Knowledge is captured by supporting the typical communication and coordination activities that occur during collaborative design.

Two research areas related to the computational aspects of design reuse deserve special attention:

- Case-based and model-based reasoning (AEC industry)
- Code reuse (software engineering)

The principle that "all design is redesign" expresses the idea that designers are inevitably influenced by things that they or others have designed in the past. The term "redesign" implies that new designs can be created by modifying old designs. This is the premise behind using *case-based reasoning* to automate some aspects of the design process.

The differences between this research and case-based reasoning are summarized in Table 1.

[1] It has been argued that component reuse should not be restricted to standard parts coming from catalogs but should also include reuse of designed components (Culley and Theobald 1997).

Table 1: Differences between this research and case-based reasoning.

	This research	**Case-based reasoning**
FUNDAMENTAL DIFFERENCES:		
Design is...	Collaborative reflection	Rational problem solving
Research objective is...	To support the design process	To automate the design process[2]
CONSEQUENCES:		
Knowledge representation:	Informal, facilitate collaboration	Formal, a priori schema
Role of human:	To do design (evolution captured transparently)	To input previous design cases (high overhead)
Reuse mechanism:	Human designer explores corporate memory – knowledge in context	Automated reasoning based on previous cases

ARCHIE is a case-based reasoning tool for aiding architects during conceptual design (Domeshek and Kolodner 1993). ARCHIE breaks down previous design cases into "chunks", and uses indexes such as issues, building space, and life cycle phase to identify automatically the chunks that are the most useful to the architect. CASECAD enables designers to retrieve previous design cases based on formal specifications of new design problems (Maher 1997).

This research contrasts with the above efforts in that it is centered on the human designer and the natural reuse process as it is observed in professional practice. As a consequence, the proposed approach is to support interaction with a corporate memory of knowledge, rather than formal representation of cases and automatic case retrieval. Increasingly, research in case-based and model-based reasoning is converging with the approach adopted in this research that computer systems should support rather than automate design reuse (Simoff and Maher 1998, Popova et al. 2002).

In the field of software design, code reuse is an active research topic. A small subset of these efforts is dedicated to the development of applications that use visualization to assist in the retrieval and reuse of reusable software components. Table 2 gives some examples and compares them to this research.

[2] Some research in case-based reasoning is more geared towards design assistance, relying on the human designer to guide the processes of case retrieval and case adaptation.

Table 2: Related research in software reuse.

Project	Why?	What?	How?
This research	Find, understand → reuse	Projects, discipline subsystems, components (hierarchy)	Treemaps, fisheye views, node-link histories
MODIMOS	Monitor, reuse	Software components, class hierarchies (hierarchy)	Treemaps, node-link diagrams
Dali	Understand → reuse	Files, functions, variables (network)	Node-link diagrams
Vizbug++	Understand → debug	Program execution events (network)	Node-link diagrams
Jerding et al. 1997	Understand → reuse, reverse engineer	Interactions between classes, objects, functions, etc.	Time-series graphs, node-link diagrams, various others
CodeBroker	Find, understand → reuse	Software components	Latent semantic indexing (not visual), information delivery ("push" rather than "pull")

MODIMOS (Zielinski et al. 1995) allows the designer to monitor software applications made up of heterogeneous components, and indirectly supports reuse. It uses both node-link diagrams as well as treemaps for visualizing hierarchical structures such as class hierarchies.

Dali (Kazman and Carrière 1998) visualizes software systems using networks of files, functions, and variables (the nodes), as well as relationships between them (the links). They propose operations such as aggregation for reducing the complexity of these displays.

VizBug++ (Jerding and Stasko 1994), with an emphasis on development rather than reuse, also uses node-link visualizations of networks of events such as *class define* or *instance create*. Ware et al. (1993) extend these ideas from 2D to 3D.

Jerding et al. (1997) propose the use of animated node-link diagrams and time-series graphs to visualize interactions in program executions.

All these projects emphasize the importance of the *understanding* of archived components (Jerding and Stasko 1994, Kazman and Carrière 1998). Retkowsky (1998) lists the steps for software reuse as finding, understanding, adapting, and integrating.

Codebroker (Ye and Fischer 2002) is a code reuse system that autonomously suggests code fragments for reuse as the designer works.

Chapter 3

DESIGN AND RESEARCH METHODOLOGY

This chapter describes how ethnographic findings were used to design *CoMem* (Corporate Memory), a prototype system for supporting design knowledge reuse in the AEC industry, and how CoMem was evaluated.

Summary of Ethnographic Findings and their Design Implications

Knowledge reuse in current AEC design practice occurs largely through social knowledge networks. Even when reuse from an external repository occurs, a human expert is usually needed to provide proactive input on what to reuse and contextual information on the designs being reused. These observations are attributed to the effectiveness of *internal knowledge reuse*, the reuse of knowledge from one's personal experiences. Internal knowledge reuse is effective because the designer can *find* items to reuse, and can recall the context of these items and can therefore *understand* them and reuse them appropriately.

This suggests that an external repository of design knowledge should, insofar as this is possible, emulate the characteristics of design knowledge as it occurs in the designer's internal memory[3]. In other words, the reuse system will be a *corporate memory*, a rich, detailed repository of *knowledge in context*. The system should support the same activities observed during internal knowledge reuse, i.e. the corporate memory should support *finding* and *understanding*.

To support finding, particularly for novice designers unacquainted with the contents of the corporate memory, the system should be able to generate some measure of relevance between the designer's current task and each item in the corporate memory.

Understanding can be brought about by enabling the designer to explore the *project context* and *evolution history* of the found item. These explorations will also help the designer to manage the tradeoff between productivity and creativity by facilitating reuse at the appropriate levels of granularity and abstraction.

Tasks in Current Practice: Retrieval and Exploration

Two main kinds of reuse tasks were observed in current practice. *Retrieval* occurs when the designer is looking for a specific item: "I am looking for the cooling tower frame (component) from the structure (discipline subsystem) of the Bay Saint Louis Hotel (project) that we worked on five years ago". *Exploration* occurs when the designer has no idea what to look for, only that it should be a relevant item or that it should satisfy certain conditions: "I am stuck trying to design a hotel cooling tower, is there anything in the system that can help me get started?" In between the two extremes of retrieval

[3] A discussion of the possibility, desirability, and consequences of capturing context digitally, is presented by Grudin (2001).

and exploration there lie a whole range of tasks, for example when the designer might have some notion that there is a specific item in the system that would be helpful, but cannot remember exactly where it is: "I remember designing a hotel cooling tower a few years ago… what project was that for and where in the system can I find it?"

Stakeholders in Current Practice: Novices, Experts, and Mentors

Three groups of stakeholders in an external reuse system can be identified from the ethnographic study: novices, experts, and mentors.

The novice is a young designer with less than five years of experience. He/she will have worked at the company for only a few years and so will be unfamiliar with the contents of the corporate memory. The novice is more likely to *explore* the corporate memory than retrieve specific items. The novice rarely knows exactly what to look for and so is unable to formulate an explicit query. Some measure of relevance between the novice's current task and each item in the corporate memory would be extremely useful in guiding the novice's exploration. Context is extremely useful to the novice, as he/she will probably use the corporate memory as a learning resource, and so the rationale or decision process behind a reusable design would be just as important as the design itself.

The expert is a designer with five to fifteen years of experience. He/she will be quite familiar with the contents of the corporate memory. The expert will consider the corporate memory to be a productivity tool rather than a learning resource. The expert is more likely to retrieve specific items from the corporate memory, although in a large company, the expert might find it useful to explore projects in which he/she was not involved. The expert might prefer to formulate explicit queries to search the corporate memory, and so he/she will rely less heavily on the relevance measure. Like the novice, the expert will need contextual information, even when retrieving specific items that he/she identified as reusable from memory. However, in this case, the expert will probably retrieve specific items of context, rather than explore the context in general. In terms of the entire knowledge life cycle, the expert will probably be a net producer (rather than consumer) of knowledge, and so from the expert's point of view it will be important to minimize the overhead for knowledge capture.

Finally the mentor is an expert designer with many years of experience. The mentor is responsible for managing and overseeing the design work of several expert and novice designers. In relation to the corporate memory, the mentor will be concerned about the quality of the designs in the corporate memory. He/she will want poor designs to be excluded or somehow marked as "poor". He/she will also want refinements to the designs to be captured in the corporate memory (i.e. knowledge refinement). The mentor can act as a "coach" who encourages the young designers to think critically about the designs in the system and to learn from them and improve them. He/she might occasionally direct novices to specific items in the corporate memory, but he/she would expect the novices to be able to interact directly with the system without his/her intervention.

Research Methodology

Figure 4 shows an overview of the methodology adopted in this research. From the ethnographic study, it was observed that internal reuse is effective because the designer can find and understand the item he/she is reusing from his/her internal memory. In other words, relationship 1 in Figure 4 was empirically observed, that the ability to find and understand internally leads to effective internal reuse. These ethnographic observations address the internal reuse aspects of the research questions listed in Chapter 1.

Based on these observations, CoMem is designed specifically to support finding and understanding. This design process addresses the external reuse aspects of the research questions listed in Chapter 1: how can finding and understanding (through project context exploration and evolution history exploration) be supported in external knowledge reuse? Therefore relationship 2 in Figure 4 expresses the design rationale behind the design of CoMem. This is based on relationship 4 in Figure 4, which was posed as the hypothesis of this research.

The purpose of the formal evaluation described in Chapter 5 is to test relationships 3 and 4 in Figure 4, thereby testing the hypothesis of this research.

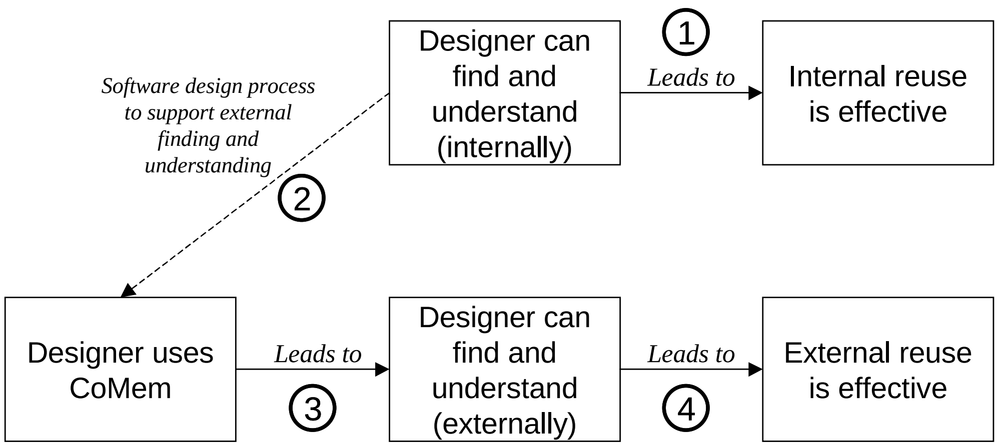


Figure 4: The research methodology.

COMEM – A CORPORATE MEMORY ENVIRONMENT

CoMem Modules for Supporting Reuse

CoMem is based on the principle of "*overview first, zoom and filter, and then details-on-demand*" (Shneiderman 1999). Based on the three reuse activities identified above – find, explore project context, explore evolution history – CoMem has three corresponding modules: an *Overview*, a *Project Context Explorer*, and an *Evolution History Explorer* (Figure 5).

Reuse step	→	User interaction
Find reusable item	"overview first, zoom and filter, and then details-on-demand"	Overview
Explore item's evolution history		Evolution History Explorer
Explore item's project context		Project Context Explorer

Figure 5: CoMem HCI experience. Transformation from observed reuse steps to user interactions.

Figure 6 shows the views that are generated of the corporate memory for each of the three modules. For each module, various metaphors were investigated, as well as possible visualization and interaction techniques. *Metaphor* here is used in a human-computer interaction sense. Metaphors increase the usability of user interfaces by supporting understanding by analogy. Modern operating systems use the *desktop metaphor*. Online services use shopping cart and checkout metaphors to relate the novel experience of buying online to the familiar experience of buying at a bricks and mortar store. For a discussion of the advantages and pitfalls of using metaphors, see Nelson (1990).

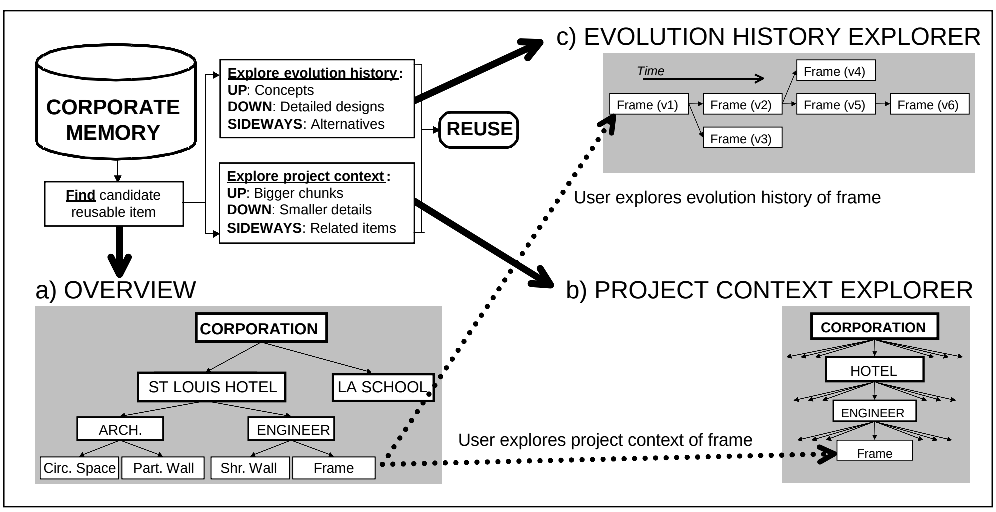


Figure 6: Views of SME data that are generated for each of the CoMem modules. (a) The Overview shows the entire corporate memory. (b) The Project Context Explorer takes a single item (in this case a structural frame) as its focal point. (c) The Evolution History Explorer shows the versions of a single item.

The Overview supports the designer in finding reusable items. The objective is to enable the designer to view the entire corporate memory at a glance. The Overview gives the designer an indication of which “regions” of the corporate memory contain potentially reusable items. The Overview might be extremely dense. Filtering tools are used to avert information overload and help the designer focus by adding emphasis to certain items.

The Overview presents the projects, disciplines, and components as nested rectangles using the *squarified tree map* visualization (Bruls et al. 1999). The size of each rectangle denotes the amount of content contained in that project, discipline, or component (number of versions, annotations, linked documents, etc.). The color of each rectangle denotes that item’s relevance to the current design task based on text analysis.

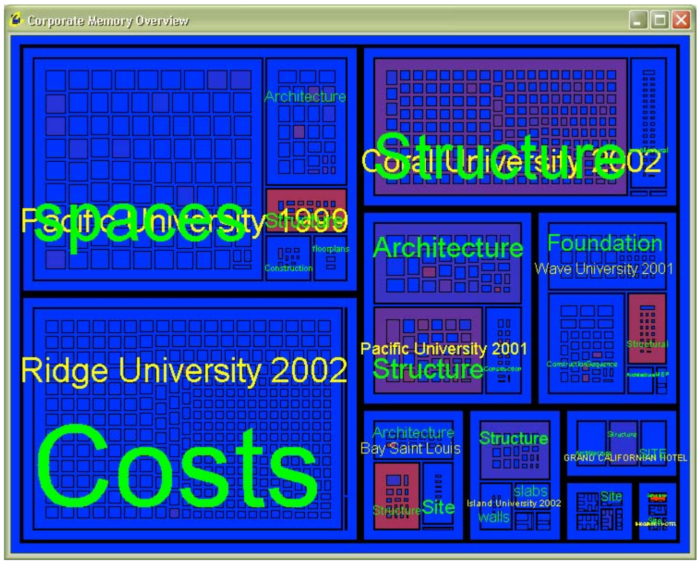


Figure 7: The CoMem Overview. A map of the corporate memory using the squarified treemap technique (Bruls et al. 1999).

Once the user has selected an item from the Overview, the *Project Context Explorer* supports the designer in exploring this item’s project context. This shows the project and discipline to which this item belongs, as well as related components and disciplines that would help the designer understand the found item. The item selected from the Overview becomes the *focal point* of the Project Context Explorer. The focal point and its related items are arranged in a two-dimensional space where the vertical axis represents level of granularity (from entire corporation down to a single component) and the horizontal axis represents the degree of interest (how closely related is an item to the focal point).

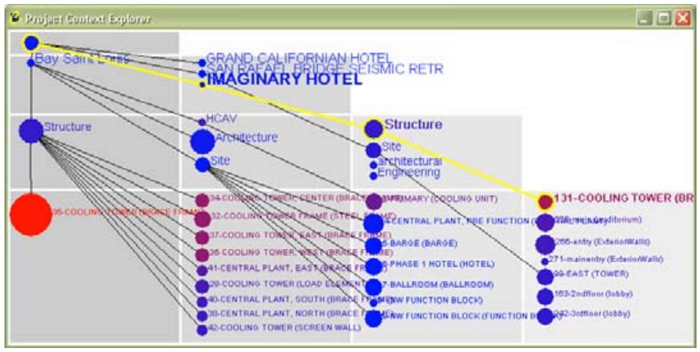


Figure 8: The CoMem Project Context Explorer. Each object is positioned in the vertical axis according to its level of granularity, and in the horizontal axis according to its degree of interest with respect to the focal point.

In the third module, the *Evolution History Explorer*, the designer can explore the evolution history of any item selected from the Overview. This view tells the story of how this item evolved from an abstract idea to a fully designed and detailed physical artifact or component.

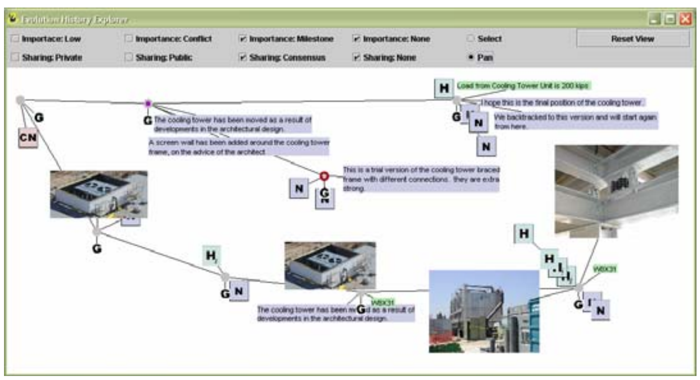


Figure 9: The CoMem Evolution History Explorer, showing the version history of the item selected from the Overview.

In addition to the Overview, Project Context Explorer, and Evolution History Explorer, CoMem also includes a *content viewer*, which displays all the disparate content associated with an item (text description, CAD file, hyperlinks, notes, notifications, data) in a single web page.

CoMem is described through an interaction scenario where Matthew (an expert) and Nick (a novice) are working on a ten-storey hotel that has a large cooling tower unit and Nick is assigned the task of designing the frame that will support this cooling tower. They are using the ProMem system (Figure 10). Nick gets stuck, but Matthew is not around to help. Nick clicks on the Reuse button in ProMem, which brings up CoMem (Figure 11). CoMem displays a map of the entire "X Inc" corporate memory. Items on the map are color-coded according to how relevant they are to his current project. Nick uses sliders to filter out irrelevant projects, disciplines, and components from the map (Figure 12). Most of the rectangles in the map are now grayed out. Of the few items that remain highlighted, Nick notices the Bay Saint Louis project. It has a relevant Engineering discipline, and several relevant components within that discipline. He clicks on the component labeled Cooling Tower Frame.

The project context and evolution history of the Bay Saint Louis cooling tower frame appear in two separate displays (Figure 13). Nick examines the evolution of the frame. He chooses to see only milestone versions of the evolution (Figure 14). He sees that it started as a composite steel-concrete frame but was later changed into a steel frame. He sees several notes that were exchanged between the architect and engineer that help to explain this change. Nick clicks on one of the versions, and a detailed view of this version appears (Figure 15). He finds a useful early sketch of the composite frame, which he saves to his local hard drive.

Next, Nick begins to explore the project context of the Bay Saint Louis frame. He clicks on the Engineering discipline object in the Project Context Explorer and sees that the Bay Saint Louis structural design criteria are similar to those in his current project (Figure 16). He notices a related component under the HVAC discipline: it is labeled Cooling Tower. This is the air conditioning unit that is supported by the frame. Nick finds a specifications sheet attached to this component (Figure 17). It gives him an idea of the loads for which he must now design his cooling tower frame.

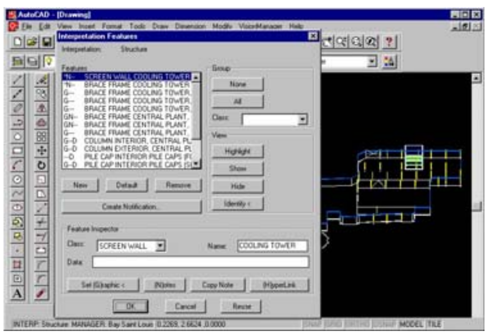


Figure 10: Nick is working in the ProMem system when he gets stuck. He presses the *Reuse* button.

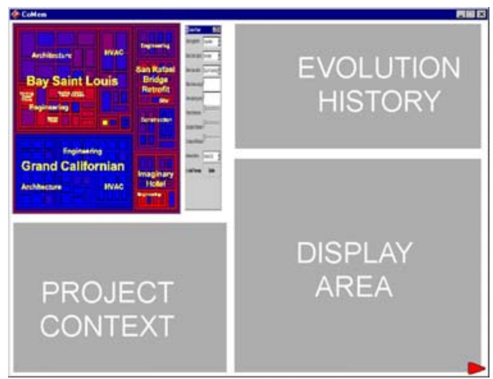


Figure 11: CoMem pops up on the screen and displays the Corporate Map.

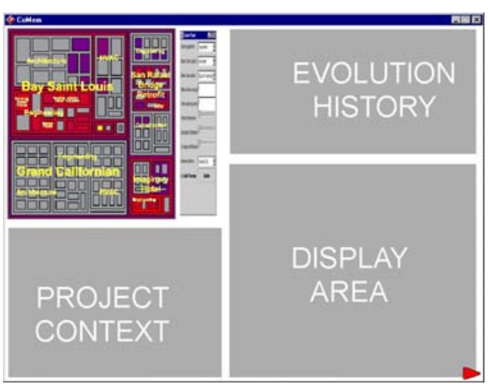


Figure 12: Nick filters out some items from the map using the sliders. He notices the cooling tower frame and clicks on it.

Figure 13: The project context and evolution history of the cooling tower are displayed.

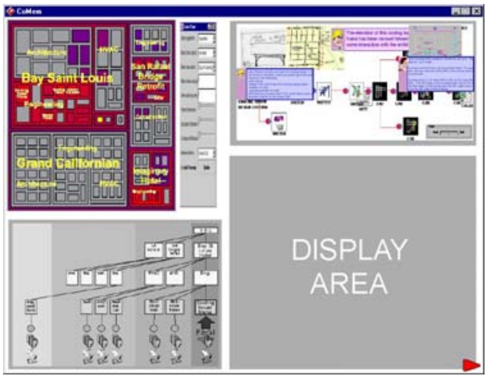


Figure 14: Nick filters out unimportant versions from the cooling tower evolution using the slider and enlarges several thumbnails.

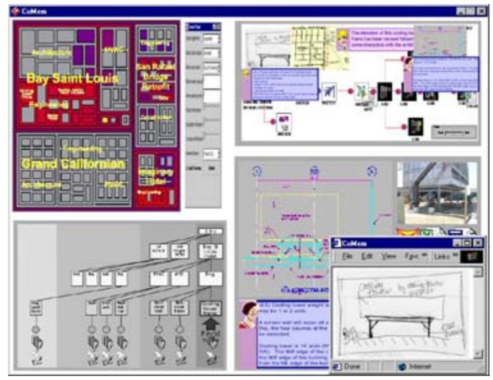


Figure 15: Nick clicks on a particular version from the Evolution History Explorer. The details of this version appear in the display area.

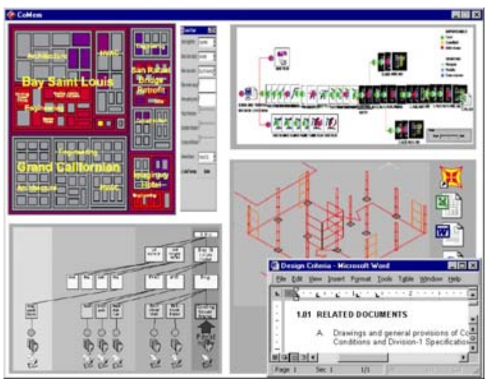


Figure 16: Nick uses the Project Context Explorer to view information about the structural system. He clicks on the document icon to bring up the design criteria document in a separate window.

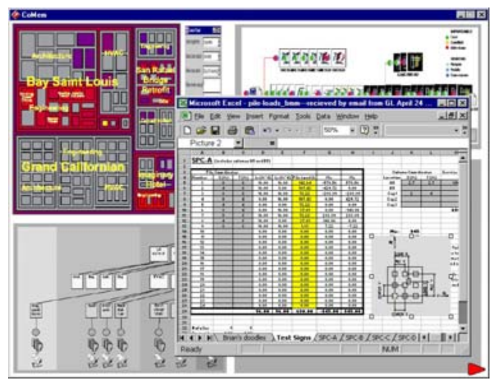


Figure 17: Nick uses the Project Context Explorer to view information about the cooling tower unit. He views a spreadsheet attached as a hyperlink to the cooling tower object.

Chapter 5

COMEM USABILITY EVALUATION

Evaluation Approach

This chapter presents the evaluation of CoMem that assesses the extent to which it 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 (Figure 4 in Chapter 3).

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 this evaluation is to identify metrics for the validity of such statements and then to compare these metrics for CoMem versus "traditional tools", as shown in Figure 18. Traditional tools are tools that reflect the current state of practice of design reuse in industry. In addition, a set of variables are introduced into the comparisons to identify specific circumstances under which CoMem leads to more effective external reuse.

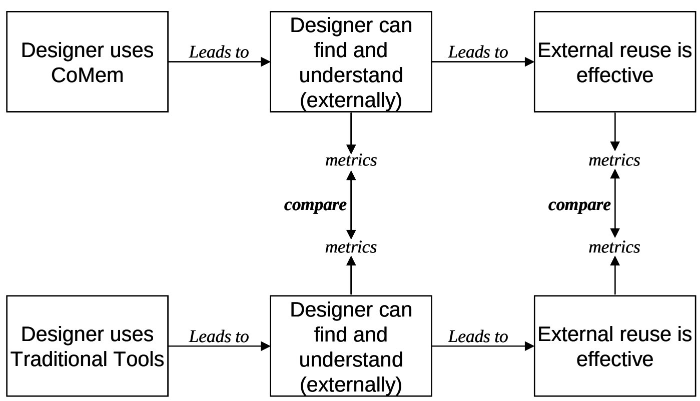


Figure 18: Wvaluation of CoMem. CoMem is compared to traditional tools in current practice.

CoMem Versus Traditional Tools

CoMem offers the following tools for *finding* and *understanding* items from the corporate memory:

- Overview (Corporate Map metaphor, Figure 7)
- Project Context Explorer (fisheye lens metaphor, Figure 8)
- Evolution History Explorer (storyteller metaphor, Figure 9)

The following tools were developed for the purpose of evaluating CoMem, and were used by the test participants 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 (Figure 19 (a)). 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 (the same relevance that is indicated on the CoMem Overview module, as shown in Figure 19 (b)). When the user selects an item from the Outline Tree, the versions of this item are displayed in a table similar to a spreadsheet or database program (Figure 19 (c)). The table displays the version number as well as the parent version and other ancestors. It can also display any textual information attached to that version (notes, notifications, and data).

- **Hit List**. This is a prototype web interface (Figure 20) that returns a list of hits in the same format as a web search engine, such as Google (Brin and Page 1998). 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 (for exploration tasks). The user can also search the corporate memory by keyword, which is the mechanism expected to be used in the case of retrieval tasks. The user may select any item from the Hit List to displays all the versions of that item in a web-based table similar to a spreadsheet or database program (Figure 20 (b)).

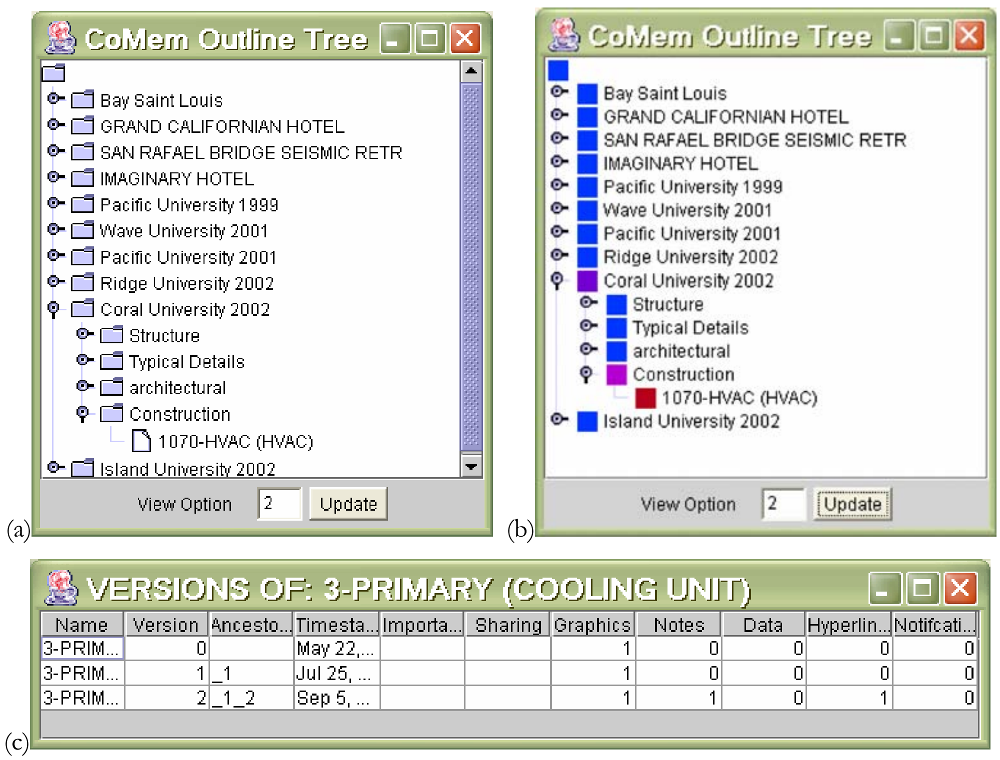
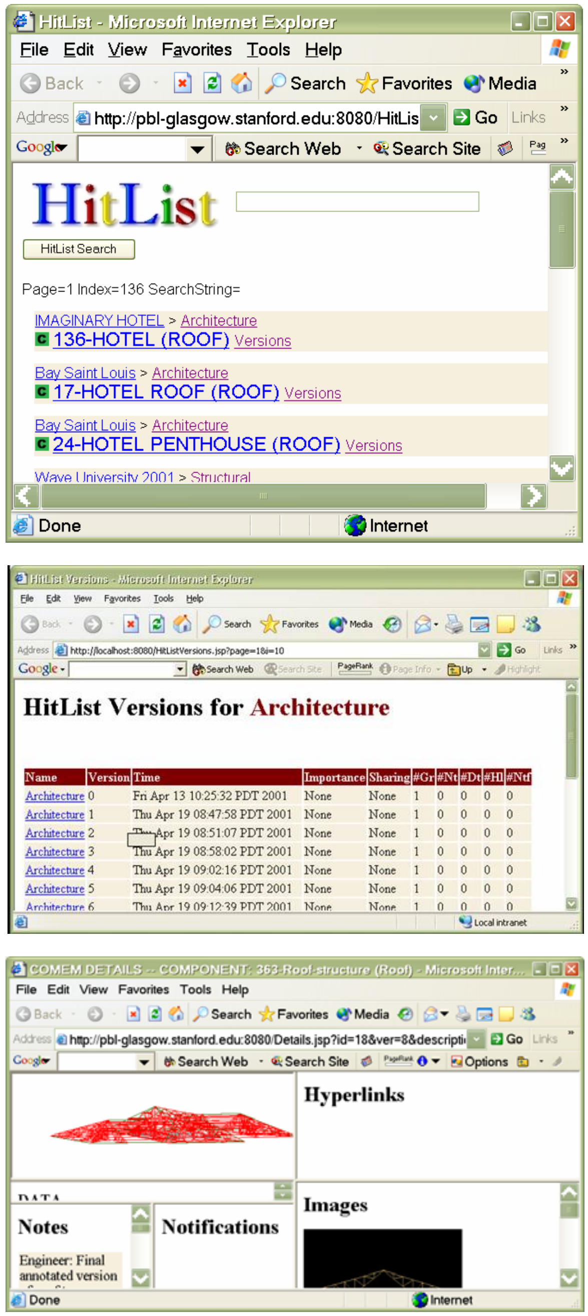


Figure 19: Outline Tree prototype. (a) Outline Tree with generic icons. (b) Outline Tree with colored icons to indicate relevance, used for exploration tasks. (c) Version table which lists all versions of an item in a table.



(a) (b) (c)

Figure 20: Hit List prototype. (a) Main page of Hit List for searching the corporate memory. (b) Web based version table. (c) Web view of an item select from Hit List.

Variables and Metrics

The aim of this evaluation is not merely to determine whether CoMem offers improved support for reuse, but also to identify the specific circumstances under which traditional tools break down and CoMem offers genuine added value (and vice versa). The following variables are pertinent.

Type of finding task. There are two main kinds of finding tasks that need to be supported: *retrieval* and *exploration*, as described above.

Size of the repository. CoMem was designed with large repositories in mind, as this is where traditional tools often fail in supporting the finding and understanding of useful information. To what extent does CoMem also support smaller repositories, and what is the repository size for which traditional tools break down?

The following metrics for *effective finding* were measured:

- Retrieval: Time to find the desired item. In the case that the user is looking for a specific item, the time taken to find that item is the most important metric.
- Exploration:

1. Number of relevant items found. For each exploration 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 user divided by the total number of useful items as judged by the human expert.
2. The test subject was instructed to continue exploring the corporate memory and listing 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.

The following metrics for *effective understanding* were measured:

- Ability to answer a set of questions after exploring the project context and evolution history, such as: "Why did the design team choose that building material?" A *context score* was generated for each user by dividing the number of correctly answered questions by the total number of questions asked. This was intended to measure the extent to which the tool enabled the user to understand why that item was designed the way it was. The purpose of the questions was to test the ability of the user to *understand* content retrieved with that tool, rather than to test the user's domain expertise. Overly technical questions about architecture, engineering, or construction were avoided.

For *effective external reuse*, the extent to which the user agrees with the following statements was used as a measurable metric that assesses the effectiveness of the reuse process:

- *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.*

CoMem was used in the context of synthetic experiments. If CoMem were used for a real project, possible metrics would have been:

- Percentage of designed artifact based on reused components.
- Quality of final design.

For each metric that was measured in the experiment, a 90% confidence interval was calculated and is displayed in the charts in this chapter. The Student-t distribution was applied, with the sample standard deviation used as an estimate of the population standard deviation and the number of degrees of freedom was estimated as the sample size minus one.

Method

Participants. Twenty participants were recruited from amongst students and researchers in the Department of Civil and Environmental Engineering at Stanford University, as well as professionals from local design offices. The participants were chosen to be as close as possible in age, computer experience, and design experience to eliminate any variability in the data due to these factors.

Materials. Three different software prototypes were tested:

- Outline Tree: indented list of projects, disciplines, and components, with versions of items displayed in tables.
- Hit List: web search engine with versions of items displayed in web-based tables.
- CoMem: Overview, Project Context Explorer, Evolution History Explorer.

Procedure

1. **Brief.** A standard passage describing each of the prototypes, the nature of tasks, and the objective of the user tests was read to the participant.
2. **Warm up.** The participant was invited to familiarize him/herself with the prototypes by exploring data unrelated to the tasks for about five minutes. During this time, he/she was able to ask questions about how the prototypes work. After this warm-up, the formal experiment started.
3. **Retrieval tasks.** The participant was asked to complete three different randomly chosen retrieval tasks with CoMem, the Outline Tree, and the Hit List.
   Retrieval tasks are simple: "find the component called … which is in the discipline called … in the project called …". For each participant, the task chosen to be completed using each prototype was randomly chosen. All retrieval tasks used were of comparable difficulty (for example, they were all component items from sub trees of the corporate memory with similar branching factors). For each retrieval task the following were measured:
   - Time to complete the task, and
   - Correctness of final answer.
4. **Exploration tasks.** A standard passage describing a randomly-chosen synthetic scenario and a related exploration task based on the projects in the test bed repository was read to the participant. The participant was asked to explore the repository using CoMem and list all reusable items, until he/she feels confident that he/she has found all the reusable items in the repository. This was repeated for Outline Tree, and then the Hit List with different scenarios and tasks.
   Exploration tasks are of the type: "you are working on this problem, find anything you think would be helpful in the corporate memory to help you complete your design task." An example exploration task is shown in Figure 22. There were a total of 6 previously-prepared exploration tasks, all of which were designed to be comparable in difficulty (for example, having the same number of reusable items and contextual questions). For each participant, the task chosen to be completed using each prototype was randomly chosen. The participant was asked to explore the corporate memory and make a list of all potentially reusable items found. After the task was completed, the participant was asked to answer some simple questions about each of the items

listed, such as: "why did the design team choose that building material?". For each task, the following were measured:

- **Recall score**: the proportion of potentially reusable items as judged by a human expert that were actually found by the participant.
- **Context score**: the proportion of questions about helpful items that could be correctly answered by the participant.
- **Time taken**: the time taken to feel confident that all helpful items had been found.

The exploration tasks and the retrieval tasks were run first with a large repository, and then repeated with a small repository in the cases of CoMem and the Outline Tree.

5. **Questionnaire.** The participant was asked to complete three questionnaires, one for each of the prototypes, asking them about their subjective reactions to the prototype (shown in Figure 21, loosely based on Brooke 1996).
6. **Debrief.** Short, informal interview.

For both exploration and retrieval, the order of testing the three prototypes was randomly chosen, in an attempt to eliminate the effects of learning and increased familiarity with the data.

Questionnaire

The questionnaire given to test subjects to solicit subjective feedback on CoMem at the end of the test is shown in Figure 21 below. Test subjects were given similar questionnaires for Hit List and Outline Tree, but with questions 13, 14, and 15 omitted, as those questions are specific to CoMem.

CoMem **Questionnaire**	**Strongly Disagree**						**Strongly Agree**
1. I think that I would like to use this system frequently	1	2	3	4	5	6	7
2. I found the system unnecessarily complex	1	2	3	4	5	6	7
3. I thought the system was easy to use	1	2	3	4	5	6	7
4. I would imagine that most people would learn to use this system very quickly	1	2	3	4	5	6	7
5. I found the system very cumbersome to use	1	2	3	4	5	6	7
6. I felt very confident using the system	1	2	3	4	5	6	7
7. I needed to learn a lot of things before I could get going with this system	1	2	3	4	5	6	7
8. I would feel very confident reusing some content that I found using this system.	1	2	3	4	5	6	7
9. In the exploration tasks I completed using this system, I felt I had a good understanding of the items I was exploring.	1	2	3	4	5	6	7
10. In the exploration tasks I completed using this system, I felt that I was able to find all potentially reusable items in the corporate memory in the given time.	1	2	3	4	5	6	7
11. If I had this system in my work, I would reuse content from previous projects **more frequently** than I do currently.	1	2	3	4	5	6	7
12. If I had this system in my work, I would reuse content from previous projects **more appropriately** than I do currently.	1	2	3	4	5	6	7
13. I think the **Overivew / Map** would be very useful in my work.	1	2	3	4	5	6	7
14. I think the **Storyteller / Evolution history** would be very useful in my work.	1	2	3	4	5	6	7
15. I think the **Fisheye Lens / Project context** would be very useful in my work.	1	2	3	4	5	6	7

COMMENTS:

Figure 21: The CoMem questionnaire. Test subjects were given similar questionnaires for Hit List and Outline Tree, but with questions 13, 14, and 15 omitted.

The Data

The *large repository* tests described in this chapter were conducted on a pilot corporate memory consisting of 10 project objects, 35 discipline objects, and 1036 components. Of the 1036 component objects, approximately 30% were annotated with note objects. The *small repository* tests were conducted with the smallest possible subset of projects in the large repository that would include all the data required for the exploration tasks.

Attention was paid to ensure that the repositories were densely populated in several areas related to each exploration task. For example, if the exploration task involved roof design, care was taken to ensure that at least 5 or 6 projects had rich content related to roof design: annotations, hyperlinked documents, team interactions, images, design alternatives, and so on.

There was a pool of six standard exploration tasks from among which a task was randomly chosen for each prototype and repository size. Those were:

- Roof design
- Post-tensioned slab
- Shear walls
- Atrium
- Elevator
- HVAC System

Figure 22 shows an exploration task where the user is working on a roof design.

TASK I = ROOF DESIGN
ProblemIndex=136

Reusable	Reusable Items	Context Questions
Pacific 1999	X	
Pacific 2001	Pacific 2001>Structure>449-Roof	What material was used for this roof? (Metal panels)
	Pacific 2001>Structure>444-PT Slabs	Why? (Lighter than concrete, simpler connections than steel, ease of construction)
	Pacific 2001>Construction>481-Roof System	What did the roof look like over the auditorium? (Pyramid)
		Why will the roof be expensive? (Because of the curvature)
		Which other building component had to be coordinated with the roof? (PT slabs)
Wave 2001	Wave2001>Arch>366-Roof	Can you name some architectural concepts that were considered? (Gable, mansard/French gable)
	Wave2001>Eng>363-Roof structure	What was the CM's feedback on the architect's ideas? (Complicatated, hieght restriction, snow/rain accumulating)
	Wave 2001>Construction>404-Air Handling Unit	What materials were considered? (Timber, steel)
		What equipment will go on the roof? (Air handling unit)
		What impact will this have on the structure? (Larger columns)
Coral 2002	Coral 2002>Structure>890-roof1(columns)	Can you describe the roof system? (prefab roof truss, elevated on columns and beams, prestressed roof slab)
	Coral 2002>Structure>888-roof1(beams)	Why was the roof truss elevated? (Natural ventilation, aesthetics)
	Coral 2002>Structure>892-slab1(roof)	
	Coral 2002>Structure>894-rooftrusses1(rooftrusses)	
Ridge 2002	X	
Island 2002	X	
Bay Saint Louis	BSL>Arch>25-Ballroom (roof)	
	BSL>Arch>17-Hotel roof (roof)	
	BSL>Arch>24-Hotel Penthouse (roof)	
Grand Californian Hotel	GCH>Structure>59-Disney Store Roof (Steel dome)	
	GCH>Structure>46-Area1 roof (roof truss)	
	GCH>Structure>49-Area2 roof (roof truss)	
	GCH>Structure>55-Area3 roof (roof truss)	
San Rafael Bridge Retrofit	X	
Imaginary Hotel	X	

Figure 22: A sample exploration task, where the user is searching for reusable items in roof design.

Results

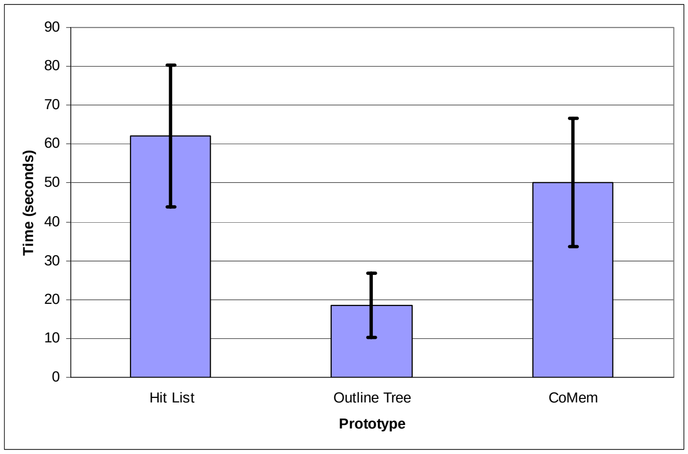


Figure 23: Time to complete a simple retrieval task with 90% confidence intervals displayed.

The time to complete a simple retrieval task is shown in Figure 23 above. The best performance in the case of retrieval was achieved by the Outline Tree, which allowed retrieval tasks to be completed in the shortest time. The Outline Tree is effective for retrieval in the same way that binary search is effective for sorted arrays. By first selecting the project and discipline from much smaller lists than the list of all component objects in the corporate memory, the list of components that need to be visually scanned is greatly reduced.

After the Outline Tree, CoMem allowed retrieval tasks to be completed in the next shortest time. In spite of the fact that it was not developed with retrieval tasks in mind, CoMem still provides support for such tasks. Future research should investigate the role CoMem can play in retrieval tasks.

The average time to complete an exploration task was comparable for the three prototypes CoMem, Outline Tree, and Hit List (14-18 minutes), even though, as discussed below, the user's performance in terms of recall score and context score varied considerably from tool to tool.

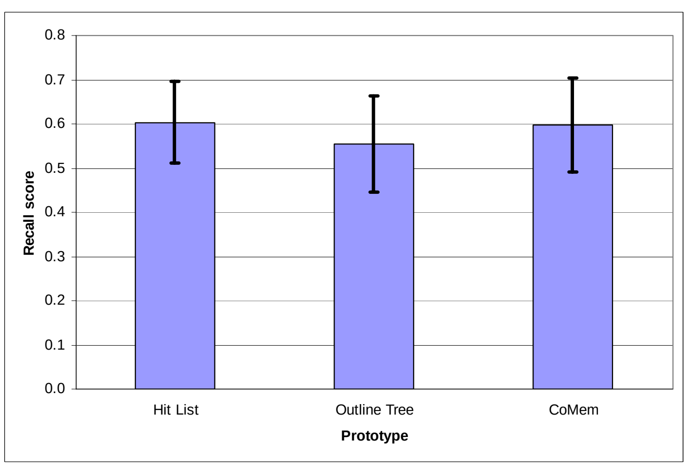


Figure 24: Recall score during exploration tasks with 90% confidence intervals displayed.

Figure 24 shows the fraction of relevant items successfully recalled by the test participants during exploration tasks. CoMem performed well in exploration recall. The Outline Tree had the poorest performance in exploration recall. This can be explained by the fact that in most cases reusable items were buried deep inside the hierarchy (i.e. at the component level) and left very little information scent at the higher levels that appear initially in the Outline Tree. Information scent is the user's perception of the value, cost, or access path of information sources. In the Outline Tree, projects and disciplines are displayed first and must be expanded by the user to display their component children. This requires that, for a relevant component, that component's parent discipline and grandparent project objects must also be relevant in order to encourage the user to expand those sub trees and find the reusable component. This is rarely the case in the CoMem relevance measure.

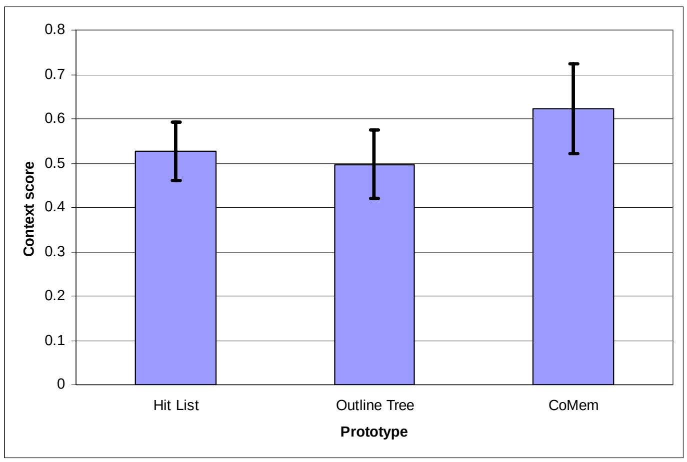


Figure 25: Context score during exploration tasks with 90% confidence intervals displayed.

Figure 25 shows the fraction of contextual questions that could be answered correctly by test participants about the items they retrieved. CoMem performed better than the Outline Tree and Hit List although it also had a slightly larger confidence interval. Most of the contextual questions were based on interactions between the designers, and the resulting version history of the item in question. The CoMem Evolution History Explorer was rated very highly by test participants. It was used during exploration tasks much more extensively than the Project Context Explorer, and repeatedly praised by the participants during the debriefing interview.

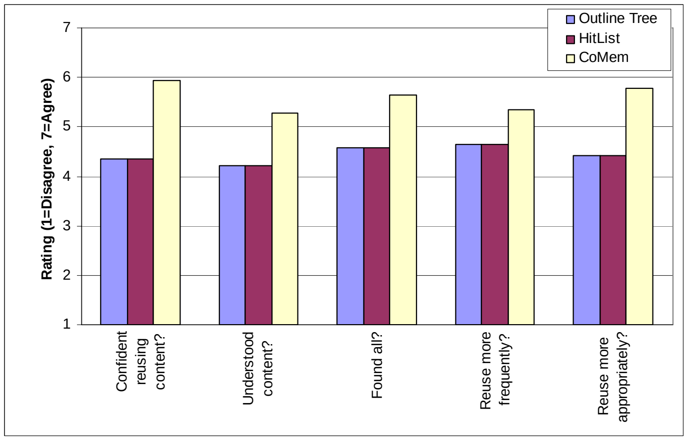


Figure 26: A selection of the questionnaire results.

Figure 26 shows the subjective feedback of the test participants about CoMem, the Outline Tree, and Hit List as gathered from the questionnaires. For the questions regarding general usability characteristics (learnable, complicated, cumbersome), which are not displayed in Figure 26, CoMem attained comparable scores to the Hit List and Outline Tree. This is in spite of the fact that CoMem uses radically different interaction techniques, whereas the other two prototypes are tools with which any average computer user would be very familiar and experienced.

CoMem received higher scores particularly for questions 8-12 (Figure 21). Questions 11 and 12 are the main metrics for the extent to which external reuse is effective: does the user feel that if he/she had that prototype in his/her work, he/she would reuse designs more **frequently** and more **appropriately** (last two questions in Figure 26).

Questions 8, 9, and 10 (first three questions in Figure 26) measure the user's perceived ability to find and understand:

- I would feel very confident reusing some content that I found using this system.
- I had a good understanding of the items I was exploring.
- I felt that I was able to find all potentially reusable items in the corporate memory in the given time.

The high score awarded to CoMem in these questions supports the higher recall and understanding performance measures achieved by the test subjects when using CoMem for exploration tasks.

The users were asked to rate the three CoMem modules: the Overview, the Project Context Explorer, and the Evolution History Explorer. The highest-rated module is the Overview, which validates the claim that providing a succinct overview of the entire corporate memory is extremely valuable, and that a treemap is a good visualization for this purpose. The Evolution History Explorer was also rated very highly. By observing the users during the tests, it is clear that this module enables the users to reconstruct the evolution of the designs and understand the rationale behind this evolution much more effectively than a list of versions or displays of single versions one at a time. The lowest-rated module, although very slightly, is the Project Context Explorer. Many users found it unclear because it shows the same items as those in the Overview, but positioned and colored differently. Further development is needed to couple the Project Context Explorer more tightly with the Overview, so that a change in one display triggers a corresponding change in the other. It is expected that advanced users of CoMem would make more use of the Project Context Explorer.

Discussion

At a global (macro) level, the results test the hypothesis of this research. Traditional tools 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 the claim that the steps of *find* and *understand* lead to effective reuse, as shown in Figure 27.

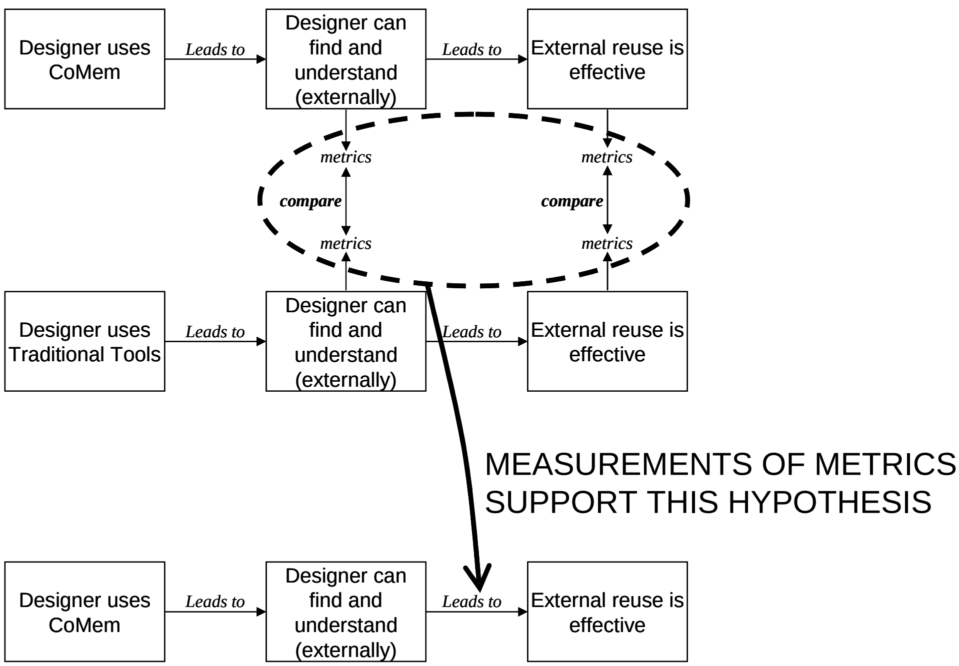


Figure 27: Macro evaluation to test the hypothesis of this research.

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

The other variable that was introduced into the evaluation is repository size.

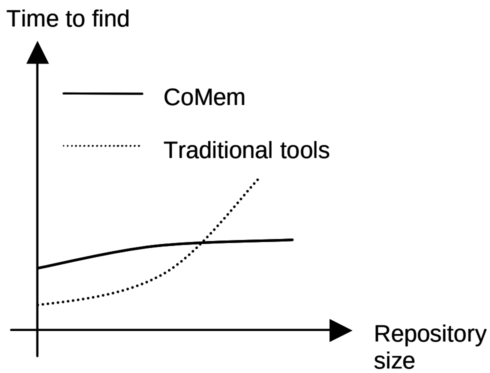


Figure 28: A diagrammatic representation of the hypothesized effect of repository size on the performance of information tools.

Figure 28 shows the hypothesized effect of repository size on the performance of CoMem and traditional tools. Figure 29 and Figure 30 show the actual effects observed on exploration time and retrieval time. In the case of exploration (Figure 29), the size of the repository seems to have little effect. A more subtle aspect such as the amount of text that needs to be read to complete the task is more likely to have an effect on exploration time than the relatively simple count of the number of items in the repository. In the case of retrieval (Figure 30) the results are more similar to the hypothesized effect. As the repository size is increased, the performance of CoMem is assumed to stay approximately constant[4], while that of the Outline Tree begins to deteriorate (takes more time for the larger repository). By simple extrapolation, it can be imagined that a point would be reached beyond which CoMem outperforms the Outline Tree.

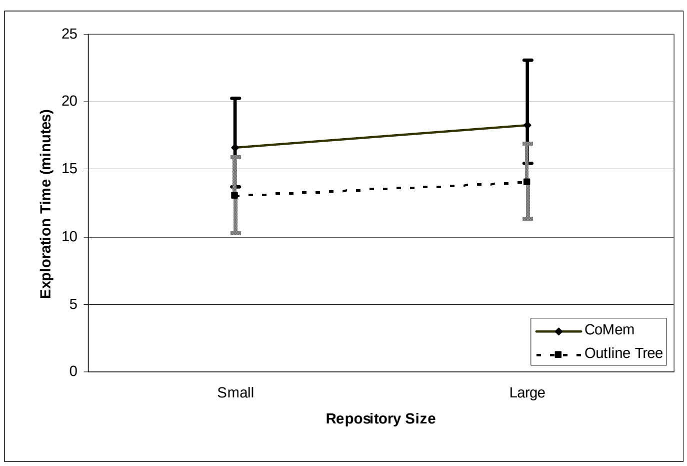


Figure 29: The effect of repository size on exploration time with 90% confidence intervals displayed.

[4] As can be seen from Figure 30, the retrieval time is actually shorter for the larger repository. It can be seen from the 90% confidence interval that the reliability of this result is uncertain and logic dictates that it would take just as long, if not longer, to retrive an item from a larger repository as from a smaller repository so it can be assumed that the performance of CoMem is approximately constant for both repository sizes.

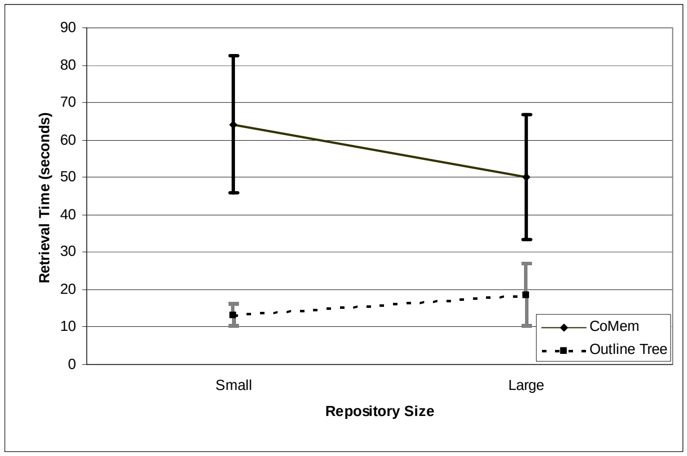


Figure 30: The effect of repository size on retrieval time with 90% confidence intervals displayed.

C h a p t e r   6

DISCUSSION AND CONCLUSIONS

This chapter presents a final discussion of this research as a whole, and particularly the results in light of the stated research hypothesis and research questions. It highlights the contributions of this research, and discusses the conclusions that can be drawn from it.

The objective of this research is to *improve* and *support* the process of design knowledge reuse in the AEC industry. Ethnographic observations show that the three key activities in *internal* knowledge reuse process are:

- *Finding* a reusable item
- Exploring this item's *project context* which leads to *understanding*
- Exploring this item's *evolution history* which leads to *understanding*

The hypothesis is that if the designer's interaction with the external repository enables him/her to:

- Rapidly *find* relevant items of design knowledge
- View each item *in context* in order to *understand* its appropriateness, specifically:
    - Explore its project context
    - Explore its evolution history

⇨ Then the process of reuse will be improved.
This improved reuse will lead to higher quality design solutions, and save time and money.

The internal knowledge reuse aspects of the research questions of this research were addressed through an ethnographic study. Internal knowledge reuse can be formalized into *finding* and *understanding*. Finding occurs by simultaneously comparing data at the three levels of granularity: project, discipline, and component.

The CoMem Overview explores how finding reusable design knowledge may be supported in external repositories using an innovative graphical user interface. The Corporate Map presents a succinct snapshot of the entire corporate memory that enables the user to make such multi-granularity comparisons and quickly find reusable items. In order to provide direct value to the users, and their search tasks, items on the map are color-coded based on their relevance to the current design task.

The *understanding* step in internal reuse occurs by exploring the project context and evolution history of the item being reused. The CoMem Project Ceontext Explorer and the Evolution History Explorer address how this exploration may be supported in external knowledge reuse. The Evolution History Explorer draws from the effectiveness of comic books for telling stories, and explores how this effectiveness can be carried over to the presentation of version histories. The Project Context Explorer combines the relevance measure with the classic fisheye formulation to aid the user in identifying and exploring related items in the corporate memory.

The usability evaluation results presented in Chapter 5 support the hypothesis of this research, that the ability to find and understand does lead to more effective reuse. CoMem offers greater support for finding and understanding than traditional tools, and reuse using CoMem is consistently rated to be more effective by test participants.

Contributions

The main contribution of this research is the recognition that reuse consists of the two tasks of *finding* and *understanding*, and the formalization of the reuse process. An ensuing contribution is the *decoupling* of find and understand, in terms of the tasks that need to be supported, interaction metaphors for supporting these tasks, and processing of the knowledge in the corporate memory to facilitate finding and understanding.

The CoMem prototype constitutes a substantial contribution to information technology in the form of an innovative design of human-computer interface. The domains that it can be applied to are not limited to engineering design, but CoMem can be generalized to the task of finding and using content from large hierarchical repositories.

The spectrum between exploration and retrieval is underlined in this research. Retrieval is disproportionately favored and exploration is commonly neglected in traditional tools. CoMem addresses this imbalance by recognizing the importance of exploration, and appreciating the radically different interfaces that are needed to support it.

This research also makes a methodological contribution through the evaluation of CoMem. The CoMem usability evaluation represents a useful framework for evaluating information interfaces. The same data can be explored using different interfaces. Hit List, Outline Tree, and CoMem cover the spectrum of information interfaces, from traditional to innovative. The important dimensions of the evaluation space are the size of the repository, the type of task, and the user's familiarity with the data. Search engines and expandable/collapsible folder trees can be used to represent traditional information interfaces.

Conclusions

CoMem started with the observation that, whereas designers reusing designs from their personal experiences (*internal memories*) is an extremely effective process, designers reusing designs from digital or paper archives of content from previous projects often fails. From extensive ethnographic studies of practicing designers, we identify two reasons for the effectiveness of internal knowledge reuse:

1. Even though the designer's internal memory is usually very large, he/she is always able to *find* relevant designs or experiences to reuse.
2. For each specific design or part of a design he/she is reusing, he/she is able to retrieve a lot of contextual knowledge. This helps him/her to *understand* this design and apply it to the situation at hand. When describing contextual knowledge to others, the designer explores two contextual dimensions: the *project context* and the *evolution history*.

Armed with these observations, CoMem was developed, an external reuse system that enables designers to:

1. *Find* reusable items in large corporate archives
2. Explore the project context of these items in order to *understand* them
3. Explore the evolution history of these items in order to *understand* them

Based on the three reuse steps identified above – find, explore project context, explore evolution history – CoMem has three corresponding modules: an Overview, a Project Context Explorer, and an Evolution History Explorer.

Future Research

From anecdotal evidence observed during the user tests, the labeling of treemaps plays a very important role in their support for retrieval tasks. Very few of the test subjects used the keyword search function in CoMem during the retrieval tasks. Further research is needed to develop the labeling of treemaps and to understand the role of labeling in retrieval.

CoMem is poised to be generalized to a wide variety of domains. Work is already underway on an interactive workspaces version of CoMem which runs in technology-rich spaces with computing and interaction devices on many different scales (Johanson et al. 2002). CoMem prototypes are being developed for search in textual databases. New functions are being added that exploit concepts from the merging field of chance discovery. This research has laid the foundation for stimulating future research into knowledge capture and reuse, treemaps, measuring relevance, and evaluating information interfaces.

More work is required to investigate the effect of familiarity with the contents of the repository. CoMem must support novice users who are unfamiliar with the contents of the corporate memory as well as advanced users who are able to formulate explicit queries. In practice it will be impossible to be completely familiar with the corporate memory because it is constantly growing and evolving. Further studies should focus specifically on the user's familiarity.

For further readings on this research, refer to Demian 2004, Demian and Fruchter 2004, and Fruchter and Demian 2003, 2002, 2002(a), 2002(b), 2002(c).

REFERENCES

Ahmed S., Blessing L., and Wallace K., 1999. "The relationships between data, information and knowledge based on a preliminary study of engineering designers", Proceeding of the Eleventh International Conference on Design Theory and Methodology (DTM), ASME Design Engineering Technical Conferences (DETC).

Alonso O. and Frakes W. B., 2000. "Visualization of reusable software assets", Lecture Notes in Computer Science, Proceedings of the Sixth International Conference on Software Reuse (ICSR), Advances in Software Reusability, (Frakes W. B., Ed.), Springer-Verlag, Vienna, Austria, pages 251-265.

Altmeyer J. and Shürmann B., 1996. "On design formalization and retrieval of reuse candidates", Proceedings of the Fourth International Conference on Artificial Intelligence in Design (AID), pages 231-250.

Baudin C., Underwood J., and Baya V., 1993. "Using device models to facilitate the retrieval of multimedia design information", Proceedings of Thirteenth International Joint Conference on Artificial Intelligence (IJCAI), pages 1237-1243.

Bilgic T. and Fox M. S., 1996. "Case-based retrieval of engineering design cases: Context as constraints", Proceedings of the Fourth International Conference on Artificial Intelligence in Design (AID), pages 269-288.

Brin S. and Page L., 1998. "The anatomy of a large-scale hypertextual web search engine", Computer Networks and ISDN Systems, Volume 30, Issue 1-7, pages 107-117.

Brooke J., 1996. "SUS: A quick and dirty usability scale", *Usability Evaluation in Industry*, (Jordan P. W., Thomas B., Weerdmeester B. A., and McClelland I. L., Eds.), Taylor and Francis, London, UK, pages 189-194.

Brown J. S. and Duguid P., 2000. *The Social Life of Information*, Harvard Business School Publishing, Boston, MA.

Bruls D. M., Huizing K., and van Wijk J. J., 1999. "Squarified treemaps", 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), (de Leeuw W. and van Liere R., Eds.), Springer-Verlag, Vienna, Austria, pages 33-42.

Chung P. and Goodwin R., 1994. "Representing design history", Proceedings of the Second International Conference on Artificial Intelligence in Design (AID), pages 735-751.

Cross N., 1989. *Engineering Design Methods*, John Wiley and Sons, Chichester, NY.

Culley S. J. and Theobald G., 1997. "Dealing with standard components for knowledge intensive CAD", *Knowledge Intensive CAD*, (Mäntylä M., Finger S., and Tomiyama T., Eds.), Volume II, Chapman and Hall, London, UK, pages 235-255.

Culley S. J., 1998. "Design reuse of standard parts", Proceedings of the Engineering Design Conference on Design Reuse, pages 77-88.

Culley S. J., 1999. "Classification approaches for standard parts to aid design reuse", Proceedings of the Institution of Mechanical Engineers, Part B, Journal of Engineering Manufacture, Volume 213, Issue 2, pages 203-207.

Demian P., 2004. "CoMem: Design knowledge from a Corporate Memory", Doctoral Thesis, Stanford University.

Demian P. and Fruchter R., submitted for publication 2004. "Measuring relevance in support of design reuse from archives of building product models", ASCE Journal of Computing in Civil Engineering.

Dingsøyr T., 1998. "Retrieval of cases by using a Bayesian network", Papers from the AAAI Workshop on Case-Based Reasoning Integrations, pages 50-54.

Domeshek E. and Kolodner J., 1993. "Finding the points of large cases", Artificial Intelligence for Engineering Design, Analysis and Manufacturing (AI EDAM), Volume 7, Issue 2, pages 87-96.

Dorst K., 1997. "Describing design: A comparison of paradigms", Doctoral Thesis, Delft University of Technology, The Netherlands.

Finger S., 1998. "Design reuse and design research – Keynote paper", Proceedings of the Engineering Design Conference 1998: Design Reuse, (Sivaloganathan S. and Shahin T. M. M., Eds.), pages 3-10.

Fruchter R. and Demian P., 2002(a). "CoMem: Design knowledge reuse from a corporate memory", ASCE Proceedings of the Ninth International Conference on Computing in Civil and Building Engineering (ICCCBE-IX), Volume 2, pages 1145-1150.

Fruchter R. and Demian P., 2002(b). "Corporate memory in action", Proceedings of the ASCE International Workshop on Information Technology in Civil Engineering, Computing in Civil Engineering, pages 90-102.

Fruchter R. and Demian P., 2002(c). "Knowledge management for reuse", Proceedings of the CIB W78 Conference, Distributing Knowledge in Building, Volume 1, pages 93-100.

Fruchter R. and Demian P., 2002. "CoMem: Designing an interaction experience for reuse of rich contextual knowledge from a corporate memory", Artificial Intelligence for Engineering Design, Analysis and Manufacturing (AI EDAM), Volume 16, Issue 3, pages 127–147.

Fruchter R. and Demian P., accepted for publication 2003. "Corporate memory", *Knowledge Management in Construction*, (Anumba C. J., Ed.), Thomas Telford, London, UK.

Fruchter R., 1996. "Conceptual, collaborative building design through shared graphics", IEEE Expert: Intelligent Systems, AI in Civil and Structural Engineering, Volume 11, Number 3, pages 33-41.

Fruchter R., Clayton M. J., Krawinkler H., Kunz J., and Teicholz P., 1996. "Interdisciplinary communication medium for collaborative conceptual building design", Journal of Advances in Engineering Software, Computing in Civil and Structural Engineering, Volume 25, Issues 2-3, pages 89-101.

Fruchter R., Reiner K., Leifer L., and Toye G., 1998. "VisionManager: A computer environment for design evolution capture", Journal of Concurrent Engineering: Research and Applications (CERA), Volume 6, Number 1, pages 71-84.

Gerbé O., 1997. "Conceptual graphs for corporate knowledge repositories", Lecture Notes in Artificial Intelligence, Proceedings of the Fifth International Conference on Conceptual Structures (ICCS), Conceptual Structures: Fulfilling Peirce's Dream, (Lukose D., Delugach H., Keeler M., Searle L., and Sowa J., Eds.), Springer-Verlag, Vienna, Austria, pages 474-488.

Gero J. S., 1990. "Design prototypes: A knowledge representation schema for design", AI Magazine, Volume 11, Number 4, pages 26-36.

Grudin J., 2001. "Desituating action: Digital representation of context", Human-Computer Interaction, Special Issue on Context-Aware Computing, Volume 16, pages 269-296.

Jerding D. F. and Stasko J. T., 1994. "Using visualization to foster object-oriented program understanding", Georgia Institute of Technology, Atlanta, GA, Graphics, Visualization, and Usability Center (GVU), Technical Report GIT-GVU-94-33.

Jerding D. F., Stasko J. T., and Ball T., 1997. "Visualizing interactions in program executions", Proceedings of the Nineteenth International Conference on Software Engineering (ICSE), pages 360-370.

Johanson B., Fox A., and Winograd T., 2002. "The interactive workspaces project: Experiences with ubiquitous computing rooms", IEEE Pervasive Computing, Volume 1, Number 2, pages 67-74.

Kazman R. and Carrière S. J., 1998. "View extraction and view fusion in architectural understanding", Proceedings of the Fifth International Conference on Software Reuse (ICSR), pages 290-299.

Kuffner T. A. and Ullman D. G., 1990. "The information requests of mechanical design engineers", Proceedings of the Second International Conference on Design Theory and Methodology (DTM), ASME Design Engineering Technical Conferences (DETC), pages 167-174.

Lakin F., Wambaug H., Leifer L., Cannon D., and Sivard C., 1989. "The electronic design notebook: Performing medium and processing medium", Visual Computer: International Journal of Computer Graphics, Volume 5, Number 4, pages 214-226.

Leifer L., 1997. Design Project Laboratory (ME310), Stanford University, Course Notes.

Maher M. L. and Gómez de Silva Garza A., 1996. "Developing case-based reasoning for structural design", IEEE Expert: Intelligent Systems, AI in Civil and Structural Engineering,, Volume 11, Number 3, pages 42-52.

Maher M. L., 1997. "CASECAD and CADSYN: Implementing case retrieval and case adaptation", *Issues and Applications of Case-Based Reasoning in Design*, (Maher M. L. and Pu P., Eds.), Lawrence Erlbaum Associates, Mahwah, NJ, pages 161-185.

McCall R., 1987. "PHIBIS: Procedurally Hierarchical Issue-Based Information Systems", Proceedings of the ASME Conference on Architecture at the International Congress on Planning and Design Theory, pages 17-22.

Nelson T. H., 1990. "The right way to think about software design", *The Art of Human-Computer Interface Design,* (Laurel B., Ed.), Addison-Wesley, Reading, MA, pages 235-244.

Polanyi M., 1966. *The Tacit Dimension*, Doubleday, Garden City, NY.

Popova M., Johansson P., and Lindgren H., 2002. "An integrated platform for case-based design", Proceedings of the CIB W78 Conference, Distributing Knowledge in Building, Volume 2, pages 99-106.

Regli W. C., Hu X., Atwood M., and Sun W., 2000. "A survey of design rationale systems: Approaches, representation, capture and retrieval", Engineering with Computers, Volume 16, Numbers 3-4, Springer-Verlag, Vienna, Austria, pages 209-235.

Reiner K. and Fruchter R., 2000. "Project memory capture in globally distributed facility design", ASCE Proceedings of the Eighth International Conference on Computing in Civil and Building Engineering (ICCCBE-VIII), Volume 2, pages 820-827.

Retkowsky F., 1998. "Software reuse from an external memory: The cognitive issues of support tools", Proceedings of the Tenth Workshop on Psychology of Programming Interest Group (PPIG).

Schank R. C., 1990. *Tell Me a Story: A New Look at Real and Artificial Memory*, Scribner, New York, NY.

Schön D. A., 1983. *The Reflective Practitioner: How Professionals Think in Action*, Basic Books, New York, NY.

Shahin T. M. M., Sivaloganathan S., and Gilliver R., 1997. "Automation of feature-based modelling and finite element analysis for optimal design", Proceedings of the Eleventh International Conference on Engineering Design (ICED).

Shneiderman B., 1999. "Supporting creativity with advanced information-abundant user interfaces", University of Maryland, College Park, MD, The Institute for Systems Research (ISR), Technical Report 1999-73. Also available from *Frontiers in Human-Centred Computing, Online Communities and Virtual Environment*, 2001, (Earnshaw R., Guedj R., Van Dam A., and Vince J., Eds.), Springer-Verlag, London, UK, pages 469-480.

Simoff S. J. and Maher M. L., 1998. "Data mining in hypermedia case libraries", Proceedings of the Fifth International Conference on Artificial Intelligence in Design (AID), Machine Learning in Design (MLinD) Workshop.

Simon H. A., 1969. *The Sciences of the Artificial*, MIT Press, Cambridge, MA.

Ullman D. G., 1994. "Issues critical to the development of design history, design rationale and design intent systems", Proceeding of the International Conference on Design Theory and Methodology (DTM), ASME Design Engineering Technical Conferences (DETC), Volume 68, pages 249-258.

Ware C., Hui D., and Franck G., 1993. "Visualizing object oriented software in three dimensions", Proceedings of the Conference of the Centre for Advanced Studies on Collaborative Research: Software Engineering (CASCON), IBM Centre for Advanced Studies Conference, Volume 1, pages 612 – 620.

Ye Y. and Fischer G., 2002. "Supporting reuse by delivering task-relevant and personalized information", Proceedings of the Twenty-Fourth International Conference on Software Engineering (ICSE), pages 513-523.

Zack M. H., 1999. "Managing codified knowledge", Sloan Management Review, Volume 40, Number 4, pages 45-58.

Zieliński K., Laurentowski A., Szymaszek J., and Uszok A., 1995. "A tool for monitoring heterogeneous distributed object applications", Proceedings of the Fifteenth International Conference on Distributed Computing Systems (ICDCS), pages 11-18.