Engineering with Computers (2005) 21: 177–192
DOI 10.1007/s00366-005-0003-9

ORIGINAL ARTICLE

Maria C. Yang · William H. Wood III
Mark R. Cutkosky

Design information retrieval: a thesauri-based approach for reuse of informal design information

Received: 26 July 2004 / Accepted: 9 February 2005 / Published online: 10 November 2005

Abstract Information is integral to the engineering design process, and gaining access to design knowledge is critical to effective design decision-making. This paper considers the indexing and retrieval of informal, unstructured information captured from electronic design logbooks. One of the key observations of informal design information is its evolutionary nature over time. While this characteristic makes informal information a rich source for reuse, it also makes it difficult to employ traditional information retrieval (IR) approaches. The work described in this paper is based on a framework developed specifically for the information handling requirements of designers. This manual method for indexing information is adapted to meet the evolutionary nature of design through the development of thesauri for design context. Several approaches to building thesauri are examined, including manual and automated methods. It is found that manual methods provide a high level of IR performance, but also have high overhead requirements. Machine methods, however, may provide a viable, low overhead alternative.

Keywords Engineering design · Design information · Design process · Information retrieval

Maria C. Yang (✉)
Daniel J. Epstein Department of Industrial and Systems Engineering, University of Southern California,
3715 McClintock Avenue, GER 201, Los Angeles,
CA 90089, USA
E-mail: maria.yang@usc.edu
Tel.: +1-213-7403543
Fax: +1-213-7401120

W. H. Wood III
Department of Mechanical Engineering,
University of Maryland Baltimore County,
1000 Hilltop Circle, Baltimore, CA 21250, USA

M. R. Cutkosky
Department of Mechanical Engineering, Terman 523, Stanford University, Stanford, CA 94305, USA

1 Introduction

Information is integral to engineering design [1, 2]. The engineering design process generates and transforms a wide variety of information, from quantitative specifications data to informal knowledge about design process and decision making. In fact, Hales [3] has found that the majority of time spent by designers and engineers on a large-scale design project was on accessing, processing, or sending out information rather than on traditional "design" activities. Likewise, Court et al. [4] concluded that 50% of designers' time was focused on managing the information that stems from engineering design process. In particular, information from the early, conceptual phases of design can strongly influence the direction of the later stages of design [5], as well as the considerations of future designers. Indeed, Simpson et al. [6] describe the importance of maintaining design freedom while increasing design knowledge in the early stages of design. This paper investigates information retrieval (IR) approaches to accessing conceptual design information for future reuse.

In an analysis of ways in which information is accessed, Court et al. [7] found that designers rely heavily on their past experiences during the design process. Design reuse is the transfer of that knowledge of designers and engineers to other designers and engineers [8–10]. Knowledge of past design cycles can provide insight on past design alternatives, decisions, and rationale, and can prevent costly duplications in design effort. In short, by gaining appropriate access to past knowledge, designers can achieve better designs.

Before design information can be reused, it must first be captured. Here, capture involves the archiving of information as it is generated, without imposing structure. This allows the full richness of informal information to be documented. We turn to a digital form of the traditional engineering logbook, the electronic design notebook, as a way to collect information design information.

Numerous methods exist for searching and retrieving archived information, but many of these assume the information is formal and structured. While the informal nature of design information makes it valuable to future designers, it also makes its retrieval challenging. The lack of formal representation for informal information is a fundamental challenge [11]. In design, the vocabulary used to describe an artifact is linked to the artifact itself, and in the same way an artifact evolves over the life of a project, the language used to describe that artifact changes as well. For example, early on in a project, an ambiguous part might be referred to as "thingamajig." Later on, however, the part may acquire a more standard name such as "lead screw." This condition in which multiple terms can represent the same concept is a well-known problem in natural language known as *anaphora* [12].

We base the retrieval work in this paper on an approach drawn from the information requirements of engineering designers known as Dedal [13]. Dedal has been shown to be a highly effective method for retrieving design information, with performance that is far better than typical. However, in its original form, it is an entirely manually indexed endeavor that requires a great deal of human effort and overhead. As we know from the domain of usability, it is important for designers to retrieve past design work, but if that work is not easily accessible, its value is soon overshadowed by the effort required to find it [14]. The implicit information needs of engineers and designers are also characterized by using quantitative methods of Song et al. [15].

One strategy we investigate to address the evolving terminology of design is domain-specific thesauri. Thesauri improve IR performance by expanding queries to include synonyms of terms. This has shown to be the case both in design IR [16] and in IR in general [17]. The best thesaurus to use is one that will not add terms to the query which might hurt performance. For this reason, it is important to limit the scope of a thesaurus to the domain of interest. In this work, we develop representations of informal design information based on artifact models that are applied as thesauri. We compare the IR performance of manually built thesauri drawn from both informal and formal information sources. We further examine the viability of extracting thesauri automatically using Latent Semantic Analysis [18], thereby reducing overhead. This work is cumulative of earlier work from Refs. [16, 19, 20].

The questions we seek to answer are:

- How does the evolution of design information affect its retrieval?
- Can thesauri be a viable approach to improving retrieval?
- How useful is informal information compared to formal information as a source of thesauri?
- How well do machine-generated thesauri compare to hand-generated thesauri for retrieving information from electronic notebooks?

2 Related work

2.1 Formality and informality in design information

The approach this paper takes to design information grows from observations about the structure of design information itself, and about notions of formality and informality. Informal design information is unstructured text, captured as it is generated. Figure 1 shows a formality spectrum of design information. At the formal end are highly structured, detailed documents, such as final reports, patents, and CAD drawings, while at the informal end are unstructured, fragmentary documents, such as those captured in design logbooks. In the middle is semiformal information, which is essentially informal information with a limited amount of structure imposed, such as design rationale systems or case studies.

2.2 Formal information

One of the most well-known structured information paradigms for design is the issue based information system (IBIS) [21]. The IBIS structure of issue–position–argument has been a fruitful basis for many other design information systems [22, 23]. They classify information by node type (issue, position, or argument), with links to other types of supporting information. Regli et al. [24] provide a comprehensive overview of structured design rationale capture. Other work models the design process as a formalized exchange of design information [25, 26]. These systems utilize ontologies of engineering design language [27, 28] to provide a shared understanding of a design.

Formal design information research includes work in CAD, including Smart Drawing [29], Interdisciplinary Communication Medium (ICM) [30], design information infrastructure [31], the Learning Shell for Iterative Design (LSID) that utilizes design histories in routine, parametric design [32], and tools for extracting relationships between geometric entities [33]. These systems rely on traditional CAD representations of information

Fig. 1 Spectrum of formality of design information

as a base, linking informal information about a design to parts of the CAD model using databases and other information management tools.

Work has been done in converting short, informal notes that annotate CAD drawings into a frame/slot format using lexical analysis [34], but Boujut [35] argues such approaches must be able to represent complex notions with sufficient detail. Jacobsen et al. [36] have developed the beginnings of a framework for describing functional requirements in terms of an engineering-specific hierarchy of verbs. In addition, major CAD vendors offer data "safes" in an effort to store information related to a design. Unfortunately, most CAD applications capture geometric data without capturing information as to why something is shaped or arranged the way the designer has specified. Parametric CAD (e.g., Pro/Engineer, AutoCAD Mechanical Desktop, etc.) provides a way of capturing design relationships as mathematical constraints among geometrical entities but falls far short of rationale capture.

2.3 Informal information

Informal design information is valuable because it reflects many important aspects of the design process that are not found in formal documentation [3, 37–39]. As found in the extensive work by Court et al. [7] and Culley et al. [40], a great deal of important informal information is generated before it is later processed into formal information. Potentially valuable informal information may be lost in its translation into formal representations of information. Furthermore, little of this type of information has been previously documented in electronic form. Liang et al. [41] observed the IR patterns of student designers of both formal and informal design information. It was determined that 85% of the information retrieved dealt with design *process*, while only 15% of the retrieved information referred to the product. It was also found that designers tend to refer back to their own engineering notes, but not to the notes of others. Two conjectured reasons: (a) search capabilities for informal engineering notes are limited, and (b) it is difficult for designers to contextualize the informal work of others [42]. Taken together, this suggests that informal information is valuable to designers, but is more difficult for both humans and machines to access specific pieces of information when it is unstructured. The issue is how to access this information best.

One approach for observing informal design practice in real time is protocol analysis. Cross et al. [43] collected a number of protocol studies that examine the interaction that takes place during design activity. Based on a similar set of protocol studies, Tomiyama [44] developed a computational design process model. Ullman et al. [45] and Kuffner and Ullman [46] have done a number of protocol studies of the tasks in design and the types of information that is sought by mechanical designers during the process. Baudin et al. [47] studied the question-asking behavior of engineers, codifying the types of questions posed during design. This work, known as Dedal, serves as the foundation for the work on informal design IR in this paper.

A semiformal approach to design information is design case studies. Wood and Agogino [9] and Kolodner [48] take a case-based approach to accessing informal design information, concentrating on conceptual design information found in case studies.

2.4 Design information capture and reuse

Before informal design information can be analyzed, it must be first acquired. There are many ways of documenting design, and information capture is distinct in that it archives information as it is generated, in its original context [49]. Analysis of different communication methods used in the class suggests that informal methods of concept generation, like brainstorming and electronic notebooks, include more ideas than more formal methods, like presentations and final documents. Both Petroski [2] and Kolodner [48] tell us that in order for design information to be useful it must be embedded in the process that generated it. Reuse is then a matter of "replaying" the design under new constraints. The capture of design information is relevant to design IR because of the informality of archived information. There are many potential sources of captured design information available. Yen et al. [50] determined which communication methods generate more ideas for design.

Design logbooks are of special interest because they capture information as it is created, covering design information comprehensively with a richness not found in more formal documentation [51]. In this paper, we examine electronic design notebooks which offer electronic capture of design process knowledge.

2.5 Trade-offs of informal and formal design information

As observed in artificial intelligence [52], there is an inherent trade-off between formal and informal systems for computers and how the overhead associated with formality affects users [53]. Computers need to be given explicit steps in order to compute or think effectively. However, humans tend to perform complex tasks intuitively. In design information capture, formality must be considered in the effort required to capture it [22, 54].

This paper focuses on both manual and automated approaches of providing structure to informal information. The rationale for automation is based in part on work by Baya and Leifer [55], who showed that in activities like brainstorming, which emphasize the fluid generation of concepts, information type changes rapidly, in a matter of seconds. Tools that require users to

structure their ideas in a quick-changing environment may not be as effective, forcing designers to slow down or focus on fewer concepts than if they did not have to stop and classify ideas. Structure imposed by the needs of the computer can be costly to implement and limit the capture of information. The goal is to structure information with limited overhead to the designer.

2.6 Information retrieval basics

There are two main steps in IR [56]: indexing a collection and retrieving the documents from the index. Documents are represented as vectors, with all the terms present in the collection determining the length of the vector.

Figure 2 shows an example corpus matrix for a collection of mechanical engineering documents. The rows are titles of documents in the collection, and the columns are the terms that appear in the documents in the collection. The matrix values c_{ij} are weights that represent the importance of terms in documents. One traditional scheme is term-frequency inverse document frequency (tf-idf) weighting in which the calculation of weights is rooted in empirical studies [17] that show the relevance of a term is related to the frequency with which it appears in a document. For example, if the word "motor" appears many times in a single document, then "motor" is a concept likely relevant to that document. However, if "motor" appears in every document in the collection, then "motor" is less unique as an identifying term. Weights are directly proportional to the number of times a particular term appears in a document (term frequency or tf_j), and inversely proportional to the number of documents that the term appears in (the document frequency df_j). Very common but meaningless words, such as "the" and "and," called stopwords, are stripped out.

The task of mapping a query into a set of possibly relevant documents is referred to as IR. Eschewing natural language interpretations of the query and the documents, the common technical method of IR is to map textual language into symbol vectors which can be easily manipulated mathematically. The result set generated by IR is a rank ordered list of documents which likely contain information that the user has specified.

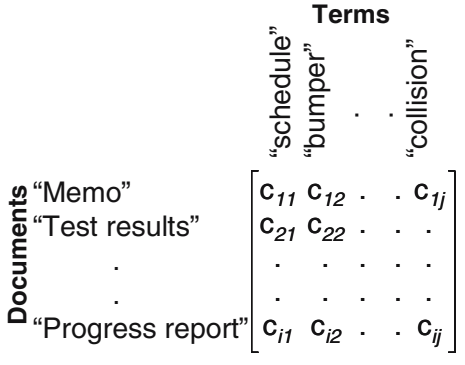

Fig. 2 Vector space representation of a document collection, or corpus matrix

Examples of common IR systems include Web search engines such as Google (http://www.google.com) [57] and Altavista (http://www.altavista.com), both of which use schemes that include tf-idf. The IR systems provide fast, but not always accurate, answers to the questions posed by Web users. Search engines are designed to handle generic collections of text, based on word frequency, regardless of the content of the collections. Articles about rapid prototyping are handled in exactly the same way as pages on biochemistry. In this research, the goal is to make search more effective in the design domain through the addition of relevant context, such as a thesaurus of design terms.

Although there are many tools that can make search more intelligent through syntactic and semantic analysis, these methods have the drawback of being computationally expensive and difficult to maintain. Simple "word count" methods for IR like vector clustering have proven to be ubiquitous for search on the Web. This is not to say that current search engines are particularly effective at search, but only that speed, scalability, and maintenance are sometimes more important than actual search engine effectiveness.

2.7 Performance measures for information retrieval

Two metrics are usually used to describe the quality of an IR system: *recall*, the proportion of relevant documents retrieved by the system; and *precision*, the proportion of retrieved documents that are relevant. Precision is an accuracy measure, while recall is a measure of how much good information is retrieved. Poor precision means users may have to wade through mountains of bad information, while poor recall means that much good information is missed. Perfect performance would constitute 100% precision and 100% recall, meaning that all the possible correct documents are retrieved, with no incorrect retrievals. In practice, however, this is rare. In fact, the two measures are coupled—an increase in precision is usually made at the expense of recall and vice versa. In large part, it is up to the user's preferences of recall versus precision to determine which overall strategies are most useful.

Precision and recall are components of another indicator of performance for an IR, the system constant K:

$$K = \text{Precision} \times \text{Recall} \tag{1}$$

Empirically, this value K tends to be constant for a given IR scenario, so performance is exhibited in constant curves. Precision and recall also tend to relate inversely. When precision is improved, it is often at the expense of recall, and vice versa.

2.8 Thesauri

Design IR uses domain specific thesauri to aid in the search for design information. The design process is artifact based, and artifacts inherently change over the life of a project. The context of design artifacts provides an opportunity to customize the basic methods of IR. When searching across contexts, the thesaurus improves performance two-fold. Yang and Cutkosky [20] demonstrate results from applying generic thesauri in a Boolean term matching mode. Dong and Agogino [58] use machine learning techniques over IR representations of design documents to induce a directed graph of relationships among design concepts. The method tags design information by part of speech, extracts noun phrases (generally considered to carry most of the information in the text), and finds co-occurrences among them. These co-occurrences are then fed into a Bayes-network learner to extract relationships among concepts. In more recent work, Dong et al. [59] have employed latent semantic indexing (LSI) [18] to characterize team coherence.

3 Methods

3.1 Dedal framework

The framework on which this work in design IR is based is called Dedal [13]. Dedal was derived from protocol studies of designers at work. Designers were observed as they voiced the questions that emerged while engaged in a redesign task. It was found that the questions tended to focus on various functional and developmental aspects of the physical artifact being designed. The research team codified these questions into a two-part format consisting of a *descriptor* and a *subject* (Fig. 3). A *descriptor* is a generic engineering concept that crops up repeatedly in design discourse, and set of ten of the most common descriptors was defined. Engineers want to consider *alternatives*, for example, and examine *assumptions*. The *subject* is a specific part of the device model, such as a *motor* or a *linkage*.

3.2 Dedal performance and trade-offs

In early versions of Dedal without a thesaurus, final report documentation was indexed by hand for descriptors and subjects. This resulted in precision of up to 70% and recall up to 90%, considered unusually good performance for an IR system. The cost for this approach, however, was both a large expense for hand-indexing design information and the static nature of the indexing model which does not easily adapt to an evolving design model, so the basic Dedal methodology does not lend itself to in-process retrieval unless the hand-indexing process is regularly repeated.

The heart of the work in this paper lies in automating Dedal indexing through the use of thesaurus terms. Methods of creating effective thesauri for Dedal terms are explored in detail, including both manual and machine methods of development. Manually built thesauri are composed of design terms that are based on models of the artifacts being designed. These terms relate to design structure, components, and function. The approach to automatically generating thesauri eschews labor and knowledge intensive model building for the more tenable problem of extraction of related terms using modal analysis.

In this paper, we strive to capitalize on the power of the Dedal's design IR model while mitigating the resource demands of hand-indexing design text. In essence, we use a semiformal IR framework to retrieve informal information. We extend the effectiveness of design IR by taking aspects three lines of research. From Ref. [13], it is observed that designers' questions tend to fall into distinct categories, we take the model of descriptor–subject querying and the notion that there are generic design descriptors. From Ref. [16], we take the IR focus and reinforce the idea that a generic thesaurus is of value across design contexts. Finally, from Ref. [58], we take the notion that we can extract meaningful design representations from design text.

Dedal's descriptor–subject pair, shown in Fig. 3, presents two distinct points of introduction for thesaurus terms which might improve IR performance. This paper describes two general experiments on the

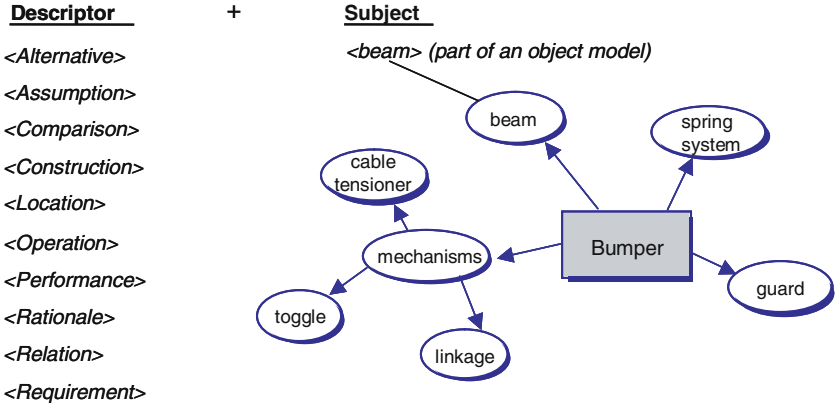

Fig. 3 The Dedal framework

development of a *subject* thesaurus. The first experiment is designed to identify the best source of information from which to hand construct a thesaurus and the generality of information which should be included in it. The second experiment builds on the first, automatically generating a thesaurus from the best information source. Finally, we compare the performance differences between hand- and machine-generated thesauri.

Application of a generic thesaurus for the descriptor query component was explored by Yang and Cutkosky [20]. This thesaurus was constructed manually, using terms drawn from both general-purpose thesaurus and the text of design notebooks themselves. While this is a time consuming process, the thesaurus does not have to be continually generated because of the generic, stable nature of descriptors. An example synonym for the descriptor *Alternative* would be *possibilities*. In this case, the Dedal query *Alternative* of *actuator* would return any block of text containing both the word *possibilities* and the manually indexed subject *actuator*. Tests on three different electronic notebooks with three different descriptors improved retrieval precision between 30 and 50% over nonthesaurus searches.

Two sources for *subject* thesaurus terms were examined in these studies: final project reports (including CAD drawings and diagrams) and informal project design notebooks. As discussed earlier, the formality of the final reports eases the task of creating a thesaurus, but because these reports are generated specifically due to the academic nature of the design projects they may not generalize to nonacademic design (although certainly the CAD portions of final reports are generic). On the other hand, the design notebooks represent the generic communication that takes place in the process of team design.

The electronic design notebooks employed in this study were created in PENS (Personal Electronic Notebook with Sharing) [60], a tool for generating collaborative, Web-based design notebooks using text and graphics. Design teams documented much of their work in these design notebooks, from to-do lists to formal reports. As a result, the design notebooks capture much of each team's design process. Notebook entries vary widely in nature, from fragmentary to well organized. The overriding tenor of all of this research activity is the ability to capture design nuances through much richer representations (most commonly free text) that can be easily accomplished through machine-understandable data coding. Comprehensive final reports for the class were generated by teams at the end of the year. Reports contained detailed information on the final design, such as CAD drawings and diagrams, as well as content drawn from the design notebooks.

The generation of models from CAD drawings and other diagrams from a final report is straightforward. Part names and relationships are relatively unambiguous. In the design notebooks, informal, partial device models are generated constantly throughout the design process. These models are usually fragmentary, with the team concentrating on only a portion of the design at a time. Figure 4 shows how the view of a design that emerges from final documentation differs substantially from that seen in day-to-day documentation. The language used to describe parts of a design in these notebooks can be very different than the language used in a final report or CAD drawing, potentially changing with each design iteration. Immersed in the design task, the language of discourse can also be very general (i.e., calling the fountain assembly the "prototype").

The question is: what is the better source of information for generating a subject thesaurus, informal in-process documents or formal final design documents? In addition, we must understand how we might customize the retrieval process for each notebook and the trade-off between the effort required for customization and its impact on retrieval performance.

The queries used in these experiments were drawn from those expressed in the design notebooks, and then translated into the two-part Dedal format. For example,

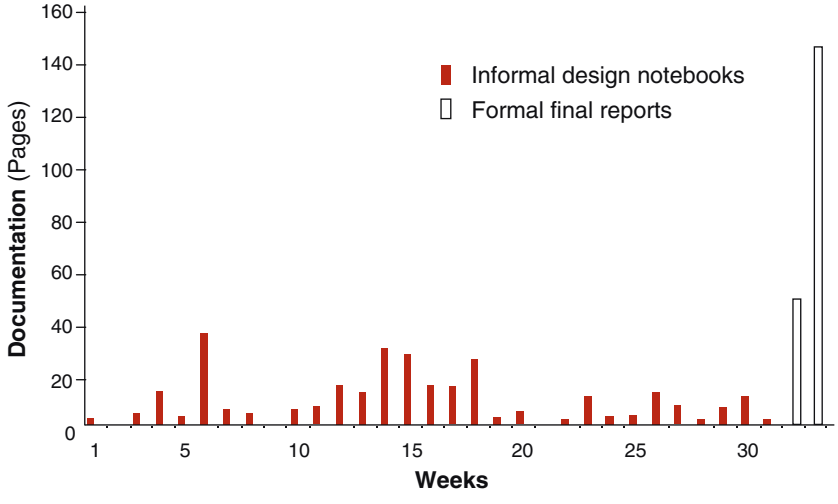

Fig. 4 Comparison of documentation quantity between informal and formal sources

Table 1 Selected natural language queries and their Dedal query equivalents

Natural language query	Dedal query	
	Descriptor	Subject
Where does the bolt go?	< Location >	< bolt >
Which motor is better	< Comparison >	< motors >
What mechanisms were considered?	< Alternative >	< mechanism >

the natural language question "what actuator alternatives were considered?" becomes the Dedal query *alternative* of *actuator*. The set of relevant documents for each of the queries was determined through an exhaustive search of each notebook. Some other queries appear in Table 1.

Precision and recall both depend on knowing, for each query, which documents from the corpus are relevant. Thus, it is necessary to use human judges to determine the set of documents that a query should return. This set can then be used to calculate a retrieval system's precision and recall. Both user queries and the documents held within a corpus are each represented in the same way, through a vector of weights associated with the words they contain.

These studies focused on three sets of project documentation, including the electronic design notebooks from all designers and the final report, from a graduate level course in electromechanical design at Stanford University. Students work in teams of three or more over a 9-month period, starting from conceptual design to working prototype. All projects in the course were industry sponsored and funded, and the representative projects examined here include the design of a car bumper, a water fountain, and a personal digital assistant. These projects cover different types of design: redesign of an existing product and new conceptual design. Typical corpus size is ~5,000 documents (~2 MB of ASCII text).

In the research described here, the SMART system [17] is used to test out several strategies for gaining access to design information. The basics of the system include query/document representation, indexing, and similarity measurement:

Indexing: Documents within a corpus are indexed as a group. As each document is read, each term that occurs is inspected. If it is a common word, it may be discarded. If it is kept, it is stemmed (common prefixes and suffixes are removed) and entered into the dictionary. The term count for the document (tf) is then incremented. If it is the first occurrence of the term in a document, the document count (df) for that term is incremented. Finally, all of this information is synthesized into a weight with which each term in the dictionary represents each document in the corpus. In SMART, these weights are calculated as follows:

$$c_{ij} = \log(1 + \mathrm{tf}_{ij})/\mathrm{df}_j \tag{2}$$

where i is the document index, j is the term index, tf_{ij} is the count of term j in document i, and df_j is the number of documents containing term j.

In mapping a query to documents within the corpus, a simple matrix multiplication is used to measure similarity. The document corpus matrix (C, m rows of n-dimensional document vectors) is multiplied by the query vector (q, n term weights in column form), resulting in a column vector of query–document similarities (s, m document similarities) as given in

$$C \cdot q = s \quad \begin{bmatrix} c_{11} & c_{12} & c_{13} & \cdot & \cdot & c_{1j} \\ c_{21} & c_{22} & c_{23} & \cdot & \cdot & \cdot \\ c_{31} & c_{32} & c_{33} & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ c_{i1} & c_{i2} & c_{i3} & \cdot & \cdot & c_{ij} \end{bmatrix} \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ \cdot \\ \cdot \\ q_j \end{bmatrix} = \begin{bmatrix} s_1 \\ s_2 \\ s_3 \\ \cdot \\ \cdot \\ s_1 \end{bmatrix} \tag{3}$$

Technically, this form of IR is both simple and elegant. However, much additional work is directed toward circumventing the assumption that the meaning of a document can be determined solely from the profile of words that occur in it.

3.3 Study 1: manually derived subject the-sauri—exploring information sources

Two artifact models were created for each project, one drawn from *formal* information found in final reports another from *informal* design model fragments found throughout the design notebooks. The level of formality of these models is sufficient only for determining system decomposition and for assigning names to subsystems. Synonyms for both form and function were assigned to each node. Examples of formal and informal models for the car bumper project are shown in Fig. 5. The Dedal subject is a "legform impactor," which is a device used to test car bumpers. A simulated mechanical leg is suspended from above then a test bumper collides with the legform impactor. Sensors on the legform impactor characterize the impact.

To better understand the difference in language between formal and informal models, we can compare the components. In the formal model, the legform impactor includes an "acceleration sensor," but in the informal model, the legform impactor contains an "accelerometer." These phrases clearly represent the same notion, but because the words are not exactly the same, a traditional IR search for the terms would product different results.

Form synonyms and functions for the informal model were generated by examining the design notebooks carefully for references to the part. Two distinct ways of referring to parts were found: (1) domain specific synonyms and (2) generic design words. Domain specific synonyms for the supporting structure of the fountain are "skeleton," "box," or "cradle"; their

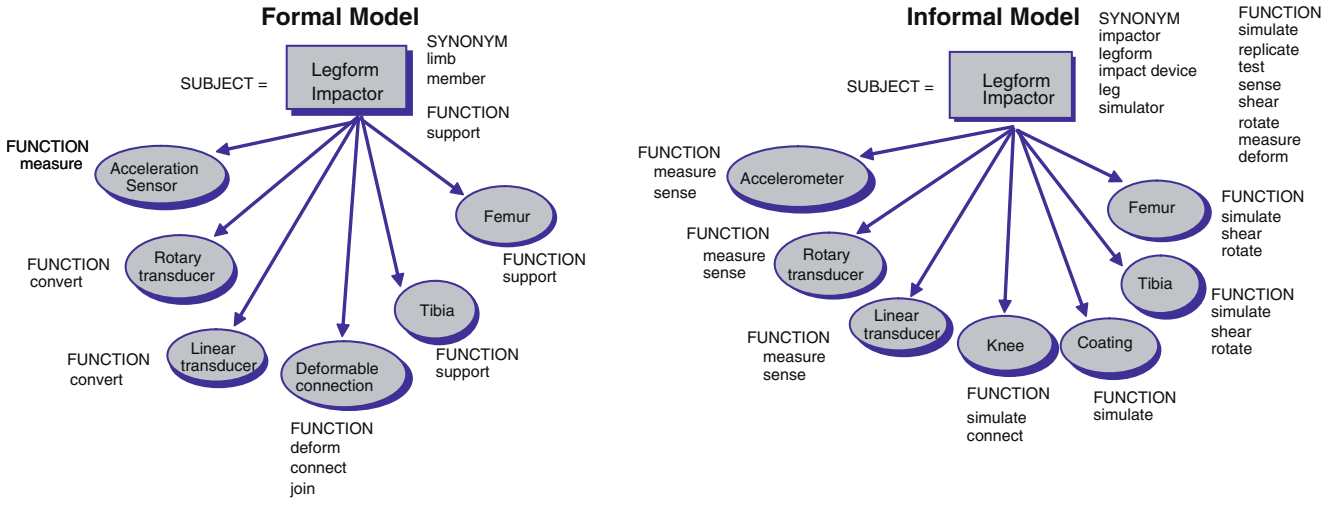

Fig. 5 Difference in component naming in models drawn from formal sources (*left*) and informal sources (*right*)

association requires some understanding of the project and its jargon. Generic design words were used like pronouns to describe a design: the fountain project's nozzle is variously called a "system," a "device," a "prototype," and a "design."

The models further include information about synonyms and function of the components that will be used to expand search. Synonyms and function for parts of the formal model were found by using a Web-based dictionary and thesaurus engine (Hypertext Webster Gateway at UCSD). This dictionary includes searches on Webster's Dictionary and WordNet, a semantic net thesaurus, and is presumably an objective, repeatable way of finding such information. While searching for these synonyms and functions, however, it quickly became obvious that many of these terms, like "Pedestrian Impact Guard", are too specific to be found in a general-purpose dictionary. In these cases, the most general form of that part name, such as "guard," was used.

Creation of models for the informal thesaurus was less straightforward than for the formal models because of the implicit and incomplete nature of model description in informal sources such as design logbooks. While some diagrams and drawings are provided in the electronic logbooks, they are not contextualized as thoroughly as might be found in a final report. Ideally, neat models could be extracted from a notebook from time to time to show "snapshots" of a changing design, but it is difficult to create a single model for an evolving design. For this reason, fragments from various stages of design are linked together.

4 Results

Retrieval runs were performed for a set of questions for each design notebook. Various strategies for applying the subject thesaurus are given below. In each case the same descriptor thesaurus was used to add synonyms to the descriptor element:

- *Subject only (baseline)*: search using the Dedal subject alone (no subject thesaurus)
- *+Component thesaurus*: add form synonyms for the subject
- *+Function thesaurus*: add function terms for the subject
- Parent–child: add parent/child termsNote that relevance is a Boolean decision and no specific ordering is given for the return set; however, as Eq. 3 shows, SMART gives a retrieval set that can be ordered by its similarity measure. Thus, precision and recall are calculated for various thresholds of similarity. We will show precision at distinct recall levels.

Figure 6 shows the results of using subject thesauri drawn from formal documentation. These curves illustrate precision at specific values of recall, and from the shape of the curve we see that precision and recall are generally inversely related. Better performance means that a particular curve is further up and to the right. At the lower values of recall, we see that the addition of component and function synonyms boosts precision slightly.

Figure 7 shows the use of thesauri drawn from informal documentation. It is clear that, for all levels of recall, precision is higher than the baseline without a thesaurus. Furthermore, we see that there is slight increase in performance at points due to the addition of function-related terms.

The effect of function and component terms as synonyms is detailed in Fig. 8. Function terms improve search over the baseline only slightly, and boost the effect of component terms. However, the use of component terms in the subject thesaurus clearly has the biggest effect.

Figure 9 shows a direct comparison between the best performing set of thesauri drawn from formal

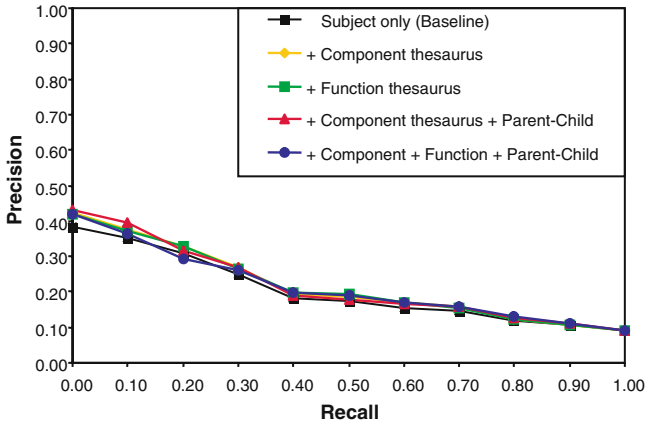

Fig. 6 Information retrieval using thesauri drawn from formal documentation

documents and the best of the informal. At all points, we see that informal thesauri outperform formal thesauri. Overall, the difference between the informal and formal thesaurus performance is nearly 40%.

The difference in terminology between in-process and final documentation is significant; a thesaurus used to access actual in-process documentation must be derived from it. This result is not surprising but implies that final design documents do not provide a shortcut useful for representing the evolution of ideas throughout the design process.

Another measure that can be used to understand the difference in performance of IR systems is the system constant K, which is the product of precision and recall. K roughly indicates the overall performance of a particular IR approach. Figure 10 compares K values for the informal thesaurus against those for the formal thesaurus for each retrieval episode. Because the majority of points fall above the $K_{\text{informal}} = K_{\text{formal}}$ line, it appears that the informal thesaurus is more effective at finding correct answers than the formal thesaurus. Of particular note is the set of points aligned near the vertical axis. This indicates that the informal thesaurus is effective in cases where the formal thesaurus is almost

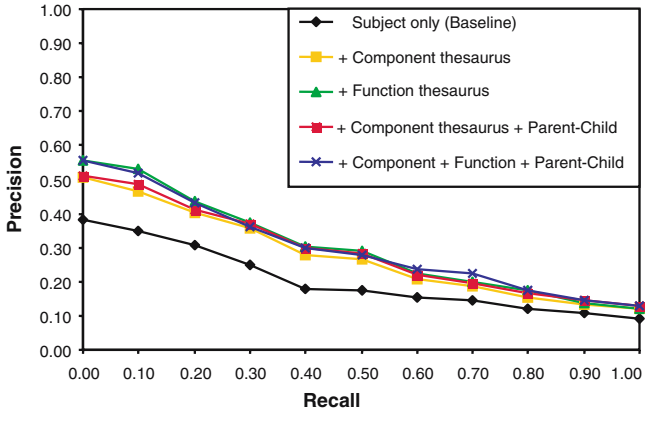

Fig. 7 Information retrieval using thesauri drawn from informal documentation

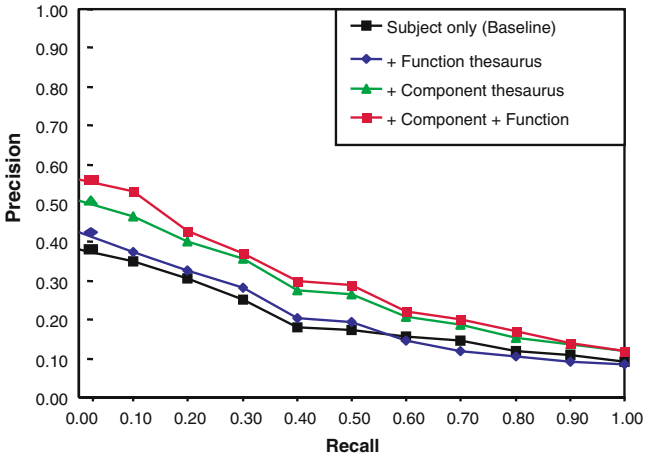

Fig. 8 Comparison of the usefulness of function terms

completely useless, a strong result which might indicate a change in language from the design process to the final design documentation.

Figure 11 details the results of Fig. 10, showing the difference in precision and recall performance between the two thesauri. The informal thesaurus almost always outperforms the formal thesaurus with respect to recall and generally improves on its precision as well. Again, if one wants access to informal design documentation it appears to pay to align queries with the less formal language found there. The largest average gap between the two thesauri comes when only subject + component or subject + function terms are added, showing a 50% difference in recall with a 10% improvement in precision.

4.1 Study 2: automatically generated thesauri—methods for creating thesauri

In the previous study, we examined which type of information source, informal or formal, was better for

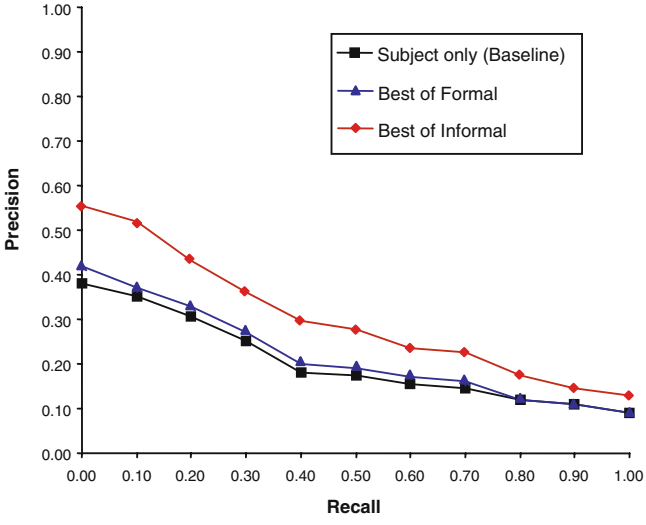

Fig. 9 Direct comparison of formal and informal thesauri

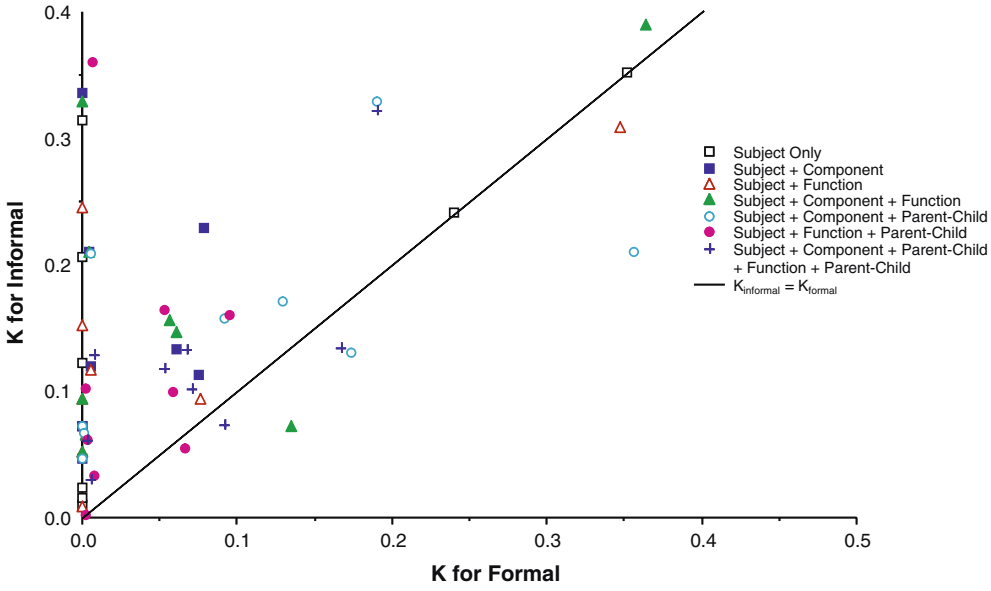

Fig. 10 Comparison of *K* values for formal and informal thesauri

design IR. We now turn our attention to other *methods* for generating thesauri which might reduce the extra overhead required to create a thesaurus. A less knowledge intensive approach is taken here for generating thesauri, not only to avoid the overhead and cost issues of hand generating thesauri, but also because it is the approach favored by IR systems currently in use.

Much work has been done to suggest that artifact models may be automatically extracted from text. Dong and Agogino [58] have explored ways of creating belief net models of important concepts in a text, although this system has not been used to create hierarchical device models. Riloff [61] and Kim and Moldovan [62] have developed dictionaries by first creating domain specific frames. Grefenstette [63] takes the middle ground between knowledge poor and knowledge rich approaches for extracting thesauri from text relating to several domains. Grefenstette's approach relies on syntactic analysis of text to determine term associations. It should also be noted that the goal of mode extraction is *not* to create an actual thesaurus such as Roget's thesaurus [64]. Grefenstette has shown that comparisons between domain-specific thesauri and general-purpose thesauri like Roget's are ill advised. The point of the thesauri created here is not to automatically create Roget's, but to increase the performance of an IR system. The above methods are quite powerful, with interesting results. However, they all require a great deal of manually annotated data to "learn" patterns for term association.

4.1.1 *Extraction of term modes from text*

The general approach for thesaurus generation is by extracting groups of words from text that are often found in the same document. In this scheme, co-occurring sets of terms in a corpus are extracted from a

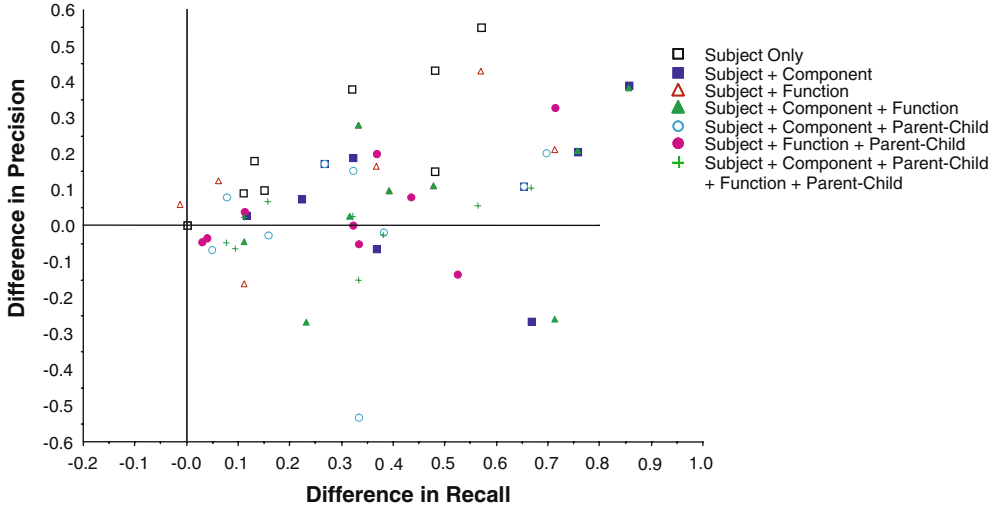

Fig. 11 Comparison of differences in precision and recall for informal and formal thesauri

document collection using a method called singular value decomposition (SVD). The SVD of the corpus matrix produces term ''modes'' that characterize a collection. These modes are used in an integrated approach to IR called LSI [18, 65]. However, in this work, SVD is utilized solely to create a thesaurus that will enhance the effectiveness of an existing IR system. The process that is used is relatively automatic: the SVD of each corpus yields a set of orthonormal ''modes'' whose peaks are extracted and used to populate a thesaurus database. In this capacity, it is hoped that these corpus themes can serve as an effective thesaurus.

Recall that the corpus matrix represents a collection of documents with term-document vectors. The SVD of the corpus matrix results in three matrices as shown in

$$C = USV^{\mathrm{T}} \tag{4}$$

where C is the $m \times n$ corpus matrix (rows are document vectors), U is an $m \times m$ orthonormal matrix that reconstructs C from the term modes, S is an $m \times n$ diagonal matrix of singular values, and V is an $n \times n$ orthonormal matrix of term modes in the corpus.

The term modes in V represent the partitioning of the corpus into a set of ''modes,'' ordered by the strength of the mode in the corpus given by its associated diagonal value from S. Singular values are closely related to eigenvalues [66]. In fact, if C is a symmetric matrix, then the SVD of C yields its eigenvalues. However, the usual form for a corpus matrix is rectangular, in which case, C can be factored as above. In the mechanical engineering domain, physical structures vibrate at a resonant frequency characterized by its eigenvalues. Similarly, a corpus matrix is characterized by its term modes. The use of SVD on a corpus matrix passes the modes through a filter to eliminate minor modes, thereby extracting more important themes from the documents.

4.1.2 *Granularity of documents*

The scheme outlined above extracts terms from whole documents in a collection. However, there may be some advantage to extracting themes from smaller document sections so that modes are likely to be contextually close. The determination of topic structure and term distribution within a document to improve information access has been addressed by Hearst [67]. In that work, topics were found by looking for boundaries between episodes of text by comparing their lexical similarity. We take a simpler approach to term distribution, assuming only that there may be some advantage to extracting themes from smaller document sections so that modes are likely to be contextually close. To better understand this, imagine that two words repeatedly appear together in separate, full-length documents. There is a reasonable probability that these words are related. However, if two words from a 50-word block of text appear together often, the likelihood of the two terms being related together increases. The notebook entries range from a few words long to 3,800 words, averaging 216 words each. Extensive work has been done in exploring the use of subdivided passages in IR [68–70]. For this reason, documents are divided into smaller ''blocks'' in which term co-occurrence is limited (Fig. 12). The same theme extraction techniques are then used over these smaller, more closely related blocks that are 50 words long. To better understand if proximity improves theme extraction, we further look at the effect of breaking documents down into even smaller blocks of 25, 10, and 5 words.

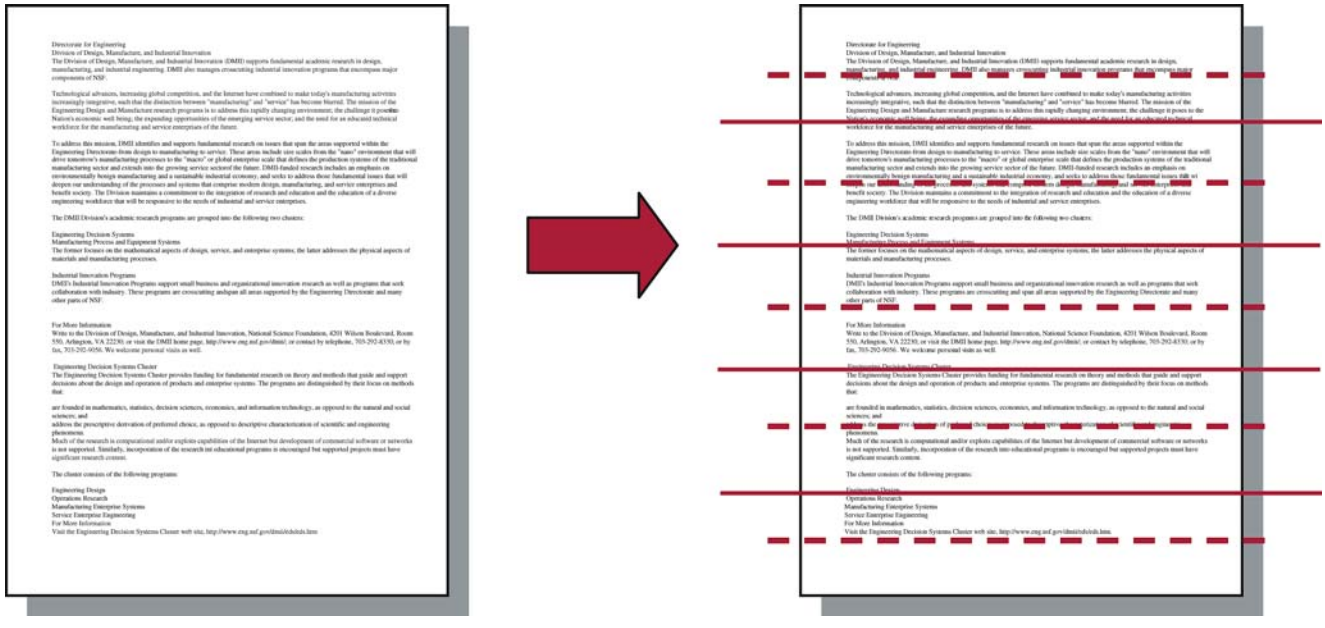

Fig. 12 Conversion of whole documents to fixed length, overlapping blocks

These blocks are further overlapped so as not to lose any text at the beginning or end of a block.

While these modes contain related terms and their variations, there is no ready mapping between the modes and Dedal subjects that can directly serve as a subject thesaurus. To institute such a mapping, the modes are indexed using another IR system, WAIS (wide area information system) [71], a vector-based system similar to SMART.[1] The original set of Dedal queries is applied to this WAIS index of term modes to identify themes in which both terms appear with the same numerical sign. The original Dedal *descriptor–subject* query is applied to a SMART index of the database to identify themes in which both terms appear with the same numerical sign.[2] These themes are ordered with respect to SMART similarity scores and divided into quartiles for selective augmentation of the original Dedal query.

Below is an example of one pair of many modes extracted from 50-word fixed length, overlapping blocks from one of the electronic notebooks about car bumpers. Modes can be positive or negative. Although both the corpus matrix and the affinity matrix derived from it have only positive entries, matrix modes contain both positive and negative terms. For this reason, modes contain information both about which terms should appear together but also that they should not appear along with terms of opposite sign.

The mode above differentiates between the shock absorbing bumper and testing of the legform impactor. In general, an effort was made to match the extracted concepts by looking for modes that matched the query on the positive side but did not match the mode on the negative side, and vice versa.

positive: shock, shock, pedestr, pedestrians, pedestrian, pedestrian's, pedestrians's, bump, bumpers, bumper, bumper's, activ, activities, actively,active, system, systems, system, system's, energ, energies, energy, absorb, absorbers, absorbs, absorption, absorbing, absorber, absorbed, absorb

negative: impact, impact, impactor, impacted, impacts, prototyp, prototype, prototyping, prototypes, legform, legform, test, tested, tester, testor, tests, test, testing

Which quartile or combination of quartiles is better for IR? In Fig. 13, a root of "Block" denotes that the thesaurus generated from the blocked corpus was used, "Whole" denotes that the whole document corpus was used. Appended to each of these is the lowest quartile of thesaurus terms added into the query. There are two definitive trends shown here: (1) the blocked thesaurus performs much better (~70%) than the whole document thesaurus demonstrating that contextual proximity improves theme extraction; and (2) that nearly all of the thesaurus terms retrieved by SMART are useful for IR. In fact, the first two quartiles produce inferior performance without the addition of the last two which produce nearly identical results. Effective thesaurus terms for IR come from throughout the mode generation process, not just from the main themes of the corpus. A possible explanation for this phenomenon may be that lower quartiles happen to be sufficiently relevant for inclusion in the thesaurus, and that a higher threshold for relevancy eliminates useful terms. Clearly, blocks are more effective than whole documents for thesauri extraction.

Figure 14 shows the effect of using different combinations of quartiles of relevance in the machine-generated thesauri for all block sizes. Average precision values are plotted across all block sizes. It was hypothesized that using only terms from the top quartiles (Q1 and Q12) might improve retrieval, but the graph shows a general pattern that using terms from all four quartiles (Q1234) produces significantly better precision over all values of recall. In fact, using only the top quartiles produces markedly poorer results. This result is consistent with the findings shown in Fig. 13 because it underlines that thesaurus terms from all quartiles are valuable for IR, and in this case, it is true for blocks of all sizes.

Now that it has been determined that using all quartiles is the best strategy, we next compare the results of testing the various block lengths on retrieval using all quartiles of the thesauri. The results in Fig. 15 show that 50 word long blocks performed the poorest, followed by 25 word long blocks. 5 and 10 word long blocks perform still better, but the two block lengths appear to have very similar precision and recall. In fact, along middle values of recall, 10 word blocks perform noticeably better than the smaller blocks.

This finding illustrates that term modes mode together in the same block are probably more contextually close, but only up to a certain block size. The performance of 5 and 10 word blocks is quite close, suggesting there is a limit to how small a block can be and still improve IR. One possible reason for this is the "zoom" effect. Imagine looking for New York City on a globe of the world. One approach is to focus the search, first on countries, then on states, and then counties. By using smaller block sizes, we hoped to zoom in on meaningful terms. However, when we used very small blocks, we found that we had zoomed in too closely, something akin to looking for New York City on a map of Central Park. Part of the reason for this could be that the natural coherent unit of documentation for these electronic design notebooks (the length of a sentence or paragraph, perhaps) is no shorter than 5 words, and no longer than 10 words. That is, a 5-word block may be insufficient to include two associated terms as may be found in a longer sentence or paragraph.

Figure 16 shows results of testing both 5 and 10 word blocks, along with results for other quartiles for the

[1]There was no performance advantage or disadvantage to using WAIS over SMART for indexing modes, except that existing code for the procedure was already set up for use with WAIS.

[2]Although both the corpus matrix and the affinity matrix derived from it have only positive entries, matrix modes contain both positive and negative terms. For this reason, modes contain information both about which terms should appear together but also that they should not appear along with terms of opposite sign.

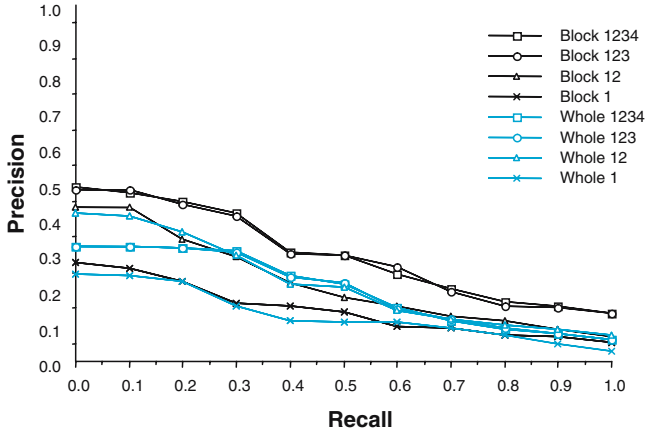

Fig. 13 Block indexing versus whole document indexing for generating thesauri

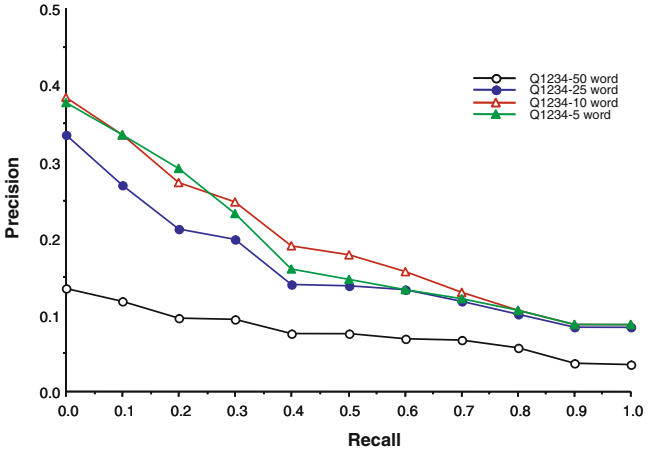

Fig. 15 Results of decreasing block size

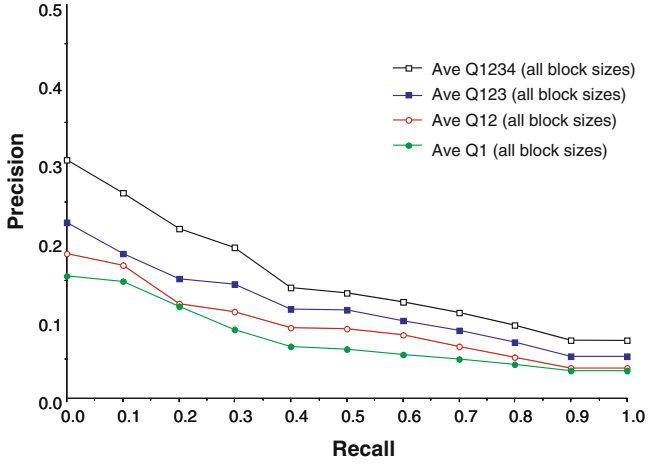

Fig. 14 Comparison of the effect of quartiles of relevance on performance

same blocks. With the exception of one curve, the graph shows that 10 word length bocks do indeed perform better than 5 word long blocks.

4.2 Comparison of manually derived and machine-extracted thesauri

We have now established that the actual informal design documentation is the best starting point for a thesaurus and that if a machine-generated thesaurus is to be mined from this corpus, the corpus should be broken into smaller blocks. Now the best performers of both manually built thesauri and machine generated are compared: thesauri from the informal information, and thesauri from the "blocked" data.

Figure 17 directly compares the two thesaurus methods. The conclusion here is that performance of the hand-created thesauri is superior to the machine generated ones, particularly at low values of recall (~34%) but less so at middle values of recall (~12%). In one sense, this is not surprising because taking advantage of domain specific knowledge would intuitively improve recall and precision over a knowledge poor approach such as simple term extraction. However, the clear trade-off between the two approaches to building thesauri is the effort and resources required to create them. Hand-constructed thesauri may provide better performance, but at the same time they require knowledge and time to create and maintain. In contrast, machine-generated thesauri are easily extracted and updated. And while machine-generated thesauri fare worse overall than hand-generated thesauri, Fig. 17 shows that the machine-generated thesauri still perform better (about 10–15%) than the baseline *subject only* in all but the lowest recall, highest precision setting, suggesting that this low overhead method is still better than no thesaurus at all.

5 Conclusions

This paper explores the use of a thesaurus to augment search for design information in design notebooks. The inherent challenge of design information is its evolving nature. We examine several methods of generating thesauri: hand building and machine generation. For both types of thesauri, we experimented to see what types of source material would be most effective for developing thesauri.

This section discusses the trade-offs among hand- and machine-generated thesaurus performance, generation effort, and maintenance issues. In this paper, the results of using thesauri drawn from the informal information found in design logbooks were compared with those from thesauri created from formal information found in final reports. It was believed that creating thesauri from formal material would be a more straightforward procedure than from informal material, and would make a hand generation of thesauri a more palatable option for

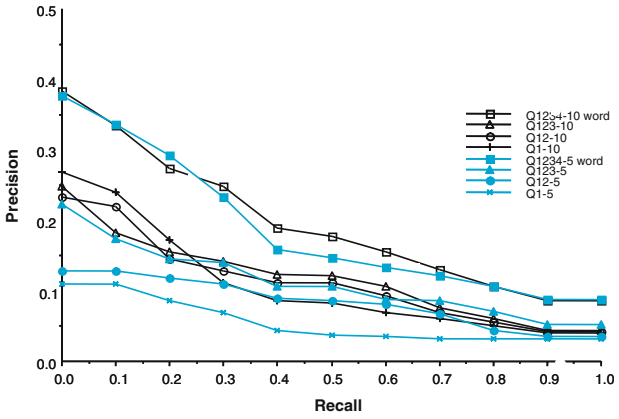

Fig. 16 Detailed comparison of 5 and 10 word block performance

IR. However, it was demonstrated that using informal information produced markedly (10–50%) better precision at the same values of recall than formal information. This result was not surprising, given the disparity between the types of languages in informal and formal design documentation, and that the material being searched is also informal information.

An alternative to hand building of thesauri is machine extraction of terms using the corpus matrix of a collection. We compared the use of fixed length blocks of documents, and whole documents as sources for thesauri. It was hypothesized that using blocked documents would produce better results because of the way blocking limits the proximity of terms. Experiments bore this out, showing blocking to be the better method.

Finally, the best of each type of thesauri, hand and machine generated, was compared directly. Machine thesauri require little human overhead to generate, making it an attractive alternative to hand built thesauri. However, testing demonstrated that manually created thesauri from informal information clearly produce better precision and recall than machine-generated thesauri from blocked material. These results are summarized in Table 2.

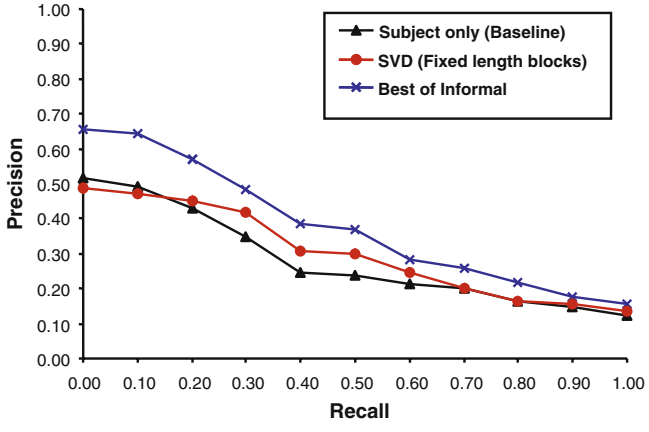

Fig. 17 Manually constructed thesauri compared with automatically extracted

Table 2 Comparison of thesauri generation methods and effectiveness for information retrieval

		Information retrieval performance	Ease of creation	Ease of maintenance
Manual thesauri	Informal	High	Low	Low
	Formal	Med	Med	Med
Machine thesauri	Singular value decomposition (SVD)	Low	High	High

These results illustrate a classic intelligence trade-off. That is, using richer domain knowledge that includes more context produces better results than using nondomain knowledge. In the work presented in this paper, domain knowledge is equated with increased human effort to generate thesauri.

The research questions posed at the beginning of this paper are now addressed below:

- *How does the evolution of design information affect its retrieval?* Traditional methods for indexing and retrieval are based primarily on term frequency, and do not take into account meaning or the change in terminology over time that occurs in design.
- *Can thesauri be a viable approach to improving retrieval?* Work in this paper shows, however, that one effective way of handling the changes in design language is to use a thesaurus of design terms to account for these changes in language.
- *How useful is informal information compared to formal information as a source of thesauri?* Informal information is more effective at IR than formal, final report information as a source for thesauri. However, there is a cost for manually constructing thesauri from informal information in terms of the human overhead required to build and maintain them.
- *How well do machine-generated thesauri compare to hand-generated thesauri for retrieving information?* Machine-generated thesauri are a potentially attractive alternative to hand-constructed thesauri because of the savings in human overhead and ability to maintain an up-to-date index. Despite these advantages, however, machine-generated thesauri do not perform as well at IR as hand built thesauri.

6 Future work

Future work in thesauri for design IR will concentrate on filling in the gap between manually built and machine-generated thesauri. We will examine other approaches to machine generation of thesauri, using different schemes for creating the modal matrix, such as covariance matrices. Future work will involve analyzing the modes that are extracted from documentation over the life of a project in order to better

understand the evolution of subject models over time. For hand-generated thesauri, we will test the utility of using machine techniques for creating the Dedal descriptor thesaurus. It is felt that the descriptor thesaurus is a generic and stable set of terms, ideal for machine approaches. Finally, in an effort to reduce the human effort required to produce manual thesauri, we will continue to look for more structured methods for creating these thesauri.

In a larger context, this work has clear implications for the development of collaborative tools for design teams. Studies of concurrent design teams, particularly those that are dispersed [41], with access to electronic communication tools suggest they will take advantage of good tools, and generate increasing amounts of electronic design information. In the context of concurrent engineering, it is believed that informal design IR can expedite the design process by providing designers with efficient, effective access to information needed for design.

Information access is a particularly relevant issue in engineering because of the changing practice of design and manufacturing. Global design teams require a shared understanding of a design among stakeholders throughout the development cycle.

One of the trademarks of dispersed multidisciplinary teams is the disparate vocabulary among disciplines. As described earlier, one method to translating vocabularies between terms involves the development of prespecified ontologies [27, 28]. The IR and thesaurus construction techniques presented in this paper may be a less knowledge intensive approach for finding domain specific synonyms for product design and development, thereby facilitating the sharing of information among design teams. The possibility of improving collaboration among design teams certainly warrants further exploration into design IR.

Acknowledgements Portions of this work were performed under a DARPA DSO-sponsored project, administered with the assistance of ONR, under the RaDEO program and is partially funded by Navy contract N00014096-1-0679. Their support is gratefully acknowledged.

References

1. Bucciarelli LL (1994) Designing engineers. MIT Press, Cambridge
2. Petroski H (1996) Invention by design: how engineers get from thought to thing. Harvard University Press, Cambridge
3. Hales C (1987) Analysis of the engineering design process in an industrial context. Ph.D. thesis, Department of Engineering, University of Cambridge
4. Court AW, Culley SJ, McMahon CA (1993) The information requirements of engineering designers. In: Proceedings of the international conference on engineering design
5. Winner RI, Pennell JP, Bertrand HE, och Slusarczuk MMG (1988) The role of concurrent engineering in weapon systems acquisition. Institute for Defense Analysis (IDA)
6. Simpson TW, Rosen D, Allen JK, Mistree F (1998) Metrics for assessing design freedom and information certainty in the early stages of design. ASME J Mech Des 120:628–635
7. Court AW, Culley SJ, McMahon CA (1996) Information access diagrams: a technique for analyzing the usage of design information. J Eng Des 7 (1):55–75
8. Baya V, Gevins J, Baudin C, Mabogunje A, Toye G, Leifer L (1992) An experimental study of design information reuse. In: Proceedings of the international conference on design theory and methodology
9. Wood WH, Agogino AM (1996) A case-based conceptual design information server. Comput Aided Des 28(5):361–369
10. Duffy AHB, Duffy SM (1996) Learning for design reuse. AI EDAM—Artif Intell Eng Des Anal Manufact 10(2):139–142
11. Szykman S, Sriram RD, Regli WC (2001) The role of knowledge in next-generation product development systems. J Comput Inform Sci Eng 1(1):3–11
12. Hirst G (1981) Anaphora in natural language understanding: a survey. Springer, Berlin Heidelberg New York
13. Baudin C, Gevins J, Baya V, Mabogunje A (1991) Dedal: using domain concepts to index engineering design information. In: Proceedings of the ninth national conference on artificial intelligence
14. Trigg RH, Blomberg J, Suchman L (1999) Moving document collections online: the evolution of a shared repository. In: Proceedings of the European conference on computer-supported cooperative work ECSCW'99
15. Song S, Dong A, Agogino A (2002) Modeling information needs in engineering databases using tacit knowledge. J Comput Inform Sci Eng 2(3):199–207
16. Wood WH, Yang MC, Cutkosky MR, Agogino A (1998) Design information retrieval: improving access to the informal side of design. In: Proceedings of the 1998 design engineering technical conferences 10th international conference on design theory and methodology
17. Salton G, McGill M (1983) Introduction to modern information retrieval. McGraw-Hill, New York
18. Deerwester S, Dumais ST, Furnas GW, Landauer TK, Harshman R (1990) Indexing by latent semantic analysis. J Am Soc Inform Sci 41:391–407
19. Yang M, Cutkosky M (1999) Machine generation of thesauri: adapting to evolving vocabularies in design documentation. In: Proceedings of the international conference on engineering design ICED99
20. Yang M, Cutkosky M (1997) Automated indexing of design concepts for information management. In: Proceedings of the eleventh annual international conference on engineering design (ICED97)
21. Kunz W, Rittel HWJ (1970) Issues as elements of information systems. Universitt Stuttgart, Institut far Grundlagen der Planting
22. Ullman DG (1994) Issues critical to the development of design history, design rationale, and design intent systems. In: Proceedings of the 1994 ASME design technical conferences
23. Qureshi SM, Shah JJ, Urban S, Harter E, Bluhm T (1997) Integration model to support archival of design history in databases. In: Proceedings of the ASME design engineering technical conferences
24. Regli WC, Hu X, Atwood M, Sun W (2000) A survey of design rationale systems: approaches, representation, capture and retrieval. Eng Comput 16:209–235
25. Cutkosky MR, Engelmore RS, Fikes RE, Genesereth MR, Gruber TR, Mark WS, Tenenbaum JM, Weber JC (1993) PACT: an experiment in integrating concurrent engineering systems. IEEE Comput 26(1):28–37
26. Yoshioka M, Sekiya T, Tomiyama T (1998) Design knowledge collection by modelling. In: Proceedings of the tenth international IFIP WG 5.2/5.3 conference PROLAMAT 98
27. Gruber T (1993) A translation approach to portable ontology specifications. Knowl Acquis 5(2):199–220
28. Gruninger M, Fox MS (1995) The design and evaluation of ontologies for enterprise engineering. In: Proceedings of the IJCAI-95: workshop on basic ontological issues in knowledge sharing

29. Dong A, Agogino AM (1998) Managing design information in enterprise-wide CAD using 'smart drawings'. Comput Aided Des 30(6):425–435
30. Fruchter R, Reiner K, Leifer L, Toye G (1996) Vision-Manager: a computer environment for design evolution capture. In: Proceedings of the artificial intelligence in design '96
31. Shah J, Blizankov P, Urban S (1993) Development of a machine understandable design process representation language. In: Proceedings of the ASME design theory conference
32. Jamalabad VR, Langrana NA (1993) Extraction and reuse of design information. In: Proceedings of the 5th international conference on design theory and methodology
33. Anthony L, Regli WC, John JE, Lombeyda SV (2001) An approach to capturing structure, behavior, and function of artifacts in computer-aided design. J Comput Inform Sci Eng 1(2):186–192
34. Tyhurst JJ (1986) Applying linguistic knowledge to engineering notes. In: Proceedings of the winter annual meeting of the American Society of Mechanical Engineers
35. Boujut J-F (2003) User-defined annotations: artefacts for coordination and shared understanding in design teams. J Eng Des 14(4):409–419
36. Jacobsen K, Sigurjonsson J, Jakobsen O (1991) Formalized specification of function requirements. Des Stud 12(4)
37. Bucciarelli LL (1988) An ethnographic perspective on engineering design. Des Stud 8: 159–168
38. Tang JC (1989) Listing, drawing, and gesturing in design: a study of the use of shared workspaces by design teams. Doctoral Thesis, Department of Mechanical Engineering, Stanford University
39. Minneman SL (1991) The social construction of a technical reality: empirical studies of group engineering design practice. Ph.D. Thesis, Department of Mechanical Engineering, Stanford University
40. Culley SJ, Boston OP, McMahon CA (1999) Suppliers in new product development: their information and integration. J Eng Des 10(1)
41. Liang T, Cannon DM, Leifer LL (1998) Augmenting the effectiveness of a design capture and reuse system based on direct observations of usage. In: Proceedings of the international conference on design theory and methodology
42. Liang T, Leifer LJ (2000) Re-use or re-invent? Understanding and supporting learning from experience of peers in a product development community. In: Proceedings of the 30th ASEE/IEEE frontiers in education conference
43. Cross N, Christiaans H, Dorst K (eds) (1996) Analysing design activity. Wiley, New York
44. Tomiyama T (1995) Design process model that unifies general design theory and empirical findings. In: Proceedings of the 1995 ASME design engineering technical conference
45. Ullman DG, Dietterich TG, Stauffer LA (1988) A model of the mechanical design process based on empirical data. AI EDAM—Artif Intell Eng Des Anal Manuf 2(1):33–52
46. Kuffner TA, Ullman DG (1990) Information requests of mechanical design engineers. In: Proceedings of the international design engineering technical conferences
47. Baudin C, Underwood JG, Baya V (1993) Using device models to facilitate the retrieval of multimedia design information. In: Proceedings of the thirteenth international joint conference on artificial intelligence
48. Kolodner J (1993) Case-based reasoning. Morgan Kaufmann, San Francisco
49. Ruecker LSF, Seering WP (1996) Capturing design rationale in engineering contexts. In: Proceedings of the 1996 ASME design engineering technical conferences and computers in engineering conference
50. Yen SJ, Fruchter R, Leifer L (1999) Facilitating tacit knowledge capture and reuse in conceptual design activities. In: Proceedings of the ASME design engineering technical conferences, 11th international conference on design theory and methodology
51. Sobek DK (2002) Preliminary findings from coding student design journals. In: Proceedings of the 2002 American Society for Engineering Education annual conference and exposition
52. Winograd T, Flores F (1987) Understanding computers and cognition: a new foundation for design. Addison-Wesley, Reading
53. Grudin J (1994) Groupware and social dynamics: eight challenges for developers. Commun ACM 37(1):92–105
54. Viste MJ, Cannon DM (1995) Firmware design capture. In: Proceedings of the ASME design theory and methodology conference
55. Baya V, Leifer LJ (1995) Understanding design information handling behavior using time and information measure. In: Proceedings of the seventh international conference on design theory and methodology
56. Baeza-Yates R, Ribeiro-Neto B (1999) Modern information retrieval. Addison-Wesley, Reading
57. Page L, Brin S (1998) PageRank, an eigenvector based ranking approach for hypertext. In: Proceedings of the 21st annual ACM/SIGIR international conference on research and development in information retrieval
58. Dong A, Agogino AM (1996) Text analysis for constructing design representations. In: Proceedings of the artificial intelligence in Design '96
59. Dong A, Hill AW, Agogino AM (2004) A document analysis method for characterizing team-based design outcomes. J Mech Des 126(3):378–385
60. Hong J, Toye G, Leifer L (1994) Using the WWW for a team-based engineering design class. In: Proceedings of the second WWW conference
61. Riloff E (1996) An empirical study of automated dictionary construction for information extraction in three domains. Artif Intell 85(1–2):101–134
62. Kim J, Moldovan D (1993) Acquisition of semantic patterns for information extraction from corpora. In: Proceedings of the ninth IEEE conference on artificial intelligence for applications
63. Grefenstette G (1992) Use of syntactic context to produce term association lists for text retrieval. In: Proceedings of the fifteenth annual international ACM SIGIR conference on research and development in information retrieval
64. Kipfer BAE (2001) Roget international thesaurus indexed edition. HarperCollins, New York
65. Schutze H, Silverstein C (1997) Projections for efficient document clustering. In: Proceedings of the SIGIR 1997
66. Strang G (1988) Linear algebra and its applications. Harcourt Brace Jovanovich College Publishers, New York
67. Hearst M (1994) Context and structure in automated full-text information access. Ph.D., Computer Science, UC Berkeley
68. Hearst MA, Plaunt C (1993) Subtopic structuring for full-length document access. In: Proceedings of the 16th annual international ACM SIGIR conference
69. Stanfill C, Waltz D (1992) Statistical methods, artificial intelligence, and information retrieval. In: Text based intelligent systems: current research and practice in information extraction and retrieval, Lawrence Erlbaum Associates, Mahwah, pp 215–226
70. Moffat A, Sacks-Davis R, Wilkinson R, Zobel J (1994) Retrieval of partial documents. In: Proceedings of the second text retrieval conference (TREC-2)
71. Kahle B (1991) An information system for corporate users: wide area information servers. Thinking Machines, Cambridge