Combining CBR with Interactive Knowledge Acquisition, Manipulation and Reuse*

David B. Leake and David C. Wilson

Computer Science Department
Indiana University, Lindley Hall
150 S. Woodlawn Ave
Bloomington, IN 47405, U.S.A.
{leake,davwils}@cs.indiana.edu

Abstract. Because of the complexity of aerospace design, intelligent systems to support and amplify the abilities of aerospace designers have the potential for profound impact on the speed and reliability of design generation. This article describes a framework for supporting the interactive capture of design cases and their application to new problems, illustrating the approach with a discussion of its use in a support system for aircraft design. The project integrates case-based reasoning with interactive tools for capturing expert design knowledge through "concept mapping." Concept mapping tools provide crucial functions for interactively generating and examining design cases and navigating their hierarchical structure, while CBR techniques provide capabilities to facilitate retrieval and to aid interactive adaptation of designs. The project aims simultaneously to develop a useful design aid and more generally to develop practical interactive approaches to fundamental issues of case acquisition and representation, context-sensitive retrieval, and case adaptation.

1 Overview

Aerospace design is a complex process that requires designers to address complicated issues involving numerous specialized areas of expertise. No single designer can be an expert in every relevant area, and becoming proficient may require years of experience. Consequently, intelligent systems to support and amplify the abilities of human designers have the potential for profound impact on the speed and reliability of design generation. An appealing approach, which has been applied in systems such as (Domeshek *et al.*, 1994), is to augment the designers' own design experiences with relevant information from prior designs: to provide support with case-based reasoning.

* This research is supported in part by NASA under award No NCC 2-1035. The authors gratefully acknowledge support from Northwestern University while on leave and many contributions by Alberto Cañas, Mary Livingston, and James Newkirk.

Ideally, case-based design support tools will include three related capabilities to aid design reuse: capture of and access to specific design experiences, support for new designers as they try to understand the lessons of those prior experiences; and support for adapting prior designs to fit new design goals. For practical application, the tools must not depend on extensive domain knowledge; for designer acceptance, they must leave the designer in control. This article describes principles for addressing these goals and their application in the case-based design aid DRAMA (Design Retrieval and Adaptation Mechanisms for Aerospace).

The DRAMA project integrates case-based reasoning with interactive tools for capturing expert design knowledge through *concept mapping* (Novak and Gowin, 1984), with the goal of leveraging off the strengths of both approaches. We are applying concept mapping tools from the Concept Mapping group at the University of West Florida, led by Dr. Alberto Cañas, to provide an interactive interface and crucial functions for generating and examining design cases, as well as navigating their hierarchical structure. CBR techniques provide the capabilities to facilitate retrieval and to aid interactive adaptation of designs. The implemented DRAMA system supports browsing of prior design knowledge and proactively provides designers with concrete examples of designs and design adaptations from similar prior problems. At the same time, it unobtrusively acquires new examples from the user's interactive design process.

The project develops "knowledge-light" (Wilke *et al.*, 1997) interactive approaches to addressing fundamental CBR issues of case acquisition, case adaptation, and context-sensitive retrieval. The system demonstrates that fully integrating a CBR system into the design environment enables the system to dynamically adjust the relevance criteria used to retrieve prior experiences, exploiting task-based information without requiring the user to provide it explicitly. In addition, the system illustrates the benefits of interactively capturing and manipulating cases at a "middle level" between traditional highly structured cases with fixed representations, and unstructured textual cases.

The system differs from previous approaches in allowing multiple case representations that users themselves can develop and revise. In interactive CBR systems, a user's ability to understand and apply a prior case may depend not only on its content, but on how its representation matches the user's conceptualization of the domain: A seemingly more distant case may be more useful to the user if it is more understandable. This raises interesting research questions about supporting user-defined representations and reconciling the divergent benefits of flexibility, customization, and case standardization as the case library grows.

2 The Task Domain

A significant concern at NASA is "knowledge loss:" that critical aerospace design expertise is the domain of a few experts and will be lost when they retire. This has given rise to knowledge preservation efforts, a number of which have employed CBR. For example, the RECALL tool at the NASA Goddard Space Flight Center was developed to store and access textual reports of important lessons (Bagg, 1997). However, different experts may conceptualize designs very differently, making it hard for others to interpret descriptions of prior designs.

To make records more comprehensible, new projects have investigated the use of concept mapping. The goal of concept mapping for design is to capture not only important features of the designs themselves, but also the designers' conceptualizations of those designs—the relationships and rationale underlying their components. This raises the question of how to organize and access the knowledge that concept maps capture, and how to facilitate its reuse. Our framework uses interactive CBR techniques to support retrieval and reuse of designs represented as concept maps.

3 Tenets of the Approach

Our tenets shaping the DRAMA framework are:

- **The system should leverage a designer's knowledge, rather than attempting to replace it.**
  This requires interactivity and support rather than autonomous design generation. All parts of the process must accept user control.
- **The system should support multiple conceptualizations of the design space.**
  The system must both allow multiple (potentially idiosyncratic) representations and support standardization when that does not impose a burden.
- **Support information should automatically be focused on the current task.**
  This requires that the system monitor the task context in order to anticipate information needs and to determine how to fulfill them.
- **Learning must play a central role, both at the design level and at the level of design manipulation.**
  This requires the capability to capture and reuse cases both for designs and design processes.

All the examples in this paper focus on cases containing designs, but the framework could be applied to representations of processes as well. Core system methods provide a domain-independent framework for interactive capture, graphical manipulation, and experience-supported reuse of design knowledge.

4 Background

Case-based Design Support: Case-based reasoning is widely used in design-aiding systems. The Clavier system (Hinkle and Toomey, 1995), for example, is a case-based advisory system put into production use to suggest and critique designs of autoclave layouts at Lockheed. Research systems address support for tasks such as architectural design (Goel *et al.*, 1991; Hua and Faltings, 1993; Gebhardt *et al.*, 1997; de Silva Garza and Maher, 1996; Smith *et al.*, 1995), circuit design (Vollrath, 1998), and conceptual design of aircraft subsystems (Domeshek *et al.*, 1994). Many of these systems display impressive capabilities, but at the expense of considerable development effort to tailor them to domain-specific needs. We instead provide a framework for building up case knowledge and indexing criteria. Specific case representations, rather than being predefined, are developed incrementally through interactions with users as the system is applied. Users can easily augment and adjust the case representation as needed, with simple analogical mapping processes allowing disparate types of cases to be retrieved.

Concept Mapping: Concept mapping is a process to reveal an individual's internal cognitive structures by developing external representations of concepts and propositions. A concept map (CMap) is a two-dimensional representation of a set of concepts constructed so that the interrelationships among them are evident. Individual concepts are linked to related concepts through one or two-way links, each link associated with a label/proposition describing the relationship. The vertical axis generally expresses a hierarchical framework for the concepts; for example, a concept map of design problems might represent a hierarchy of abstract and more specific problems. However, we stress that there is no requirement that they represent particular relationships; they are compatible with *any* structured representation.

Semantic networks are a form of concept map, but concept maps are not constrained by syntactic rules and have no associated semantics; they are normally seen as a medium for informally "sketching out" conceptual structures. The visual presentation of information in concept maps provides a natural starting point for organizing and accessing information in multiple forms (e.g., images or video clips), which also are contained in the CMap. For example, Figure 1 shows a sample CMap describing the basic structure of the Boeing 777 aircraft, annotated with an associated image and diagram. This CMap is displayed by the CMap tools described in a later section.

Concept mapping has been used in educational contexts to help students clarify and compare their understanding. A recent effort integrated concept mapping into a set of knowledge construction and sharing tools linking over a thousand schools in Latin America (Cañas *et al.*, 1995). It is currently being used to capture a NASA Mars expert's knowledge in CMaps organizing multimedia resources, to be made available to the public on the World Wide Web, and for knowledge construction and sharing among astrobiologists. The application of concept mapping to design is intended both to help an expert clarify his or her own conceptualizations and to make those conceptualizations available for examination by the expert or others (e.g., members of a design team seeking to understand the expert's design to evaluate or modify it, or novices seeking to increase their own understanding). Through differences in maps that different designers generate for the same concepts (whether in the features and relationships they include, or in the level of granularity they use), concept mapping can illuminate their different perspectives. For example, a designer specializing in airflow might include features such as wing or surface shapes and operational constraints that dictate them (e.g., the need for short-field landings), while an avionics designer would focus on aspects such as aircraft control systems.

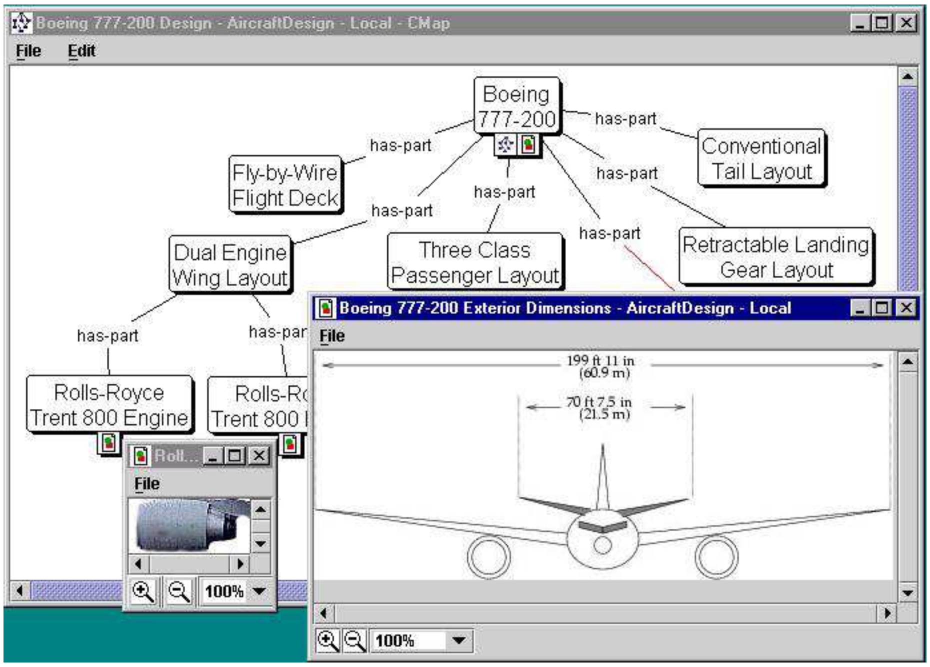

Fig. 1. Sample screen images from the CMap Editor.

Manual procedures have been developed to aid the initial generation of CMaps (e.g., Jonassen *et al.*, 1993, pp. 138–139; Novak and Gowin, 1984, pp. 24–36), and computerized tools have been developed to facilitate this process and to capture its results. The CMap tools, developed at the Institute for Human and Machine Cognition of the University of West Florida, support interactive definition and arrangement of initial maps, and manual browsing through concept map-based multimedia environments and case libraries.[1] The system also allows concept maps to be defined hierarchically, so that the nodes of any map can be associated with complete maps describing them at a finer-grained level.

Motivations for Integrating CMaps and CBR: The integration of CBR methods with interactive CMap tools provides benefits for both. Existing CMap tools provide an interactive medium for representing and examining designs, but their framework does not provide facilities for retrieval of relevant CMaps.

[1] The CMap tools are publicly available from http://cmap.coginst.uwf.edu/.

Likewise, although the tools provide capabilities for interactively defining new CMaps and manipulating their structure by adding, deleting, or substituting components, they provide no support for the decision-making required by that adaptation process. Consequently, their usefulness can be extended by the addition of automatic aids for retrieving relevant CMaps, for navigating CMaps and locating relevant information, and for reusing prior CMaps.

Conversely, case-based reasoning can leverage off the interactive case definition and revision capabilities of the CMap tools. The CMap tools provide a convenient method for entering case information in an intermediate form between textual descriptions (which are easy to generate but hard for systems to reason about) and rich structured representations (which are hard to generate but support complex reasoning). In our domain, the push to use concept mapping to understand the design process means that CMap cases will be available at low cost as "seed cases" for the CBR system. In addition, the CMap tools already provide crucial functions for interactively generating and examining these cases and navigating their hierarchical structure.

5 The DRAMA System

In the DRAMA system, concept maps are used to organize acquired aerospace design cases in a form that can be browsed by other designers in order to leverage their own expertise by profiting from stored prior experiences. The system uses concept mapping tools as a method for initial capture, manual browsing, and manual modification of design cases represented as concept maps. It uses interactive CBR techniques to retrieve relevant prior cases and to retrieve alternatives to support adaptation. In addition, it uses CBR to manage and present cases that record the rationale for particular decisions and cases that suggest adaptations of designs. The following sections discuss the main features of the system.

5.1 Using CMaps to organize and represent design information

In DRAMA, CMaps represent two types of information. First, they represent user-definable/modifiable hierarchies of aircraft and part types. This information is used to organize specific design cases and to guide similarity assessment during case retrieval. Such organization provides the designer with browsable hierarchies of aircraft (e.g. dividing military and commercial aircraft), aircraft components (e.g. specific wings, engines, fuel tanks), and component configurations (e.g. fuel tanks inside or outside the aircraft) for reference during the design process.

Second, CMaps represent specific information about particular designs such as their components and component relationships. Each component is represented as a CMap, enabling interactive viewing and manipulation of hierarchical designs at different levels of granularity.

5.2 How the system supports design

To illustrate the design process, the following sections present a simple example involving the coarse-grained configuration of an airliner after an initial set of "seed case" designs has been provided to the system, along with hierarchies of aircraft types organizing those designs. The steps described include retrieval of a similar prior design as a starting point, retrieval support for adaptation and refinement of system suggestions, and the capture of a new adaptation for future use. The concept maps used in the following figures are simple examples; those used by expert designers would include finer-grained technical details at lower levels of the hierarchy. NASA domain experts are currently developing richer concept maps to explore the framework as applied to a design initiative for reusable spacecraft.

Retrieving a relevant prior design: The case-based design process begins by selecting a similar example as a starting point. In addition, or if no sufficiently similar prior example exists, the designer is free at any point to develop designs from scratch and add them to the CMap library for future use.

The designer may choose either of two interfaces for the initial search process, one non-interactive and the other interactive. The first (non-interactive) option, the "Design Finder," is a simple and traditional CBR retrieval interface. The interface presents selection boxes for choosing the desired features of a design from a pre-defined set of standard attribute types (e.g., aircraft type, manufacturer, model number, etc.). Currently the system uses a standard predefined feature set, but features could also be derived automatically from the set of designs. Given the list of features, the system performs nearest-neighbor retrieval, according to a predefined feature weighting scheme, to retrieve references to potentially-relevant CMaps. These are presented to the designer along with a match score. The designer can browse and select from the alternatives to bring up the CMap for a particular design.

The second interface allows the designer to interactively navigate the hierarchy of concept maps, exploring alternative "views" of aircraft and aircraft component types. In our sample scenario, the designer is considering alternatives for increasing the fuel efficiency of a large airliner. The first step is to establish a context for the design by locating the CMap node for an aircraft similar to the one envisioned; the designer then chooses to consider possible engine types. The designer could also simply navigate to and browse specific engines, but in that case less contextual information would be available to aid in adaptations.

The designer first navigates through the types of aircraft to select an aircraft, and pulls up the top-level concept map for its design. The designer then selects (by clicking on the concept map) the particular part to adapt. In this example, the selection is the engine. If no CMap is already present for the component selected (e.g., the designer wishes to fill in a sketchy design by specifying its engine), the designer can use the interactive CMap tools to create a new CMap from scratch, or can browse the CMaps for designs, import a design, and then adapt as desired. If a CMap is already present for the part and it has been defined at a sufficient level of detail, the designer may also decompose the part representation into its component CMaps and make the revisions in the sub-components (with CBR support). Alternatively, the designer may define new component substructures, making the representation more detailed.

When the previous case has been retrieved, the designer has four choices, as shown in Figure 2: to *adapt* it (changing the representation in memory, e.g., when continuing work on a design begun in a previous session); to *derive* a new design, by having the system make a copy to adapt; to ask the system to use its hierarchy of aircraft parts to form an abstraction of the current design's structure as a template to fill in; or to ignore the proposed design and begin a new design from scratch.

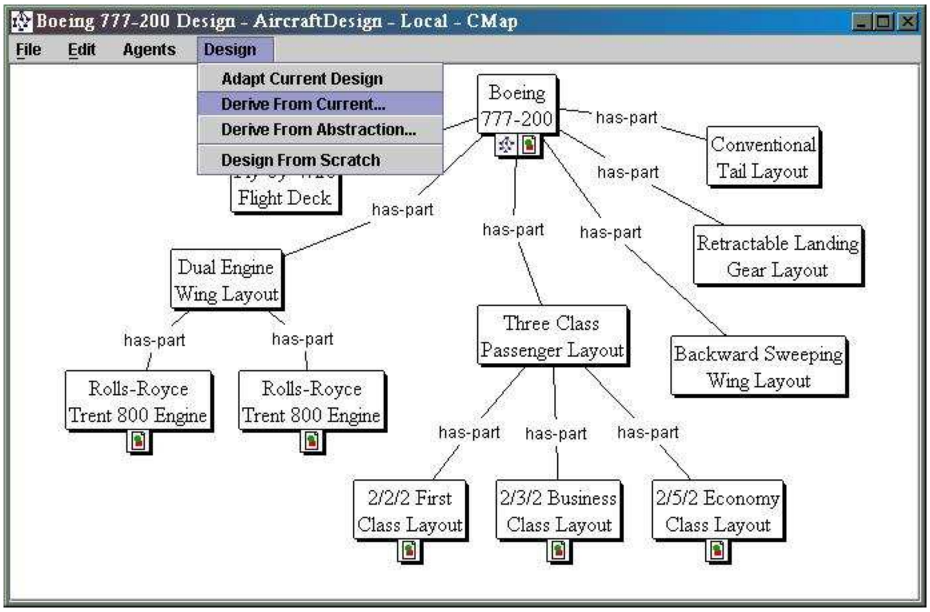

Fig. 2. Beginning derivation of a new design from a prior case.

Adapting designs: Once the designer has navigated, for example, to the engine of a particular aircraft, the system supports three ways of examining why the engine was used and the alternatives that may exist. First, the designer may simply interactively browse stored information, following links in the CMap to examine associated information such as finer-grained concept maps, video clips of explanations from previous designers, photographs, or specifications for the engine. Second, the designer may request information about similar designs. The designer may request to have this retrieval targeted to either:

– Focus on designs with components similar to the one that is currently of interest (e.g., CMaps that show aircraft using similar engines)
– Focus on designs that provide similar contexts for the current type of component (e.g., CMaps that show the engines of similar aircraft)

The algorithms underlying this retrieval are described in Section 6.3.

Retrieved alternatives are listed in order of goodness of match according to the chosen focus. The designer may also enter additional criteria to be matched against any textual annotations of rationale recorded by previous designers. For example, the designer may request that fuel-efficient engines be weighted more heavily. This prompts a re-sorting of options, using simple text matching techniques from information retrieval to decide which prior rationale to consider most relevant.

Suggesting prior adaptations: When the designer selects a component of an aircraft to adapt, the system has access to the following information: the component affected, any designer input of additional retrieval criteria, and the design itself. This information is used to index into stored records of prior adaptations to suggest adaptations that have been previously performed in similar contexts to address similar issues. Note that this adaptation process does not assume knowledge of complex constraints. DRAMA's method reduces the amount of knowledge that must be encoded, requiring the designer to evaluate the possibilities suggested.

Performing adaptations: The designer may select any of the suggested engines to browse further or to substitute for the engine in the design. The designer may also simply delete or add a component to the representation using the CMap tools. Adaptations of concept maps can be thought of as falling into three general categories corresponding to the support that they require: additions, deletions, and substitutions. DRAMA's framework supports the designer's performance of these operations as follows:

– Additions: The designer may use the hierarchical browser or plain-text retrieval capability to retrieve potentially-relevant components to be linked into the design.
– Deletions: The system can warn of potential deletion issues by proactively retrieving similar deletions, checking them for problems, and presenting those problems to the designer.
– Substitutions: The system can support substitution by retrieving and suggesting candidate substitutions, using both the explicitly-stated criteria and contextual information from the current map to guide the retrieval. It retrieves these from two sources: From stored adaptation cases encapsulating prior substitutions, and from analogous nodes in similar designs.

When the designer states a goal and finds a suitable substitution, the system learns adaptation cases, following research on case-based adaptation learning (Leake *et al.*, 1997; Sycara, 1988). These package the query, information about the CMap that was used as context for the search, and the selected result.

Storing rationale and design cases: After the designer performs a substitution, the designer is prompted to enter an optional textual annotation of why the new alternative is preferable to the old. This question focuses rationale capture: The designer does not record a rationale for the component as a whole (which could involve countless factors), but simply for why it is the *better* component in the current context. Focusing the explanation process in this way is related to the common idea in CBR of aiming explanations at expectation failures (Hammond, 1989; Leake, 1992; Schank, 1982). During future adaptations, this rationale will be provided with other information about the component, and it can also be used as an additional index when retrieving possible substitutions. Adapted cases are placed into the system's hierarchies of cases at the point where the designer found the most similar previous case.

This approach to rationale capture differs strongly from traditional rule-based or model-based approaches. The information in CMaps and additional learned features corresponds to the "weak explanations" advocated by Gruber and Russell (1992), providing just enough information to guide a designer's own reasoning process towards inferring important aspects of the design.

6 Perspective on issues and methods

DRAMA's approach is relevant to a number of fundamental issues for developing practical case-based applications. This section highlights its contributions on addressing these issues.

6.1 Interactive case acquisition

Experience deploying CBR has shown that CBR may require significant "case engineering" effort (Aha and Breslow, 1997; Kitano and Shimazu, 1996; Mark *et al.*, 1996; Voß, 1994). Research CBR systems often use carefully-structured case representations, which enable powerful reasoning at a high knowledge acquisition cost (Kolodner, 1993). At the other end of the spectrum, current projects in *textual case-based reasoning* (Lenz and Ashley, 1998) address how to exploit case information already stored in textual form. For such systems, case acquisition cost is negligible, but exploiting case context is much more difficult.

CMaps provide a middle ground. CMap representations include structural information and are intended to concisely represent key concept properties, facilitating their use by AI systems. However, concept maps do not necessarily use any standard syntax or standard set of attributes. This places them at a middle point between classic structured case representations and purely textual cases. It makes them more difficult to manipulate autonomously within an AI system, but also makes them more flexible if experts use distinctions that were not anticipated, and "forgiving" when non-experts in AI are called upon to encode their knowledge. Domain experts who use the CMap tools seem to have few problems adapting to the concept mapping process.

6.2 Guiding the user towards useful representations

Although users of DRAMA are free to change existing representations or devise new representations if needed, the system uses two methods to help standardize representations. First, when a user draws a CMap and is about to fill in a new link or node, it presents the user with menu of alternatives from previous maps. If one of these is suitable, the user may select it. This builds up a set of standard link types and concept types over time. The second is that the baseline process for generating new design CMaps is modification of previous designs. The system is intended to begin with a set of CMaps that reflect the conceptualizations of a particular expert designer, reflecting that designer's coherent view of the factors important in a design. When new designs are generated by adaptation, significant portions of old representations are reused for new tasks, resulting in representations with similar structure and content. The two approaches facilitate the case engineering task while guiding accumulated design knowledge towards a coherent representation scheme that includes structural information. We intend to perform empirical tests to determine the additional value of the CMap structure, compared to, for example, applying pure information retrieval techniques on the concept map's textual content alone.

6.3 Similarity assessment for semi-structured information

Retrieving candidate design components for making suggestions requires comparing the current concept map to those in memory. Concept maps afford both structural and content information. Link structures can be viewed with or without consideration of their labels (because not all corresponding labels are guaranteed to have been assigned the same names, requiring all names to match may be too strong a constraint). Their structural properties may be compared by, e.g., applying structure-mapping approaches from analogical reasoning (e.g., (Falkenhainer *et al.*, 1989)). The DRAMA system is beginning to address these issues by considering a simple model of structure and content in retrieval.

The current system retrieves candidate design components in a two-stage process: retrieving relevant designs (e.g., designs for similar aircraft) and choosing relevant concepts (e.g., the engines) from those designs. The second step is required because the corresponding roles of concept map designs may not provide direct indications of how the components should be mapped (e.g., whether a link designated "tail engine" in one concept map should correspond with one designated simply "engine" in another).

Given a user-selected component (e.g. a particular engine) to be adapted and the goal of finding other engines from similar designs, DRAMA first retrieves similar designs, using a matching procedure that compares map structure and content (based on the distance of corresponding concepts in hierarchical concept memory), when they are included in the set of concepts. Second, DRAMA chooses the closest matching concept from each of the retrieved maps. Because concept maps lack a rigid semantic structure, the concept is selected both by matching available role structure (an abstraction of the component in question, if available, represents a type of slot to be filled) and by distance in concept memory (where the closest-matching concepts are successively paired). The results are ranked by the inverse of each map's summed distance. This gives an indication of the relative goodness of each design suggestion within the overall pool of suggestions.

Once candidate concepts have been retrieved and displayed, the user can adjust the relative ranking by entering textual descriptions of desired properties. The system compares these with the properties annotating the candidates using simple IR methods. Suggestions for candidates that are supported by similar textual rationale are given added weight in the ranking.

6.4 Interactive indexing and retrieval

Ideally, the CBR retrieval process takes into account both high-level goals and concrete design features. Applied CBR systems tend to rely on the user to explicitly provide this information (whether all at once or incrementally). Conversational case-based reasoning (CCBR) systems guide the retrieval process through an interactive dialogue of questions (Aha and Breslow, 1997). However, because poor questions or question organization may prevent retrieval or slow identification of the right cases, a substantial case engineering effort may be required to craft the set of questions.

DRAMA's alternative approach is to attempt to integrate the CBR process tightly enough into the user's task process that it can infer a substantial part of the needed contextual features directly from monitoring the user's task. The system has access not only to the user's retrieval request (e.g., to find a substitute engine), but also to a significant part of the context surrounding the request that will determine the relevance of the retrieval (e.g., the aircraft for which the engine is needed). The designer may also augment this context with additional information (e.g., that the goal is to find a more fuel-efficient engine that could substitute), but is not required to do so. When the designer does provide information, the system learns new rationale-based indices, by storing the information that the selected substitution is believed to satisfy the designer's constraint. We note that in itself, a feature such as "high fuel efficiency" is not enough to fully specify a retrieval—an airliner designer seeking a high efficiency engine would not consider the high-efficiency engine from a Cessna. In DRAMA, the features stored from designer queries are used only to filter candidates that are already believed to fit the task context.

DRAMA also differs from existing CCBR systems in what it retrieves. Initially, both DRAMA and CCBR systems are aimed at retrieving the most appropriate complete solution from previous cases. However, in its retrieval to support adaptation, DRAMA provides the ability to perform retrievals focused on subparts of the problem for the user to compose. As the user adapts part of the design, the retrieval context changes automatically, loosely corresponding to CCBR systems' adjusted rankings as more information becomes available.

7 Future Directions

The DRAMA system is an ongoing project. The CMap tools are already in use for concept mapping at NASA, and the goal of the project is to test the system in the context of a design project for the next generation of reusable spacecraft. The concrete experience from this test will provide feedback and data to adjust details of the interface, functionality, and indexing algorithms. It will also provide data for conducting controlled tests of the quality of recommendations provided by the system. Because the system lacks the knowledge to evaluate the quality of the designs produced, the designer using the system bears the responsibility of assuring that adaptations are reasonable; the key question is how well the system aids designers in their work. However, knowledge-based tools could be developed to provide some verification, and this would be highly desirable.

Because the CMap tools provide the capability to share CMaps across the World Wide Web, designs from multiple designers and sites can be imported into the system's design process. Work is under way at the University of West Florida to develop CMap facilities for managing concurrent CMap generation and modification. Ideally, the design context for a particular engine, for example, could be updated as other designers make other changes in the specifications.

The system's capability to deal with non-uniform representations is being enhanced by the use of IR methods such as thesaurii to aid matching. In addition to refining the system as an aid to recording and reusing design information, we see a long-term opportunity to apply it to reuse of information about *design processes*. A CMap-style interface could be used to capture traces of the steps used in generating a design (e.g., conceptual design, specification, numerical simulations, etc.), to capture how a design was formulated and to guide reasoning throughout the design process.

8 Conclusions

Our experience with the DRAMA system provides a case study of some central issues for interactive CBR systems. Our integration of CBR with CMaps was motivated by the complexity of aerospace design, for which autonomous intelligent design tools are currently infeasible. However, the framework applies to other design tasks as well. It provides a general "knowledge-light" model for flexible graphically-based case acquisition, manipulation, and reuse.

The DRAMA project has identified a number of principles that we expect to have broad implications for integrations between CBR components and interactive systems:

- Representations should be easily comprehensible and interactively adaptable by end users; visually-based representations may be especially useful.
- Support for representation generation should help assure consistent representations, but must not prevent the users of interactive systems from developing new representations or representational elements when needed.

– CBR's "retrieve and adapt" process to build new cases can facilitate standardization by reusing prior representational components. This can naturally build up the case library and the representational vocabulary in parallel.
– The same types of similarity considerations used to guide retrieval can be used to suggest representational vocabulary as cases are built.
– Retrieval should tolerate representational discrepancies.
– Interactive support systems should be sufficiently integrated into the processes they support to be able to unobtrusively monitor and exploit information about the task context.

The overall conclusion is that interaction must be across all parts of the CBR system—initial knowledge capture, representation, retrieval, and adaptation—and across the larger task. Frameworks that allow the user and system to support each other in a shared task context, building up and using shared knowledge, have the potential to leverage off the strengths and alleviate the weaknesses of both system and user.

References

[Aha and Breslow, 1997] D. Aha and L. Breslow. Refining conversational case libraries. In *Proceedings of the Second International Conference on Case-Based Reasoning*, pages 267–278, Berlin, 1997. Springer Verlag.

[Bagg, 1997] T. Bagg. RECALL: Reusable experience with case-based reasoning for automating lessons learned. http://hope.gsfc.nasa.gov/RECALL/-homepg/recall.htm, 1997.

[Cañas *et al.*, 1995] A Cañas, K. Ford, J. Brennan, T. Reichherzer, and P. Hayes. Knowledge construction and sharing in quorum. In *World Conference on Artificial Intelligence in Education*, 1995.

[de Silva Garza and Maher, 1996] A. Gómez de Silva Garza and M. Maher. Design by interactive exploration using memory-based techniques. *Knowledge-Based Systems*, 9(1), 1996.

[Domeshek *et al.*, 1994] E. Domeshek, M. Herndon, A. Bennett, and J. Kolodner. A case-based design aid for conceptual design of aircraft subsystems. In *Proceedings of the Tenth IEEE Conference on Artificial Intelligence for Applications*, pages 63–69, Washington, 1994. IEEE Computer Society Press.

[Falkenhainer *et al.*, 1989] B. Falkenhainer, K. Forbus, and D. Gentner. The structure-mapping engine: Algorithm and examples. *Artificial Intelligence*, 41:1–63, 1989.

[Gebhardt *et al.*, 1997] F. Gebhardt, A. Voß, W. Gräther, and B. Schmidt-Belz. *Reasoning with complex cases.* Kluwer, Boston, 1997.

[Goel *et al.*, 1991] A. Goel, J. Kolodner, M. Pearce, and R. Billington. Towards a case-based tool for aiding conceptual design problem solving. In R. Bareiss, editor, *Proceedings of the DARPA Case-Based Reasoning Workshop*, pages 109–120, San Mateo, 1991. DARPA, Morgan Kaufmann.

[Gruber and Russell, 1992] T. Gruber and D. Russell. Generative design rationale: Beyond the record and replay paradigm. Knowledge Systems Laboratory KSL 92-59, Computer Science Department, Stanford University, 1992.

[Hammond, 1989] K. Hammond. *Case-Based Planning: Viewing Planning as a Memory Task.* Academic Press, San Diego, 1989.

[Hinkle and Toomey, 1995] D. Hinkle and C. Toomey. Applying case-based reasoning to manufacturing. *AI Magazine*, 16(1):65–73, Spring 1995.

[Hua and Faltings, 1993] K. Hua and B. Faltings. Exploring case-based design - CADRE. *Artificial Intelligence in Engineering Design, Analysis and Manufacturing*, 7(2):135–144, 1993.

[Jonassen *et al.*, 1993] D. Jonassen, K. Beissner, and M. Yacci. *Explicit methods for conveying structural knowledge through concept maps*, chapter 15, page 155. Erlbaum, Hillsdale, NJ, 1993.

[Kitano and Shimazu, 1996] H. Kitano and H. Shimazu. The experience sharing architecture: A case study in corporate-wide case-based software quality control. In D. Leake, editor, *Case-Based Reasoning: Experiences, Lessons, and Future Directions*, pages 235–268. AAAI Press, Menlo Park, CA, 1996.

[Kolodner, 1993] J. Kolodner. *Case-Based Reasoning.* Morgan Kaufmann, San Mateo, CA, 1993.

[Leake *et al.*, 1997] D. Leake, A. Kinley, and D. Wilson. A case study of case-based CBR. In *Proceedings of the Second International Conference on Case-Based Reasoning*, pages 371–382, Berlin, 1997. Springer Verlag.

[Leake, 1992] D. Leake. *Evaluating Explanations: A Content Theory.* Lawrence Erlbaum, Hillsdale, NJ, 1992.

[Lenz and Ashley, 1998] M. Lenz and K. Ashley, editors. *Proceedings of the AAAI-98 workshop on textual case-based reasoning.* AAAI Press, Menlo Park, CA, 1998.

[Mark *et al.*, 1996] W. Mark, E. Simoudis, and D. Hinkle. Case-based reasoning: Expectations and results. In D. Leake, editor, *Case-Based Reasoning: Experiences, Lessons, and Future Directions*, pages 269–294. AAAI Press, Menlo Park, CA, 1996.

[Novak and Gowin, 1984] J.D. Novak and D.B. Gowin. *Learning How to Learn.* Cambridge University Press, New York, 1984.

[Schank, 1982] R.C. Schank. *Dynamic Memory: A Theory of Learning in Computers and People.* Cambridge University Press, Cambridge, England, 1982.

[Smith *et al.*, 1995] I. Smith, C. Lottaz, and B. Faltings. Spatial composition using cases: IDIOM. In *Proceedings of First International Conference on Case-Based Reasoning*, pages 88–97, Berlin, October 1995. Springer Verlag.

[Sycara, 1988] K. Sycara. Using case-based reasoning for plan adaptation and repair. In J. Kolodner, editor, *Proceedings of the DARPA Case-Based Reasoning Workshop*, pages 425–434, San Mateo, CA, 1988. Morgan Kaufmann.

[Vollrath, 1998] I. Vollrath. Reuse of complex electronic designs: Requirements analysis for a CBR application. In P. Cunningham, B. Smyth, and M. Keane, editors, *Proceedings of the Fourth European Workshop on Case-Based Reasoning*, pages 136–147, Berlin, 1998. Springer Verlag.

[Voß, 1994] A. Voß. The need for knowledge acquisition in case-based reasoning – some experiences from an architectural domain. In *Proceedings of the Eleventh European Conference on Artificial Intelligence*, pages 463–467. John Wiley, 1994.

[Wilke *et al.*, 1997] W. Wilke, I. Vollrath, K.-D. Althoff, and R. Bergmann. A framework for learning adaptation knowedge based on knowledge light approaches. In *Proceedings of the Fifth German Workshop on Case-Based Reasoning*, 1997.