



# Integrating diverse information resources in a case-based design environment

David B. Leake<sup>a,\*</sup>,  
Larry Birnbaum<sup>b</sup>, Kristian Hammond<sup>b</sup>, Cameron Marlow<sup>c</sup>, Hao Yang<sup>d</sup>

<sup>a</sup>Computer Science Department, Indiana University, Lindley Hall, 150 S. Woodlawn Ave, Bloomington, IN 47405, USA

<sup>b</sup>Intelligent Information Laboratory, Computer Science Department, Northwestern University, 1890 Maple Avenue, Evanston, IL 60201, USA

<sup>c</sup>The Media Laboratory, Building E15, Massachusetts Institute of Technology, 77 Massachusetts Ave, Cambridge, MA 02139, USA

<sup>d</sup>PDC, MD 270, GB-D68, Vehicle Operations, Ford Motor Co., 21500 Oakwood Blvd, Dearborn, MI 48124, USA

Received 1 December 1998; accepted 1 August 1999

---

## Abstract

The success of case-based design aids depends both on the case-based reasoning processes they apply and on effectively integrating those processes into the larger task context: on making the case-based reasoning component present case information at the right time and in the right way, on exploiting additional information resources as needed to supplement the case library and to guide case application, on capturing useful information from current reasoning and providing it to up- and down-stream designers, and on unobtrusively learning new cases during the design process. This article presents a set of principles and techniques for integrated case-based design support systems and illustrates their application through a case study of the Stamping Advisor, a system to support feasibility analysis for sheet metal automotive parts. © 1999 Elsevier Science Ltd. All rights reserved.

*Keywords:* Case-based reasoning; Design; Information integration; Information retrieval; Just-in-time retrieval; Knowledge management

---

## 1. Introduction

An experienced designer's memory of prior design experiences can be a powerful aid to the design process. When a designer facing a new task is reminded of similar previous tasks, those reminders may suggest potential solutions to explore or potential problems to avoid. Case-based design support systems aid this process by augmenting the designer's own experience: they provide suggestions and warnings by retrieving relevant cases from a stored library of prior episodes and presenting them to the designer. They also contribute

to the capture and sharing of design knowledge by adding new cases to the library as problems are solved, and by making the accumulated cases and their lessons available to other designers.

Case-based design has long been an active area of case-based reasoning research, and numerous case-based design aids have been implemented to support a wide range of design tasks (see Kolodner, 1993, for examples of some of these systems). Fully realizing the benefits of such systems, however, requires addressing additional issues beyond the case-based design support process itself. In order to maximize the usefulness of case-based design aids, they must be designed not as stand-alone systems but as integral parts of a single unified framework that supports all phases of the design process, that supports the multiple actors that those phases often involve, and that exploits the diverse information resources relevant to design decisions. Thus developers of case-based design-support

---

\* Corresponding author. Tel.: +1-812-855-9756; fax: +1-812-855-4829.

*E-mail addresses:* leake@cs.indiana.edu (D.B. Leake), birnbaum@ils.nwu.edu (L. Birnbaum), hammond@ils.nwu.edu (K. Hammond), cameron@media.mit.edu (C. Marlow), hyang1@ford.com (H. Yang).

systems must address crucial questions of integrating the case-based reasoning (CBR) system into the larger task context: how to make the CBR component provide information at the right time and in the right form, how to exploit other information sources in concert with case information, how to capture and transmit accumulated information for use up- and downstream, and how to acquire new cases unobtrusively. This paper presents a set of design principles and techniques addressing these questions. It illustrates their application through a case study of the Stamping Advisor, a tool to support feasibility analysis for sheet metal automotive parts by capturing, managing and presenting relevant design knowledge.

## 2. The Stamping Advisor domain

Automotive body design is a crucial task in automobile development. Body design has a profound impact on the vehicle's appeal and function, and the body is the most expensive component of the vehicle to manufacture. Stamped body parts, which make up the major portion of the body subsystem, are designed under constraints arising from aesthetic considerations, structural and functional requirements, cost concerns, and the availability of manufacturing resources.

Body styles are developed in an iterative process between initial designers and *feasibility engineers* who examine the design for potential manufacturing issues. These include formability issues, which may result in splitting or wrinkling of the metal after the stamping process; manufacturing process complexity issues, such as shapes that must be stamped with a large number of dies (increasing costs), and quality concerns due to material properties and feature shapes, which may add significant cost to die testing or affect the quality or consistency of the final product. Integrating these considerations into the design process serves the goal of *designing for manufacturability*.

The feasibility engineer's task is to identify potential problems, to justify why they are likely to occur, to estimate the costs that will be incurred if they are not addressed, and to propose design revisions to remedy them. Expert feasibility engineers report that they often base their judgments on specific experiences with prior designs. However, new engineers begin their work without the expert's library of experiences, and even experienced engineers may not have had experience with the most relevant designs for a particular problem. Multiple information resources exist to aid the feasibility analysis task, such as records of experiences with prior designs, stored in paper and electronic forms. However, it may be difficult or excessively time-consuming for engineers to locate the information needed. Likewise, communicating their decisions and

justifications is often cumbersome: The standard method for communicating their decisions downstream is to fill out and send a paper form.

Key questions for improving this process are how to provide better access to experiences and other engineering knowledge, and how to improve the usefulness of the information when it is reapplied. A collaboration was established between the Intelligent Information Laboratory at Northwestern University and the Vehicle Operations and Visteon divisions at the Ford Motor Company to investigate integrated case-based design support systems to address these questions. The company already had captured paper records of feasibility assessment issues and decisions, some of which had been placed in a database, providing a library of seed cases. The research question was how, given a set of feasibility analysis cases and the standard manuals used by feasibility engineers, to access and present them to maximize their usefulness to the design process.

Thus one goal of the project was *information integration* (Knoblock and Levy, 1998): to develop methods for satisfying the designer's information needs using cases and other information sources, for automatically producing the information needed by designers up- and down-stream, and for supporting unobtrusive case acquisition from available information resources. The Intelligent Information Laboratory developed the *Stamping Advisor* system, described in this paper, to demonstrate a framework for this design support process, and its approaches are now being applied to new systems at the Ford Motor Company.

## 3. Principles for integrated intelligent design support

The Stamping Advisor system embodies five general principles for the integration of case-based design support systems into the design environment. These principles are as follows:

- Seamless interaction. Interaction with the combined system must parallel the feasibility engineer's own problem-solving process.
- Just-in-time retrieval. The system must proactively anticipate information needs and automatically provide the right information when it is needed, rather than placing the burden on the user to formulate requests.
- Integration with other knowledge sources. The system must link all available information resources, presenting prior cases, supplementary information to help understand the cases or apply their lessons, and additional information as appropriate to the task.

- Integration across tasks. The system must support not only the immediate reasoning task but also the downstream tasks it serves. The system should automatically access information about the previous tasks to provide a context for its reasoning, and should produce products that can be used by the reasoning processes downstream.
- Experience capture. Each processing episode must provide new cases in a usable form.

These principles are related to two basic tenets of the case-based reasoning cognitive model (Kolodner, 1994; Leake, 1998; Schank, 1982): that accessing and storing cases is a natural part of task performance, and that models of knowledge access must reflect the task context. The design-support framework presented here extends these principles to anticipate the user's needs, accessing relevant information wherever it is available, and extends the target of support beyond the current user to capture and transmit relevant information downstream.

### 3.1. Realizing these principles

Achieving a design support system that respects the previous principles requires addressing a number of CBR issues. Integrating the system with the feasibility engineer's reasoning and providing just-in-time support requires modeling his or her reasoning process, and especially modeling when and why particular cases and other information resources are retrieved. Integrating multiple knowledge sources depends both on appropriate task-based indexing and on methods for similarity assessment and retrieval that can be applied to preexisting documents and other information sources that differ from traditional cases. Experience capture depends on methods for case acquisition. After an overview of the system architecture and case representation, the remainder of this paper discusses how the Stamping Advisor system links these processes and addresses each of these issues.

## 4. The Stamping Advisor's architecture

The Stamping Advisor is designed to provide a unified framework for accessing, collecting, and storing design information. It contains components to anticipate information needs during the design process and retrieve prior cases, a graphical interface to present information that highlights potential problems, and additional query generation and retrieval components to retrieve task-relevant information from on-line resources.

The system also provides support for communication between design processes. During the design

process, the user can request design changes (from upstream users), provide justifications for keeping the design intact (for future users), and flag problematic areas (for down-stream users). Once the problems have been addressed, the system generates a set of reports as part of the work product. These reports, as well as case information built up incrementally, are available for adding to system databases.

## 5. Representing feasibility analysis cases

The Stamping Advisor currently uses a feature-vector case representation that includes three types of information. The first is general reference information about the part, such as the model and year of the vehicle for which it was designed and the Ford part number. The second is a coarse-grained numerical encoding of the part's style, used to compare parts in order to retrieve parts with similar shapes. The third is information about the problem involved, with features describing the aspects of the part that cause the problem, the problem they cause, and how the problem was solved. This includes a textual description of the problem identified or predicted by the feasibility analyst (e.g. "the attaching flange is too wide"), a description of the predicted result of the problem (e.g. insufficient rigidity), and a textual description of how the problem was resolved (e.g. "added stiffening beads").

The existing representation is sufficient for current purposes, but two refinements are envisioned to enable finer-grained choices as the size of the case library increases. The first will add structured representations of geometric features to the case representation. For each part type, this information can be captured as a table with numeric fields for key geometric features associated with the part type, such as fields for "radius of curvature" for headlamp openings. The second refinement will replace current textual descriptions with a taxonomy of standard semantic elements for describing part characteristics, problems and repairs. For example, elements for describing part characteristics might include "tight radii" and "wrinkles"; elements to describe problems might include terms such as "springback", a common problem in which a panel returns to its original shape after stamping (e.g. because of the amount of stretching required and the material used); and elements to describe repairs might include standard operations such as "loosen radii" or "add alignment points". Developing such a vocabulary — and supporting the entry of case information in this standardized form — will improve the effectiveness of retrieval compared to text-based methods. However, the textual information in the current cases has proven

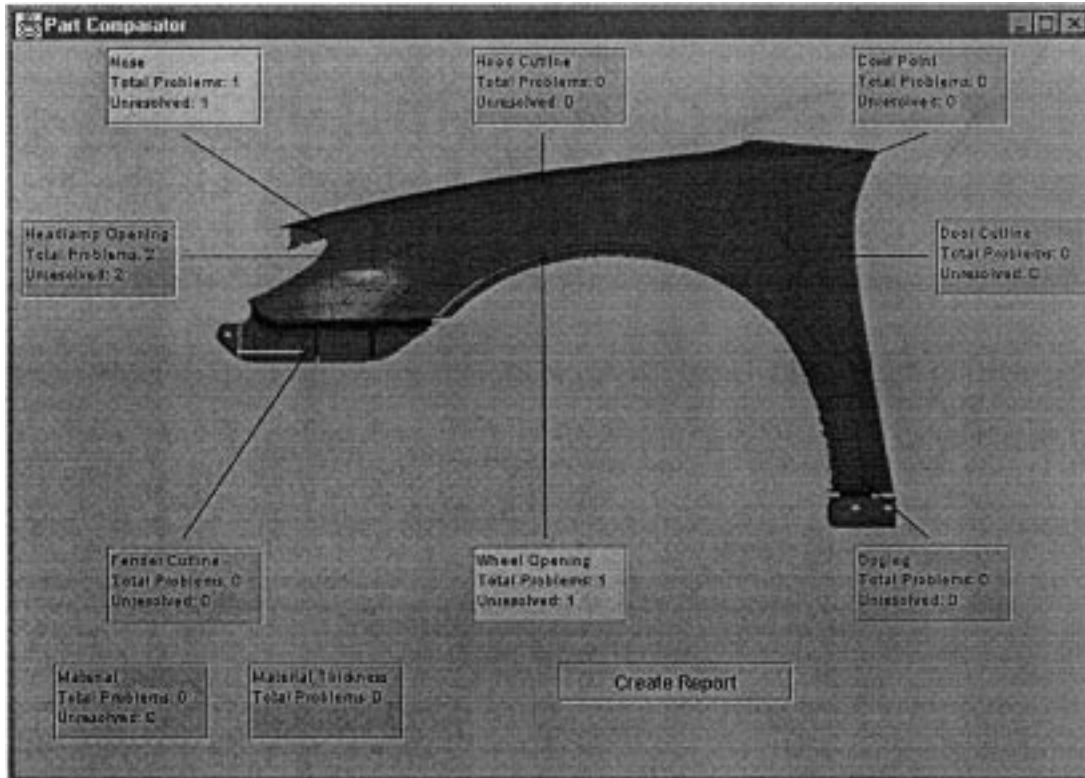


Fig. 1. Screen image from the Stamping Advisor's issue summary screen.

sufficient for good retrieval performance in the current system.

## 6. Coordinating case presentation with the reasoning of feasibility engineers

One of the goals of the Stamping Advisor project was to make the presentation of cases fit the engineer's own reasoning. This is done in two ways: by designing the case presentation interface to fit the engineer's reasoning style, and by using knowledge of the engineer's task to anticipate the engineer's information needs and provide the right information proactively.

In the traditional feasibility analysis process, feasibility engineers are presented with an image of the part to evaluate, produced by the computer-aided design (CAD) system used to generate the design. They then examine the image to identify feasibility issues. Interviews with feasibility engineers established that one of their reasoning styles is to scan the image sequentially, tracing around the border of the part looking for portions of the design that raise feasibility issues. To support this process, the Stamping Advisor's primary interface provides a CAD image of a part, with different regions annotated by information about relevant cases, and with the capability to select particular regions for additional scrutiny. This makes it easy

for the engineer to profit from relevant cases while following his or her normal scanning process.

To support the decisions that the feasibility analyst must make, the system proactively provides the designer with the following three types of case information:

- the closest parts with similarly-shaped regions that have issues, and the descriptions of those issues, to warn about potential issues for the current design;
- the resolutions that previous designers chose for those issues, in order to suggest possible design changes;
- the closest parts with similarly-shaped regions that did not have issues, to help the designer to assess whether the issues apply.

Given a part design, the system summarizes the cases retrieved for each region by annotating the display of the part image with summaries of the number of issues found for each region, and their resolutions. The graphical interface organizes case information geometrically according to the regions of the part. For each region, it provides a summary of the cases found that involve issues for that region. The summaries of the issues for each area of the design are highlighted with color-coded warnings to identify the most problematic regions (green when surrogates support feasibility, yellow for limited problems, red for more serious problems). Fig. 1 shows the issue summary interface for

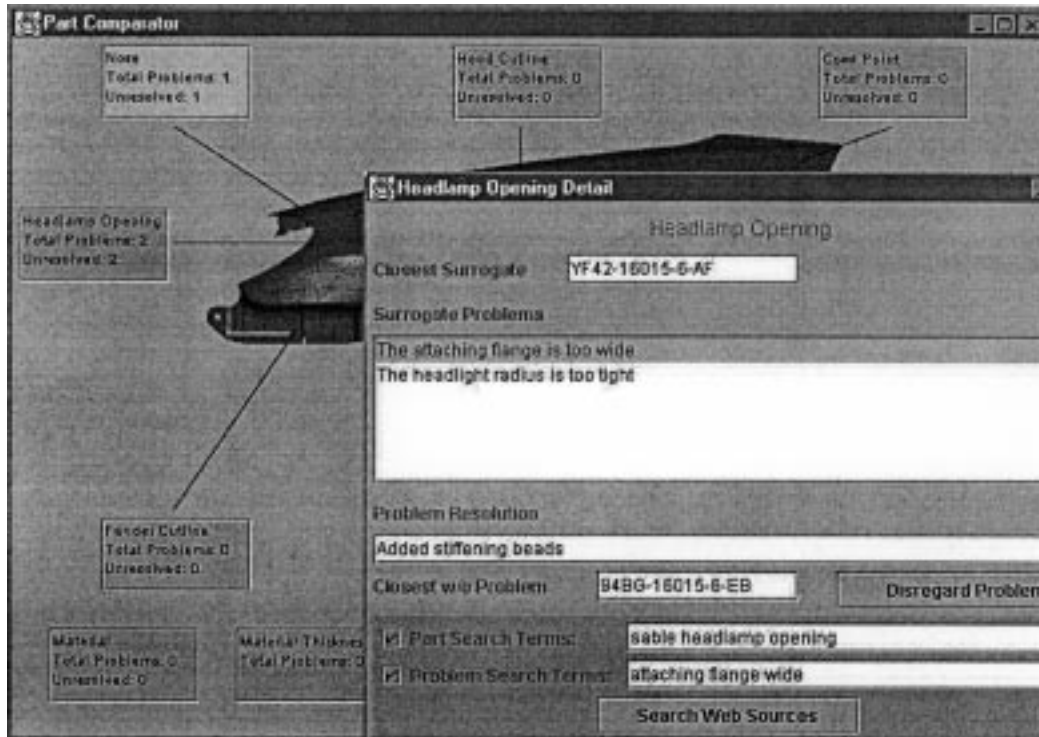


Fig. 2. Presentation of relevant surrogates, issues, and resolutions.

an automobile fender. In the screen display, the left-most box, describing the headlamp opening, is highlighted in red because previous cases identified two potential issues that could not be resolved. The boxes for the nose (upper left) and wheel opening (bottom center) are highlighted in yellow, because each one includes one unresolved problem. No other problems were found, so all other boxes are highlighted in green.

To see additional information for a region, the feasibility engineer clicks on the boxes for the displayed issue sets to select a region of interest. A window appears with information about issues in the most relevant prior cases (in the interface, the prior cases are called “surrogate parts” in accordance with the engineers’ terminology). The engineer can select issues from this list to see how they were resolved. In some instances, the design will have been revised to repair the problem, suggesting a possible revision to consider. In others, the previous engineer may have detected mitigating factors that were originally overlooked, which made the problem inapplicable; these suggest factors for the engineer to check in the current design. In some cases, the prior engineer may have decided that the problematic design feature was so valuable aesthetically that it counterbalanced the extra production costs; in that situation the old case contains information about the estimated costs to consider when weighing whether to allow the potential problem to remain. The interface for this process is shown in Fig. 2.

## 7. The case-retrieval process

In order to evaluate a new part for feasibility, feasibility engineers must consider multiple aspects of the part. The Stamping Advisor supports this process by retrieving multiple sets of cases, each set targeted towards aiding the evaluation of a particular aspect of the part. For a fender there are ten aspects of the part to consider. Eight of these involve the shapes of particular regions of the fender (e.g. the headlamp opening or fender cutline). Two others concern characteristics of the material used (e.g. stamping aluminum parts instead of steel parts involves special feasibility issues concerning sheet metal thickness). Each of the ten aspects corresponds to one of the boxes shown in Fig. 1. Each aspect to be considered is associated with a set of lower-level features relevant to determining its feasibility. For example, lower-level details for the headlamp opening would include features such as the radius of curvature for the opening.

When the Stamping Advisor retrieves prior cases, it selects cases according to three criteria. The system first discriminates between cases by the type of part being analyzed (e.g. when examining the feasibility of a fender, only prior experiences with fenders are considered for retrieval) and then the high-level aspect being examined (e.g. when retrieving information to support feasibility analysis of headlamp openings, it only retrieves cases for headlamp openings). After cases have been filtered by the part and the type of

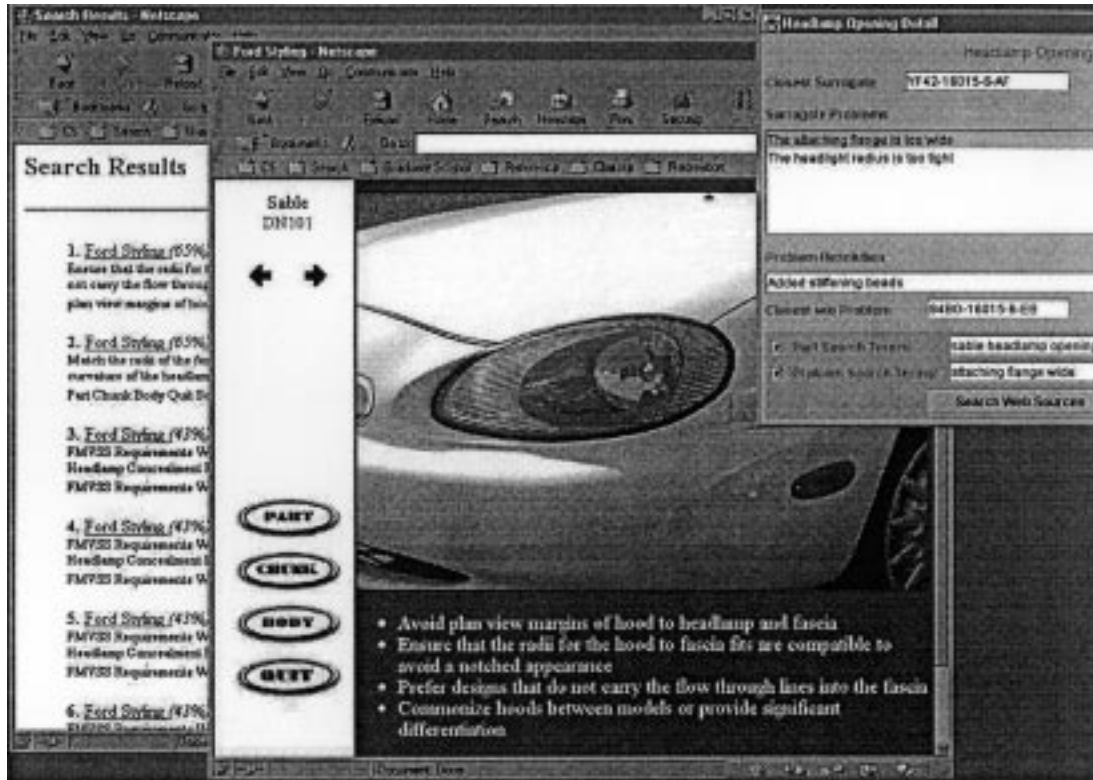


Fig. 3. Style guide page retrieved as relevant to the problem of a headlamp opening being too tight.

part feature under consideration, it then discriminates between the remaining cases according to finer-grained features. The basic retrieval process is a nearest-neighbor retrieval algorithm using feature weightings developed for the domain (see, for example, Watson, 1997).

In some instances, relationships may exist between distinct types of attributes, so that simply considering the regions independently is not sufficient. For example, the amount of springback may depend on the relationships between the shapes of two adjacent regions. In such cases, the relationship across types of features is recorded and used to adjust the weighting of retrieved cases. If both adjacent regions have features that suggest springback, the weight of the cases suggesting springback is increased compared to the weights that were derived from looking at each region alone before considering the supporting relationship between them.

## 8. Integrated information access

Cases are helpful for warning of potential problems and suggesting prior solutions. However, additional information may be needed to assess the relevance of prior issues, to determine the applicability of old solutions, or to develop new solutions reflecting changed constraints. For example, Ford maintains on-line manuals with design recommendations for keeping stamp-

ing costs reasonable and for maintaining consistent styling. Given that these information sources will often be required to supplement the retrieved cases, access to this information is important.

Keeping with the philosophy of integrating the CBR system, a project goal was to use knowledge of the user's task and task context to automatically guide the search for this information: to automatically present the engineer with the supplementary information that is useful, given the knowledge that it is being retrieved in response to specific issues in a specific case. To provide this support, the Stamping Advisor uses tracking information about the current task to automatically formulate targeted queries that can go against any documents indexed by standard search engines. The system demonstrates this capability by automatically generating queries to retrieve relevant style guidelines from the *Ford Advanced Feasibility Guidelines for Styling*.

### 8.1. Query generation and document retrieval

As a product of the manual feasibility analysis process, textual information such as part names, part numbers, problem descriptions, feature names, and the vehicle name are recorded on paper forms. This information has been encoded into the database from which the cases are retrieved, and consequently is available for every part handled by the Stamping

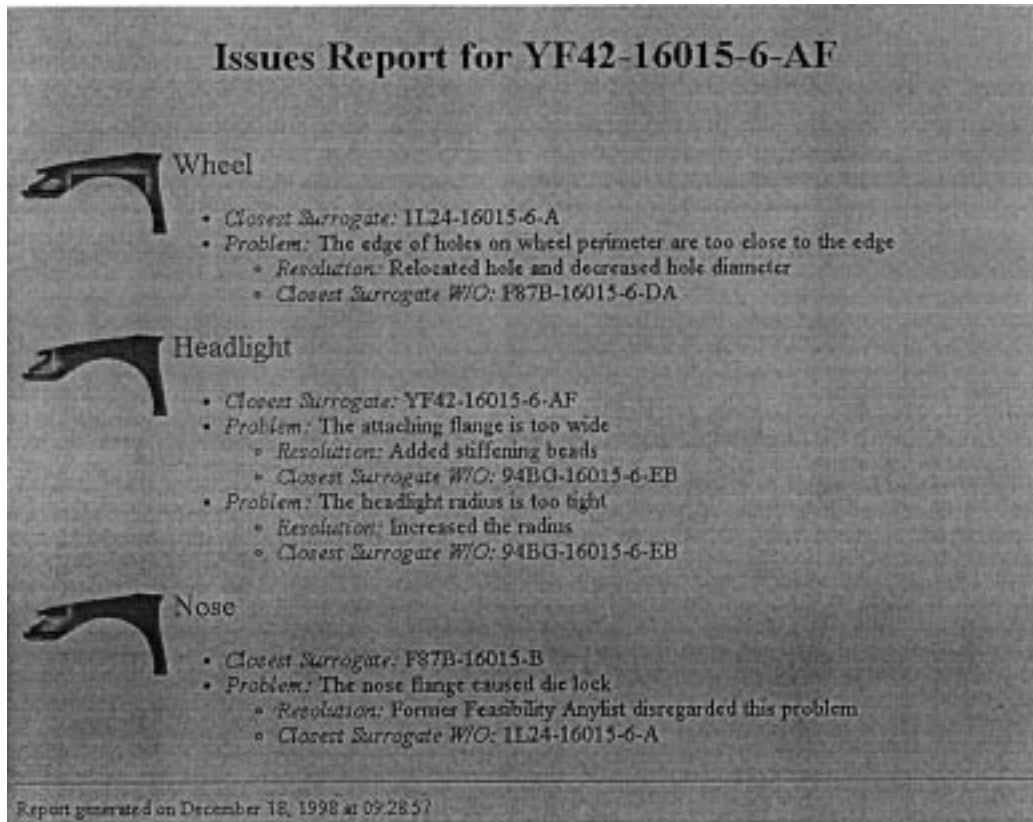


Fig. 4. Final report document.

Advisor. This text is sufficient to distinguish parts at a textual level.

The Stamping Advisor uses this descriptive information, combined with a model of task relevance, to form queries to other information resources. Specifically, when a feasibility engineer is considering a feature, the system automatically forms queries to gather additional information about related features or problems from on-line resources. The system forms its queries by extracting textual strings for four pieces of information from the current task context (the records of the part being designed and the prior case suggesting a problem): the name of the vehicle for which the current part is being designed, the name of the current part, the problematic feature, and a textual description of the problem in question. The system processes each textual string by removing any words contained in a standard “stop list” of frequent words with low information content. It then generates a list of the remaining terms. This list becomes the query to use in searching textual information sources.

For example, when the feasibility engineer considers the design for the fender for a Sable, and examines the headlamp opening problems highlighted in Fig. 1, a retrieved case suggests that there may be a problem with the attaching flange being too wide. The Stamping Advisor generates a query containing “Sable

headlamp opening” for the part under consideration, and “attaching flange wide” for the problem. Upon the feasibility engineer’s request, this query is used to search for relevant guidelines in on-line manuals. Before initiating a search, the engineer can request that the query be focused on only similar parts or similar problems, and can edit the query text as desired (e.g. to replace “Sable” to compare the styling on a different line of car). The query presentation interface is shown at the bottom right of Fig. 2.

Once created, this query can be passed to any typical Internet search engine to search selected resources. The current implementation uses the document indexing system Verity to index the on-line Ford Style Guide, as illustrated in Fig. 3. Verity processes queries by stemming each of the given words, broadening the search to other possible forms of the terms, and assigning a numerical score. This score is based first on the number of word matches and then on the density of those matches within a given document.

A ranked list of matches is presented to the feasibility engineers, who can select documents to retrieve. Evaluating and refining this query-generation method are topics for future research, but informal tests suggest that it is sufficient to retrieve relevant information from the task-focused resources being searched.

## 9. Integration across tasks

Previous case-based design support tools have a natural goal: aiding a designer in his or her task. However, in industrial settings, the designer's task is only one step in an extended process. For example, in stamping design, one or more designers initially formulate the design, a feasibility engineer critiques the design and makes suggestions, and the design is refined through an iterative cycle of changes and critiques. When a design is finalized, down-stream design team members may need to evaluate the design, its potential issues and the designers' justifications for why they matter (or do not matter), and how they were resolved. Ideally, design aids should support this entire *process* rather than supporting only one individual step. This requires the sharing of information across tasks.

A tenet of the Stamping Advisor project is that the design-support system for any particular task should automatically access information about the previous tasks to provide a context for its reasoning, and should produce products that can be used by the reasoning processes downstream. Work is under way on augmenting the CAD system used for initial design to automatically capture the specification information used in feasibility analysis cases (e.g. to capture the part number, part type, vehicle, and a pointer to the CAD file), to be passed automatically to the Stamping Advisor at the start of feasibility analysis. This will provide additional integration between the tasks of the initial designer and the feasibility engineer.

At the close of the feasibility assessment process, the system generates a Final Report Document to aid upstream or down-stream design team members who need to understand or evaluate the feasibility engineer's work, replacing documentation generated by hand. In the task model developed for the Stamping Advisor, the information needed by the evaluation process is: (1) the part being examined, (2) the issues considered, (3) how they were disposed of, and (4) the surrogates providing evidence relevant to the issues and decisions. A sample Final Report Document is shown in Fig. 4.

## 10. Case capture

Ford maintains an extensive library of reports of feasibility analysis problems and solutions in paper form. However, as is often the case in developing applications systems, there is a bottleneck in translating this paper information into a usable case form. The ability of the Stamping Advisor to create Final Report Documents suggests a way to alleviate this bottleneck. In the Stamping Advisor, a user's decisions about appropriate surrogates, the problems they predict, and

the ultimate disposition of the problems are captured by the system during the feasibility assessment process. These are used to create the Final Report Document. This document is produced as the by-product of the user's decision-making and requires no additional effort on his or her part beyond that already required to convey the needed information down-stream. This document automatically combines information captured from the user with other background information, gathering all the information needed to generate a new feasibility assessment case.

This case capture framework gathers data when they are available at each phase of the design process, not just during feasibility analysis. The growing record is made available to each down-stream process for reasoning from existing data and addition to the record. In particular, information is built up during initial part design, feasibility analysis, and final decision-making on how to proceed on a part.

Information used to characterize part designs in the CAD system (e.g. (the model, year, and part number, and a pointer to the CAD file) provide an initial record of the design. Current seed cases rely on hand-coded coarse-grained geometric features to determine part similarity, but ideally, automatic geometric matching procedures could be applied to the designs. Development of these procedures could draw on the procedures for geometric feature matching already developed in the context of case-based design (Coulon and Steffens, 1994), as well as on the considerable research addressing the problem of how to recognize machining features (such as holes, slots and grooves) in CAD models (see Shah et al., 1994, for some examples of this work). Alternatively, because the engineer must already document the important geometric features when describing problems to generate the down-stream report, it would be comparatively simple to tag these features according to a predefined vocabulary of standard features that can then be used for matching.

When the Final Report Document is provided electronically to the person who determines the final disposition of the request, that person can enter the final decision to complete the case information. By controlling the information that can be entered at each step of the process (e.g. through menus), cases can be standardized. However, the system's ability to do textual searches using information-retrieval techniques provides the additional capability to search through free-form comments, etc.

In summary, this framework integrates case capture across different parts of the design process, and uses cases as a vehicle both for sharing knowledge as it is gathered and for long-term knowledge capture. In particular, this case-capture framework is based on the principles that case content should:

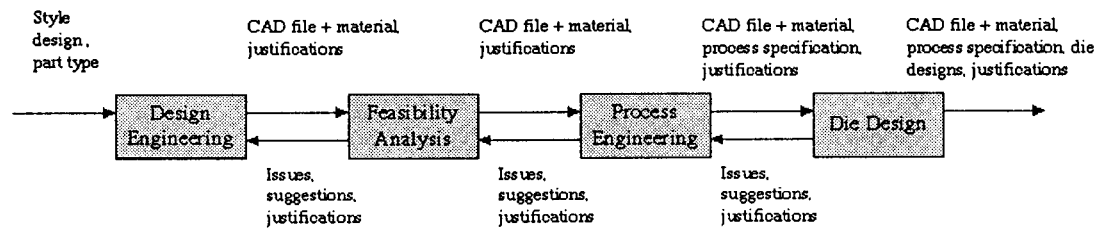


Fig. 5. The four phases of stamping design with their up- and down-stream information flow.

1. be built up incrementally as a natural part of the problem solving process;
2. be used incrementally during the process, as soon as it has been generated;
3. provide a full record of relevant information at the end of the process, in the form needed for future use.

This supports rapid growth of the case library and the standardization of case information.

## 11. Generality of the approach

The previous sections illustrate how each of the design support principles of Section 3 are realized for the Stamping Advisor's task of supporting feasibility analysis. Beyond this application, however, the Stamping Advisor system can be regarded as instantiating a general approach to case-based design support that can be replicated and linked to support the tasks and communication needs of multiple phases of the design process.

For example, feasibility analysis is only one of four design phases involved in developing the part and process designs for stamped automobile body parts. As illustrated in Fig. 5, the feasibility analysis phase is preceded by the *engineering design phase*, which determines how the body surface will be divided into panels, what hidden surfaces and fasteners are needed to enable attachment of adjacent panels, etc. Feasibility analysis is followed by *die process design*, which determines the processes to produce the designed part (e.g. piercings, stamping depth and angles, etc.), in the light of the available production resources, and *die design*, which determines the design of the actual dies to be used in the stamping process. Designers involved in each of these design phases can be supported by proactively suggesting solutions, revisions, and critiques as demonstrated for feasibility analysis by the Stamping Advisor.

In addition, cases storing information captured or augmented during each design phase provide a vehicle for the automatic accumulation of information and communication up- and down-stream. Cases capturing

the up-stream designer's decisions and the justifications for those decisions provide the down-stream designer with a starting point for the next design phase. When problems are detected or possible optimizations are noticed down-stream, the down-stream designer can request changes up-stream, justifying the need for changes with references to prior experiences and other information. The up-stream designer then responds to the change requests, providing justifications for both the changes made and the suggestions that were not followed.

Across all these phases, stored cases can collect and package lessons for communication and for cross-context use — any lesson may be retrieved and reapplied at multiple points in this design sequence, not only in the processes adjacent to those where it was detected. For example, problems revealed during the die design process for a part may suggest lessons about problematic shapes to be applied during future feasibility analysis.

## 12. Relationship to previous work

### 12.1. Case-based design support

A wide range of case-based design support tools has been developed for numerous tasks such as architectural design (Gebhardt, et al., 1997; Goel et al., 1991; Hua and Faltings, 1993; Maher et al., 1995; Smith et al., 1995), conceptual design of aircraft subsystems (Domeshek et al., 1994; Leake and Wilson, 1999), autoclave layout design (Hinkle and Toomey, 1995), device design (Goel, 1989; Sycara et al., 1991), and circuit design (Vollrath, 1998). A central task of many of these systems is to augment the user's memory by providing records of relevant prior successes and failures. For example, in the ARCHIE system (Goel et al., 1991), the user specifies a set of design goals and is presented with past designs providing suggestions and warnings relevant to those goals. The Stamping Advisor's just-in-time retrieval process replaces user specification of goals with automatic anticipation of

information needs, to present the user with relevant cases without requiring the user to make requests.

The Stamping Advisor's task is closely related to that of the load validator in the system Clavier (Hinkle and Toomey, 1995), which warns users about potential problems in new autoclave layouts by presenting users with similar prior layouts and their outcomes. A crucial issue in autoclave layout design is the interacting effects of components of the layouts, and these interactions are hard to explain and separate. Consequently, Clavier based its predictions on the similarity of the entire previous layouts with entire current designs. In the Stamping Advisor domain, problems can be localized by the feasibility engineer. As a result, Stamping Advisor cases can represent problems at the level of the individual regions they affect (with additional checks for interactions that span multiple regions). Using cases at the level of subparts facilitates the transfer of problem information to new contexts (e.g. headlamp opening problems can be predicted on the basis of prior experiences with the headlamp openings in very different styles of fenders). The Stamping Advisor also differs in using cases not only to advise the designer, but also to capture and communicate the rationale that underlies design decisions taken in response to its advice, and in automatically generating queries to retrieve additional information to augment its cases, to help clarify their significance, and to suggest possible repairs.

The Stamping Advisor integrates case-based support with the CAD tools already used to create and examine designs for stamped parts. This approach is similar to those taken by the FABEL (Gebhardt et al., 1997) and CADRE (Hua et al., 1996) projects, both of which integrate the CBR system with existing CAD systems. It differs, however, in using a very specific task model to automatically determine both the types of information to provide and when to provide information with just-in-time retrieval. In contrast, FABEL provides a "virtual construction site" that the engineer can navigate, and a tool kit from which the designer selects tools to perform particular types of retrievals. The Stamping Advisor uses its model of how the feasibility analysis task is done to anticipate specific information needs and proactively determine what information is needed and how to retrieve it.

### 12.2. Integrating CBR and IR

The Stamping Advisor also goes beyond case-based support to integrate multiple knowledge sources. There is considerable current interest in the use of CBR for textual cases, and in the use of information-retrieval methods to access them (Lenz and Ashley, 1998). A challenging question is how to maintain the strengths of CBR — the pragmatic focus that traditional CBR

provides — while exploiting the generality of IR methods for assessing the similarity of documents. This depends on bridging the gap between the task-relevant indexing traditionally used in CBR and methods that can be applied to unstructured textual data. (Rissland and Daniels, 1996) present one method for this integration in the retrieval of legal cases. Their system first performs a feature analysis to do a traditional CBR retrieval of the most relevant cases from a case library represented in a carefully structured form. It then uses the textual descriptions of those cases as seed examples for the relevance feedback mechanism of a text-based information retrieval system, which generates queries to retrieve similar texts from a larger library of textual case descriptions. The Stamping Advisor uses task-based characterizations more directly: rather than going from cases to their texts to queries, it directly generates a search engine query by using its characterization of relevant problem features to extract information from the current problem situation. The Stamping Advisor also provides the user with the capability to revise this query before a search to reflect additional information goals that may not be known to the system.

In addition, the Stamping Advisor's IR retrieval is not only used to extend capabilities to access textual cases, but is also a method for focused access to information in style guides, etc., to help understand the cases' significance.

### 12.3. Case capture

A crucial issue for scaling up CBR applications is knowledge capture. The Stamping Advisor system is designed to facilitate this through knowledge capture during use. Feasibility analysis is a "natural" CBR domain (Mark et al., 1996), in that the manual feasibility analysis process includes extensive paper documentation for each design case. However, it is anticipated that the primary source of new cases will be automatic case acquisition during system use. Even if *no* cases were available in the system case library, the system would be useful as a convenient interface for recording feasibility information (now recorded on paper) and aiding searching of on-line resources. Thus feasibility engineers have the incentive to use the Stamping Advisor, and their use of the system provides cases that will increase its usefulness as sufficient data is gathered to take full advantage of the CBR component.

### 12.4. Knowledge management

Increased recognition of the value of corporate information has led to new interest in knowledge management — how to manage repositories of information

to facilitate access and reuse (see, for example Davenport and Prusak, 1997). Efforts to improve knowledge management often focus on developing tools that a user can apply to access information resources, such as data warehouses, and on developing practices that encourage employees to contribute to knowledge repositories (O’Leary, 1998). CBR techniques provide a basis for the representation and retrieval of experiential information (Becerra-Fernandez and Aha, 1999), and have already been successfully applied for large-scale corporate knowledge sharing (Kitano and Shimazu, 1996; Klahr, 1997), as well as for local knowledge capture and reuse in many contexts.

A primary contribution of the Stamping Advisor approach to knowledge management is its integration of knowledge capture and access *into the user’s task*. Although current knowledge management systems are designed with particular tasks in mind, it is normally the user who must make the actual link between tasks and information: the user must decide that he or she needs to seek information, must determine the type of information to seek to further the task, and must select tools to access knowledge repositories to seek that information. In addition, after the task is complete, the user must often take extra steps to document the task process for capture. In contrast, the approach described in this paper performs knowledge access and capture automatically during the design process, according to a model of how designers perform their tasks. In addition, the Stamping Advisor’s just-in-time information access anticipates the designer’s needs, instead of requiring the designer to specify them. The knowledge management perspective on this project is discussed in (Leake et al., 1999b).

### 13. Conclusions

The Stamping Advisor project illustrates a set of principles for integrating case-based reasoning systems into the larger task context. The system was designed to provide an open architecture for case and other information retrieval based on features of the current design, and to exploit and support the flow of information from successive steps of the design process. To make the system natural to use, the interaction is designed to parallel the feasibility engineer’s own problem-solving process, and to automatically provide just-in-time access to the right cases, rather than placing the burden on the user to formulate requests. The system uses its task model to generate focused IR queries to access additional knowledge sources, retaining the capability for the user to adjust those queries to explore additional topics. The system performs automatic knowledge capture, gathering in-

formation about each interaction and using it for dual purposes: to provide the information needed downstream of the reasoning task and to provide new cases for future use. Feasibility engineers who have seen the system expect it to be a valuable aid to their problem-solving, and work is underway on the next phase of system scale-up.

The central lesson of this project is that the development of successful case-based design aids must depend not only on the CBR processes themselves, but also on crucial questions of integrating the CBR system into the larger task context and supporting exploitation of the diverse information sources the task involves: making the system automatically provide information when it is needed and in the right form, accessing relevant information from additional information sources, and communicating and capturing information. Work is continuing to strengthen this integration as the current system is refined. One goal, for example, is to fully integrate the Stamping Advisor into the initial CAD design process, to immediately warn the original designer of potential problems while the design is being generated. The authors believe that CBR fits naturally into a new mode of knowledge management that not only tracks where documents *are*, but tracks *how they are used* and *where they are needed* to access multiple information sources to provide the right information at the right time.

### Acknowledgements

This research was supported by the Ford Motor Company under award No. 0970-355-A200. It was conducted while David Leake was a Visiting Professor at Northwestern University, on sabbatical leave from Indiana University, and he thanks the Intelligent Information Laboratory and the Northwestern Computer Science Department for their generous support. His research is also supported in part by NASA under award No. NCC 2-1035. This paper is extended and revised from (Leake et al., 1999a).

### References

- Becerra-Fernandez, I., Aha, D., 1999. Case-based problem solving for knowledge management systems. In: Proceedings of the Twelfth Annual Florida Artificial Intelligence Research Symposium. AAAI, Menlo Park, CA, pp. 219–223.
- Coulon, C.-H., Steffens, R., 1994. Comparing fragments by their images. In: FABEL Report 13, Gesellschaft für Mathematik und Datenverarbeitung mbH, pp. 36–44.
- Davenport, T., Prusak, L., 1997. Working Knowledge: How Organizations Manage What They Know. Harvard Business School Press, Cambridge, MA.
- Domeshek, E., Herndon, M., Bennett, A., Kolodner, J., 1994. A case-based design aid for conceptual design of aircraft subsystems.

- tems. In: Proceedings of the Tenth IEEE Conference on Artificial Intelligence for Applications. IEEE Computer Society Press, Washington, pp. 63–69.
- Gebhardt, F., Voß, A., Gräther, W., Schmidt-Belz, B., 1997. Reasoning with Complex Cases. Kluwer, Boston.
- Goel, A., 1989. Integration of case-based reasoning and model-based reasoning for adaptive design problem solving. Ph.D. thesis, The Ohio State University.
- Goel, A., Kolodner, J., Pearce, M., Billington, R., 1991. Towards a case-based tool for aiding conceptual design problem solving. In: Bareiss, R. (Ed.), Proceedings of the DARPA Case-Based Reasoning Workshop. DARPA, Kaufmann, San Mateo, CA, pp. 109–120.
- Hinkle, D., Toomey, C., 1995. Applying case-based reasoning to manufacturing. *AI Magazine* 16 (1), 65–73.
- Hua, K., Faltings, B., 1993. Exploring case-based design — CADRE. *Artificial Intelligence in Engineering Design, Analysis and Manufacturing* 7 (2), 135–144.
- Hua, K., Faltings, B., Smith, I., 1996. CADRE: case-based geometric design. *Artificial Intelligence in Engineering* 10, 171–183.
- Kitano, H., Shimazu, H., 1996. The experience sharing architecture: a case study in corporate-wide case-based software quality control. In: Leake, D. (Ed.), *Case-Based Reasoning: Experiences, Lessons, and Future Directions*. AAAI Press, Menlo Park, CA, pp. 235–268.
- Klahr, P., 1997. Knowledge management on a global scale. In: Gaines, B., Musen, M., Uthurusamy, R. (Eds.), Proceedings of the 1997 Spring Symposium on Artificial Intelligence in Knowledge Management. AAAI, Stanford, CA, pp. 82–85.
- Knoblock, K., Levy, A. (Eds.), 1998. Proceedings of the AAAI-98 Workshop on AI and Information Integration. AAAI Press, Menlo Park, CA.
- Kolodner, J., 1993. *Case-Based Reasoning*. Morgan Kaufmann, San Mateo, CA.
- Kolodner, J., 1994. From natural language understanding to case-based Reasoning and beyond: a perspective on the cognitive model that ties it all together. In: Schank, R., Langer, E. (Eds.), *Beliefs, Reasoning, and Decision Making: Psycho-Logic in Honor of Bob Abelson*. Erlbaum, Hillsdale, NJ, pp. 55–110.
- Leake, D., 1998. Cognition as case-based reasoning. In: Bechtel, W., Graham, G. (Eds.), *A Companion to Cognitive Science*. Blackwell, Oxford, pp. 465–476.
- Leake, D., Birnbaum, L., Hammond, K., Marlow, C., Yang, H., 1999a. Integrating information resources: a case study of engineering design support. In: Proceedings of the Third International Conference on Case-Based Reasoning. Springer, Berlin, pp. 482–496.
- Leake, D., Birnbaum, L., Hammond, K., Marlow, C., Yang, H., 1999b. Task-based knowledge management. In: Proceedings of the AAAI-99 Workshop on Exploring Synergies of Knowledge Management and Case-Based Reasoning. AAAI Press, Menlo Park, pp. 35–39.
- Leake, D., Wilson, D., 1999. Combining CBR with interactive knowledge acquisition, manipulation and reuse. In: Proceedings of the Third International Conference on Case-Based Reasoning. Springer, Berlin, pp. 203–217.
- Lenz, M., Ashley, K. (Eds.), 1998. Proceedings of the AAAI-98 workshop on textual case-based reasoning. AAAI Press, Menlo Park, CA.
- Maher, M., Balachandran, B., Zhang, D., 1995. *Case-based Reasoning in Design*. Erlbaum, Hillsdale, NJ.
- Mark, W., Simoudis, E., Hinkle, D., 1996. Case-based reasoning: expectations and results. In: Leake, D. (Ed.), *Case-Based Reasoning: Experiences, Lessons, and Future Directions*. AAAI Press, Menlo Park, CA, pp. 269–294.
- O’Leary, D., 1998. Enterprise knowledge management. *Computer* 31 (3), 54–61.
- Rissland, E., Daniels, J., 1996. The synergistic application of CBR to IR. *Artificial Intelligence Review* 10, 441–475.
- Schank, R., 1982. *Dynamic Memory: A Theory of Learning in Computers and People*. Cambridge University Press, Cambridge.
- Shah, J., Mäntylä, M., Nau, D. (Eds.), 1994. *Advances in Feature Based Manufacturing*. Elsevier Science, Amsterdam.
- Smith, I., Lottaz, C., Faltings, B., 1995. Spatial composition using cases: IDIOM. In: Proceedings of First International Conference on Case-Based Reasoning. Springer, Berlin, pp. 88–97.
- Sycara, K., Guttal, R., Koning, J., Narasimhan, S., Navinchandra, D., 1991. CADET: a case-based synthesis tool for engineering design. *International Journal of Expert Systems* 4 (2), 157–188.
- Vollrath, I., 1998. Reuse of complex electronic designs: requirements analysis for a CBR application. In: Cunningham, P., Smyth, B., Keane, M. (Eds.), Proceedings of the Fourth European Workshop on Case-Based Reasoning. Springer, Berlin, pp. 136–147.
- Watson, I., 1997. *Applying Case-Based Reasoning: Techniques for Enterprise Systems*. Kaufmann, San Mateo, CA.
- David Leake** is an Associate Professor of Computer Science and member of the Cognitive Science faculty at Indiana University. He received his Ph.D. in Computer Science from Yale University in 1990. His primary research interests are case-based reasoning, goal-driven learning, and strategic information search. He Co-Chaired the Second International Conference on Case-Based Reasoning (ICCB-97) and edited the book *Case-Based Reasoning: Experiences, Lessons, and Future Directions* (AAAI Press, 1996). He is the Editor of *AI Magazine*.
- Larry Birnbaum** is Chair of the Computer Science Department and Associate Professor of Computer Science and the School of Education and Social Policy at Northwestern University. He received his Ph.D. in Computer Science from Yale University in 1986. His research interests include interface design and intelligent interfaces; computers and education; natural language understanding; computer vision; and machine learning.
- Kristian Hammond** is a Professor of Computer Science at Northwestern University. He received his Ph.D. in Computer Science from Yale University in 1986. Through 1998 he was the Director of the Artificial Intelligence Laboratory at The University of Chicago, where he led projects investigating intelligent agents in a wide range of domains. He recently founded the Intelligent Information Laboratory to focus on issues of information access and management arising out of the high-speed computer connectivity of the modern world. The Info Lab charter is to invent and creatively exploit new information technologies that respond to and support human goals within complex computer environments.
- Cameron Marlow** is a doctoral student at the MIT Media Laboratory. He received a B.S. degree at The University of Chicago in 1999. His primary research interests involve methods for using context to better satisfy users’ information needs, intelligent indexing, and integrating case-based reasoning with information retrieval.
- Hao (Howie) Yang** is a process development engineer at the Vehicle Operations division of the Ford Motor Company. He received a Ph.D. in Mechanical Engineering from the University of Missouri-Rolla in 1994. His research interests are in the areas of CAD/CAM, feature-based design, intelligent design and manufacturing systems, case-based reasoning, and knowledge management.