Deutsches
Forschungszentrum
für Künstliche
Intelligenz GmbH

Research Report
RR-94-08

Conserving Corporate Knowledge for Crankshaft Design

Otto Kühn, Björn Höfling

March 1994

Deutsches Forschungszentrum für Künstliche Intelligenz GmbH

Postfach 20 80
67608 Kaiserslautern, FRG
Tel.: + 49 (631) 205-3211
Fax: + 49 (631) 205-3210

Stuhlsatzenhausweg 3
66123 Saarbrücken, FRG
Tel.: + 49 (681) 302-5252
Fax: + 49 (681) 302-5341

Deutsches Forschungszentrum für Künstliche Intelligenz

The German Research Center for Artificial Intelligence (Deutsches Forschungszentrum für Künstliche Intelligenz, DFKI) with sites in Kaiserslautern and Saarbrücken is a non-profit organization which was founded in 1988. The shareholder companies are Atlas Elektronik, Daimler-Benz, Fraunhofer Gesellschaft, GMD, IBM, Insiders, Mannesmann-Kienzle, Sema Group, and Siemens. Research projects conducted at the DFKI are funded by the German Ministry for Research and Technology, by the shareholder companies, or by other industrial contracts.

The DFKI conducts application-oriented basic research in the field of artificial intelligence and other related subfields of computer science. The overall goal is to construct systems with technical knowledge and common sense which - by using AI methods - implement a problem solution for a selected application area. Currently, there are the following research areas at the DFKI:

Intelligent Engineering Systems
Intelligent User Interfaces
Computer Linguistics
Programming Systems
Deduction and Multiagent Systems
Document Analysis and Office Automation.

The DFKI strives at making its research results available to the scientific community. There exist many contacts to domestic and foreign research institutions, both in academy and industry. The DFKI hosts technology transfer workshops for shareholders and other interested groups in order to inform about the current state of research.

From its beginning, the DFKI has provided an attractive working environment for AI researchers from Germany and from all over the world. The goal is to have a staff of about 100 researchers at the end of the building-up phase.

Dr. Dr. D. Ruland

Director

Conserving Corporate Knowledge for Crankshaft Design

Otto Kühn, Björn Höfling

DFKI-RR-94-08

This report is a slightly modified version of the paper to be published by Gordon and Breach Publishers in the Proceedings of the Seventh International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems, (IEA-AIE-94), Austin, Texas, May 31 - June 4,1994.
This work has been supported by a grant from The Federal Ministry for Research and Technology (FKZ ITWM-9304/3).

Conserving Corporate Knowledge for Crankshaft Design

Otto Kühn and Björn Höfling
German Research Center for Artificial Intelligence (DFKI)
P. O. Box 2080, D-67608 Kaiserslautern, Germany
Phone: +49-631-205-3463, Fax: +49-631-205-3210
Email: (kuehn,hoefling)@dfki.uni-kl.de

Abstract

A company's technical know-how which constitutes one of its most important assets often exists only in the heads of a small number of human experts. This limits the availability of this crucial resource and puts considerable strain on the respective experts. In cooperation with a German company [1] which produces motor-powered tools and vehicles a prototypical knowledge conservation system was developed which captures an individual expert's know-how about the design of crankshafts and makes it available to the whole design team. The knowledge-based system provides explanations of previous designs and supports unexperienced designers by suggesting viable alternatives. By checking the consistency between new cases and previously stated general design constraints, the system supports a continuous evolution of the stored knowledge which thus always reflects the current state of a company's technical know-how.

[1] The company wishes not to be named in this publication in order to not provide hints to competitors. For the same reason, the examples given in this paper had to be slightly modified.

Contents

1 Introduction **3**

2 Why Knowledge Conservation for Crankshaft Design? **4**

3 Development and Architecture of the KONUS System **5**

3.1 **Knowledge Acquisition** 5

3.2 **Knowledge Representation** 6

3.3 **Overview of the System Architecture** 8

4 Using the KONUS System **9**

4.1 **Using the Design Informer** 9

4.2 **Using the Design Aide** 11

4.3 **Using the Design Librarian** 11

5 Evaluation and Discussion **12**

1 Introduction

A company's accumulated experience in designing and manufacturing a particular product is essential for a successful competition on the marketplace. It thus constitutes a valuable asset which should be carefully fostered and preserved, and utilized as efficiently and effectively as possible.

Even today, it can more often than not be found that such corporate knowledge exists only in the heads of a small number of human experts. This not only entails a certain risk of loss with fluctuations of personnel but, more importantly, imposes a serious restriction on the accessibility of this vital information and puts a high strain on the involved experts. As a consequence, development times for new products are often protracted due to insufficient flow of information, and the introduction of technical innovations is delayed, since the responsible experts have to spend most of their time doing routine tasks.

Conserving corporate knowledge and making it effectively available to the various specialists who are engaged in product development, production and evaluation poses many interesting challenges to knowledge and software engineering which go beyond those of traditional expert systems and approach those of database systems. The relevant knowledge has to be acquired, stored and processed not for the automatic solution of one particular task which is specified in advance, but for supplying a large number of different services which might be requested by the users. Such services comprise: the automatic solution of simple routine tasks, supporting human experts solving complex design and diagnosis tasks, and answering all sorts of questions for which pertinent information is available. In addition, it is of utmost importance that the stored knowledge can be modified and extended so that it always reflects the current state of a company's technical know-how.

In the last five years, the working group on knowledge-based engineering systems at the German Research Center for AI has developed tools and techniques for the acquisition, representation, and compilation [?], for the management [?] and for the evolution [?] of real world knowledge in the domain of mechanical engineering. In the current paper, we will describe how these techniques were applied to build a prototypical knowledge conservation system for crankshaft design in cooperation with design experts from a German company.

In the next section, we will give a brief introduction into the target domain and point out the needs for knowledge conservation in this particular area. In the third section of the paper, we will discuss some knowledge acquisition and knowledge representation problems, and present the architecture of the developed knowledge conservation system. In the fourth section, we will show how the stored knowledge is employed to provide different services which fall into three categories: the design informer who answers questions about previous designs, the design aide who supports the development of new designs, and the design librarian who enables the evolution of the knowledge base. In the last section, we will give a preliminary evaluation of the developed prototype and compare it to some related approaches.

2 Why Knowledge Conservation for Crankshaft Design?

The crankshaft is one of the core pieces of a combustion engine. When designing a new engine or modifying an existing one, the crankshaft will almost always be affected. As a consequence, there is a very high demand on the crankshaft specialist whose advice is urgently needed.

The crankshaft expert is usually a mechanical engineer who has many years of experience with the design and production of crankshafts in the particular company. His expertise consists not only of very deep mechanical engineering knowledge in this special field, but to an even larger extent of his ability to take into account all company-specific circumstances. In order to design a high-quality crankshaft which can be manufactured at low costs, he must consider what machines are available for manufacturing, what quality has previously been achieved on this machinery with particular manufacturing procedures, what raw parts and components are offered at sufficient quality and at reasonable prices, what are the cheapest materials that have the required mechanical properties and can be effectively processed with the available machines and tools, etc. All these factors may affect the geometry of the crankshaft, and must thus be reflected in the technical drawings of the individual components at various stages of processing.

Figure 1 shows the major tasks in which a crankshaft expert is involved. First of all, the crankshaft specialist is a member of several development teams, which devise new products for different segments of the market. His participation is required in all phases of product development, from initial feasibility studies of newly conceived designs, to subsequent elaborations and final evaluations [?].

Besides his participation in the development of new products, the crankshaft specialist is constantly involved in the continuous enhancement and improvement of products already on the market. He has to handle frequent requests for modifications from the manufacturing department which arise from drawbacks of a given design and opportunities for cost reduction being noticed only during manufacturing. The quality assurance department reports flaws which are detected after a product has been on the market. The necessary improvements may again require the expertise of the crankshaft expert. Finally, the marketing department may observe that a competitor is now offering a comparable product at a lower price and may call for a cost reduction for the own product. Such calls for cost reductions entail specific questions about the reasons for the selection of a particular crankshaft design and about the feasibility of particular, cost-cutting modifications.

Since for cost reasons the number of personnel must be kept at a minimum, there are not enough crankshaft experts available to handle all requests immediately. Furthermore, many questions can only be answered by the specialist who designed the particular crankshaft. Due to insufficient documentation (caused by time pressure and work overload), he is often enough the only one who has the required information. This bottleneck in the availability of expertise on crankshafts causes severe delays in the development of new products and in improving the cost efficiency and quality of existing products.

A computer system which provides adequate support in the described situation must capture parts of the expertise of the crankshaft specialist and make it available to the whole development team. For this, it is not sufficient to just store the information on some medium, be it a printed document, a database or a hypertext, but it must be easily at hand in various situations. Otherwise everybody still will bother the human expert instead of consulting the medium. The captured knowledge must thus be stored in such a way that it can be automatically processed so that *"The right information is provided at the right time."* as one of the intended users put it.

Since an essential portion of the to be conserved knowledge consists of company specific experiences and guidelines which are based on the current engineering technology and market situation, it will be of little use, if the current situation changes. For instance, the devaluation of a currency, may affect the competitiveness of various suppliers, and the previously favored crankshaft design is no longer optimal with respect to cost-efficiency. The knowledge base must thus be open for revision, adaptation and extension, in short for evolution [**?**]. Whereas major changes and additions will require the participation of a knowledge engineer, frequently required updates of the captured corporate knowledge should be done by the users of the system and during use, i.e. as soon as a deficiency is detected.

In order to demonstrate that a computer system which meets these requirements can be successfully built and applied in practice, the prototypical knowledge conservation system KONUS [2] was developed. The expected benefits of KONUS are twofold: By answering questions about previously designed crankshafts KONUS relieves the human expert from some of his routine tasks and enhances the availability of his expertise. By also supporting the design of new crankshafts, the refinement of designs can now be done more quickly and possibly also by less experienced designers. This should give the crankshaft specialist more time for the conception of innovative designs which will give the company an edge on the market.

3 Development and Architecture of the KONUS System

3.1 Knowledge Acquisition

Since our general approach to knowledge acquisition has been described elsewhere [**?**], we will mention here only those aspects which are of particular importance for building a knowledge conservation system.

The primary source of information was a young mechanical engineer with three years of experience in crankshaft design. Technical drawings of previously designed crankshafts were used in some of the interviews for illustration purposes, and mechanical engineering textbooks (e.g. on combustion engines and crankdrives) were only used by the knowledge

[2]KONUS is a German acronym for conservation and construction support.

engineer in order to get an initial understanding of the domain.

Contrary to knowledge acquisition for a traditional expert system the focus was not on a problem solving method or model of expertise [?] but on a sort of domain model which was based on the analysis of the information flow between the crankshaft expert and his colleagues (see Fig. 1). Since the to be developed system does not have to solve a major task automatically, it is not essential that the knowledge base be complete and neither are redundancies disastrous. The knowledge base can thus more or less contain exactly that knowledge which is considered important by the expert.

In the performed interviews, the expert mentioned various constraints which should be met by a good design, gave justifications for some of these constraints, and made some suggestions about the strategy which should be pursued when designing a new crankshaft. All this knowledge was entered to the KONUS system, since it is obviously relevant for the expert's everyday work and is thus supposedly also worth being conserved and made available to his colleagues.

Since the primary function of the stored knowledge consists in enhancing communication, particular care must be taken so that both the structure of the entire knowledge base as well as the contents of individual knowledge items are transparent to the intended users. In order to achieve this, the structure of the KB was discussed with the expert and an explanation of all potentially ambiguous terms and non-evident knowledge items was included.

3.2 Knowledge Representation

The primary purpose of a knowledge conservation system is to make captured knowledge available so that it can be communicated to advice-seeking users. On the other hand, it should also be amenable to automatic problem solving in order to give the users better support in the solution of routine tasks. These two requirements cannot be met by a single knowledge representation scheme, since there is a well-known trade-off between the cognitive and computational adequacy of any knowledge representation formalism.

In the KONUS system, the same information is therefore represented at different levels, which are either more cognitively or more computationally adequate. In sequence of decreasing cognitive adequacy, the following four knowledge representation levels are distinguished (see Figure 2):

The text level: At this level the knowledge is represented as natural language text, basically as extracted from the transcriptions of the expert interviews. A considerable amount of knowledge is only represented at this level, but gets linked to formal knowledge items to which it provides explanations and justifications.

The user level: This is the level at which those knowledge items (e.g. design rules) which can be automatically processed are presented to the user. In order to make the logical content and the system's interpretation of the knowledge items obvious to the user without bothering him with an impenetrable syntax, the knowledge is displayed in a graphical window, with cryptic symbol chains being translated into partial sentences (phrases), and the logical structure made visible by familiar graphical means, such as highlighting and indenting.

The storage level: This level contains the representations which are stored on disk and which can be easily mapped both to the user and to the processing level. It contains also all the informal documentation for the different types of knowledge items. Only the part of this information which is relevant for performing inferences is translated into a representation at the processing level. The code at this level is quite easily readable for a knowledge engineer, and in fact the knowledge was entered at this level, since a tool for knowledge entry at the user level is only now being developed as part of the KONUS system.

The processing level: This level contains efficiently executable knowledge items which can be readily employed for performing the different kinds of inferences, thus enabling short response times.

Figure 2 shows the representation of a typical design rule at the four different levels. The user level representation is part of the graphical interface which is implemented in TCL/TK. The storage level employs a LISP syntax with the informal documentation attached as strings. The processing level shows the design constraint without its annotation as a bidirectional rule in the knowledge representation formalism CoLab [**?**].

The knowledge base contains four different types of knowledge (see figure 3) which are represented as follows.

The collection of previous designs (or case base) contains descriptions of crankshafts and their components (e.g. connecting rod). Each component is described by attribute value pairs. Contrary to the technical drawings of the crankshafts and their components which are already available in the company, the attribute value representation provides a more abstract and mostly qualitative description (design features) which plays a central role in the experts reasoning when solving a design problem.

The relevant reference objects such as materials and bearings are also described by attribute value pairs. Even though they are not designed but selected, their properties must be taken into account in the design of the crankshaft components.

The concept definitions and explanations can be regarded as an ontology of all relevant terms together with their specifications and additional information like the mentioned informal documentation. The following example shows how a term occurring as an attribute in a case description is defined and annotated at the storage level:

```
(SYMBOL     thick-hard-layer
 USERSTRING "Thickness of hardened layer"
 TEXT       "The thickness of the hardened layer of the crankshaft
             component on the whole surface except the covered parts."
 USE        *attribute-name*
 VALUE-TYPE pos-real
```

```
 UNIT       "mm"
 RELEVANCE  (heat-treatment = case-hardening)
)
```

SYMBOL specifies a valid LISP symbol which can be used at the processing level. The USERSTRING is displayed in the graphical user interface and should be sufficiently understandable to a mechanical engineer. TEXT serves as an additional explanation which can be shown on demand to clarify the meaning of the term. USE indicates whether the term is used as a concept or an attribute and may provide some other relevant information.

The design rules contain general criteria for a good design which are the guidelines for the development of new crankshafts and constitute the up-to-date knowledge of the company's know-how with respect to construction, cost-reduction, manufacturing etc. On the processing level they are implemented as bidirectional rules (see figure 2), which can be used both bottom-up with forward chaining (for suggesting alternatives) and top-down with backward chaining (for finding explanations).

Two types of rules are used in our application: strict constraints which should always be met and recommendations which may occasionally be violated. Every rule has a name by which the processing level can pass information to the storage level and vice versa. The symbols marked with a "#" at the end in figure 2 indicate references to other objects such as the connecting rod or a bearing. On the processing level these references are resolved by the means of variables and coreferencing. Attribute value pairs are represented as binary predicates with the attribute name as the predicate name and the involved object and the value as arguments.

Not all design rules are as simple as the one shown in the example. Other types of design rules which are allowed in KONUS comprise: rules without premisses, rules with multiple conclusions, rules with a disjunction of attribute values both in conclusions and premisses, rules with relations between an attribute and a value, rules with relations between two attributes. Such complex rules are included in KONUS since they occasionally occurred in the expert interviews. We felt, that in a knowledge conservation system the expert should be allowed to state arbitrary chunks of knowledge he deems adequate. Such complex rules can sometimes be broken down into several simple rules at the processing level, but at the higher levels meaningful units of knowledge should always be maintained, even if this causes some computational problems.

We had originally intended the knowledge base to also include **design strategy rules** which contain suggestions for the sequence in which the various attribute values should be specified when designing a new crankshaft. Since only a few such rules were found in the expert interviews, they would be hardly of any practical use, and they are thus not included in the current KONUS system.

3.3 Overview of the System Architecture

The global architecture of the KONUS system is shown in figure 3. The knowledge captured in the knowledge base is used to provide a wide range of services to various users who interact with the KONUS system via a graphical user interface.

The functionalities offered by the KONUS knowledge conservation system for crankshaft design can be grouped into three categories which were named according to [?]:

The design librarian supports the management and evolution of the design knowledge by providing functionalities for knowledge retrieval, modification and validation.

The design informer answers different types of questions about previous designs which are stored in the knowledge base. It relieves the crankshaft expert from responding to routine queries, and reduces the information bottleneck by making the expertise about crankshafts better available.

The design aide provides direct support for the construction and modification of crankshafts, so that this activity can now also be performed by engineers who are not dedicated crankshaft specialists.

The functions of the design librarian, design informer, and of the design aide which are marked by white boxes in figure 3 were implemented first, since our partner considered them most important for a demonstration of the benefits of a knowledge conservation system. These functions will be described in more detail in the next section of the paper.

Even though shown in three distinct columns, the design librarian, design informer, and design aide are embedded in a uniform graphical user interface which makes the functionality of each component available from the others, and also supports a concurrent utilization and evolution of the captured knowledge.

4 Using the KONUS System

4.1 Using the Design Informer

As indicated in figure 3, the design informer supports three basic services which provide information about a given design. Each of these services is invoked, by asking a question about a particular attribute (or attribute value pair) of a crankshaft, a crankshaft component, or a reference object.

First of all, the design informer does **provide relevant information** by supplying explanations of attribute and value descriptors. Such explanations may be needed by some users, since it is hardly possible to find reasonably short (for the sake of conciseness) and self-explanatory description terms. Besides giving these explanations which just consist in displaying the associated explanation texts, the design informer can also provide relevant information which is encoded in design rules. A rather general question returns all design rules which impose some restriction for the value of a particular attribute. Such a question can be read as: *What must be taken into account when determining the value of this attribute (e.g. the type of main end bearing)?* By showing the user all knowledge which the system has about a particular topic, the users may assess the competence of the system and gain confidence into its suggestions. The design informer can also be asked a more specific question which returns only those design rules which apply in the given problem context (e.g. the rules for determining the type of main end bearing in high-power machines which use forged connecting rods, etc.). In order to identify these rules, the design informer just has to check whether all premisses of the rules referring to the questioned attribute are satisfied by the given case data.

One of the core functions of the design informer, is its ability to **explain a given solution** by giving (plausible) reasons of "Why" a particular attribute value was selected (e.g. *Why is the main end bearing a caged needle bearing?*). In order to answer such a question, KONUS searches all design rules which are applicable in the given case and either support or contradict the questioned attribute value combination. KONUS then informs the user of how many design rules argue pro the given solution, and displays these rules on request. If all applicable rules contradict the given solution, KONUS informs the user that in fact a different solution seems more adequate, and again displays the respective rules upon request. The answering of "Why" questions is analyzed in more detail in [**?**]. It should only be noted here that KONUS takes a completely different approach to explanation than traditional expert systems. Whereas the latter rely on the solution trace for constructing an explanation, KONUS adopts a purely reconstructive explanation philosophy [**?**]. This approach offers the advantage that a plausible explanation for a proposed solution can also be given even if the system lacks complete knowledge for deriving the solution or if its knowledge is partially contradictory.

The design informer's ability to **check alternative solutions** is of great practical importance. As previously mentioned, requests to revise a given design decision in favor of a supposedly more favorable alternative solution are quite common in industrial practice. A typical question might be: *Why should a cheaper cageless roller bearing not be used instead of the given caged needle bearing?* In order to answer such a question, the design informer gives the user a rather detailed account of how many applicable design rules support both, only the original, only the suggested alternative, or none of the two solutions. Furthermore, the user is told how many rules which are currently not applicable would support the suggested alternative. By inspecting the different sets of rules (usually most sets are empty or contain only one or two rules) the user may assess whether the proposed alternative has to be ruled out, or can be adopted, if appropriate other modifications are made. The latter can be determined from those rules which are currently not applicable but do support the newly suggested solution.

A screendump of a sample session with the design informer is shown in figure 4. The left window shows the global crankshaft data together with a sketch of a crankshaft which allows to access the descriptions of individual components by a mouse click. The respective data are then displayed in a separate window, as shown on the right side of the figure. When clicking on an attribute, a menu pops up which offers the different questions which can be asked about the particular attribute value pair. The answer to the question is presented in a new window, which pops up and offers the user the option to request additional information.

4.2 Using the Design Aide

The users' interaction with the design aide is very similar to the interaction with the design informer. All functions of the design informer can also be accessed from within the design aide, since they may be useful designing or modifying a crankshaft. Contrary to the design informer which opens a case description in a "read only" mode, the design aide allows the modification of attribute values (or the filling in of unspecified ones), and supports this activity by offering some additional services.

Since many attribute values are qualitative with a limited range of possible values, the desired value can be selected from a box of possible values which is popped up, if the user clicks on an icon next to the attribute. The selected value is then automatically entered into the respective field.

Whenever an attribute value is changed, the design aide verifies whether some design rules have been violated and provides the user an automatic **critique** if this is the case. This constant monitoring of the completion or modification of the new design was explicitly requested by the crankshaft expert and considered to be very useful. Besides this automatic critique which only reports rule violations, the user can also request an explicit critique which provides more detailed information about the design rules supporting or contradicting the proposed solution.

Instead of trying out different attribute values and waiting for critique, the user can request the design aide to **suggest possible values** for an attribute which are consistent with the already specified values of the other attributes. Again the user can inspect the rules suggesting particular values, so that he can make a reasonable choice if several possible values are suggested.

The design aide currently only informs the user, when an inconsistency between an entered attribute value and the stored design rules is detected. In a future version, the user will be prompted to resolve the conflict by taking one of the following actions:

1. Revise the criticized entry so that it is consistent with the design rules.

2. Update the design rules by calling the design librarian. Some rules may have become obsolete or need refinement. Such a deficiency in the knowledge base is fixed best, as soon as it has been detected.

3. Add an annotation to the criticized attribute value which explains why the general guidelines were violated in that particular circumstance. Such annotations would thus provide an explicit documentation of a design decision, which is now needed since the implicit documentation provided by the general design rules is no longer inadequate.

4.3 Using the Design Librarian

The design librarian is that component of the KONUS system which was implemented last and is not yet completed. We first wanted to demonstrate that the KONUS system can indeed provide useful support for crankshaft design, before bothering about supporting knowledge maintenance and evolution. Nevertheless knowledge evolution by the end user is of crucial importance for the success of the entire knowledge conservation system.

The basic goal of the design librarian is to enable the user to make additions and modifications to the design rules and to the concept definitions. Such updates can only be managed by the user, if they can be performed at a knowledge representation level which is cognitively adequate. We expect that the previously described user level will be sufficient for this task (or at least with some minor user training).

The design librarian thus provides a structure editor for design rules, which allows the user to add and delete premisses and conclusions, to build attribute chains and value sets, all by menu selection and mouse clicks. The rule editor thus guarantees that the constructed rules are syntactically correct, and semantically at least not total nonsense, since only those items are offered for selection which might at least principally make sense.

Of course, such a design editor which allows only the construction of rules within a predefined vocabulary, will be of limited value, if the vocabulary itself cannot be modified and extended. The design librarian therefore will also provide a concept editor, which allows the addition and deletion of attributes to objects, the assignment of possible values to attributes, and the definition of new value types and individual values.

In the long run, the design librarian could be extended to provide all functionalities of a knowledge conservation system "shell", which could then be applied to a different domain.

5 Evaluation and Discussion

A prototype of the described KONUS system was recently delivered to our industrial partner. Their general reaction was, that a system which offers the functionalities of KONUS might be very useful in overcoming the information bottleneck which often delays the development and enhancement of products. It was noted in particular, that KONUS could indeed provide adequate and comprehensive answers to many questions which arise in everyday practice. This is mainly due to the combination of natural language explanations together with formal knowledge items which can be automatically processed, but can also be presented to the user in an intelligible form.

On the other hand, it became also obvious that the development and maintenance of a practically useful knowledge conservation system will require considerable investments of which the ultimate payoff cannot be foreseen. The current prototype, which took about one manyear to build, will need considerable qualitative and quantitative enhancements in order to become a usable and useful system. Whereas the integration with the company specific hard and software environment (in particular databases) is mostly just a technical problem, one critical difficulty is the integration with the currently used CAD system. It offers too unstructured and too low-level data to make an automatic extraction of those high-level features feasible which are referred to in the design rules.

Furthermore, KONUS will need to encompass much more knowledge in order to attain that critical mass which makes users want to use and enhance a computer system [?]. Whether this knowledge can be entered and maintained by the intended users with the help of the design librarian is currently being investigated. Based on some initial feed-back, this system component is now being improved and completed. Whereas the KONUS prototype clearly showed the potential usefulness of a full-blown knowledge conservation system, the question of the ultimate payoff of such a system, can only be answered by a detailed requirements analysis and feasibility study which will be conducted in close cooperation with our industrial partner. One central goal of this study will be the identification of a relatively small aspect of the knowledge conservation problem for which the actual benefits of a computer system can be demonstrated most convincingly.

Although tailored to a specific application, the described KONUS system owes much to other systems which have been described in the literature. The closest relatives of KONUS are perhaps the knowledge based systems developed in the working group of Gerhard Fischer at the University of Colorado, the best known of which is probably the JANUS kitchen configuration system [?]. This system integrates Hypertext with a knowledge-based design environment for the support of cooperative problem solving. It lets the users determine the course of action, only providing critique [?], and making suggestions. The conjoint utilization and evolution of a knowledge base has also been suggested by this working group as well as by several authors within the Hypertext community (e.g. [?]). The overall architecture of the KONUS system and its organization of the knowledge base which includes a domain ontology together with explanations of the employed terms was inspired by recent suggestions for knowledge sharing and reuse [?].

Acknowledgement

This research was done in cooperation between an industrial company and the DFKI projects ARC-TEC (supported by grant FKZ ITW-8902 C4 from the German Federal Ministry for Research and Technology) and IMCOD/VEGA (grant 413-5839-ITW9304/3). The authors wish to thank the crankshaft expert and his colleagues who showed much patience during knowledge acquisition and helped us understand the real requirements for a practically useful knowledge conservation system. Concerning the implementation of the system, we owe thanks to Volker Becker, Thomas Malik, and Michael Sintek, as well as the developers of the CoLab knowledge representation system.

Figure 1: Workload of the crankshaft expert and support provided by KONUS.

Figure 2: Levels of knowledge representation in KONUS.

Figure 3: Overview of the KONUS knowledge conservation system.

Figure 4: Using the design informer