The model-based construction
of a case-oriented expert system

Franz Schmalhofer, Jörg Thoben

June 1992

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 by the shareholder companies Daimler-Benz, IBM, Insiders, Fraunhofer Gesellschaft, GMD, Krupp-Atlas, Digital-Kienzle, Philips, Sema Group Systems, Siemens and Siemens-Nixdorf. 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
- ❑ Intelligent Communication Networks
- ❑ Intelligent Cooperative Systems.

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.

Prof. Dr. Gerhard Barth
Director

The model-based construction of a case-oriented expert system

Franz Schmalhofer, Jörg Thoben

The Model-Based Construction of a

Case-Oriented Expert System

Franz Schmalhofer & Jörg Thoben
German Research Center for Artificial Intelligence
University Bldg 57
Erwin-Schroedinger Str.
D-6750 Kaiserslautern

email: schmalho@informatik.uni-kl.de

Abstract. Second generation expert systems should be based upon an expert´s high level understanding of the application domain and upon specific real world experiences. By having an expert categorize different types of relevant experiences and their components, hierarchies of abstract problems and operator classes are determined on the basis of the expert´s accumulated problem solving experiences. The expert´s global understanding of the domain is integrated with the experiences by a model of expertise. This model postulates problem classes at different levels of abstractions and associated skeletal plans. During a consultation with the expert system previously unseen types of input may be used to delineate a new problem. The application of the expert system can thus be situated in changing environments and contexts. With increasing dissimilarity between the cases that were analyzed during knowledge acquisition and the specific problem that is processed at the time of the application of the system, its performance gracefully degrades by supplying a more and more abstract skeletal plan. More specifically, the search space which is represented by the skeletal plan increases until the competence of the system is exceeded. This paper describes how such a case-oriented expert system is developed for production planning in mechanical engineering.

1. Introduction

Previous expert systems research has clearly shown that model-based knowledge engineering such as the KADS (Breuker & Wielinga, 1989), the generic task (Chandrasekaran, 1986) or role-limiting approaches (McDermott, 1988) are very essential in order to meet the demands of second generation expert systems (Steels, 1987). Since the new generation expert systems must be easier to explain and easier to maintain than their first generation predecessors, they should be constructed according to some knowledge-level rationale (Newell, 1982), which can be expressed by a general problem solving model or model of expertise (Breuker & Wielinga, 1989) rather than being solely implemented as some prototypical system.

The recent successes of case-based approaches (Riesbeck & Schank, 1989) furthermore promoted the idea of applying case-based reasoning (CBR) in expert systems. Possibilities for developing such systems in a model-based fashion were for example discussed by Bartsch-Spörl (1991). Janetzko & Strube (in press) elaborate these suggestions and show how

conventional rule-based reasoning can be combined with case-based reasoning techniques by a specific turn taking procedure. The impetus of the CBR research on expert system developments comes mostly from the observation that case descriptions are situational memories (Schank, 1980, p. 260) about specific real world episodes.

Since the real world (unlike the microworld of hypothetical blocks, i.e. the blocksworld) refuses to be represented (once and for all times) by some general and formal specification, the future application situations of a system can only be partially predicted and the real world itself (with all its richness and surprises) must be allowed to enter new and unpredictable types of input into the expert system, at the time when the system is used to solve an application task. The user of the expert system would obviously provide some interpretation of the new types of input and would thus function as a filter. Rather than the expert system itself being situated, one may therefore better describe the user-system tandem to be situated (Pfeifer & Rademakers, 1991). Since the expert system is situated according to the user's perceptions, the system itself may be seen as a cognitive tool for processing the perceptions of its user (Schmalhofer, 1987).

Real world experiences are often good predictors for similar real world situations in the future (Riesbeck & Schank, 1989; chap. 2), especially when they are interpreted according to the perception of human experts. Human experts have highly developed perceptual abilities (Shanteau, 1984; 1988). They are able to extract information that novices (as well as uninformed machine learning programs) either overlook or are unable to see (Chase & Simon, 1973). When novices are given already extracted information, however, they are often capable of making decisions that are nearly as good as experts. When computer systems are given the same information, they may perform equally well.

The difference is that experts actually seem to be able to see or evaluate what others cannot. For some experts this difference is primarily perceptual, for others it appears to be more attentional. In either case, experts have developed conceptual systems for extracting information which are superior to novices and they have the ability to simplify what to novices appear to be highly complex problems. " An expert is someone who can make sense out of chaos. ... Thus they have an enhanced ability to get to the crux of the problem. " (Shanteau, 1984). In other words, experts form the right kinds of abstractions. Such conceptual systems of human experts may at least partially be formed through the nonanalytic-automatic abstraction of concepts (Kellogg & Bourne, 1989).

We are therefore proposing to develop expert systems that are case-oriented: A set of prototypical cases from the real world is analyzed according to a human expert´s abstractions, so that future problems can be associated to a more or less specific or abstract class and processed accordingly. Similar to case based reasoning itself, previous solutions or their abstractions can thus be reused when (according to the user's perception) the new problem is related to the previous case in a concrete or abstract way.

In the next section, we will present a general overview of case-oriented expert systems and how the integrated knowledge acquisition method for analyzing different information sources (Schmalhofer, Kühn, & Schmidt, 1991a) is applied for constructing such systems. This overview is based on the knowledge acquisition research which was conducted within the ARC-TEC project (Bernardi et al. 1991) over the last two years. This project is concerned with solving planning problems in technical domains, such as mechanical engineering. In the third section we will describe the specific mechanical engineering application. In section four, the generation and modification of general production plans by a human expert will be presented. In the conclusion of the paper, the situated application of a case-oriented expert system will be discussed and compared to competing approaches.

2. Description of Case-Oriented Expert Systems

When stated as a search problem the tasks of an expert system are typically intractable. This is certainly true for synthetic tasks such as planning in a complex real world domain (Georgeff 1987). Human experts can, nevertheless, handle such tasks quite well. Human experts solve complex real world tasks by relying on chunked problem solving experiences (Laird, Rosenbloom & Newell, 1984), which have been indexed by the respective problem descriptions. Human experts have consequently been found to classify problem descriptions according to their solution method (Chi, Feltovich & Glaser, 1981). The different problem classes may furthermore be hierarchically organized (Chase & Ericsson, 1982).

Despite of any intractability one may still delineate different problem classes, so that a solution can at least be found for all those problems, which belong to one of these classes. When such a problem classification is based upon the experiences and the high level understanding of a human expert, a powerful expert system may be developed. When the problem classification is additionally performed on the basis of real world episodes, the system development itself may be said to be situated in the real world (Suchman, 1987).

The search space for finding a solution can be defined by a skeletal plan, that is associated to each class (Bergmann, in press; Schmalhofer, Bergmann, Kühn & Schmidt, 1991b). A skeletal plan is a complete or possibly only partially ordered sequence of abstract operators, which when appropriately specialized will solve the problem at hand (Friedland & Iwasaki, 1985).

The problem classes can furthermore be used to define the competence of the expert system: The tasks which do not belong to any problem class can be said to lie outside the area of competence of the expert system. Supposedly, such problems would only arise rather infrequently. For problems, that fall into a more abstract class, the system would be considered less competent than for problems that belong to a more specific class.

2.1 A General Model of Expertise for Case-Oriented Expert Systems

For developing a model of expertise for planning or synthesis tasks (Breuker & Wielinga, 1989, p. 286), it is thus proposed, that real world episodes are grouped into classes, so that according to an expert´s high level understanding a uniform rationale about finding a solution can be associated with each class. Such uniform rationales are represented by skeletal plans and related domain knowledge. Since the different problem classes are furthermore partially ordered by an assumed abstraction hierarchy (Knoblock, 1990), the theory about finding a solution for one set of problems is nested within more abstract problem classes.

The lower part of Figure 1 outlines how a case-oriented expert system would process planning tasks: A problem description consisting of the specification of the available *environment*, the relevant *context*, the *initial state* and the *desired state* to be produced by the expert system is classified into the *most specific problem class* that subsumes the given problem description. The associated *skeletal plan* is then retrieved and further *domain knowledge* is used to refine the skeletal plan to a *concrete plan* for accomplishing the transition from the initial to the desired state. The task- and inference structures of this system are described in more detail by Kühn & Schmalhofer (1992).

Case-oriented expert systems may be constructed with a previously developed integrated knowledge acquisition method (Schmalhofer et. al., 1991a) and a set of coordinated knowledge acquisition tools: The Case Experience Combination System or CECoS (Bergmann & Schmalhofer, 1991), the Case-Oriented Knowledge Acquisition Method from Text or COKAM+ (Schmidt & Schmalhofer, 1990; Schmalhofer & Schmidt, 1991) and the Skeletal Plan Generation Procedure or SP-GEN (Bergmann, in press; Schmalhofer, et. al.,

1991b). The three knowledge acquisition tools are coordinated on the basis of the structure of the expert system which is represented by the model of expertise.

The input-output relations among the different knowledge acquisition tools are shown in the upper part of Figure 1. CECoS elicits the expert´s *high level judgements* about a number of *real world episodes or cases* so that an *abstraction hierarchy of problem classes* can be constructed. For the different *operators, which are contained in the specific plans* and related documentation (*text*), an *operator abstraction hierarchy and abstraction and refinement rules* are constructed with COKAM+, by which the operators at the different levels of the hierarchy are related to one another. SP-GEN takes *one or several cases as input* and constructs *skeletal plans* with the domain knowledge for all those problem classes, which subsume the specific cases.
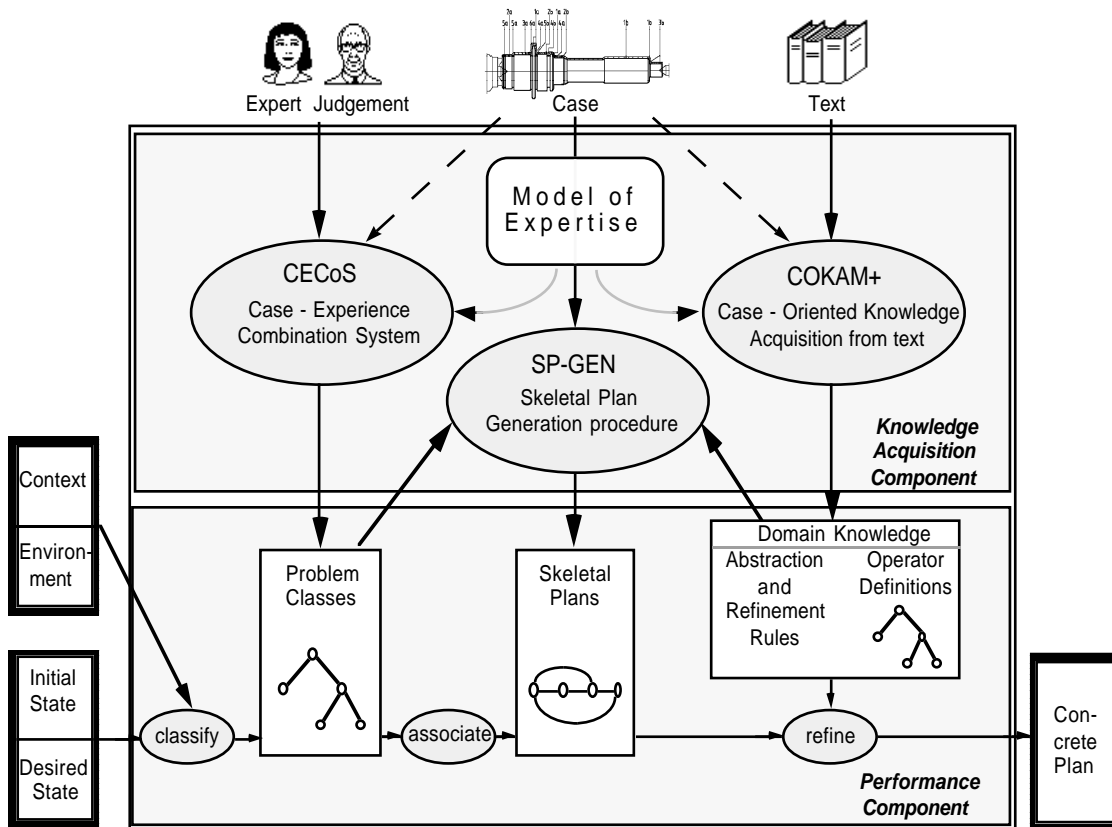

Figure 1: General structure of the knowledge acquisition and performance components of a case-oriented expert system for situated planning tasks.

2.2 Using Cases to Define an Abstraction Hierarchy of Problem Classes

Real world cases consist of authentic problems and successfully applied plans. In order to delineate the desired competence of a future expert system, those cases which follow some common rationale are selected. In order to delineate the desired area of competence the selected cases must be sufficiently different. When there are not enough complete cases available problem descriptions without a solution may be used in addition.

Each case or problem description is viewed as an instance of some specific problem class called terminal problem class. By combining terminal problem classes, larger and more abstract problem classes are obtained. The problem classes are thus partially ordered by an abstraction hierarchy. Since a more abstract problem class must at least subsume two less

4

abstract classes, the maximum number of non-terminal (or abstract) classes is equal to the number of internal nodes in a full binary tree: For n problem descriptions, there can at most be n terminal and (n-1) abstract problem classes.

After the various concrete and abstract problem classes have been delineated by representative problem descriptions (extensional delineation), an intensional definition of each class is additionally constructed in the terms (terminology) of the respective application domain. The knowledge acquisition tool CECoS supports experts in transferring their problem categorizations to a computer system according to these principles. CECoS was described in more detail by Bergmann & Schmalhofer (1991), Tschaitschian (1990) and Reinartz (1991).

2.3 Defining an Operator Abstraction Hierarchy from the Operators of the Representative Plans

In order to construct skeletal plans for the various problem classes, the operators which occur in the respective plans must also be described at different levels of abstraction. The operators may be described in terms of preconditions for their application and the resulting consequences in a STRIPS-like notation. For at least two distinguished levels of this abstraction hierarchy (i.e. a concrete and an abstract level) the STRIPS-assumption (Georgeff, 1987) is postulated to hold. For obtaining such operator descriptions from the written documents (texts) which are more or less related to the specific cases under consideration, the knowledge acquisition tool COKAM+ has been developed. COKAM+ furthermore relates the operator descriptions to the model of expertise, which guides the development of the expert system. COKAM+ has been described by Schmidt & Schmalhofer (1990) and Schmalhofer & Schmidt (1991). Kühn, Linster & Schmidt (1991) investigated the construction of domain knowledge about clamping operators with COKAM and their representation language OMOS (Linster, 1992).

2.4 The Construction of Skeletal Plans for the Different Problem Classes

With the hierarchy of problem classes resulting from CECoS and the hierarchy of operator descriptions resulting from COKAM+, a skeletal plan can be constructed for each problem class. The skeletal plans of terminal problem classes are obtained by the explanation-based generalization procedure SP-GEN. For the more abstract problem classes, similar knowledge-based learning mechanisms may be applied for forming the appropriate abstractions. Abstractions require the transition from one description language to a more abstract language (Korf, 1987).

In order to apply SP-GEN, at least one specific plan of a problem class must be available. SP-GEN basically analyzes the dependencies (or protection intervals) between the operators of the concrete plan in terms of the more abstract operator descriptions. The most significant dependencies are represented at the abstract level by a dependency graph together with respective abstract operators. A prototypical implementation of SP-GEN was presented by Bergmann (in press).

The comprehensive overview of the described knowledge acquisition and performance components in Figure 1 shows that cases play a central role in the knowledge acquisition phase: Cases from the real world are used to elicit the expert´s high level understanding. The operators contained in those cases are similarly employed to elicit operator abstraction hierarchies and other domain knowledge from text. Finally skeletal plans are abstracted from the specific cases. The analysis of cases is always guided by the specific model of expertise.

3. An Expert System for the Production Planning in Mechanical Engineering

We will now describe the progress we have made in the construction of a case-oriented expert system for production planning in mechanical engineering. Its application domain is the manufacturing of rotational parts from different work piece materials where one of several different lathe machines may be used. This domain will first be outlined, before abstraction hierarchies and the generation of skeletal plans are discussed.

3.1. Description of the Application Domain

The technique for manufacturing a rotational part is best understood by a comparison to pottery. To manufacture a pot one puts or attaches a piece of clay to a potter´s wheel and shapes the clay to a specific form, *only by removing* some parts of the clay while the potter´s wheel is turned. Contrary to the soft clay, which also allows a potter to push some material to a neighboring position, a rotational part or workpiece (metals) is shaped, *solely by removing materials with a hard cutting tool*.

Section a) of Figure 2 graphically represents a workplan for a rotational part. The geometric form of the mold and the target workpiece are overlayed and shown in the middle of the top part of the figure. During the manufacturing process the chucking fixture (seen as the black area on the left and the black triangle on the right side) is rotated with the attached mold (a 500 mm long cylinder indicated by the shaded area) with the longitudinal axis of the cylinder as the rotation center. The sequence of cuts are indicated by the numbers 1 to 7. For each cut the cutting tool, the cutting parameters, and the cutting path are also shown in the figure. For example, the cutting tool number 1 has the specification "CSSNL 3232 C15 SNGN151016 TO 3030". It is applied to remove a part of the upper layer of the cylinder with a rotation speed of v_c = 450 m / min, a feed of f = 0,45 mm/U and a cutting depth of a_p = 5 mm. A complete description of the real world operations would also include further technological data of the workpiece (surface roughness, material, etc.) and precise workshop data (CNC machines with their rotation power and number of tools and revolvers, etc.).

The production plan must fit the specific CNC machine which is used for manufacturing the workpiece. For each company the CNC machines are individually configured from a set of different components. The configuration of a machine depends on the spectrum of workpieces and the lot size which the company expects to produce. Therefore, hardly any two lathe machines of a company are completely identical.

There are a number of interdependencies between the tools, the CNC machines and the workpieces to be produced. CNC machines must have a large enough revolver to keep all the necessary tools. In addition, the CNC machine must have enough power to achieve the required cutting speed and force for the operations specified in the plan.

It is therefore not surprising that 1 - 2 man months are invested by human experts to specify a production plan. Although the quality of the resulting plan may be very high, the planning processes themselves are not completely knowledge-based. Even the plans developed by the best experts must be tested by executing them in the real world rather than only simulating their execution. Through these situated plan executions several incremental improvements are usually obtained.

Besides the case m5 (statement of manufacturing problem and corresponding solution plan), which is shown in the upper half of Figure 2 (geometry 5), only four additional cases were originally available. These cases are denoted by m1 to m4 and the respective geometries together with chucking and cutting operations are outlined in the lower half of Figure 2 (geometry g1 to geometry g4).
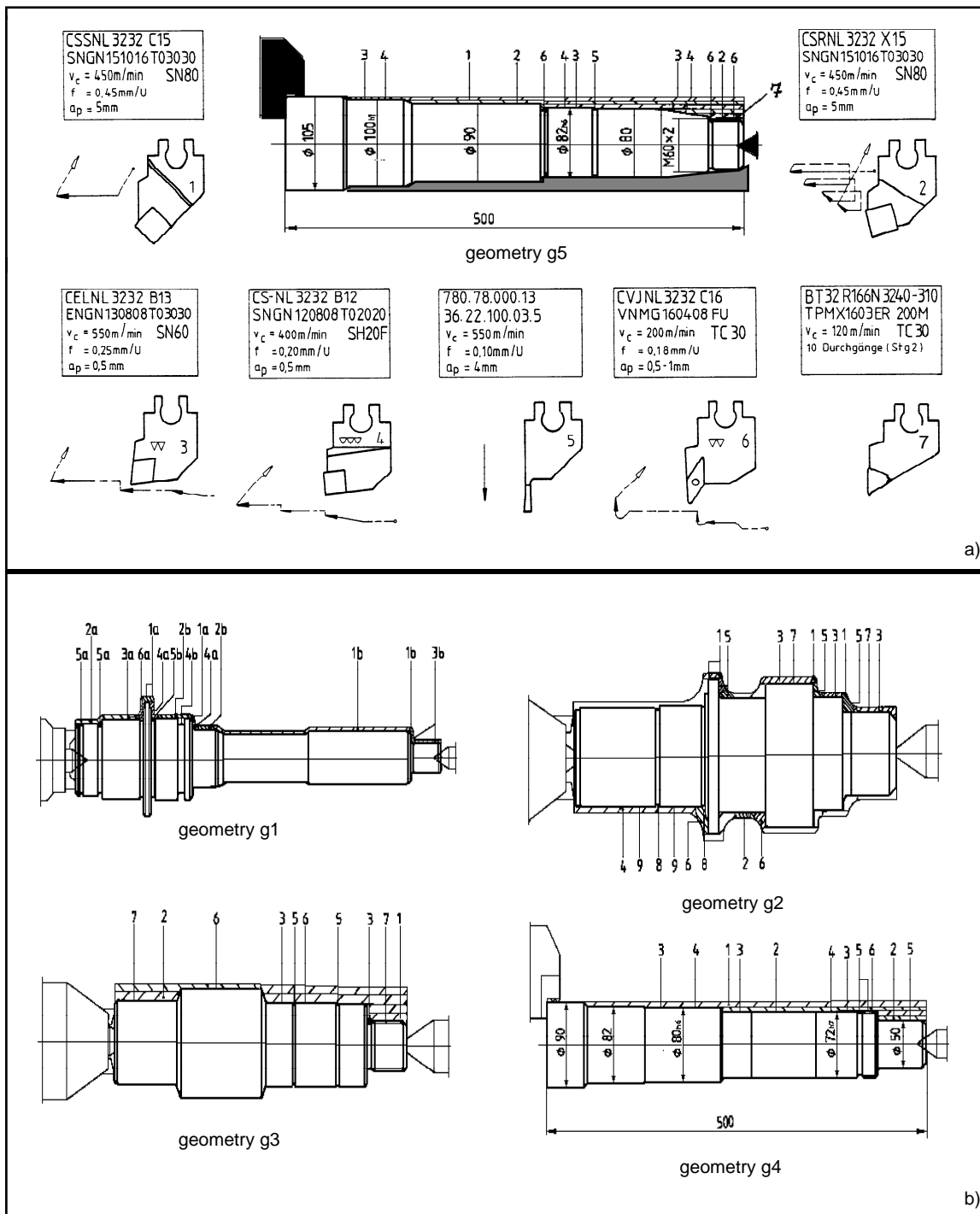
Figure 2: Section a) shows a graphical representation of a typical workplan (m5) for a rotational part and section b) shows four additional cases: m1, m2, m3 and m4. (From "Examples for Application" (pp. 25-27), Plochingen Neckar, Germany: Feldmühle AG. Copyright 1984 by Feldmühle AG. Reprinted by permission) .

The realm of competence for the desired expert system, on the other hand, should include three different manufacturing machines (lathes): A lathe at the low end of the performance spectrum (5,5 kW) with only one revolver and 4 tool holders (d1), a lathe (70 kW) with one revolver and 6 tool holders (d2), and a high performance lathe (90kW) with two revolvers and 12 tool holders (d3). In addition, one of four different work piece materials may be

requested: unalloyed steel (w1), a type of cast iron, namely GG25 (w2), a type of aluminum (w3), and alloyed steel (w4). A more precise description can be found in Reinartz (1991).

The system should be competent for problem classes, for which a representative set of instantiations is given by the factorial combination of 3 manufacturing machines (lathes), 5 geometries (two drive shafts: g1 and g2; one pinion shaft: g3; and two axle shafts: g4 and g5) and 4 workpiece materials. The system should thus be able to deal with the scope of problems denoted in Table 1.

Workpiece

Geometry	Drive Shaft g1				Drive Shaft g2				Pinion Shaft g3				Axle Shaft g4				Axle Shaft g5			
Workpiece material	w1	w2	w3	w4	w1	w2	w3	w4	w1	w2	w3	w4	w1	w2	w3	w4	w1	w2	w3	w4
d1					3	3	1		3 *	3 *	1 *	*	4	4	2		4	4	2	*
d2	5 *	5	7	9 *	5	5	7 *	9	m3 5	5 *	7	9	6	6	8	10	6	6	8 *	10
d3	m1 5 *	5 *	7	9 *	m2 5 *	5	7	9 *	5	5 *	7	9	m4 6	6	8	10	m5 6	6	8 *	10

(Left axis label: Manufacturing machines (lathes))

Table 1: A number of specific problems are used in order to delineate the competence of the future expert system. From the factorial combination of three types of manufacturing machines (d1,d2, and d3), and workpieces with five different types of geometries (g1, g2, g3, g4, and g5) and materials (w1, w2 w3, and w4) fifty-two problems were identified as meaningful. The numbers 1 to 10 refer to the terminal nodes (problem classes) shown in Panel a of Figure 3. Problems with associated plans are indicated by m1 to m5 and by asterisks. See text for further explanation.

3.2 Abstraction Hierarchies for Problem Classes and for Operator Classes

From the 60 potential problems, which are obtained by factorially combining 5 types of geometries with 4 workpiece materials and 3 different lathes, only 52 manufacturing problems are meaningful. An expert[1] grouped these 52 problems into 10 different classes, so that a skeletal plan would exist for each of them. The assignment of each problem to one of the ten classes is indicated by the numbers 1 to 10 (see Table 1). The blank cells of the matrix refer to the meaningless problems.

---

[1] We would like to thank Dipl. Ing. Ralph Legleitner for serving as the expert and Thomas Reinartz for playing the part of a knowledge engineer.
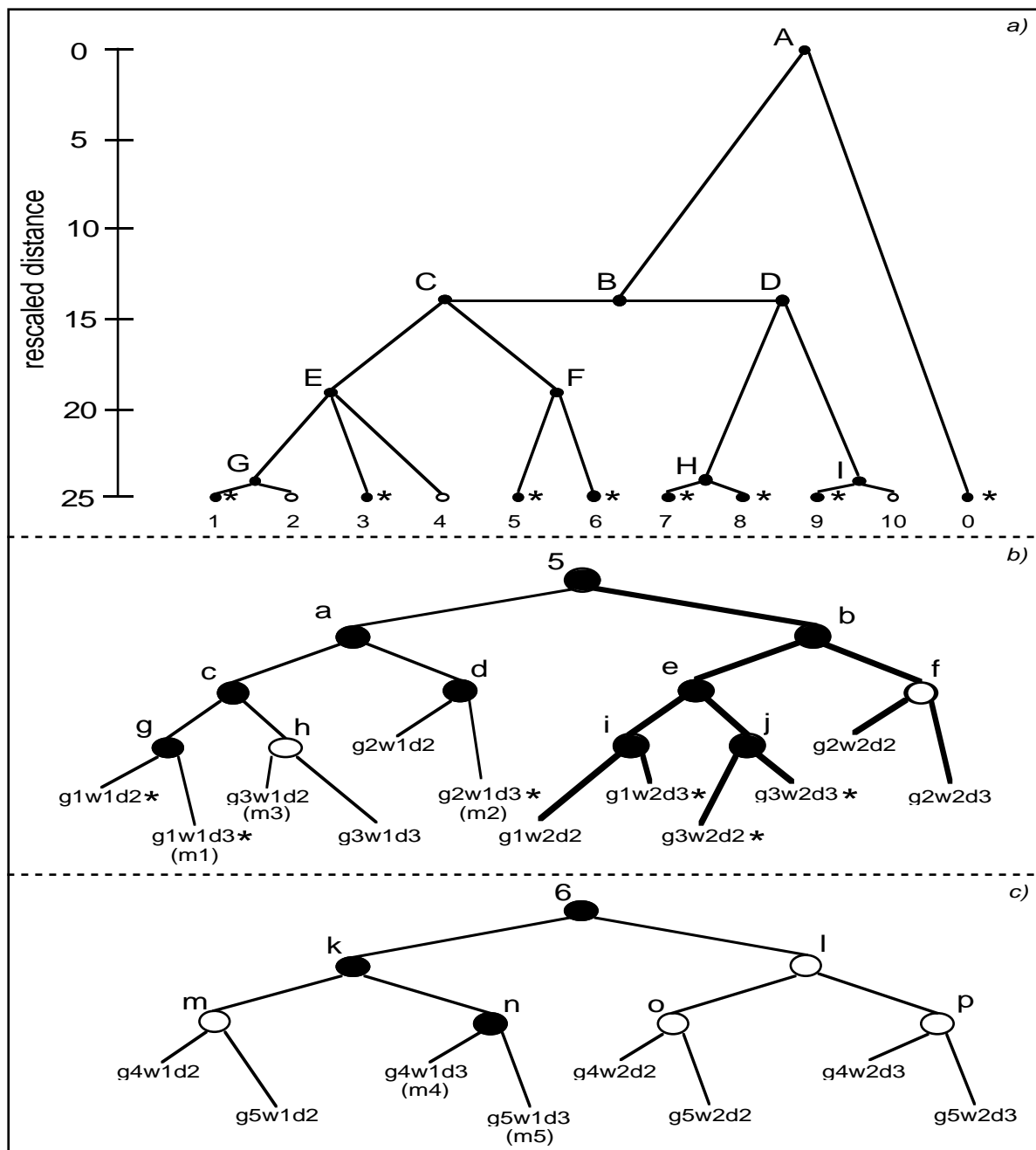
Figure 3: Panel a) shows the problem class abstraction hierarchy. The problem classes 5 and 6 are further divided into the subclasses shown in panel b) and c).

With a statistical procedure more abstract problem classes were identified by iteratively combing two or more of the ten original problem classes. Thereby the problem classes A to I and respective similarity values (rescaled distance) which are shown in the panel a of Figure 3 were obtained.

For each of the ten original problem classes (class 1 to class 10), the respective problem instances can be used to further subdivide each problem class into subclasses. As can be seen from panel b of Figure 3, the problem class 5 was subdivided into ten non-terminal subclasses (the subclasses a to j) and twelve terminal subclasses. The problem class 6 was similarly divided into six non-terminal subclasses (k to p) and eight terminal subclasses. How these

9

problem classes can be intensionally defined in mechanical engineering terms is described by Reinartz (1991) and Schmalhofer & Reinartz (1991).

In order to obtain a hierarchy of abstract operator classes the plans of the different cases are first parsed into their individual operators as well as into well established macro-operators (i.e. frequently occurring sequences of operators). From these operators and macro-operators, abstraction hierarchies are then determined similar to the abstraction hierarchies for the problem classes and STRIPS-like descriptions are constructed.

By the four-phase generation procedure SP-GEN skeletal plans can now be constructed for all those problem classes whose node is the root of a subtree with a node that contains an associated plan. In other words one of the leaves of the subtree must represent a complete case (problem <u>and</u> solution) rather than only a problem description without a solution. In Figure 3 the nodes with complete cases are indicated by the respective case numbers m1 to m5. It can easily be seen that there are not enough plans available to construct skeletal plans for all relevant problem classes.

With only five production plans (m1 to m5) being available for a total of possibly 103 different problem classes (52 terminal problem classes and a possibility for 51 non-terminal problem classes) skeletal plans can therefore only be constructed for very few problem classes. Therefore we requested a mechanical engineering student (hence also referred to as expert[1]) with extensive practical experience in the production planning of rotational parts to construct plans for several additional problems.

4. The Generation and Modification of Production Plans by a Human Expert.

While the expert could himself decide which planning problem he wanted to tackle next, he was requested to select the most appropriate old plan as a basis for constructing the new plan. The cases m1 to m5 as well as the plans which the expert would construct could be used as source cases. A knowledge engineer was employed to assist the expert in documenting and verifying the constructed plans by asking sensible questions. The whole investigation required a total of two months.

The expert formed production plans for 17 of the problems specified in Table 1, which are in detail documented in Thoben, Schmalhofer & Reinartz (1991). In Table 1 as well as in Figure 3, these 17 problems are marked by an asterisk. Fourteen of these plans were obtained by modifying one of the already existing plans. For one planning problem an old plan could be applied without any significant changes. Out of curiosity the expert also generated plans for two of the problems which were already classified as meaningless (g3w4d1 and g5w4d1). These problems were viewed as nonsense problems, because it is completely unpractical to produce these workpieces (g3w4 and g5w4) on the specific machine d1. For one of these problems the expert could not find any appropriate source case and consequently developed a plan from scratch.

The plans which were obtained by modifying an old plan were completed by an order of magnitude faster than plans which were produced from scratch. Figure 4 shows which plans were used as a source for the different target tasks. Task numbers t1 to t17 indicate the order in which the different problems were processed. The numbers and letters following the individual problem descriptions in parenthesis refer to the different problem classes (see Table 1 and Figure 3). As can be seen from Figure 4, tasks and their solutions were subsequently often used as a source case for a new task. The task numbers indicate that the expert frequently selected the next task, so that the most recently completed task could be

---

[1] We would like to thank Hans-Werner Höper for his work as expert.

used as a source. Whereas case m5 was used five times as a source, the cases m1, m2, and m4 were each used as source only once.
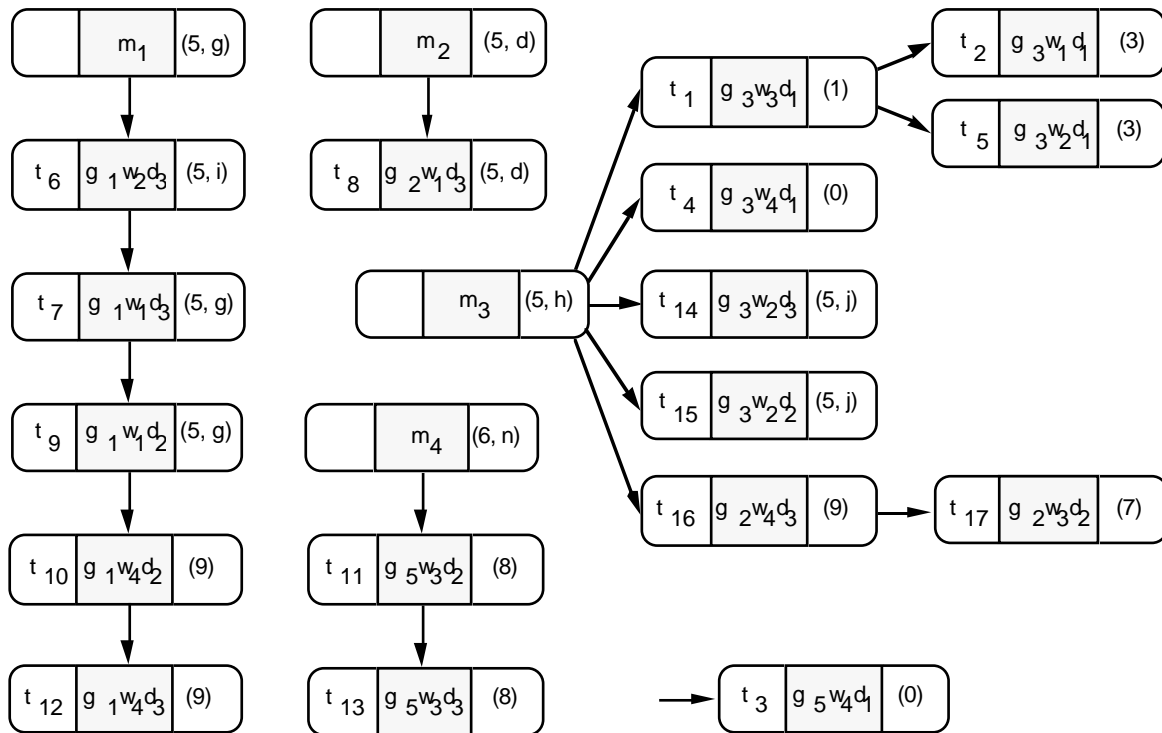

Figure 4: The source case - target case relation is shown for the 17 tasks (t1 to t17) which were solved by the expert. In parentheses the more or less specific problem classes that a specific task is associated with (see Figure 3) are noted. Task t3 was solved from scratch, so that there is no source case associated with it.

The modifications performed on the source plans ranged from changing only parameters (e.g. the cutting speed) to modifying and reordering operators and introducing or deleting operators. Often, the performed modifications were quite systematic. For example when a manufacturing machine with two revolvers (d3) was to be used instead of one with a single revolver (d2), sequences of operations could be performed in parallel instead of being serially concatenated and only one chucking was required instead of two. This systematic modification occurred with g1w4d2 as a source and g1w4d3 as a target.

Even when source and target problems were far apart according to the problem class hierarchy (see Figure 3) quite systematic changes occurred in the modifications. A change in the workpiece material (from g1w1d2 of problem class 5 to g1w4d2 of problem class 9) resulted in the use of a different cutting material, which in addition allowed to eliminate two previously required pre-cuts.

In another situation where source and target tasks were also far apart in the problem class hierarchy, quite different changes were performed: Besides changing the cutting tools and their parameters, an additional chucking had to be introduced and the various operations were reordered in a relatively unsystematic way. Since the geometry of the workpiece was identical in the source and target problems, this situation demonstrates that the workpiece material and the lathe machine have a large influence on the production plan. Production planning systems which rely exclusively on the geometry of the workpiece are therefore severely limited in their practical usefulness.

With the additional 17 plans it will now be possible to construct skeletal plans for the most relevant problem classes. For the terminal problem classes, a skeletal plan can be constructed from each of the 17 plans by simulating the plan execution with the operator descriptions represented in the operator abstraction hierarchy and subsequently constructing a dependency graph.Whereas the skeletal plans represent the most important dependencies among the various operations by a dependency graph, the expert did, however, not construct such a dependency graph (because he was not asked to do so). In Figure 3 all those problem classes for which a skeletal plan can now be constructed are marked by a black node. In panel b, black links combine those problem classes for which an intensional definition has been performed (Reinartz, 1991). More details about the refitting of the plans for the different problems are reported by Thoben, Schmalhofer, & Reinartz (1991).

5. Conclusions

First generation expert systems such as MYCIN were extremely competent for selected tasks (Harmon & King, 1985; p. 21), where they were even better than the human experts. Nevertheless, only very few of these systems were also a commercial success like for example XCON/R1 (Bachant & McDermott, 1984). Since they failed completely under those application situations, which were not foreseen by their designers, using such a system could have severe drawbacks for a professional (e.g. a physician).

Actually, there will always be application conditions which cannot be foreseen by the designers of a system (Clancey, 1989). Therefore, the knowledge (Newell, 1982) which is encoded in an expert system at the time of its development cannot completely prescribe the specific actions for all future tasks. Instead, we should look at the encoded knowledge as rational descriptions (Anderson, 1990) of past real world episodes, which are pertinent for solving future tasks. How and to what degree one can best represent knowledge in an expert system so that it can be successfully applied for future tasks thus becomes a very important question.

In the introduction we have already pointed out, that human experts have developed conceptual systems that provide them with an enhanced ability to get at the crux of (past, current and future) real world problems. We have therefore proposed to utilize an expert's high level understanding of the application domain for bootstraping a knowledge based system.

With some low-level descriptive knowledge, real world episodes can be represented as cases. For technical domains, which mostly consist of artifacts, this transition is relatively straight forward. For domains like biology or medicine, special knowledge acquisition tools may be needed for obtaining adequate case descriptions from real world episodes (Manago, Conruyt & Lerenard, in press). An expert's goal-driven similarity judgements about such representative cases may then be used to re(construct) an expert´s abstraction hierarchy of problem classes.

Methods from knowledge engineering like KADS-models, machine learning (e.g. Mitchell et al., 1986), or case-based reasoning tools, are only used for supporting the domain experts in verifying their concrete and abstract concepts. For synthetic tasks like planning these concepts consist of hierarchically structured problem classes and associated skeletal plans. The expert will thus play the most central role in the knowledge acquisition loop of the system (Wilkins, 1991). There are similarities as well as differences between such systems and CBR-systems. We will therefore compare their knowledge-base organizations, the refitting of plans and how rules and cases can be combined in these systems.

5.1 Knowledge-Base Organization

In case-oriented systems an expert's accumulated situational memories (Schmalhofer & Glavanov, 1986; Wharton & Kintsch, 1991) are used to determine general problem classes and a quite abstract model of expertise. Since an expert's situational memories contain information about specific operations in general (Schank, 1980; p.260), they provide a very good overall context for any (past, current or future) real world episode in the specific application domain.

Abstract skeletal plans resemble to a certain degree the Memory Organization Packages or MOPS (Schank, 1982), which are used for indexing cases in CBR systems. The hierarchy of problem class descriptions corresponds to the MOP abstraction hierarchy and the so called scenes of MOPS correspond to the individual operators of the case-oriented expert system. At least for the terminal problem classes, the dependency graph of skeletal plans (Schmalhofer et al. 1991b) are basically identical to PRIAR's validation structure (Kambhampati, 1990). Concerning the refitting of plans, there are two important differences, however.

5.2 The Refitting of Plans

In PRIAR, old plans are adjusted to new situations, by directly refitting the concrete old plan to a concrete new situation. There is no direct refitting of a plan in case-oriented systems. Instead, the most specific abstraction of an old case, which also meets the requirements of the new conditions is selected. This abstract description is then specialized for the particular new situation. Strictly speaking, old cases are neither adapted by structural analogy of the cases (Hammond, 1989), nor by derivational analogy to veridical problem solving processes (Carbonell, 1986; van Lehn, Jones & Chi, in press). But an expert's independently supplied rationalizations for past success cases are applied for reusing the old success cases under modified conditions. Because of the underlying hierarchical organization, the specialization of the skeletal plans to a concrete plan could, however, be viewed as a sort of derivational adaptation (van Lehn, personal communication).

New problems can thus be solved by finding some abstract characterization that fits the new problem. Completely novel types of context and environment descriptions may be added at system application time. Whenever required by the application situation, a seemingly identical problem may thus be appropriately classified for the apparently different application situation.

For example, in mechanical engineering some production problem may be solved by a completely different plan, when the new context now requires that all tools must be bought from some specific supplier, with which a sales contract was made in the meantime. Similarly, new technologies may require that some new cutting tools are to be applied.

Such novel situation specifications will not make the expert system fail completely: Although the plan from a specific case may no longer be usable, the application conditions of some more abstract skeletal plan may still indicate that the attached skeletal plan defines an appropriate search space for finding a solution to the novel problem. Since the expert's conceptions were used for partitioning the search space by defining appropriate skeletal plans, some of the adaptability of human experts may therefore also be found in case-oriented expert systems.

The more abstract the skeletal plans which fit the current situation, the less competence the system supplies for solving the new problem. The user of the system must then supplement the lacking knowledge. The least abstract skeletal plan, which subsumes the plan of the old and the new problem thus determines how competent the system is for the specific new

problem. With increasingly dissimilar new situations, the case-oriented expert system gracefully degrades until problems are recognized to lie outside its field of expertise.

PLEXUS and similar systems (Alterman, 1988, 1991) refit plans by abstracting and respecializing single operations. The case-oriented expert system on the other hand abstracts a complete plan with its dependency graph and then specializes it according to the new situational factors.

5.3 Combination of Rules and Cases

The skeletal plans can be seen as explanation-based generalizations (Mitchell, Keller & Kedar-Cabelli, 1986) or abstractions of one or several concrete plans in terms of the expert´s conceptualization. For these explanation based abstractions the expert's problem classes in the abstraction hierarchy serve as the "target concepts" and the abstraction and refinement rules together with the different operator definitions are the "domain theory". Hierarchical skeletal plans are constructed as intermediate generic action sequences or mechanisms (Klinker, Bhola, Dallemagne, Marques, and McDermott, 1990), that are more or less operational. These mechanisms lie between the expert's high level understanding of the domain and the episodic knowledge (Strube, 1989) encoded in the cases.

Case oriented systems unify case and rule based reasoning (instead of coordinating it) by building abstract and approximate theories (Chien, 1989), which are logically consistent with all those (prototypical) cases, that are subsumed in the problem class abstraction hierarchy. The system is therefore not competent for exceptional cases. The unification of a planning rationale in the form of general rules with specific cases makes the system quite robust and dependable. Heuristic approaches to combining case and rule based reasoning (e.g. Rissland & Skalag, 1989) will probably not yield the same robustness.

The model priority coordination of MOLTKE (Althoff & Wess, in press) and the turn taking coordination proposed by Janetzko & Strube (in press) do not require the cases to be consistent with the general rules of the expert system. In those approaches, rules are used for regular application conditions and cases are applied for handling the exceptions.

5.4 Developing Planning and Design Rationals through the Analysis of Success Cases

Rather than a heuristic indexing scheme for the efficient access to pertinent cases (Kolodner, 1983), we proposed to use an expert's high level understanding of the whole application domain for organizing a case-oriented knowledge-base. By analyzing success cases (Fischer & Reeves, in press) in terms of the expert's perceptions, rationales are found for the specific cases and subsequently represented in the knowledge base. We could therefore also portray the knowledge acquisition for case-oriented systems as reversed engineering.

Case-oriented expert systems are authored by a domain expert. This authoring process is supported by several tools (and possibly by a knowledge engineer as an editorial assistant). After an expert has published his knowledge in such a system, different users may situate the encoded knowledge in their specific real world contexts. The user would thus decide how similar the current problem is to some previous problem. This similarity is determined by finding the most specific abstraction in the problem hierarchy for the current problem, in terms of some general language, that is shared between the author (expert) and the user of the knowledge base.

Although the expert system can perform tasks automatically, we can still view the user of the system as a reader of the knowledge base (Schmalhofer, in preparation). After the user has

comprehended the knowledge base with respect to the specific problem at hand, he may possibly enter new knowledge into the system, so that the problem is processed according to his desires. Situated applications of expert system thus require that the system is end-user modifiable (Fischer & Girgensohn, 1990).

Acknowledgments

References

Alterman, R. (1988). Adaptive planning. Cognitive Science, 12, 393-421.

Alterman, R. (1991, January). Interaction and instruction usage. Paper presented at the second annual winter text conference, Teton Village, Wyoming.

Althoff, K.D. & Wess, S. (in press). Case-based reasoning and expert system development. In Schmalhofer, F., Strube, G., & Wetter, T. (Eds.), Contemporary Knowledge Engineering and Cognition (pp. 143-155). Berlin/Heidelberg: Springer-Verlag.

Anderson, J.R. (1990). The adaptive character of thought. Hillsdale, NJ: Lawrence Erlbaum.

Bachant, J., & McDermott, J. (1984). R1 Revisited: Four years in the trenches. AI Magazine, 5 (3).

Bartsch-Spörl, B. (1991). Wie bekommt man KADS und Case-Based reasoning (CBR) unter einen Hut? . In Ueberreiter, B. & Voß, A. (Eds.), Materials for the KADS User Meeting. München: Siemens.

Bergmann, R. & Schmalhofer, F. (1991). CECoS: A case experience combination system for knowledge acquisition for expert systems. Behavior Research Methods, Instruments, & Computers, 23, 142-148.

Bergmann, R. (in press). Knowledge Acquisition by generating skeletal plans from real world cases. In Schmalhofer, F., Strube, G., & Wetter, T. (Eds.), Contemporary Knowledge Engineering and Cognition (pp. 121-129). Berlin/Heidelberg: Springer-Verlag.

Bernardi, A., Boley, H., Klauck, C., Hanschke, P., Hinkelmann, K., Legleitner, R., Kühn, O., Meyer, M., Richter, M.M., Schmalhofer, F., Schmidt, G., & Sommer, W. (1991). ARC-TEC: Acquisition, representation and compilation of technical knowledge. In Haton, J.P. & Rault, J.C. (Eds.), Proceedings of the Eleventh International Conference Expert Systems & Their Applications (Avignon ´91) (pp. 133-145). Paris, France: Gerfau.

Breuker, J. & Wielinga, B. (1989). Models of expertise in knowledge acquisition. In Guida, G. & Tasso, C. (Eds.), Topics in expert system design, methodologies and tools (pp. 265 - 295). Amsterdam, Netherlands: North Holland.

Carbonell, J.G. (1986). Derivational analogy: A theory of reconstructive problem solving and expertise acquisition. In Michalski, R.S., Carbonell, J.G., & Mitchell, T.M. (Eds.), Machine Learning: An artificial intelligence approach (Vol. 2, pp. 371-392). Los Altos, CA: Morgan Kaufmann.

Chandrasekaran, B. (1986). Generic tasks in knowledge-based reasoning: High-level building blocks for expert system design. IEEE Expert, 1, 32 - 30.

Chase, W.G. & Simon, H.A. (1973). Perception in chess. Cognitive Psychology, 4, 55-81.

Chase, W.G. & Ericcson, K.A. (1982). Skill and working memory. In Bower, G.H. (Ed.), <u>The psychology of learning and motivation</u>. New York: Academic Press.

Chi, M., Feltovich, P., & Glaser, R. (1981). Categorization and representation of physics problems by experts and novices. <u>Cognitive Science</u>, 5, 121-152.

Chien, S.A. (1989). Using and refining simplifications: Explanation-based learning of plans in intractable domains. In <u>Proceedings of the Eleventh International Joint Conference on Artificial Intelligence</u> (pp. 590-595). Palo Alto, CA: Morgan Kaufmann.

Clancey, W.J. (1989). The frame-of-reference problem in cognitive modelling. In <u>Proceedings of the 11th Annual Conference of the Cognitive Science Society</u> (pp. 107-114). Hillsdale, NJ: Lawrence Erlbaum.

Fischer, G. & Girgensohn, A. (1990). End-User Modifiability in Design Environments, Human Factors in Computing Systems, <u>CHI'90, Conference Proceedings</u> (Seattle, WA) ACM, New York (April 1990), pp. 183-191.

Fischer, G. & Reeves, B. (in press). Beyond intelligent interfaces: Exploring, analyzing and creating success models of cooperative problem solving. <u>Applied Intelligence Journal</u>.

Friedland, P.E. & Iwasaki, Y. (1985). The concept and implementation of skeletal plans. <u>Journal of Automated Reasoning</u>, 1, 161-208.

Georgeff, M.P. (1987). Planning. <u>Annual Review in Computing Science</u>, 2, 359-400.

Hammond, K. (1989). <u>Case-based planning</u>. London: Academic Press.

Harmon, P., & King, D. (1985). <u>Expert systems</u>. New York: Wiley Press.

Janetzko, D. & Strube, G. (in press). Case-based reasoning and model-based knowledge-acquisition. In Schmalhofer, F., Strube, G., & Wetter, T. (Eds.), <u>Contemporary Knowledge Engineering and Cognition</u> (pp. 96-111). Berlin/Heidelberg: Springer-Verlag.

Kühn, O., Linster, M., & Schmidt, G. (1991). <u>Clamping, COKAM, KADS, and OMOS: The construction and operationalization of a KADS conceptual model</u> (Technical Memo No. TM-91-03). Kaiserslautern, Germany: German Research Center for Artificial Intelligence.

Kühn, O. & Schmalhofer, F. (1992). Hierachical skeletal plan refinement: Task-and inference structures. In Bauer, C. & Karbach, W. (Eds.), <u>Materials for the second KADS User Meeting: Task and inference structures</u>. München: Siemens.

Kambhampati, S. (1990). Mapping and retrieval during plan reuse: A validation structure based approach. In <u>Proceedings of the Eighth National Conference on Artificial Intelligence (AAAI-90)</u> (pp. 170-175). Cambridge, MA: MIT Press.

Kellog, R.T. & Bourne, L.E. (1989). Nonanalytic-automatic abstraction of concepts. In Sidinski, J.B. (Ed.), <u>Conditioning, cognition and methodology</u> (pp. 89-111). Landham, MD: University Press of America.

Klinker, G., Bhola, C., Dallemagne, G., Marques, D., & McDermott, J. (1990). Usable and reusable programming constructs. In Boose, J.H. & Gaines, B.R. (Eds.), <u>Proceedings of the 5th Banff Knowledge Acquisition for Knowledge-Based Systems Workshop</u> (pp. 14-1-14-20). Banff, Alberta, Canada: SRDG Publications.

Knoblock, C.A. (1990). Learning abstraction hierarchies for problem solving. In <u>Proceedings of the Eighth National Conference on Artificial Intelligence</u> (pp. 923-928). Cambridge, MA: MIT Press.

Kolodner, J.L. (1983). Maintaining organization in a dynamic long-term memory. <u>Cognitive Science</u>, 7, 243-280.

Korf, R.E. (1987). Planning as search: A quantitative approach. <u>Artificial Intelligence</u>, 33, 65-88.

Laird, J.E., Rosenbloom, P.S., & Newell, A. (1984). Towards chunking as a general learning mechanism. In <u>Proceedings of the National Conference on Artificial Intelligence (AAAI '84)</u>. Menlo Park, CA: American Association for Artificial Intelligence.

van Lehn, K., Jones, R.M., & Chi, M.T.H. (in press). A model of the self-explanation effect. <u>Journal of the Learning Sciences</u>.

Linster, M. (1992). Knowledge acquisition based on explicit models of problem solving. Dissertation/Draft, University of Kaiserslautern, Germany.

Manago, M., Conruyt, N., & Lerenard, J. (in press). Acquiring descriptive knowledge for classification and identification. In Wetter, T., Boose, J., Gaines, B., Linster, M., & Schmalhofer, F. (Eds.), Current developments in knowledge acquisition: EKAW-92. Berlin-Heidelberg: Springer-Verlag.

McDermott, J. (1988). Preliminary steps toward a taxonomy of problem-solving methods. In Marcus, S. (Ed.), Automating knowledge acquisition for expert systems (pp. 225-256). Boston: Kluwer Academic.

Mitchell, T.M., Keller, R.M., & Kedar-Cabelli, S.T. (1986). Explanation-based generalization: A unifying view. Machine Learning, 1, 47-80.

Newell, A. (1982). The knowledge level. Artificial Intelligence, 2, 87 - 127.

Pfeifer, R. & Rademakers, P. (1991). Situated adaptive design: Toward a new methodology for knowledge systems developments. In Internationaler GI-Kongress. München.

Reinartz, T. (1991). Definition von Problemklassen im Maschinenbau als eine Begriffsbildungsaufgabe (DFKI-Document No. D-91-16). Kaiserslautern, Germany: German Research Center for Artificial Intelligence.

Riesbeck, C.K. & Schank, R.C. (1989). Inside case-based reasoning. Hillsdale, NJ: Lawrence Erlbaum.

Rissland, E.L., & Skalag, D.B., (1989). Combining case-based and rule-based reasoning: A heuristic approach. In Proceedings of the International Joint Conference on Artificial Intelligence 1989, August 1989, Detroit.

Schank, R.C. (1980). Language and memory. Cognitive Science, 4, 243-284.

Schank, R.C.(1982). Dynamic memory: A theory of learning in computers and people. Cambridge University Press.

Schmalhofer, F. & Glavanov, D. (1986). Three components of understanding a programmer's manual: Verbatim, propositional and situational representations. Journal of Memory and Language, 25, 279-294.

Schmalhofer, F. (1987). Expert systems as cognitive tools for human decision making. In Mumpower, J.L., Lawrence, D.P., Renn, O., & Uppuluri, V.R. (Eds.), Expert judgement and expert systems (pp. 269-288). Berlin-Heidelberg: Springer-Verlag.

Schmalhofer, F., Kühn, O., & Schmidt, G. (1991a). Integrated knowledge acquisition from text, previously solved cases, and expert memories . Applied Artificial Intelligence, 5, 311-337.

Schmalhofer, F., Bergmann, R., Kühn, O., & Schmidt, G. (1991b). Using integrated knowledge acquisition to prepare sophisticated expert plans for their re-use in novel situations. In Christaller, T. (Ed.), GWAI-91: 15. Fachtagung Künstliche Intelligenz (pp. 62-73). Berlin: Springer-Verlag.

Schmalhofer, F. & Schmidt, G. (1991). Situated text-analysis with COKAM+. In Woodward, B. (Ed.), Sisyphus Working Papers: Text Analysis: European Knowledge Acquisition Workshop (pp. 25-34). Crieff, Scotland: University of Strathclyde.

Schmalhofer, F. & Reinartz, T. (1991). Intelligent documentation as a catalyst for developing cooperative knowledge-based systems. Manuscript DFKI.

Schmalhofer, F. (in preparation) Expert systems as intelligent communication tools - knowledge acquisition as an authoring process. In Evans, D. & Patel, V. (Eds.), Advanced models in cognition for medical training and practice. Berlin-Heidelberg: Springer-Verlag.

Schmidt, G. & Schmalhofer, F. (1990). Case-oriented knowledge acquisition from texts. In Wielinga, B., Boose, J., Gaines, B., Schreiber, G., & van Someren, M. (Eds.), Current trends in knowledge acquisition (EKAW '90) (pp. 302-312). Amsterdam: IOS Press.

Shanteau, J. (1984). Some unasked questions about the psychology of expert decision makers. In Elhawary, M.E. (Ed.), Proceedings of the 1984 IEEE Conference on Systems, Man, and Cybernetics. New York: IEEE.

Shanteau, J. (1988). Psychological characteristics and strategies of expert decision makers. Acta Psychologica, 68, 203-215.

Steels, L. (1987). Second generation expert systems. In Bramer, M. (Ed.), Research and developments in expert systems III. Cambridge: Cambridge University Press.

Strube, G. (1989). Episodisches Wissen (Arbeitspapiere der GMD No. 385, 10-26). Sankt Augustin, Germany: Gesellschaft für Mathematik und Datenverarbeitung.

Suchman, L.A. (1987). Plans and situated actions. Cambridge: Cambridge University Press.

Thoben, J., Schmalhofer, F., & Reinartz, T. (1991). Wiederholungs- Varianten- und Neuplanung bei der Fertigung rotationssymmetrischer Drehteile (DFKI-Document No. D-91-16). Kaiserslautern, Germany: German Research Center for Artificial Intelligence.

Tschaitschian, B. (1991). Eine integrative Wissenserhebung und -analyse mit CECoS: Konzepte und prototypische Implementierung. Projektarbeit, University of Kaiserslautern, Germany.

Wharton, C. & Kintsch, W. (1991). An overview of the construction-integration model: A theory of comprehension as a foundation for a new cognitive architecture. SIGART Bulletin, 2, 169-173.

Wilkins, D.C. (1991). A framework for integration of machine learning and knowledge acquisition techniques. In Boose, J.H. & Gaines, B.R. (Eds.), Proceedings of the 6th Banff Knowledge Acquisition for Knowledge-Based Systems Workshop (pp. 37-1-37-14). Banff, Alberta, Canada: SRDG Publications.

DFKI Publikationen

DFKI Publications

DFKI Research Reports

RR-92-45
Elisabeth André, Thomas Rist: The Design of Illustrated Documents as a Planning Task
21 pages

RR-92-46
Elisabeth André, Wolfgang Finkler, Winfried Graf, Thomas Rist, Anne Schauder, Wolfgang Wahlster: WIP: The Automatic Synthesis of Multimodal Presentations
19 pages

RR-92-47
Frank Bomarius: A Multi-Agent Approach towards Modeling Urban Traffic Scenarios
24 pages

RR-92-48
Bernhard Nebel, Jana Koehler:
Plan Modifications versus Plan Generation:
A Complexity-Theoretic Perspective
15 pages

RR-92-49
Christoph Klauck, Ralf Legleitner, Ansgar Bernardi:
Heuristic Classification for Automated CAPP
15 pages

RR-92-50
Stephan Busemann:
Generierung natürlicher Sprache
61 Seiten

RR-92-51
Hans-Jürgen Bürckert, Werner Nutt:
On Abduction and Answer Generation through Constrained Resolution
20 pages

RR-92-52
Mathias Bauer, Susanne Biundo, Dietmar Dengler, Jana Koehler, Gabriele Paul: PHI - A Logic-Based Tool for Intelligent Help Systems
14 pages

RR-92-53
Werner Stephan, Susanne Biundo:
A New Logical Framework for Deductive Planning
15 pages

RR-92-54
Harold Boley: A Direkt Semantic Characterization of RELFUN
30 pages

RR-92-55
John Nerbonne, Joachim Laubsch, Abdel Kader Diagne, Stephan Oepen: Natural Language Semantics and Compiler Technology
17 pages

RR-92-56
Armin Laux: Integrating a Modal Logic of Knowledge into Terminological Logics
34 pages

RR-92-58
Franz Baader, Bernhard Hollunder:
How to Prefer More Specific Defaults in Terminological Default Logic
31 pages

RR-92-59
Karl Schlechta and David Makinson: On Principles and Problems of Defeasible Inheritance
13 pages

RR-92-60
Karl Schlechta: Defaults, Preorder Semantics and Circumscription
19 pages

RR-93-02
Wolfgang Wahlster, Elisabeth André, Wolfgang Finkler, Hans-Jürgen Profitlich, Thomas Rist:
Plan-based Integration of Natural Language and Graphics Generation
50 pages

RR-93-33
Bernhard Nebel, Jana Koehler:
Plan Reuse versus Plan Generation: A Theoretical and Empirical Analysis
33 pages

RR-93-34
Wolfgang Wahlster:
Verbmobil Translation of Face-To-Face Dialogs
10 pages

RR-93-35
Harold Boley, François Bry, Ulrich Geske (Eds.):
Neuere Entwicklungen der deklarativen KI-Programmierung — *Proceedings*
150 Seiten
Note: This document is available only for a nominal charge of 25 DM (or 15 US-$).

RR-93-36
Michael M. Richter, Bernd Bachmann, Ansgar Bernardi, Christoph Klauck, Ralf Legleitner, Gabriele Schmidt: Von IDA bis IMCOD: Expertensysteme im CIM-Umfeld
13 Seiten

RR-93-38
Stephan Baumann: Document Recognition of Printed Scores and Transformation into MIDI
24 pages

RR-93-40
Francesco M. Donini, Maurizio Lenzerini, Daniele Nardi, Werner Nutt, Andrea Schaerf:
Queries, Rules and Definitions as Epistemic Statements in Concept Languages
23 pages

RR-93-41
Winfried H. Graf: LAYLAB: A Constraint-Based Layout Manager for Multimedia Presentations
9 pages

RR-93-42
Hubert Comon, Ralf Treinen:
The First-Order Theory of Lexicographic Path Orderings is Undecidable
9 pages

RR-93-44
Martin Buchheit, Manfred A. Jeusfeld, Werner Nutt, Martin Staudt: Subsumption between Queries to Object-Oriented Databases
36 pages

RR-93-45
Rainer Hoch: On Virtual Partitioning of Large Dictionaries for Contextual Post-Processing to Improve Character Recognition
21 pages

RR-93-46
Philipp Hanschke: A Declarative Integration of Terminological, Constraint-based, Data-driven, and Goal-directed Reasoning
81 pages

DFKI Technical Memos

TM-91-15
Stefan Busemann: Prototypical Concept Formation An Alternative Approach to Knowledge Representation
28 pages

TM-92-01
Lijuan Zhang: Entwurf und Implementierung eines Compilers zur Transformation von Werkstück-repräsentationen
34 Seiten

TM-92-02
Achim Schupeta: Organizing Communication and Introspection in a Multi-Agent Blocksworld
32 pages

TM-92-03
Mona Singh:
A Cognitiv Analysis of Event Structure
21 pages

TM-92-04
Jürgen Müller, Jörg Müller, Markus Pischel, Ralf Scheidhauer:
On the Representation of Temporal Knowledge
61 pages

TM-92-05
Franz Schmalhofer, Christoph Globig, Jörg Thoben:
The refitting of plans by a human expert
10 pages

TM-92-06
Otto Kühn, Franz Schmalhofer: Hierarchical skeletal plan refinement: Task- and inference structures
14 pages

TM-92-08
Anne Kilger: Realization of Tree Adjoining Grammars with Unification
27 pages

TM-93-01
Otto Kühn, Andreas Birk: Reconstructive Integrated Explanation of Lathe Production Plans
20 pages

TM-93-02
Pierre Sablayrolles, Achim Schupeta:
Conlfict Resolving Negotiation for COoperative Schedule Management
21 pages

TM-93-03
Harold Boley, Ulrich Buhrmann, Christof Kremer:
Konzeption einer deklarativen Wissensbasis über recyclingrelevante Materialien
11 pages

TM-93-04
Hans-Günther Hein: Propagation Techniques in WAM-based Architectures — The FIDO-III Approach
105 pages

DFKI Documents

D-92-23
Michael Herfert: Parsen und Generieren der Prolog-artigen Syntax von RELFUN
51 Seiten

D-92-24
Jürgen Müller, Donald Steiner (Hrsg.):
Kooperierende Agenten
78 Seiten

D-92-25
Martin Buchheit: Klassische Kommunikations- und Koordinationsmodelle
31 Seiten

D-92-26
Enno Tolzmann:
Realisierung eines Werkzeugauswahlmoduls mit Hilfe des Constraint-Systems CONTAX
28 Seiten

D-92-27
Martin Harm, Knut Hinkelmann, Thomas Labisch:
Integrating Top-down and Bottom-up Reasoning in COLAB
40 pages

D-92-28
Klaus-Peter Gores, Rainer Bleisinger: Ein Modell zur Repräsentation von Nachrichtentypen
56 Seiten

D-93-01
Philipp Hanschke, Thom Frühwirth: Terminological Reasoning with Constraint Handling Rules
12 pages

D-93-02
*Gabriele Schmidt, Frank Peters,
Gernod Laufkötter:* User Manual of COKAM+
23 pages

D-93-03
Stephan Busemann, Karin Harbusch(Eds.):
DFKI Workshop on Natural Language Systems: Reusability and Modularity - Proceedings
74 pages

D-93-04
DFKI Wissenschaftlich-Technischer Jahresbericht 1992
194 Seiten

D-93-05
Elisabeth André, Winfried Graf, Jochen Heinsohn, Bernhard Nebel, Hans-Jürgen Profitlich, Thomas Rist, Wolfgang Wahlster:
PPP: Personalized Plan-Based Presenter
70 pages

D-93-06
Jürgen Müller (Hrsg.):
Beiträge zum Gründungsworkshop der Fachgruppe Verteilte Künstliche Intelligenz Saarbrücken 29.-30. April 1993
235 Seiten
Note: This document is available only for a nominal charge of 25 DM (or 15 US-$).

D-93-07
Klaus-Peter Gores, Rainer Bleisinger:
Ein erwartungsgesteuerter Koordinator zur partiellen Textanalyse
53 Seiten

D-93-08
Thomas Kieninger, Rainer Hoch: Ein Generator mit Anfragesystem für strukturierte Wörterbücher zur Unterstützung von Texterkennung und Textanalyse
125 Seiten

D-93-09
Hans-Ulrich Krieger, Ulrich Schäfer:
TDL ExtraLight User's Guide
35 pages

D-93-10
Elizabeth Hinkelman, Markus Vonerden,Christoph Jung: Natural Language Software Registry
(Second Edition)
174 pages

D-93-11
Knut Hinkelmann, Armin Laux (Eds.):
DFKI Workshop on Knowledge Representation Techniques — Proceedings
88 pages

D-93-12
Harold Boley, Klaus Elsbernd, Michael Herfert, Michael Sintek, Werner Stein:
RELFUN Guide: Programming with Relations and Functions Made Easy
86 pages

D-93-14
Manfred Meyer (Ed.): Constraint Processing – Proceedings of the International Workshop at CSAM'93, July 20-21, 1993
264 pages
Note: This document is available only for a nominal charge of 25 DM (or 15 US-$).

D-93-15
Robert Laux: Untersuchung maschineller Lernverfahren und heuristischer Methoden im Hinblick auf deren Kombination zur Unterstützung eines Chart-Parsers
86 Seiten

D-93-20
Bernhard Herbig:
Eine homogene Implementierungsebene für einen hybriden Wissensrepräsentationsformalismus
97 Seiten

D-93-21
Dennis Drollinger:
Intelligentes Backtracking in Inferenzsystemen am Beispiel Terminologischer Logiken
53 Seiten