Knowledge Acquisition by Generating Skeletal Plans from Real World Cases

Ralph Bergmann
German Research Center for Artificial Intelligence
University Bldg 57
Erwin-Schroedinger Str.
D-6750 Kaiserslautern

email: bergmann@informatik.uni-kl.de

Abstract. Although skeletal plan refinement is used in several planning systems, a procedure for the automatic acquisition of such high-level plans has not yet been developed. The proposed explanation-based knowledge acquisition procedure constructs a skeletal plan automatically from a sophisticated concrete planning case. The classification of that case into a well-described class of problems serves as an instrument for adjusting the applicability of the acquired skeletal plans to that class. The four phases of the proposed procedure are constituted as follows: In the first phase, the execution of the source plan is simulated, and explanations for the effects of the occurred operators are constructed. In the second phase, the generalization of these explanations is performed with respect to a criterion of operationality which specifies the vocabulary for defining abstract operators for the skeletal plan. The third phase, a dependency analysis of the resulting operator effects, unveils the interactions of the concrete plan which are substantial for the specified class. In the forth phase, the concept descriptions for the abstract operators of the skeletal plan are formed by collecting and normalizing the important constraints for each operation that were indicated by the dependencies. With this procedure sophisticated planning solutions from human experts can be generalized into skeletal plans and consequently be reused by a planning system in novel situations.

1. Introduction

Many planning problems can be subdivided into more or less specific classes of problem types, in which each class has its own general solution plan (Tu, Kahn, Musen, Ferguson, Shortliffe & Fagan 1989). All plans of one class can thus be said to use the same solution idea, viewed from a higher level of abstraction. Such an abstract plan is called *skeletal plan* by Friedland and Iwasaki (1985) and is defined as follows:

A skeletal plan is a sequence of abstract and only partially specified steps which, when specialized to specific executable operations, will solve a given problem in a specific problem context. Skeletal plans exist at many levels of generality and are applicable to different classes of planning problems. They capitalize on beneficial interactions and avoid detrimental interferences between the single operations of a plan. Skeletal plan operations just need the right level of generality in the respective situation, to maintain and replay important interactions and eliminate irrelevant details which are easy to reconstruct.

Classical planning mechanisms such as specialization rules, (Shortliffe, Scott, Bischoff, Campbell, Melle & Jacobs 1981) or heuristic approaches (Friedland & Iwasaki, 1985) may be applied for obtaining a concrete plan from a good skeletal plan. A skeletal plan is refined to a concrete plan by specializing abstract operators independent of each other, so that the search in the space of concrete operators becomes feasible. This requires that the important interactions between operators, which always occur even in simple realistic planning tasks, must be taken into account during the construction of the skeletal plans. Therefore, the quality of the skeletal plan data base mainly determines the quality of the results of such a planning system.

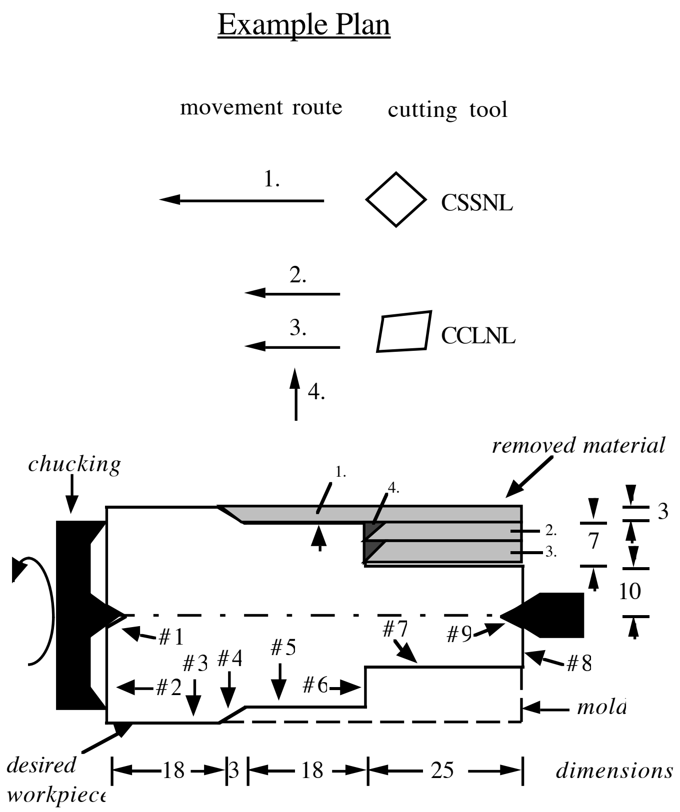
Nevertheless, the problem of the acquisition of skeletal plans has not yet been solved. In OPAL (Musen, Fagan, Combs & Shortliffe 1987), the knowledge acquisition system for the ONCOCIN expert system, oncological therapy protocols, which function as skeletal plans in this domain, must be constructed and entered manually with support of a graphical editor. For Friedlands MOLGEN planner the situation is similar. The acquisition and debugging of skeletal plans has been identified as a major problem (Stefik, 1981). This is because constructing skeletal plans is a modelling task which requires the definition of a terminology sufficiently abstracting from details which are irrelevant for the planning task. Usually neither an abstract planning terminology nor skeletal plans described in terms of such a terminology are directly available in real world domains. Schmalhofer and Thoben (this volume) have studied domain experts who are requested to construct skeletal plans for mechanical engineering. Their approach to the manual construction of skeletal plans seems to be successful but is quite time-consuming for the expert.

This current paper investigates explanation-based learning (EBL) (Mitchell, Keller, Kedar-Cabelli 1986; DeJong & Mooney 1986) in order to establish an *automatic* knowledge acquisition method for obtaining skeletal plans from real world problem solutions for a given class of problems. A theory that describes important aspects of the operator effects is used to simulate the execution of the plan and to derive an explanation of how the plan solves those problem features that define the problem class.

In the following sections, the real world application domain of mechanical engineering is introduced together with an example, and the representation of skeletal plans is discussed. The proposed method for an automatic knowledge acquisition is subdivided into four phases: a plan simulation and explanation phase, a generalization phase, a dependency analysis, and a normalization into the skeletal plan representation. In the final discussion the benefits and the limitations of this approach are evaluated in a general manner.

2. Production Planning in Mechanical Engineering

The planning domain used as the field of demonstration is mechanical engineering, more specifically the production of rotational parts on a lathe. Presently, the design of mechanical work pieces is widely supported by CAD systems, and computer controlled lathes (CNC-machines) are used for manufacturing such parts. The planning process itself cannot be performed automatically since a lot of different kinds of domain knowledge are required for the construction of good plans. The characteristics of the complexity of this domain and the planning process are presented in detail by Schmalhofer and Thoben (this volume) through a set of planning tasks from a catalogue of a supplier of machining centers and tools. For the demonstration of the automatic approach to the acquisition of skeletal plans, only a simplified version of an already existing real world planning problem is introduced for clarity. Figure 1 shows an example of the work piece to be produced together with a production plan which consists of one chucking and four cutting operations.



Formalization of the plan:

chuck (lathe_dog, '30Kg', left)
cut('CSSNL' ,
□□□□□□□form(linear,(64,17), (21,17)),
□□□□□□□speed(450,0.45))
cut('CCLNL',
□□□□□□□form(linear,(64,14),(39,14)),
□□□□□□□speed(450,0.45))
cut('CCLNL',
□□□□□□□form(linear,(64,10),(39,10)),
□□□□□□□speed(450,0.45))
cut('CCLNL',
□□□□□□form(linear,(39,10),(39,17)),
□□□□□□speed(150,0.25))

Formalization of the desired workpiece

surface(#1,form(linear, (3,0), (0,3))) ∧
centerhole(#1,'Zen3mm') ∧
surface(#2,form(linear, (0,3), (0,20))) ∧
facing_area(#2) ∧
surface(#3,form(linear, (0,20), (18,20))) ∧
surface(#4,form(linear, (18,20), (21,17))) ∧
surface(#5,form(linear, (21,17), (39,17))) ∧
surface(#6,form(linear, (39,17), (39,10))) ∧
tolerance(#6,low) ∧
surface(#7,form(linear, (39,10), (64,10))) ∧
surface(#8,form(linear, (64,10), (64,3))) ∧
facing_area(#8) ∧
surface(#9,form(linear, (64,3), (61,0))) ∧
centerhole(#9,'Zen3mm',) ∧

Figure 1:
In this example, a mold is fixed with a lathe dog, and material is removed in four steps, in which two cutting tools with different shapes are used. The material that is removed by each step is indicated by the shaded areas on the sketch of the mold. Note that this graphical sketch is only a two-dimensional sectional drawing of the three-dimenstional rotational part.

The formal description of the example is presented on the right side of this Figure. In this representation, the plan is defined as a sequence of operators together with their parameters. The workpiece is represented, similar to world states in STRIPS (Fikes & Nilsson, 1971), as a conjunction of predicate facts, in which each fact expresses one isolated attribute of the workpiece, such as a coherent surface area or a technological feature (centerhole, facing area or material) (Bergmann, Bernardi, Klauck, Kühn, Legleitner, Schmalhofer & Schmidt 1990). This representation of cases, which can easily be derived from the data representations CAD systems employ, is used as the input for the skeletal plan construction procedure.

2.1 Skeletal Plans in Mechanical Engineering

A skeletal plan consists by definition of a sequence of abstract operators or operator classes. Extensionally, an operator class is formed by grouping some concrete operators together. Intentionally, an operator class needs to be described by a combination of relevant attributes that all operators of that class have in common. A conjunction of constraints to some operator features, such as inductive methods of concept formation usually construct, are a useful manner for defining operator classes for a skeletal plan. In mechanical engineering, for example, a subclass of chucking operators may be formed by the following conjunctive description of three operator features:

chucking with centerholes **and**
chucking position on the left side **and**
two fixations.

2.2 Acquisition of Skeletal Plans

Automatic acquisition of skeletal plans by analysis of cases is itself a knowledge-intensive process. Knowledge is required to explain the functioning of the problem solution, to identify interactions between the operators, and a terminology is needed to construct the descriptions of operator classes. Therefore this automatic process is embedded into an integrated knowledge acquisition method, which has been described by (Schmalhofer, Schmidt & Kühn, 1991b; Schmalhofer, Bergmann, Kühn & Schmidt, 1991a).

Within this integrated knowledge acquisition method, an interactive tool named COKAM (Case-Oriented Knowledge-Acquisition Method from Text) (Schmidt & Schmalhofer, 1990) is used to extract information from text and the expert's common sense knowledge, guided by cases of problem solution. The formalization of this elicited knowledge serves as a model of operators which already contains the basic terms on which operator abstractions can be composed. This operator theory is required to be mostly complete and tractable to enable the application of the explanation-based learning procedure.

Another interactive knowledge acquisition tool named CECoS (Case-Experience Combination System) (Bergmann & Schmalhofer, 1991) yields a hierarchically structured set of problem classes from a set of prototypical cases through human expert judgements. The expert judgements are obtained so that a useful skeletal plan exists for each of the classes. An intensional definition of this class hierarchy, together with the classification of the origin case used for skeletal plan generation, is used to adjust the level of generality for the generated skeletal plan.

3. The Generation Procedure

The automatic generation of skeletal plans is based on an understanding of how a specific plan solves the given problem, and on recognizing those dependencies between the actions of the plan that are significant for a general solution for the whole problem class. A sequence of operator classes is constructed, so that the significant dependencies are maintained. The following detailed description of this approach is divided into four distinct phases.

3.1 Phase-I: Simulation and Explanation

This phase uses a domain theory that describes the applicability and the effects of operators for simulating the execution of the target plan. This theory formally represents each operator as a set of rules, in which the successor world state is created by the execution of the STRIPS like add- and delete actions of the rules' consequences. For example, two rules like *R1* and *R2* describe two effects of chucking operators with different generality. *R1* models the general effect of the execution of a chucking operator, whereas *R2* models the more specific consequence of a special chucking operator that requires centerholes.

R1: **IF** operator(chuck(x_{tool},x1,x2)) **THEN**
DELETE(unchucked),
ADD(chucked)

R2: **IF** operator(chuck(x_{tool},x1,x2)) ∧
requires_centerholes(x_{tool}) ∧
two_centerholes **THEN**
ADD(chuck_precision(high))

Additionally, a set of axioms is provided to infer the conditions of these rules from the descriptions of the world states. With the axioms *A1* and *A2* the condition of rule *R2* can be inferred to be true for the first operator in the example in Figure 1.

A1: $(\exists c1,c2,x1,x2,x3,x4.\ (\text{centerhole}(c1,x1,x2) \wedge \text{centerhole}(c2,x3,x4) \wedge c1 \neq c2)) \rightarrow$ two_centerholes.

A2: requires_centerholes(lathe_dog).

With a complete theory for all operators in the target plan, its execution is successfully simulated by sequentially applying all rules for the operators $Op_1,...,Op_n$ of the plan:

$$S_{ini} \xrightarrow{Op_1} S_1 \xrightarrow{Op_2} S_2 \ .... \ S_{n-1} \xrightarrow{Op_n} S_{goal}$$

From the initial state S_{ini} (the mold in mechanical engineering) all intermediate states that result after the execution of each operator, and the final state S_{goal} (the target workpiece) are computed. The proofs that exists for the applicability of each operator rule can now be seen as an explanation of each effect, which depends on operator attributes as well as on world state attributes.

3.2 Phase-II: Generalization

These proofs are generalized using EBL (Mitchell, Keller & Kedar-Cabelli, 1986; DeJong & Mooney 1986), which yields separate and more general concepts for the produced effects. The operationality criterion for EBL determines the vocabulary for expressing the generalized concepts and establishes the constraints from which the operator classes are constructed. These terms are initially provided through the application of COKAM. As a result of this generalization, a set of conditions is found which ensures more generally that the situations S1,...,Sn-1 and S_{goal} are created in the same manner. Look at the following example of four EBL generalized concepts for the effects of the first operator from Figure 1.

C1: chucked $\leftarrow$
operator(chuck(x_{tool}, x1,x2)).

C2: chucking_position(x_{pos}) $\leftarrow$
operator(chuck(x_{tool},x1,x_{pos})).

C3: chucking_precision(high) $\leftarrow$
operator(chuck(x_{tool}, x1,x2)) $\wedge$
requires_centerholes(x_{tool}) $\wedge$
centerhole(c1,x3,x4) $\wedge$
centerhole(c2,x5,x6) $\wedge$
c1 $\neq$ c2.

C4: chucking_fixations(x_n) $\leftarrow$
operator(chuck(x_{tool},x1,x2)) $\wedge$
number_of_fixations(x_{tool},x_n).

3.3 Phase-III: Dependency Analysis

The task of the dependency analysis is to identify those effects of the operators which are necessary to guarantee that those features of the workpieces which are named as relevant for the classification of the workpiece, are created by every specialization of the skeletal plan. Therefore, the interconnections between the separate concepts which were identified in the second phase are determined and analyzed. A directed graph is constructed in which all existing dependencies between the concepts are explicitly noted as arcs. A dependency arc between two concepts Cx and Cy exists, if the concept Cx describes an effect which is a necessary condition which occurs in the formation of concept Cy. Figure 2 shows a graphical representation of the dependency graph that results from the analysis of the example in Figure 1. For example, the dependency of the concept "tolerance(#6,low))" on the concept "chucking_precision(high)" states that it is necessary to have a high chucking precision in order to produce surface #6 with a low tolerance. Note that all the concepts are always related to one operator and usually require certain constraints on them. Thereby, the dependencies between two concepts also express dependencies between two operators. In the example mentioned above, the cut-operation which creates the surface #6 is dependent on the chucking operator.

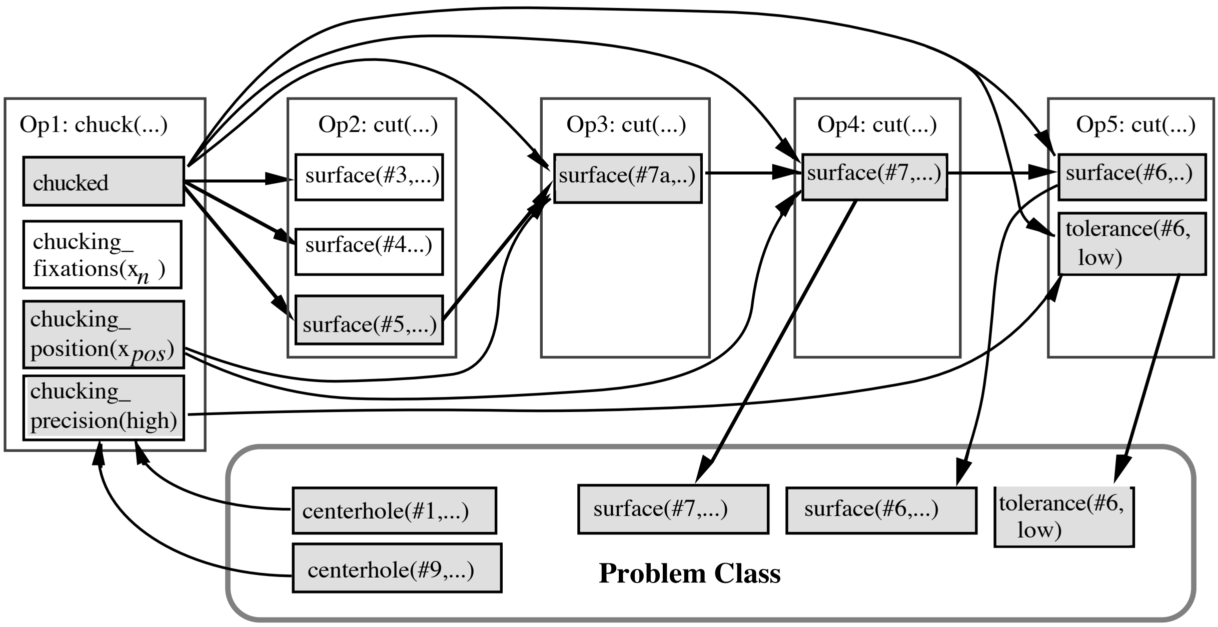


Figure 2: Dependency Graph for example case

For generating a skeletal plan, which is tailored to a definite problem class specified by important features of workpiece and mold, all concepts on which the class-relevant features are dependent, have to be identified. This is achieved by computing the least subgraph which contains all relevant features of the problem class, and in which all dependency predecessors of the concepts in the subgraph are themselves part of the subgraph. In Figure 2, the gray marked concepts, together with their links, form the subgraph, which results from a description of the classification of the examples in Figure 1. An overgeneralization that may have resulted from the independent treatment of the operators must be avoided in order to ensure that the operator classes which are to be generated for the skeletal plan can in fact be specialized independently for the solution of a new planning problem. Therefore, the concepts of the determined subgraph are unified along their dependency arcs, which yields one general concept of the plan for the whole problem class. For the example in Figure 1 a fragment of this concept is sketched as follows:

surface(#6,....) $\wedge$ surface(#7.....) $\wedge$ low_tolerance(#6) $\leftarrow$

operator(1,chuck(x_{tool1},x1,x2)) $\wedge$	; *From concept C1: "chucked"*
operator(1,chuck(x_{tool2},x3,left)) $\wedge$	; *From concept C2 : "chucking_position"*
operator(1,chuck(x_{tool3}, x4,x5)) $\wedge$	; *From concept C3: "chucking_precision"*
requires_centerholes(x_{tool3}) $\wedge$	; *From concept C3: "chucking_precision"*
centerhole(c1,x6,x7)) $\wedge$	; *From concept C3: "chucking_precision"*
centerhole(c2,x8,x9)) $\wedge$	; *From concept C3: "chucking_precision"*
c1 $\neq$ c2 $\wedge$	; *From concept C3: "chucking_precision"*
operator(2,cut(...)) $\wedge$	; *From concept "surface(#5...)"*

3.4 Phase-IV: Normalization into Skeletal Plan Representation

This phase builds the skeletal plan in its final representation by identifying independently solvable sub-formulas from the concept of the plan which expresses only local constraints on one operator. By analyzing the occurrence of variables in the conditions of the plan concept, all conditions are separated into:

- one set R_{Enable} which collects all conditions that only relate to features of the problem description,
- one set R_{Opi} for each operator Op_i where the conditions only specify parameters which directly correspond to one operator.

The set of constraints R_{Enable} formally describes the class of problems for which the skeletal plan can be used, and thus functions as an application condition for the skeletal plan. The skeletal plan itself is built of the sequence of constraints R_{Opi}, which exactly describe the required classes of operators. A further simplification of the constraint set is performed by the application of some rewrite reduction rules. Thereby, more operational descriptions of the operators classes are obtained.

For the example in Figure 1 the following skeletal plan with application conditions is generated.

1. Skeletal plan: operator(1,chuck(x_{tool}, x1,left)) $\wedge$ requires_centerholes(x_{tool})
   operator(2,cut(...)) $\wedge$ ...
   . . .
   operator(5,cut(...)) $\wedge$ ...
2. Application condition: centerhole(c1,x3,x4)) $\wedge$ centerhole(c2,x5,x6)) $\wedge$ c1 $\neq$ c2 $\wedge$ ....

A prototype of the described method was implemented on a Apple-Macintosh-II computer using the LPA-PROLOG environment (Bergmann 1990). This prototype creates skeletal plans for cases like the one in Figure 1.

4. Discussion

The automatic knowledge acquisition approach presented in this paper makes use of the idea to automatically prepare large amounts of already formally available knowledge for further use in an expert system. Especially for real world planning tasks such as mechanical engineering, the reuse of manually optimized plans in a more general way becomes possible without involving a domain expert in a time-consuming knowledge acquisition process. Qualitatively high skeletal plans can be generated if the origin plans are qualitatively good. Because of the explanation of the goal achievement, the beneficial interactions between operators are discovered and can be maintained for the abstract solution.

Since a knowledge-intensive learning paradigm such as explanation-based learning is the core of this method, a large amount of knowledge has to be provided to enable its application. The requirements on the available domain theory are very high in the sense that a correct and tractable theory is needed which is complete enough to allow the simulation of the full plan. It seems hopeless to acquire such a theory automatically, even if inductive learning methods were applied. Therefore the skeletal plan generation procedure has to be integrated with other, non-automatic methods such as COKAM and CECoS and works well if the requirements mentioned on the theory can be fulfilled in the application domain at hand.

Another question is concerned with the usefulness of the skeletal plans that are automatically acquired by this procedure. A skeletal plan is useful if it provides an abstraction that reduces the computational complexity of a planning process (Korf, 1988), and if it can be applied to a large class of problems. Since complexity reduction and wide applicability are somehow competing properties for a single abstraction we are engaged to find a hierarchy of skeletal plans. Therefore the utility based on the generality of a skeletal plan should be judged with respect to the problem class, for which the skeletal plan is constructed. If automatically obtained skeletal plans are compared with those that were acquired manually with considerable effort, a major weakness of generalization procedure can be identified. The described procedure is able to construct skeletal plans by abstracting single operations of a plan as far as the domain theory contains abstract descriptions of operator effects. If such a sufficient operator model can be supplied, the automatically generated operations can compete in utility with those constructed manually. If only a shallow operator theory such as in the case of the STRIPS domain is provided, the resulting skeletal plan for the most specific problem classes is the same as a macro-operator composed of all operators in the plan (Fikes, Hart & Nilsson, 1972).

Another kind of abstraction that appeared important for planning could not be performed by the proposed skeletal plan generation procedure. It is unable to collapse sequences of concrete operations into one single abstract operation. The bounds of the operations are always transfered from the concrete plan to the skeletal plan. Knoblock's (1990) approach to operator abstraction shows exactly the same deficits while Mädler (1991) tries to find "eyes of a needle" in the state space to combine sequences of operations into one single abstraction.

Further research to improve this automatic knowledge acquisition should consequently deal with the problem of finding more appropriate abstractions, for example by changing the plan representation language. Since this seems to be a knowledge-intensive process that cannot be applied in a isolated fashion, the interactions between such automatic and manual knowledge acquisition methods must be further examined and developed.

Acknowledgments

I would like to thank Franz Schmalhofer for many helpful discussions and for significantly contributing to this paper.

This research was supported in part by grant ITW 8902 C4 from the BMFT (German Ministry for Science and Technology) and by grant Schm 648/1 from the DFG (German Science Foundation).

References

Bergmann, R. (1990). Generierung von Skelettplänen als Problem der Wissensakquisition. *Unpublished masters thesis*. Universität Kaiserslautern.

Bergmann, R., Bernardi, A., Klauck, C., Kühn, O., Legleitner, R., Schmalhofer, F., & Schmidt, G. (1990). Formulierung von Anforderungen zur Darstellung von Werkstücken und Spezifikation einer Makrorepräsenation. *Internes ARC-TEC Diskussionspapier Nr. 8*.

Bergmann, R. & Schmalhofer, F. (1991). CECoS: A case experience combination system for knowledge acquisition for expert systems. To appear in: *Behavior Research Methods, Instruments and Computers*.

DeJong, G., & Mooney, R. (1986). Explanation-based learning: An alternative view. *Machine Learning,* **1**, pp. 145-176.

Fikes, R.E., & Nilsson, N.J. (1971). STRIPS: A new approach to the application of theorem proving to problem solving. *Artificial Intelligence,* **2**, pp. 189-208.

Fikes, R.E., Hart, P.E., & Nilsson, N.J. (1972). Learning and executing generalized robot plans. *Artificial Intelligence*, **3**, pp. 251-288.

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

Knoblock, C. A. (1990). Learning abstraction hierarchies for problem solving. *Proceedings Eight National Conference on Artificial Intelligence*, **2**.

Korf, R.E. (1988). Optimal path-finding algorithms. In: *Search in Artificial Intelligence*. Kanal, L., Kumar, V. (eds.), Springer, NY.

Mädler, F. (1991). Problemzerlegung als optimalitätserhaltende Operatorabstraktion. In: GWAI-91, 15. Fachtagung für Künstliche Intelligenz. Th. Christaller (ed.). Springer, Berlin.

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

Musen, M.A., Fagan, L.M., Combs, D.M., & Shortliffe, E.H. (1987). Use of a domain model to drive an interactive knowledge editing tool. *International Journal of Man-Machine Studies,* **26**, 1, pp. 105-121.

Schmalhofer, F., Bergmann, B., Kühn ,O. & Schmidt, G. (1991a) Using integrated knowledge acquisition to prepare sophisticated expert plans for their re-use in novel situations. In: GWAI-91, 15. Fachtagung für Künstliche Intelligenz. Th. Christaller (ed.). Springer, Berlin.

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

Schmalhofer, F., & Thoben, J. (this volume). The model-based construction of a Case-Oriented Expert System. *Contemporary Knowledge Engineering and Cognition*. F. Schmalhofer, G. Strube & Th. Wetter (eds.).

Schmidt, G., & Schmalhofer, F. (1990). Case-oriented knowledge acquisition from text. In: *Current Trends in Knowledge Acquisition*, Wielinga, B., Boose, J., Gaines, B., Schreiber, G., van Someren, M. (eds.), IOS Press, May 1990, pp. 302-312.

Shortliffe, E.H., Scott, A.C. Bischoff, M.B., Campbell, A.B., Melle, W., & Jacobs, C.D. (1981). ONCOCIN: An expert system for oncology protocol management. *Proceedings of 7th International Joint Conference on Artificial Intelligence*, *Vancouver, Canada,* pp. 878-881.

Stefik, M. (1981). Planning with constraints (MOLGEN: part 1). *Artificial Intelligence* **16**, pp.111-139.

Tu, S.W., Kahn, M.G., Musen, M.A., Ferguson, J.C., Shortliffe, E.H., & Fagan, L.M. (1989). Episodic skeletal-plan refinement based on temporal data. *Communications of the ACM,* **32,** 12, pp. 1439-1455.