Deutsches
Forschungszentrum
für Künstliche
Intelligenz GmbH

Technical Memo
TM-92-06

Hierarchical skeletal plan refinement: Task- and inference structures

Otto Kühn, Franz Schmalhofer

June 1992

Deutsches Forschungszentrum für Künstliche Intelligenz GmbH

Postfach 20 80
D-6750 Kaiserslautern, FRG
Tel.: (+49 631) 205-3211/13
Fax: (+49 631) 205-3210

Stuhlsatzenhausweg 3
D-6600 Saarbrücken 11, 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 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

Hierarchical skeletal plan refinement: Task- and inference structures

Otto Kühn, Franz Schmalhofer

DFKI-TM-92-06

A slightly modified version of this paper was printed in: C. Bauer, W. Karbach (Eds.), "Materials for the second KADS User Meeting: Task and inference structures", Siemens AG, Munich 1992.

This work has been supported by a grant from The Federal Ministry for Research and Technology (FKZ ITW-8902 C4).

Contents

Hierarchical Skeletal Plan Refinement: Task- and Inference Structures

Otto Kühn & Franz Schmalhofer

DFKI GmbH, Postfach 2080, 6750 Kaiserslautern, Germany

Phone: +49-631-205-3463,

Fax: +49-631-205-3210

email: kuehn@dfki.uni-kl.de

Abstract:
This paper presents the task- and inference structure for skeletal plan refinement which was developed for lathe production planning, the application domain of the ARC-TEC project. Two inference structures are discussed: a global inference structure which was developed in the first phase of knowledge acquisition and a more detailed inference structure which builds on the hierarchical organization of the skeletal plans. The described models are evaluated with respect to their cognitive adequacy and their scope of application. The benefits and limitations of the KADS knowledge acquisition methodology are discussed with respect to the development of the two models.

1. Background of the Inference Structure

The reported inference structure evolved from the application domain of the ARC-TEC project (Bernardi et al., 1991). The ARC-TEC project investigates and develops artificial intelligence technology so that it can be used for achieving intelligent and thereby better solutions to mechanical engineering problems, in particular computer integrated manufacturing (CIM). Artificial Intelligence is thus applied to a complex real world problem.

While previous expert systems could only solve problems from a single task category (such as technical design), the ARC-TEC project strives for an integrated approach on the basis of which two or several related problems (e.g. design of a workpiece and production planning) can be concurrently solved.

This ambitious goal can only be achieved by sophisticated 2nd generation expert system methodology with an extensive conceptualization phase, on which the actual system development is then founded. During this phase an explicit model of the future system is to be developed, so that the system performance can be easily explained. Such knowledge level analyses and the development of a corresponding knowledge acquisition method with appropriate knowledge acquisition tools is performed by the acquisition group. The integrated representations of manufacturing products and the representations for performing the various tasks in CIM are developed by the representation group. The compilation group is concerned with respective compilative implementations and expert system tools in general. The division of the ARC-TEC projects into three groups is thus inspired by the KADS methodology which suggests that conceptual, representation and implementation issues should be treated separately.

In developing the integrated expert system, ARC-TEC is pursuing the following approach: Initially, only a single task category is being analyzed and implemented, namely the task of production planning for rotational parts. This planning system is, however, being developed with the objective of later extending it to a second task category. Special attention is therefore given to the maintainablity and explainability of the intitial knowledge-based system. Through the addition of a second task category to the initial system, an integrated expert system for two task categories should then be achievable without major problems.

The technique for manufacturing a rotational part is best understood by a comparison to pottery. The manufacturing processes are similar to making a pot in the following way: 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.

The upper half of Figure 1 shows a graphical representation of a (partial) workplan for a rotational part. The geometric form of the mold and the target workpiece are overlaid and shown at the top part of the figure (in the middle). 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.).

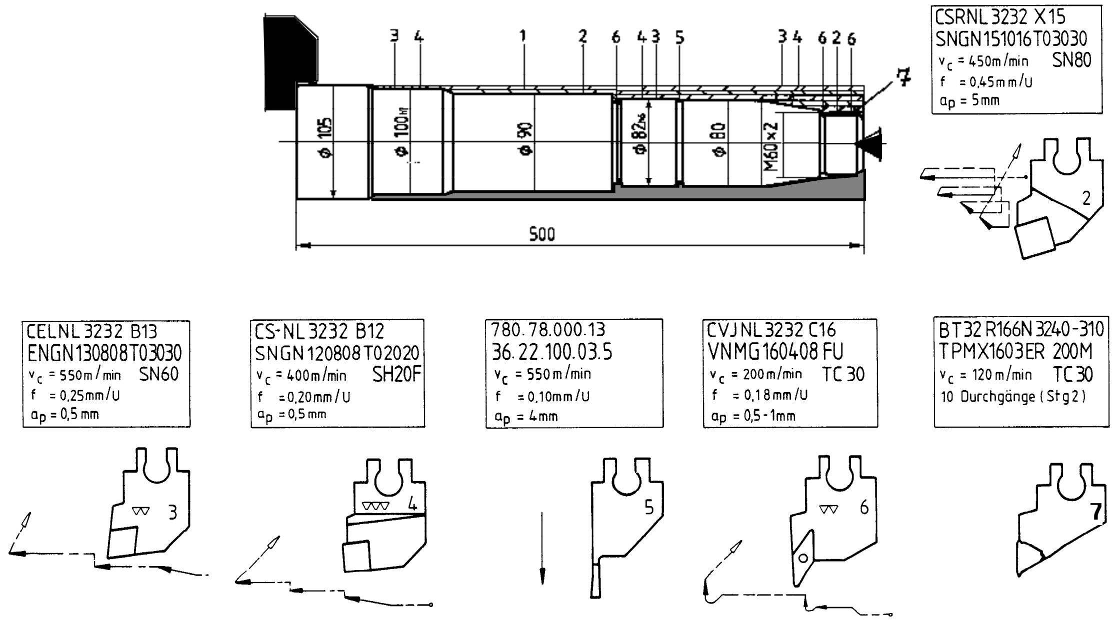

Figure 1: Workplan for manufacturing a drive shaft

The production plan must consequently 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, rarely 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. The quality of the resulting plan can be very high whereas the planning processes themselves are not completely knowledge-based in the sense that the plans have to be developed to some degree by trial and error in the real world.

An adequate problem solving method for production planning in mechanical engineering must take into account the enormous complexity of this real world domain. Traditional planning methods, such as generating and testing various sequences of actions or pure hierarchical planning, are bound to fail due to the exorbitant number of possible operations and the various requirements which a good plan must fulfill.

The planning method of hierarchical skeletal plan refinement, which will be described in this paper, not only meets the mentioned requirements, but at an abstract level it also reflects the expert's problem solving. This is important for the development of a cognitively adequate expert system which is likely to be accepted by the user (Strube, 1991). Furthermore, the knowledge which is needed for the application of this problem solving method can be readily acquired from the informational sources which are available in the domain (see Schmalhofer, Kühn, & Schmidt, 1991).

2. Inference Layer

2.1 Inference Structure

We will show two inference structures: a global inference structure, which was developed in the first phase of knowledge acquisition, and a more detailed inference structure, which builds on the hierarchical organization of the skeletal plans.

2.1.1 Global Inference Structure

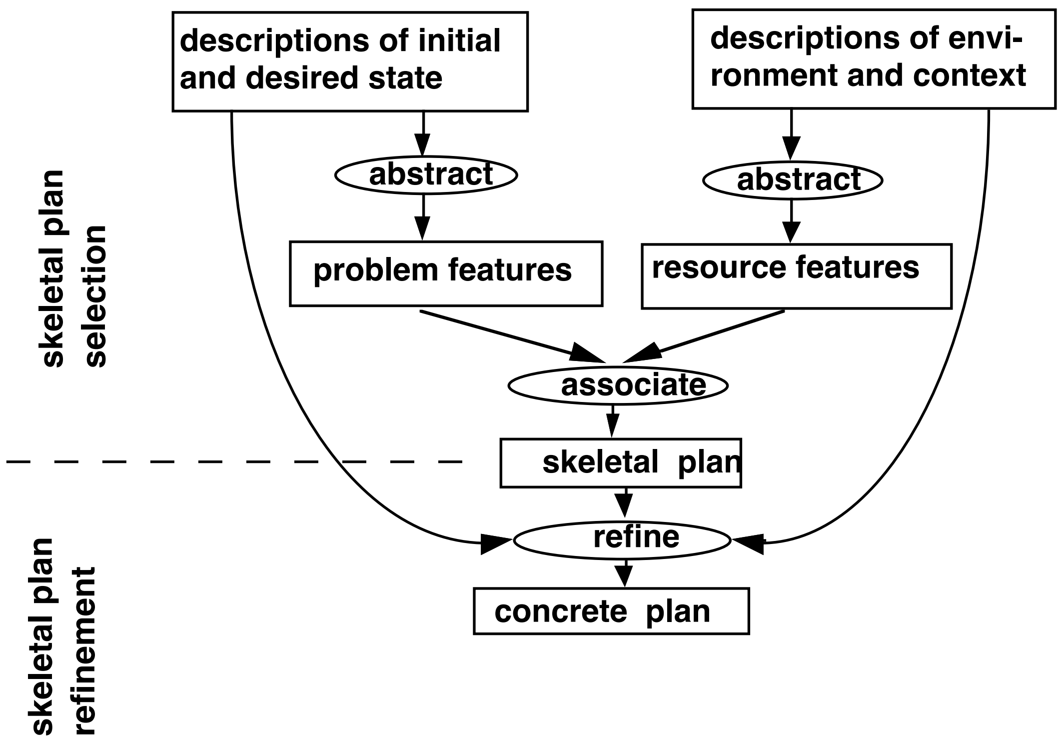

Figure 2: Global inference structure for skeletal plan refinement

The global inference reflects the expert´s problem solving which typically consists of two phases: the selection of a skeletal plan and its subsequent refinement.

When selecting a skeletal plan, the expert not only considers the descriptions of the goal workpiece and of the mold from which the workpiece is to be manufactured, but also takes into account the given environment and the context in which the manufacturing process takes place. As the expert´s verbalizations show, the selection of a skeletal plan is based on abstract features of the problem description and of the available resources specified in the environment and context description.

The selected skeletal plan is an abstract sketch of the intended manufacturing process. An executable production plan is obtained by refining the skeletal plan with respect to the concrete data given in the problem and resource descriptions. Only the detailed inference structure shows, how the hierarchies of problem classes and associated skeletal plans are used in hierarchical skeletal plan refinement.

2.1.1 Detailed Inference Structure

Figures 3a an 3b show the detailed inference structure for hierarchical skeletal plan refinement. The grey shaded metaclasses indicate what knowledge is required for applying the problem solving method. The internal structure of the more complex metaclasses is indicated by nested boxes.

The detailed inference structure was obtained through a more detailed analysis of the domain knowledge which falls into the global metaclasses and of the processes which are specified in the global knowledge sources.

For instance, it was determined that the two abstraction knowledge sources of the global inference structure should be realized as a classification of the problem description with respect to a hierarchy of problem classes. The associate knowledge source then retrieves a skeletal plan which is associated with a subset of the features defining the problem class.

The other features are relevant for the refinement of the skeletal plan which in the detailed inference structure is performed by four knowledge sources: The skeletal plan is first specified by iteratively selecting the most constrained abstract operator and specifying it according to the hierarchy of operators. The specified operators are assembled into a general plan. The parameters of the general plan are then instantiated with respect to the data provided in the problem description. Thereby, a concrete, executable plan is obtained.

2.2 Input and Output

The input to the above inference structure is given in the metaclass Problem Description which is subdivided into the four metaclasses initial state, goal state, environment, and context..

The output is the metaclass concrete plan consisting of a sequence of executable operations. These operations will transform the given initial state into the specified goal state, taking into account the limited resources of the environment and the context in which the plan is to be executed.

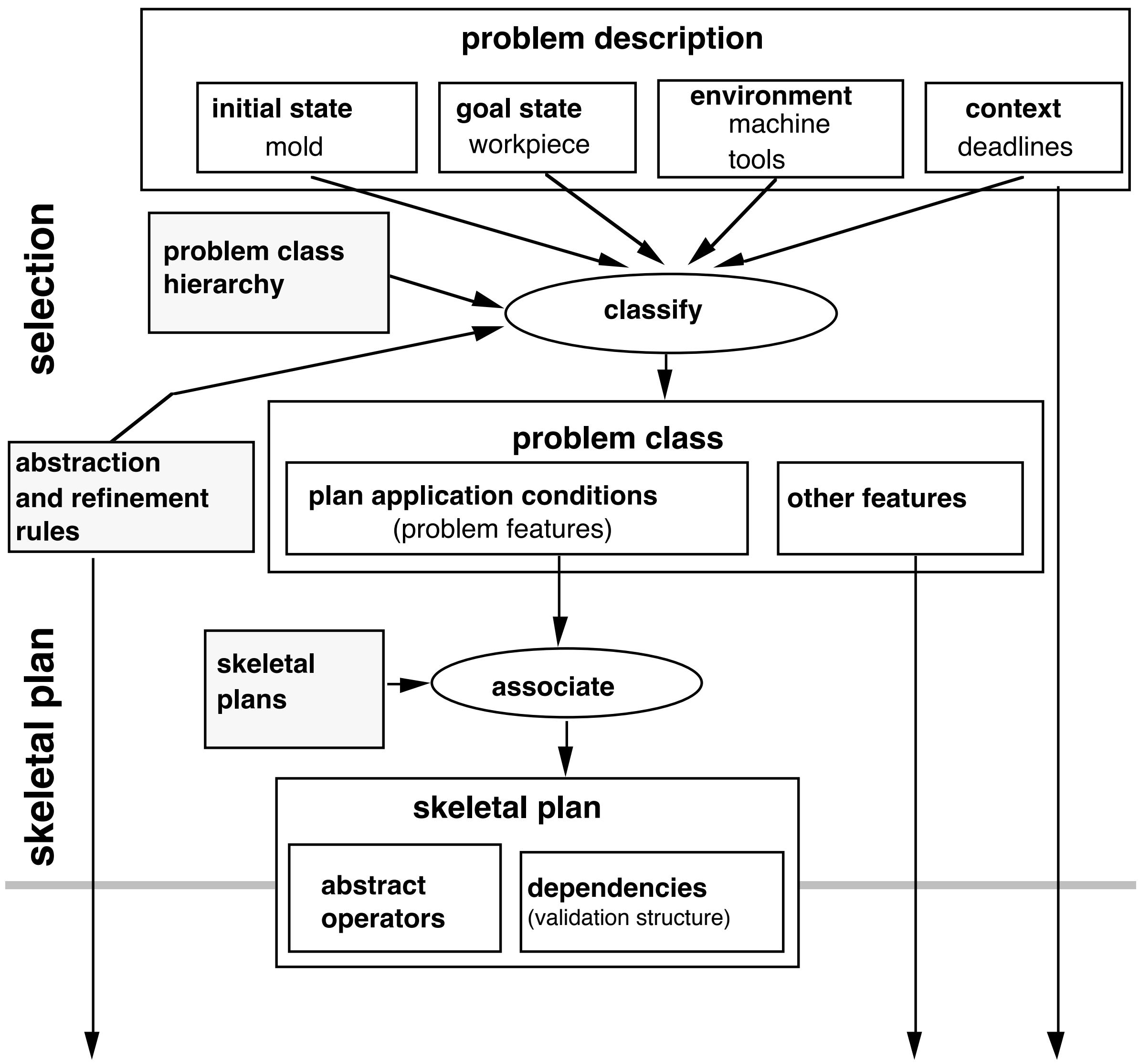

Figure 3a: Detailed inference structure for skeletal plan selection, the first subtask in hierarchical skeletal plan refinement

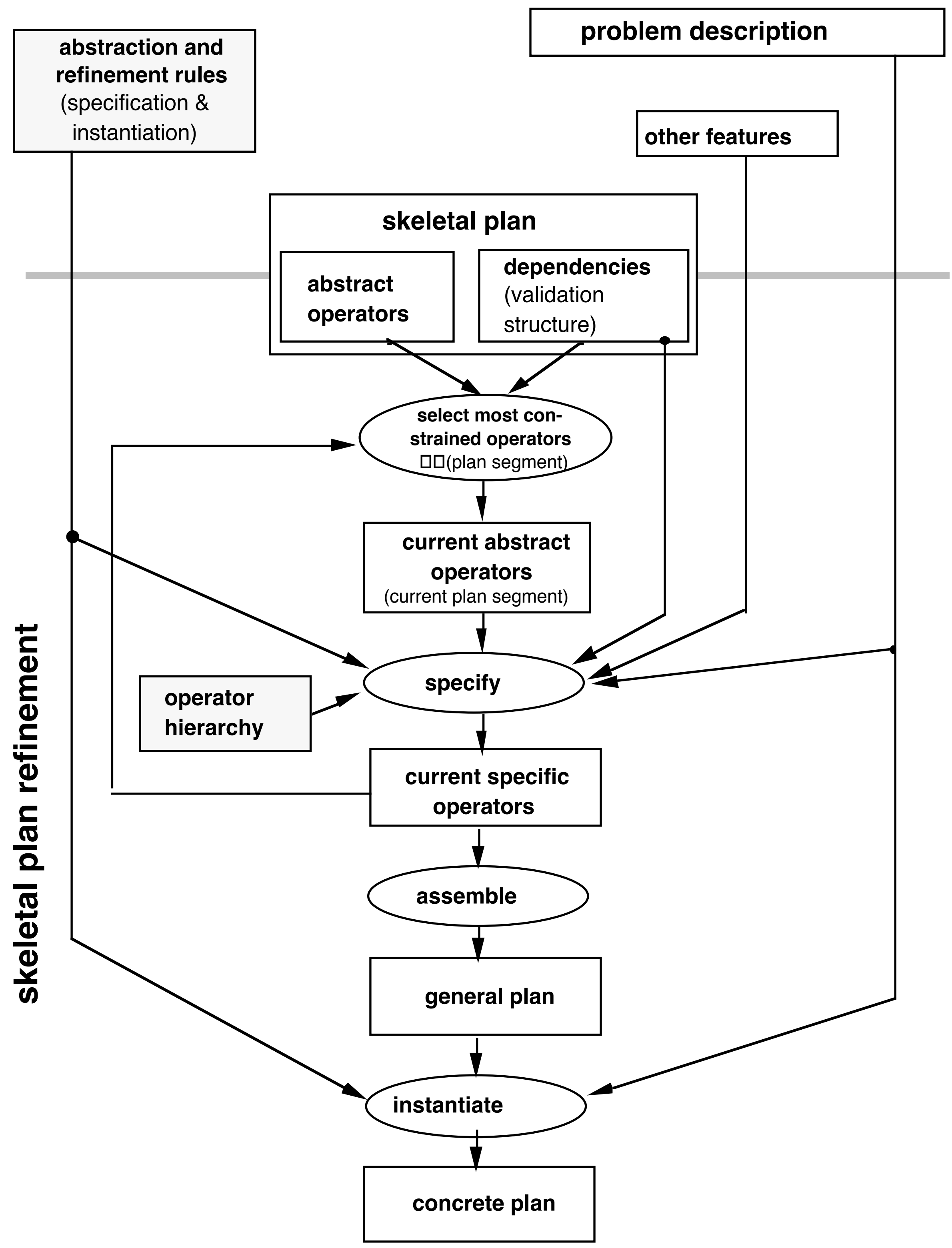

Figure 3b: Detailed inference structure for refining a skeletal plan, the second subtask in hierarchical skeletal plan refinement

2.3 Description of Metaclasses

2.3.1 Problem Description

The metaclass problem description contains all the informations which are relevant for the solution of the planning problem. Since this information falls into four different categories, the metaclass problem description is subdivided accordingly.

The metaclass initial state contains the description of the situation in which the planning problem arises. The domain elements in this metaclass are those facts which are true in the beginning but are bound to change over time, such as the description of the mold in production planning, or the initial location of the blocks in a robot planning problem.

The metaclass goal state specifies the desired state which is to be attained by the execution of the requested plan. In production planning, this metaclass contains the description of the to be manufactured workpiece. In a robot planning problem, this metaclass would contain the desired configuration of blocks on the table.

The metaclass environment contains the description of the environment in which the plan is to be executed. Contrary to the metaclass initial state, the facts in environment are static and cannot be changed by any action of the agent executing the plan. In production planning, the metaclass environment holds the description of the shopfloor, in particular, the machines and tools which are available. In a robot problem, the constellation of rooms, doors, light-switches, etc. would belong into this metaclass.

The metaclass context comprises a description of the additional requirements which must be fulfilled by a plan in order to be acceptable. Such requirements are delivery deadlines and manufacturing costs in production planning, or travel times and energy consumption in robot planning.

2.3.2 Problem Class

The metaclass problem class contains abstract features which characterize a number of cases with a common or similar solution. Some of these features are so called problem features since they can be obtained from the problem description. These features serve as plan application conditions for the identification of an appropriate skeletal plan. Typical plan application conditions are such as 'long workpiece', 'stiff machine', 'ceramic cutting tools available', etc.

The metaclass other features contains features which cannot be derived from the problem description alone, since they also refer to the solution of the problem, i.e. the plan. These features are acquired together with the problem features when the problem classes are defined with the knowledge acquisition tool CECoS (see Schmalhofer, 1991). They play an important role in refining the skeletal plan.

2.3.3 Skeletal Plan

The metaclass skeletal plan contains an abstract specification of the to be constructed plan. It consists of a set of abstract operators and a dependency graph or validation structure (Kabhampati, 1989) which specifies the dependencies and thus a partial order of the abstract operators.

2.3.4 Current abstract and specific operators

The metaclass current abstract operators contains a subset of operators from the skeletal plan with many dependencies between each other but few dependencies to other operators.

The metaclass current specific operators holds these operators after their specification.

2.3.5 General and concrete plan

The general plan consists of a sequence of operators with parameters. The concrete plan consists of the same sequence of operators with concrete values substituted for the parameters.

2.4 Description of the Knowledge Sources

2.4.1 Classify

The knowledge source classify takes as input the problem description given in the four metaclasses initial state, goal state, environment, and context and returns the most specific problem class into which the current problem falls. It requires a hierarchy of problem classes provided by the domain theory.

Classify performs a hierarchical classification. It starts with the most general problem class (the root of the tree), and at each node it tests the features of the subordinate nodes, in order to decide along which branch to proceed. Since the hierarchy of problem classes is not assumed to be complete, the classification process will not necessarily end at a leaf of the tree.

Abstraction and refinement rules are used for testing which abstract features are present or absent in the given problem. These rules relate the abstract features to the concrete problem description.

2.4.2 Associate

The knowledge source associate takes as input a problem class which is defined by a number of features and provides as output the skeletal plan which is associated with that problem class. The skeletal plans are acquired for each problem class with the help of the tool SP-GEN (Schmalhofer, Bergmann, Kühn & Schmidt, 1991) so that there will always be exactly one skeletal plan associated with a given problem class.

2.4.3 Select most constrained operators

This knowledge source takes as input the skeletal plan and the already specified operators. It returns the subset of abstract operators from the skeletal plan which has strong interdependencies and is most constrained with respect to the current specific operators.

2.4.4 Specify

The knowledge source specify takes as input the current abstract operators and returns the current specific operators. In the specification process an abstract operator may be replaced by one or more specific operators. The specification process is thus similar to task expansion in hierarchical planning approaches. The specification is guided by the dependencies in the skeletal plan, the other features in the problem class and the concrete data in the problem description. It requires an abstraction hierarchy of operators and the abstraction and refinement rules which must be part of the domain theory.

2.4.5 Assemble

The knowledge source assemble collects the specified operators and builds the general plan with the structure of the abstract skeletal plan being maintained.

2.4.6 Instantiate

The knowledge source instantiate determines the appropriate parameters for the operators in the general plan. Parameter values are determined with the help of the abstraction and refinement rules specified in the domain theory taking in to account the concrete data given in the problem description.

3. Task Layer

The subtask decomposition graph for hierarchical skeletal plan refinement is shown in figure 4, and the control flow is given in table 1 following the notation of Wielinga et al., 1991.

There are two fundamental subtasks which must be solved in hierarchical skeletal plan refinement: a skeletal plan must be found and it must be refined.

In order to find a skeletal plan, the new planning problem is classified with respect to the domain-specific hierarchy of problem classes and skeletal plan associated with the identified problem class is retrieved.

The refinement of the skeletal plan is performed in two steps: in the first step the skeletal plan is specified, in the second step it is instantiated. The specification is performed by repeatedly selecting a suitable subset of abstract operators from the skeletal plan, specifying the selected operators, and assembling them into a specific plan.

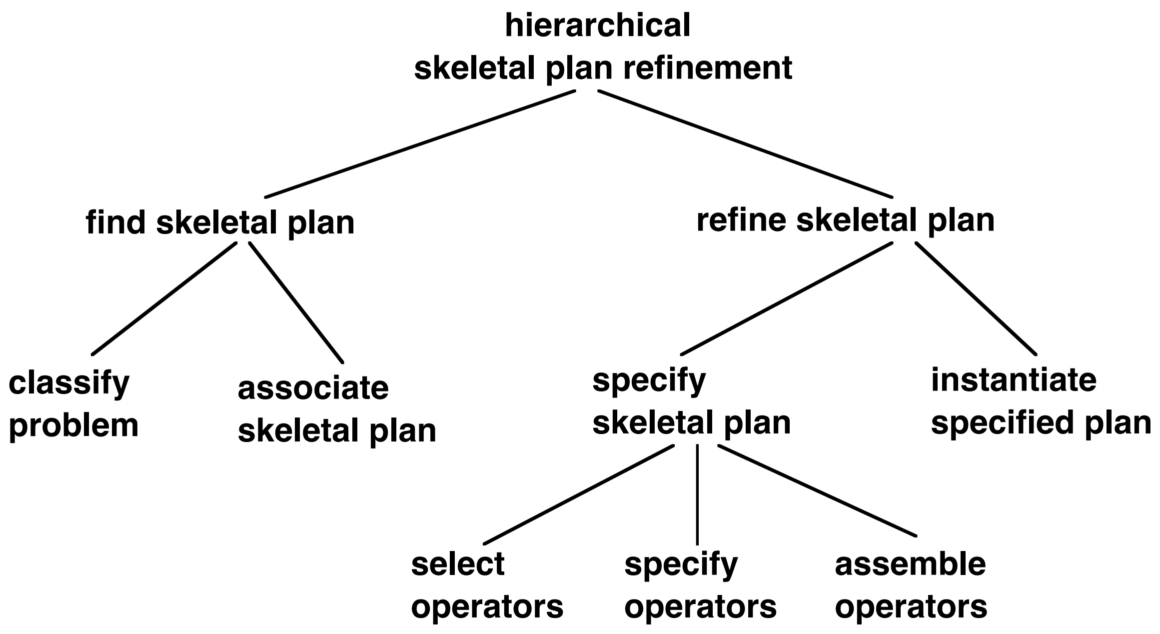

Figure 4: Subtask decomposition graph for hierarchical skeletal plan refinement

```
task  structure
   hierarchical skeletal plan refinement(problem description --> concrete plan) =
      classify(problem description --> problem class)
      associate(problem class --> skeletal plan)
      repeat
         select most constrained operators(skeletal plan --> current abstract operators)
         specify(current abstract operators --> current specific operators)
      until no more abstract operators left
      assemble(current specific operators --> specified plan)
      instantiate(specified plan --> concrete plan)
```

Table 1: Control flow for hierarchical skeletal plan refinement

4. Implementation

The described task and inference structures relate to the performance component of the ARC-TEC system. The performance components of the ARC-TEC system are implemented by the representation and compilation groups.

In the knowledge acquisition group, a procedure for generating skeletal plans with application conditions from concrete cases has previously been developed and implemented (Schmalhofer et al., 1991). The skeletal plan generation procedure shares several metaclasses and knowledge sources with the described problem solving method so that there is evidence that the implementation is feasible.

It is expected that the final implementation of the ARC-TEC system will at least globally reflect the structure of the conceptual model. Whether the refinement subtask will be implemented as described is still an open question, since efficiency considerations might suggest a slightly modified procedure.

5. Evaluation

There are only very few interpretation models for skeletal plan refinements. The interpretation model for episodic skeletal plan refinement of Linster & Musen (1991) describes how sequences of actions can be instantiated. Friedland & Iwasaki (1985) are also only concerned with the instantiation of skeletal plans.

Because of the complexity of our application domain, we needed to look at plans and the dependencies of their operators (skeletal plans) at different levels of abstraction. To our knowledge, an interpretation model for hierarchical skeletal plan refinement has not yet been developed. In developing the interpretation model for hierarchical skeletal plan refinement, we were mostly inspired by developments from case-base reasoning (Riesbeck & Schank, 1989). Rather than storing plans, our knowledge base consists of hierarchically structured problem classes and associated skeletal plans. Similar to Kambanphati's validation structures, the skeletal plans contain a dependency graph, which represents the interactions between the different plan operators at the various levels in the abstraction hierarchy. Although there are a number of similarities to case-based planning systems (in particular to Alterman's (1988) PLEXUS), which utilizes abstraction hierarchies for operators, the use of complete abstract skeletal plans which first need to be specified before they can be instantiated has not yet been developed as an AI problem solving method.

At some level of abstraction, the described inference structure describes an expert's behavior quite well (Thoben & Schmalhofer, 1991). At a more detailed level there are a number of differences (Schmidt, Legleitner & Schmalhofer, 1990). The inference structure was developed with the intention to describe the system at some level of abstraction, so that the system description would coincide with an abstract model of the expert's behavior.

The conceptual model was developed by the knowledge engineer in consultation with the domain expert. The global inference structure for hierarchical skeletal plan refinement was inspired by the inference structure for heuristic classification. This shows that an inference structure which was developed for a particular type of problems can nevertheless stimulate the development of an inference structure for a completely different problem.

The KADS typology of knowledge sources was very helpful and it was used extensively in the development of the inference structure. The described inference structure was developed by starting with a global structure and adding more and more detail. This process should continue with the evolution of the knowledge base and the implementation of the system so that the final inference and task structures are sufficiently detailed to support the maintenance of the expert system.

The suggested interpretation model is not limited to the domain of production planning in mechanical engineering. It seems applicable to a wide range of complex real world planning problems in which the effects of an agent´s actions can be reliably predicted.

Bibliography

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

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 Proceedings of the Eleventh International Conference Expert Systems & Their Applications (pp. 133-145). Avignon, France.

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

Linster, M. & Musen, M.A. (1991). Using KADS to build K-ONCOCIN: A conceptual model of ONCOCIN. In Proceedings of the 6th Banff Knowledge Acquisition for Knowledge-Based Systems Workshop (pp. 13-1-13-20). Banff, Alberta, Canada.

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

Schmalhofer, F., Bergmann, R., Kühn, O., & Schmidt, G. (1991). 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.

Schmidt, G., Legleitner, R., & Schmalhofer, F. (1990). Lautes Denken bei der Erstellung der Schnittaufteilung, der Werkzeugauswahl und Festlegung der Maschineneinstelldaten (Interner Bericht des ARC-TEC Projektes). Kaiserslautern, Germany: German Research Center for Artificial Intelligence.

Strube, G. (1992). The role of cognitive science in knowledge engineering. In Schmalhofer, F., Strube, G., & Wetter, T. (Eds.), Contemporary Knowledge Engineering and Cognition. Berlin/Heidelberg: Springer-Verlag.

Thoben, J. & Schmalhofer, F. (1990). Wiederholungs- Varianten- und Neuplanung bei der Fertigung rotationssymmetrischer Drehteile (DFKI-Document No. 16-91). Kaiserslautern, Germany: German Research Center for Artificial Intelligence.

Wielinga, B.J., Schreiber, A.T., & Breuker, J.A. (in press). KADS: A modelling approach to knowledge engineering. Knowledge Acquisition,

Deutsches Forschungszentrum für Künstliche Intelligenz GmbH

DFKI
-Bibliothek-
PF 2080
67608 Kaiserslautern
FRG

DFKI Publikationen

Die folgenden DFKI Veröffentlichungen sowie die aktuelle Liste von allen bisher erschienenen Publikationen können von der oben angegebenen Adresse oder per anonymem ftp von ftp.dfki.uni-kl.de (131.246.241.100) unter pub/Publications bezogen werden.
Die Berichte werden, wenn nicht anders gekennzeichnet, kostenlos abgegeben.

DFKI Publications

The following DFKI publications or the list of all published papers so far are obtainable from the above address or via anonymous ftp from ftp.dfki.uni-kl.de (131.246.241.100) under pub/Publications.
The reports are distributed free of charge except if otherwise indicated.

DFKI Research Reports

RR-93-14
Joachim Niehren, Andreas Podelski, Ralf Treinen:
Equational and Membership Constraints for Infinite Trees
33 pages

RR-93-15
Frank Berger, Thomas Fehrle, Kristof Klöckner, Volker Schölles, Markus A. Thies, Wolfgang Wahlster: PLUS - Plan-based User Support Final Project Report
33 pages

RR-93-16
Gert Smolka, Martin Henz, Jörg Würtz: Object-Oriented Concurrent Constraint Programming in Oz
17 pages

RR-93-17
Rolf Backofen:
Regular Path Expressions in Feature Logic
37 pages

RR-93-18
Klaus Schild: Terminological Cycles and the Propositional μ-Calculus
32 pages

RR-93-20
Franz Baader, Bernhard Hollunder:
Embedding Defaults into Terminological Knowledge Representation Formalisms
34 pages

RR-93-22
Manfred Meyer, Jörg Müller:
Weak Looking-Ahead and its Application in Computer-Aided Process Planning
17 pages

RR-93-23
Andreas Dengel, Ottmar Lutzy:
Comparative Study of Connectionist Simulators
20 pages

RR-93-24
Rainer Hoch, Andreas Dengel:
Document Highlighting —
Message Classification in Printed Business Letters
17 pages

RR-93-25
Klaus Fischer, Norbert Kuhn: A DAI Approach to Modeling the Transportation Domain
93 pages

RR-93-26
Jörg P. Müller, Markus Pischel: The Agent Architecture InteRRaP: Concept and Application
99 pages

RR-93-27
Hans-Ulrich Krieger:
Derivation Without Lexical Rules
33 pages

RR-93-28
Hans-Ulrich Krieger, John Nerbonne, Hannes Pirker: Feature-Based Allomorphy
8 pages

RR-93-29
Armin Laux: Representing Belief in Multi-Agent Worlds viaTerminological Logics
35 pages

RR-93-30
Stephen P. Spackman, Elizabeth A. Hinkelman:
Corporate Agents
14 pages

RR-93-31
Elizabeth A. Hinkelman, Stephen P. Spackman: Abductive Speech Act Recognition, Corporate Agents and the COSMA System
34 pages

RR-93-32
David R. Traum, Elizabeth A. Hinkelman: Conversation Acts in Task-Oriented Spoken Dialogue
28 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-93-43
M. Bauer, G. Paul: Logic-based Plan Recognition for Intelligent Help Systems
15 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

RR-93-48
Franz Baader, Martin Buchheit, Bernhard Hollunder: Cardinality Restrictions on Concepts
20 pages

RR-94-01
Elisabeth André, Thomas Rist: Multimedia Presentations: The Support of Passive and Active Viewing
15 pages

RR-94-02
Elisabeth André, Thomas Rist: Von Textgeneratoren zu Intellimedia-Präsentationssystemen
22 Seiten

RR-94-03
Gert Smolka: A Calculus for Higher-Order Concurrent Constraint Programming with Deep Guards
34 pages

RR-94-05
Franz Schmalhofer, J.Stuart Aitken, Lyle E. Bourne jr.: Beyond the Knowledge Level: Descriptions of Rational Behavior for Sharing and Reuse
81 pages

RR-94-06
Dietmar Dengler: An Adaptive Deductive Planning System
17 pages

RR-94-07
Harold Boley: Finite Domains and Exclusions as First-Class Citizens
25 pages

RR-94-08
Otto Kühn, Björn Höfling: Conserving Corporate Knowledge for Crankshaft Design
17 pages

RR-94-10
Knut Hinkelmann, Helge Hintze: Computing Cost Estimates for Proof Strategies
22 pages

RR-94-11
Knut Hinkelmann: A Consequence Finding Approach for Feature Recognition in CAPP
18 pages

RR-94-12
Hubert Comon, Ralf Treinen:
Ordering Constraints on Trees
34 pages

RR-94-13
Jana Koehler: Planning from Second Principles —A Logic-based Approach
49 pages

RR-94-14
Harold Boley, Ulrich Buhrmann, Christof Kremer:
Towards a Sharable Knowledge Base on Recyclable Plastics
14 pages

RR-94-15
Winfried H. Graf, Stefan Neurohr: Using Graphical Style and Visibility Constraints for a Meaningful Layout in Visual Programming Interfaces
20 pages

RR-94-16
Gert Smolka: A Foundation for Higher-order Concurrent Constraint Programming
26 pages

RR-94-17
Georg Struth:
Philosophical Logics—A Survey and a Bibliography
58 pages

RR-94-18
Rolf Backofen, Ralf Treinen:
How to Win a Game with Features
18 pages

RR-94-20
Christian Schulte, Gert Smolka, Jörg Würtz:
Encapsulated Search and Constraint Programming in Oz
21 pages

RR-94-31
Otto Kühn, Volker Becker, Georg Lohse, Philipp Neumann:
Integrated Knowledge Utilization and Evolution for the Conservation of Corporate Know-How
17 pages

RR-94-33
Franz Baader, Armin Laux:
Terminological Logics with Modal Operators
29 pages

DFKI Technical Memos

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

TM-93-05
Michael Sintek: Indexing PROLOG Procedures into DAGs by Heuristic Classification
64 pages

TM-94-01
Rainer Bleisinger, Klaus-Peter Gores:
Text Skimming as a Part in Paper Document Understanding
14 pages

TM-94-02
Rainer Bleisinger, Berthold Kröll:
Representation of Non-Convex Time Intervals and Propagation of Non-Convex Relations
11 pages

DFKI Documents

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-93-16
Bernd Bachmann, Ansgar Bernardi, Christoph Klauck, Gabriele Schmidt: Design & KI
74 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

D-93-22
Andreas Abecker:
Implementierung graphischer Benutzungsoberflächen mit Tcl/Tk und Common Lisp
44 Seiten

D-93-24
Brigitte Krenn, Martin Volk:
DiTo-Datenbank: Datendokumentation zu Funktionsverbgefügen und Relativsätzen
66 Seiten

D-93-25
Hans-Jürgen Bürckert, Werner Nutt (Eds.):
Modeling Epistemic Propositions
118 pages
Note: This document is available only for a nominal charge of 25 DM (or 15 US-$).

D-93-26
Frank Peters: Unterstützung des Experten bei der Formalisierung von Textwissen
INFOCOM:
Eine interaktive Formalisierungskomponente
58 Seiten

D-93-27
Rolf Backofen, Hans-Ulrich Krieger, Stephen P. Spackman, Hans Uszkoreit (Eds.):
Report of theEAGLES Workshop on Implemented Formalisms at DFKI, Saarbrücken
110 pages

D-94-01
Josua Boon (Ed.):
DFKI-Publications: The First Four Years
1990 - 1993
75 pages

D-94-02
Markus Steffens: Wissenserhebung und Analyse zum Entwicklungsprozeß eines Druckbehälters aus Faserverbundstoff
90 pages

D-94-03
Franz Schmalhofer: Maschinelles Lernen:
Eine kognitionswissenschaftliche Betrachtung
54 pages

D-94-04
Franz Schmalhofer, Ludger van Elst:
Entwicklung von Expertensystemen:
Prototypen, Tiefenmodellierung und kooperative Wissensevolution
22 pages

D-94-06
Ulrich Buhrmann:
Erstellung einer deklarativen Wissensbasis über recyclingrelevante Materialien
117 pages

D-94-07
Claudia Wenzel, Rainer Hoch:
Eine Übersicht über Information Retrieval (IR) und NLP-Verfahren zur Klassifikation von Texten
25 Seiten

D-94-08
Harald Feibel: IGLOO 1.0 - Eine grafikunterstützte Beweisentwicklungsumgebung
58 Seiten

D-94-09
DFKI Wissenschaftlich-Technischer Jahresbericht 1993
145 Seiten

D-94-10
F. Baader, M. Lenzerini, W. Nutt, P. F. Patel-Schneider (Eds.): Working Notes of the 1994 International Workshop on Description Logics
118 pages
Note: This document is available only for a nominal charge of 25 DM (or 15 US-$).

D-94-11
F. Baader, M. Buchheit, M. A. Jeusfeld, W. Nutt (Eds.):
Working Notes of the KI'94 Workshop:
KRDB'94 - Reasoning about Structured Objects: Knowledge Representation Meets Databases
65 Seiten

D-94-12
Arthur Sehn, Serge Autexier (Hrsg.): Proceedings des Studentenprogramms der 18. Deutschen Jahrestagung für Künstliche Intelligenz KI-94
69 Seiten