Deutsches
Forschungszentrum
für Künstliche
Intelligenz GmbH

Research
Report
RR-92-19

PIM:
Planning In Manufacturing using Skeletal Plans and Features

Ansgar Bernardi, Christoph Klauck, Ralf Legleitner

April 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 und Saarbrücken is a non-profit organization which was founded in 1988. The shareholder companies are Daimler Benz, Fraunhofer Gesellschaft, GMD, IBM, Insiders, Krupp-Atlas, Mannesmann-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

PIM:
Planning In Manufacturing using Skeletal Plans and Features

Ansgar Bernardi, Christoph Klauck, Ralf Legleitner

RR-92-19

A short version of this paper will be published in the Proceedings of the CARs & FOF '92 8th International Conference on CAD/CAM, Robotics and Factories of the Future, Metz. Some aspects of this paper are also presented in the Proceedings of the International Conference on Manufacturing Automation, August 1992, Hong Kong.

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

PIM:
Skeletal Plan based CAPP

Dipl.-Ing. R. **Legleitner,**

Dipl.-Inform. A. **Bernardi,**

Dipl.-Inform. C. **Klauck,**

ARC-TEC Project

DFKI GmbH, Postfach 2080, D-6750 Kaiserslautern, Germany

Telefon: +49631/205-3477, 205-4068

Fax: +49631/205-3210

Abstract

Abstract. In order to create a production plan from product model data, a human expert thinks in a special terminology with respect to the given work piece and its production plan: He recognizes certain features and associates fragments of a production plan. By combining these skeletal plans he generates the complete production plan.

We present a set of representation formalisms suitable for the modelling of this approach. When an expert's knowledge has been represented using these formalisms, the generation of a production plan can be achieved by a sequence of abstraction, selection and refinement. This is demonstrated in the CAPP-system PIM, which is currently developed as a prototype.

The close modelling of the knowledge of the concrete expert (or the accumulated know-how of a concrete factory) facilitate the development of planning systems which are especially tailored to the concrete manufacturing environment and optimally use the expert's knowledge and should also lead to improved acceptance of the system.

1 Advantages of knowledge-based systems - Problems in CAPP

Systems to support CAPP encounter a particular problem which arises out of the structure of this domain: In order to create a successful process plan it is not sufficient to rely on theoretically founded rules - which normally can be coded into a program quite easily - but it is also necessary to consider particularities of the production environment and special experience of human process planers. The latter is very often difficult to formalize and consequently hard to be coded, the former may change radically between different working environments and leads to a high maintenance workload.

Under this circumstances the knowledge-based programming methodology may be used: If it is possible to define domain-oriented languages facilitating the explicit representation of the knowledge in question and to provide adequate execution mechanisms for these languages the problem becomes treatable. Suitable representation languages lead to easier representation of the experience and environment-dependent knowledge. The execution mechanisms are independent of the environment and probably useful in a broader domain, thereby reducing the general maintenance workload.

In summary, knowledge-based programming may lead to higher flexibility of the system, easier adaptation to changing environments, reduced maintenance costs, and successful solutions of some problems which can't be tackled with standard techniques.

Several projects currently try to use knowledge-based methodologies in this context (cf. the ESPRIT.projects IMPACT, PLATO, FLEXPLAN [Espr89]). In the project ARC-TEC at the DFKI Kaiserslautern, these general ideas where exemplarily applied to the automatic generation of work plans for manufacturing by cutting with geometric defined cutting edge (DIN 8589 part 0), resulting in the prototypical system PIM (Planning In Manufacturing). The developed methodology is of interest for CAPP in general.

2 CAPP from the viewpoint of knowledge-based systems

In order to create a knowledge-based CAPP system, we examine the way a human expert solves the task in question. This leads to the following general model of a process-planning method:

The expert is given the description of the work piece. This description consists of all geometrical and technological data which are necessary for the generation of the process plan. In this description the expert identifies characteristic parts or areas which contain special information about the manufacturing process. According to these so-called application features, the expert selects out of his memory (or out of existing plan libraries) generalized plan fragments, the so-called skeletal plans. By combining these partial plans and by adapting them to the concrete work piece, he creates the complete production plan. [Teng84]

The combination of application features and associated skeletal plans represent the experience of the expert. (An expert may be defined as someone who knows "what to look for" and, having found this, what to do about it). It is important to realize that this observation implies that the application features and the skeletal plans depend on the concrete expert as well as on the concrete working environment and may be different for another expert or another environment.

From the viewpoint of knowledge-based programming the above description leads to the identification of several representation tasks. It is necessary to represent the work piece information,and the application features and the associated skeletal plans of an expert in a given working environment. Together with a execution mechanism which mimics the sketched method of the human expert, a flexible and reliable CAPP system is possible. [Bern91a] The following chapters describe the different representation languages which were developed in the ARC-TEC project according to figure 1.

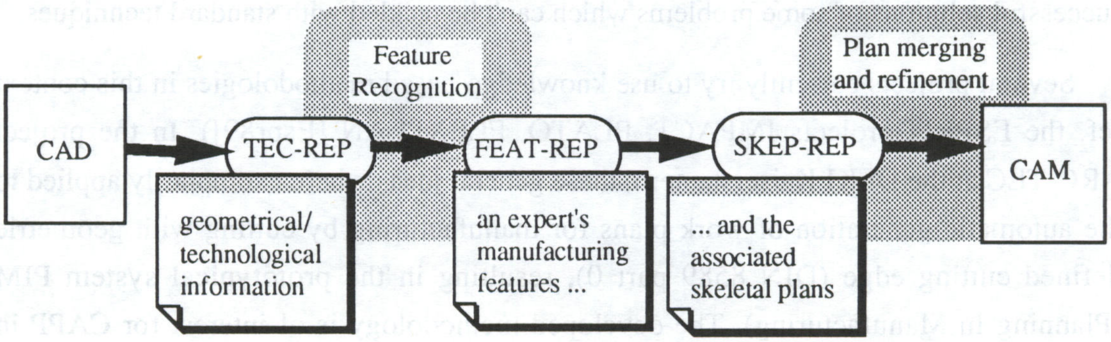


Figure 1: Basic idea and representation languages

3 TEC-REP

The representation formalism TEC-REP [Bern91b] provides the necessary constructs to describe the geometrical and technological information of the work piece. The description must contain all information about the work piece which an expert may use to create the process plan. In order to facilitate the necessary deductions the representation formalism should use symbolic expressions. As shown in a later chapter the TEC-REP formalism is well connected to the CAD-world.

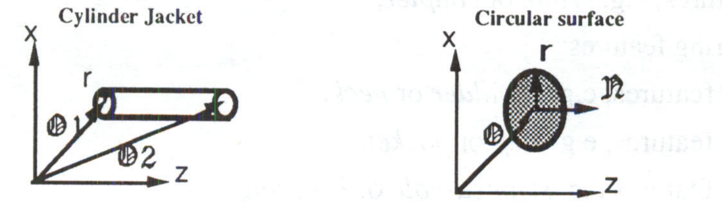
In TEC-REP, the geometry of a work piece is described by surface primitives. These predefined primitives include simple expressions for rotational symmetric parts, which are of special interest in the domain of manufacturing by turning. To be as universal as possible, primitives for non-symmetric surfaces also exist. The extensions of any surface are specified using a cartesian coordinate system. The concrete dimensions are specified for each work piece. Every surface of a work piece description can be identified by a unique identification number.

TEC-REP realizes a simplified boundary representation of the geometry of a work piece. While some surfaces like free formed surfaces which aren't considered in the ARC-TEC project currently can't be expressed, it is possible to expand TEC-REP to cover such cases e. g. by including primitives for surfaces described by B-spline-functions.

The reason to use a surface-based description method is simple: Every manufacturing operation always influences surfaces of an object, therefore the representation of these surfaces seems reasonable.

By using surface primitives, TEC-REP connects the basic data level with the world of symbolic reasoning. With respect to the intended deductions the surface primitives are augmented by information about the direction of material. A circular surface can represent the outer surface of a cylinder as well as a hole in some material, depending on the direction of material.

To facilitate reasoning about topological connections, the neighbourhood of surfaces is expressed in a separate primitive.



CJ(reference_point1: <vector>,
reference_point2: <vector>,
radius: <num>,
direction_of_material: [+,-])

C(reference_point: <vector>,
direction_vector: <vector>,
radius: <num>,
direction_of_material: [+,-])

Shape Tolerance DIN 7184

ST(surface_number: <num>,
tolerance_type: [SN, EN, RN, C, LD, PD],
tolerance_size: <num>)

Surface Finish

SF(surface_number: <num>,
surface_finish: [Rt, Rz, Rp, Ra,...],
value: <num>)

Figure 2. Some TEC-REP Primitives

The technological information is expressed by attributes to the surfaces. Some attributes (like surface finish or ISO-tolerances) belong to one surfaces, others connect different surfaces (e. g. length tolerances between different surfaces).

In summary, TEC-REP allows the description of every work piece we currently deal with by using a symbolic and attributed boundary representation.

4 Features and FEAT-REP - a closer look

The goal of the feature representation formalism is to allow the expression of a concrete expert's knowledge about characteristic aggregations of surfaces. When CAD/CAM experts view a workpiece, they perceive it in terms of their own expertise. These terms, called *features*, which are build upon a *syntax* (geometry and technology) and a *semantics* (e.g. skeletal plans in manufacturing or functional relations in design), provide an abstraction mechanism to facilitate the creation, manufacturing and analysis of workpieces. We define the term *feature* as a description element based on geometrical and technological data of a product which an expert in a domain associates with certain informations (see also [Klau91a] or [Klau91b]). They are firstly distinguished by their kind as

- functional features, e.g. *seat of the rolling bearing* or *O-ring groove*,
- qualitative features, e.g. *bars* or *solid workpiece*,
- geometrical (form-) features, e.g. *shoulder*, *groove* or *drilled hole*,
- atomic features, e.g. *toroidal shell*, *ring*, *shape tolerance* or *surface finish*.

and they are secondly distinguished by their application as

- design features, e.g. *crank* or *coupler*,
- manufacturing features:
  - turning features, e.g. *shoulder* or *neck*,
  - milling features, e.g. *step* or *pocket*,
  - drilling features e.g. *stepped-hole* or *lowering*,
  - ...
- ...

The domain-dependency of features can be illustrated by a comparison between features in the domain of CAPP and features in the domain of design. While the latter (which are investigated in many modern CAD-systems) represent higher-level informations about the ideas of a designer and describe things like functional relations between surfaces of different workpieces which touch each other, the former depend on

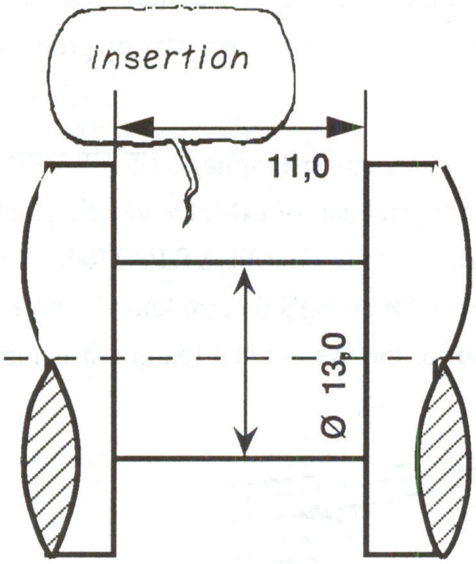


Figure 3.: Insertion

particularities of the manufacturing process and are influenced e.g. by the available manufacturing environment. In particular every feature e.g. in manufacturing will be defined by a respective expert because his area, like machines, tools or the characteristics of them, and his ideas, creativity and experience, like special tricks, are included in this definition. In this sense the features can been seen as a *language* of an expert in a domain. It is important to note that this language represents the *know-how* of the expert respectively the machine shop and that this language is an individual one ("expert in a domain" dependent). A prominent example of a such a feature in manufacturing is a insertion of certain dimensions which shall be manufactured using a certain insertion-tool. In a manufacturing environment where this tool is not available this particular insertion may be recognized as a groove or may be not recognized as a feature (figure 3).

The importance of feature recognition in manufacturing stems from the fact that each feature can be associated with knowledge about how the feature should be manufactured; this information can be used to generate a process plan. From this point of view feature recognition forms a major component of the CAD/CAM interface for CAPP [Chan90]. In our paper we concentrated on the recognition of geometrical and qualitative features; the functional features are important for design only. Working with manufacturing features means to recognize these features from the CAD data to generate a working plan. Working with design features means to construct by means of these features and to expand them to the CAD data. (cf. Klau91a)

In our paper [Klau91a] we have shown that it is possible to describe features by means of formal languages via attributed node-label-controlled graph grammars (ANLCGG's). The area of formal languages is a well established field of research and provides a powerful set of methods like parsing and knowledge about problems, their complexity and how they could be solved efficiently. The use of formal languages for feature descriptions facilitates the application of these results to the area of feature recognition and CAPP. As result ANLCGG's enables a user to define his own feature-language containing complex features and makes feature recognition a parsing process for workpiece interpretation.

To show the usability of our high-level-representation language FEAT-REP the FEAT-PATR-System was implemented as a prototypical part of PIM, by adopting a chart parser for our application in mechanical engineering. Input of our FEAT-PATR-System is a workpiece description in TEC-REP. Input is also the expert's feature knowledge about the workpiece, represented in a grammar. Output of the system is a feature-structure as shown in figure 4.

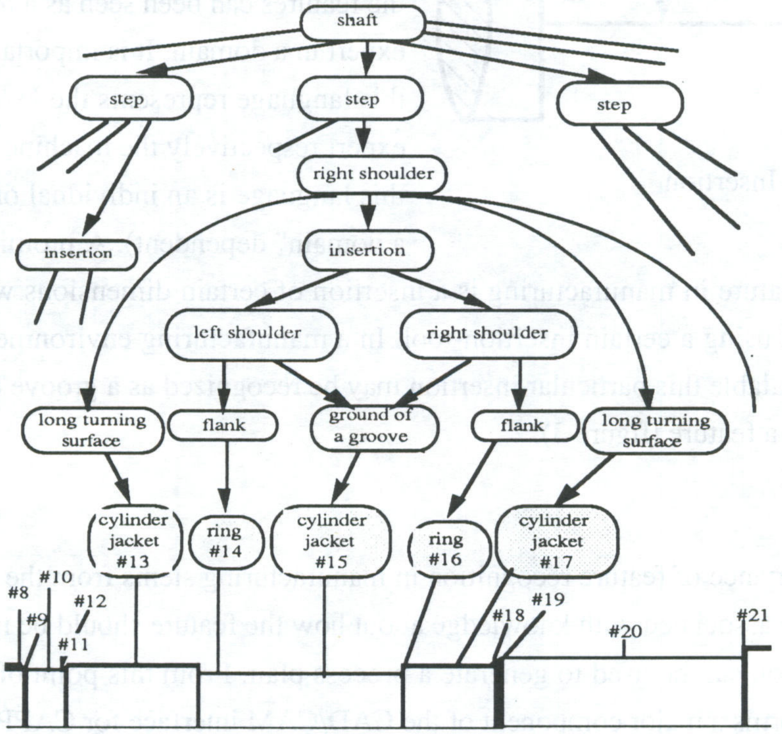


Figure 4.: Feature Structure

5 Skeletal Plans

To combine the expert's knowledge about the manufacturing process with feature structures as shown above we use skeletal plans [Richt91]|. The skeletal plan representation formalism SKEP-REP allows the expert to write down his knowledge about the process necessary for the manufacturing of his workpieces and for special parts of this workpieces (features). It also allows the expert to define how partial (skeletal) plans for special parts of workpieces (features) should be merged to complete manufacturing plans.

The idea of skeletal plans is straight forward [Frie85]: Represent knowledge about the intended solution in some partial (skeletal) plans. Find a first skeletal plan for the given goal. The plan contains some steps which are subgoals to be reached in order to solve the whole problem.

For every subgoal repeat the process until only elementary steps remain. The sequence of elementary steps is the intended plan. [Frie85]

In our domain the skeletal plans represent the expert's way of work and his knowledge. To every feature the expert associates knowledge about the manufacturing process. This may be some rather global information related to the whole workpiece, e. g. "given a workpiece of less than a certain length, use a chuck as the clamping method", as well as detailed instructions concerning small parts, e. g. "manufacture this groove using the special tool X". Such knowledge is represented in the skeletal plans. Every skeletal plan contains concrete and/or abstract information about parts of the manufacturing process or about the whole process. Every skeletal plan is associated to some feature. Feature definition and skeletal plan together represent the expert's knowledge about a certain planning task.

The concrete representation language SKEP-REP is still under development. We currently use a formalism which describes skeletal plans as shown in the following figure. A skeletal plan contains the feature or feature structure it is associated to. It then contains some context information which relates to other skeletal plans which form preconditions for the application of this particular plan. It may also contain some applicability constraints which are not expressed by the features or the context of skeletal plans. Then it contains a sequence of operations, which may result in the subroutine-like call of other skeletal plans or in the generation of concrete planning steps. In the domain of manufacturing by turning, concrete planning steps are chucking commands, cut instructions and tool selections. This steps are represented as primitives of our abstract NC-programm representation language LANCO (LAnguage for Abstract Numerical Control). Every operation may access the concrete technological and geometrical informations, especially measurements, which are represented in the TEC-REP of the surfaces which form the features associated with the skeletal plan or with the plans of the context (see above). Beside this, no information about the workpiece can be accessed. This realizes the concepts of modularity and information hiding for the skeletal plans and makes it possible to create skeletal plans for for a large bandwith of workpieces.

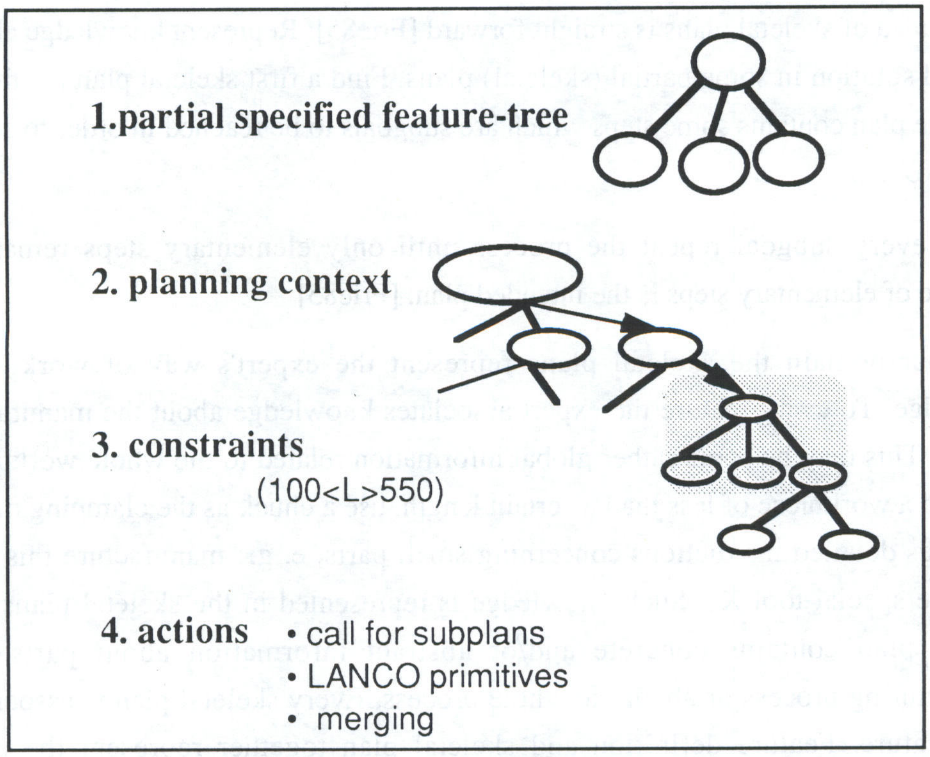


Figure 5.: Skeletal Plan Representation

Some of the operations can result in the subroutine-call of special programs for particular tasks. In our prototype, the tool selection operation uses a constraint system to find the suitable group of tools for the intended operation.

The following figure shows an example for a skeletal plan. The partial specified feature tree of this plan is a shaft which consists of crest-lstep-rtrunion. In the lower part of the figure the merging operation for the sequential planned cuts is shown.

```
(make-skp
  :special-tree
   '("shaft" shaft                                          ; partial specified feature tree
      (("crest" crest nil)
       ("lstep" lstep nil)
       ("rtrunnion" rtrunnion nil) ))
  :op-mode nil

  :constraints                                               ; list of constraints
   '(and
     (>= (Lmax *stock*) 200)
     (<= (Lmax *stock*) 600)
     (>= (Dmin *stock*) 10)
     (<= (Dmax *stock*) 150))

  :actions                                                   ; list of actions

   '((chuck shaft :kind 'in :where 'on-the-left :free-length (+ (object-length crest) 80) )
                                                                ; chucking
     (bind-global *direction* 'from-the-right)
     (plan crest)                                               ; call skeletal plan "crest"
     (chuck shaft :kind 'switch :where crest)                   ; rechucking
     (bind-global *direction* 'from-the-left)
     (center-hole shaft :kind 'create :where rtrunnion)         ; making center hole
     (center-hole shaft :kind 'move :where rtrunnion)           ; chuck with the lathe center
     (bind-local s1 (plan lstep))                               ; call skeletal plan "lstep"
   (bind-local s2 (plan rtrunnion))                             ; call skeletal plan "rtrunion"

    (merge-plans                                                ; merge the sequentialy planned
                                                                  cuts if possible
       (list (list s1 s2) :op 'roughing :kind 'lengthwise)
       (list (list s1 s2) :op 'finishing :tool-groups '(w5))
       (list (list s1 s2) :op 'recessing))
     (chuck shaft :kind 'out) ))                                ;unchucking
```

Figure 6.: A Skeletal Plan

To perform the selection and merging of the skeletal plans, a prototypical skeletal planning system was implemented as a part of our PIM system, which uses the following algorithm:

When a given workpiece description has been transformed in a feature structure according to the methods outlined in the chapter above, the skeletal plans associated with these features are found and selected according to the constraints embedded in the plans. The resulting set of skeletal plans is then merged to one final plan, and abstract variables are replaced by the concrete data of the workpiece in question. The merging of the skeletal plans is oriented on several topics: Operations using the same tool should be performed consecutively (minimalization of tool change operations). Operations in one chucking context must be performed together, minimizing the changes of chucking. Different tools belonging to a common group may be exchanged against a more general tool of this group, such that several operations using slightly different tools can be merged to one operation using only one tool. Different surfaces of a workpiece which are treated with similar operations should be grouped together.

These merging operations are supported by a hierarchical ordering of the available tools and a hierarchical grouping of the possible operations. Some heuristical approaches to skeletal plan merging are under investigation.

Figure 7 shows some output of the PIM system is shown, represented in the abstract process language LANCO. It shows the sequence of operations as well as the grouping of similarly manufactured surfaces and features which is obtained by the merging process.

```
#<Actions of Plan:
  #<CHUCK :kind IN :where ON-THE-LEFT :free-length 140 mm>
  #<ROUGHING :kind RADIAL :tool-groups (W3) :surfaces ((3) (#<Feature-Treep laneface 50>))>
  #<ROUGHING :kind LENGTHWISE :tool-groups (W2) :surfaces ((4) (#<Feature-
     Tree longturningsurface 52>))>
  #<FINISHING :kind CONTOUR :tool-groups (W5) :surfaces ((3 4) (#<Feature-
     Tree planeface 50> #<Feature-Tree longturningsurface 52>))>
  #<CHUCK :kind SWITCH :where ((3 4) (#<Feature-Tree crest 53>))>
  #<CENTER-HOLE :kind CREATE :where ((19 20 22) (#<Feature-Tree rtrunnion 8B>))>
  #<CENTER-HOLE :kind MOVE :where ((19 20 22) (#<Feature-Tree rtrunnion 8B>))>
  #<ROUGHING :kind RADIAL :tool-groups (W3) :surfaces ((10 11) (#<Feature-Tree lshoulder 66>))>
  #<ROUGHING :kind LENGTHWISE :tool-groups (W2) :surfaces ((5 6 8 9 11 12 13 15 16 17 18 19 20)
     (#<Feature-Tree lshoulder 58> #<Feature-Tree lshoulder 5E> #<Feature-Tree rshoulder 6C> #<Feature-
     Tree longturningsurface 6F> #<Feature-Tree lshoulder 77> #<Feature-Tree rshoulder 7E> #<Feature-
     Tree longturningsurface 81> #<Feature-Tree lshoulder 88>))>
  #<ROUGHING :kind RADIAL :tool-groups (W3) :surfaces ((22) (#<Feature-Tree planeface 8A>))>
  #<FINISHING :kind CONTOUR :tool-groups (W5) :surfaces ((5 6 8 9 11 12 13 18 22 19 20)
     (#<Feature-Tree lshoulder 58> #<Feature-Tree lshoulder 5E> #<Feature-Tree rshoulder 6C> #<Feature-
     Tree longturningsurface 6F> #<Feature-Tree longturningsurface 81> #<Feature-Tree planeface 8A>
     #<Feature-Tree lshoulder 88>))>
  #<FINISHING :kind CONTOUR :tool-groups (W2) :surfaces ((10 11 15 16 17)  (#<Feature-Tree lshoulder
     66> #<Feature-Tree lshoulder 77> #<Feature-Tree rshoulder 7E>))>
  #<CHUCK :kind OUT>
>
?
```

Figure 7.: Output of PIM: NC-Programm in LANCO

6 An abstract view on planning using PIM

Typical AI planning systems usually follow the methodology of STRIPS: The system searches a sequence of operators which transform a given start situation into the goal situation. Every operator is characterized by preconditions which describe the applicability of the operator and postconditions which indicate the changes of the situation resulting from the application of the operator. By standard search methods (e. g. backward chaining) it is possible to find the intended sequence of operators.

Comparing the methodology used in PIM to this, we note interesting similarities and fundamental differences: The features can be seen as aggregations of preconditions relating to geometrical/technological data of the workpiece. The workpiece description contains the description of the goal situation. The feature recognition may be seen as an efficient search method to narrow down the search space to the applicable skeletal plans. The skeletal plans then serve as indicators for the future search process. Ideally there isn't nearly any real search, as the skeletal plans determine nearly all planning steps.

In summary, the expert's knowledge, represented in features and skeletal plans, is employed to drastically reduce the search space of the planning task.

The planning task itself - to bridge the gap from CAD to NC - is exchanged against some transformation steps from between representation languages. On the level of features and skeletal plans, the gap is bridged by associations representing expert's knowledge. The found skeletal plans are then refined using the concrete data, such that the intended code can be generated. So the planning task boils down to an sequence of abstraction, association and refinement, as illustrated in figure 8, hereby eliminating the need for undirected and inefficient search.

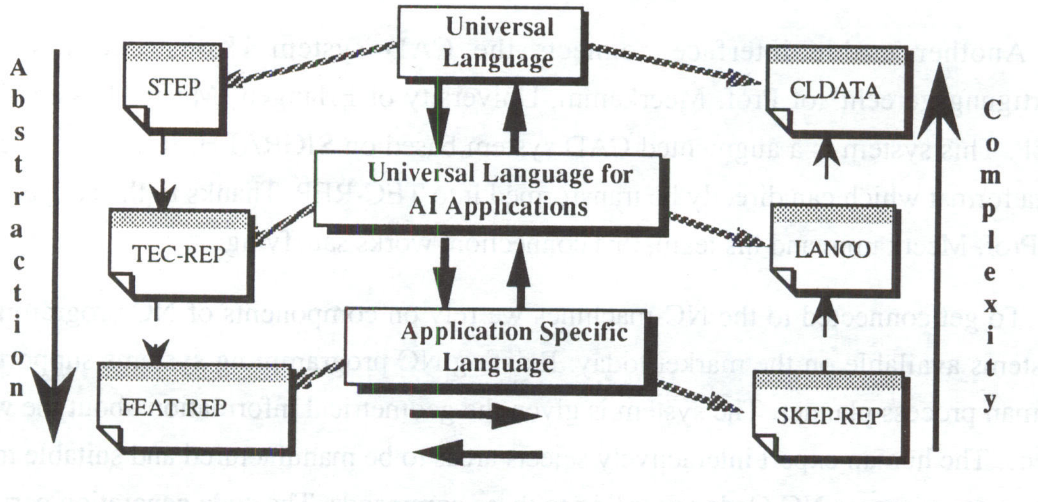


Figure 8.: Abstraction levels of PIM (Planning In Manufacturing)

Beside this it must be remembered that the main advantage of the PIM approach lies in the explicit representation of the special knowledge of an expert in his manufacturing environment. In contrast to systems which employ only general rules this approach solves the problem of the very special and environment-dependend knowledge necessary to create good plans, as motivated in the first chapter.

Intensive talks with many domain experts proved that this approach in fact resembles the way a human expert thinks, which should lead to a high acceptance of such a system by the expert.

7 Connections to the real world

Every CAPP-System serving any useful purpose must be embedded in the existing CA*-world. It has to accept the workpiece description from CAD and its output will be fed into NC machines which manufacture the workpiece. The PIM system fulfils this requirement using interfaces to CAD and NC programming systems.

The interface to the CAD world transforms the necessary geometrical/technological information about the workpiece from the CAD data into our TEC-REP. The forthcoming ISO-standard STEP [Grab 89] [Bern90] promises to provide a data exchange format covering all necessary information and providing a system-independent interface. While currently only some prototypes of CAD systems are able to produce STEP files, this will certainly change in the near future. Consequently, a prototypical processor was developed to create TEC-REP files out of a given STEP data file using the information of the geometry, topology and technology models of STEP.

Another usable interface connects the CAD system "Konstruktionssystem Fertigungsgerecht" of Prof. Meerkamm, University of Erlangen [Meer91], with TEC-REP. This system is a augmented CAD system based on SIGRAPH. It uses an internal data format which can directly be transformed into TEC-REP. Thanks to the cooperation of Prof. Meerkamm and his team, this connection works satisfying.

To get connected to the NC-machines we rely on components of NC programming systems available on the market today. Existing NC programming systems support the human process planner. The system is given the geometrical information about the work piece. The human expert interactively selects areas to be manufactured and suitable tools. The system creates NC-Code according to these commands. The code generation part of a NC programming system doesn't rely on human experts giving the commands,t he output of a planning system like PIM can serve as a command sequence as well. So a commercially available NC programming system is connected to PIM as follows:

The output of the PIM system is compiled into the command language of the NC programming system.Then, the NC programming system can generate CLDATA Code as well as machine specific NC-Code, without any further human interaction.

8 Some tools to support our approach

The basic idea behind the PIM system is the explicit representation of a concrete expert's knowledge in his given manufacturing environment in order to closely mimic the expert's problem solving behavior. This methodology leads to special requirements in the area of knowledge acquisition as well as in the area of AI reasoning methods.

According to the intended use of the PIM system, a lot of knowledge acquisition must be performed in the field. In any given manufacturing environment, the expert's knowledge must be acquired and represented separately. This is especially important for any proprietary knowledge and special manufacturing tricks of a factory. When the manufacturing environment changes during the life time of the system, the adequate knowledge must be acquired and represented again in order to adapt to new possibilities. (New tools can lead to new manufacturing methods and consequently to new manufacturing features). The acquisition group of the ARC-TEC project provides tools to support the necessary knowledge acquisition for every step in the systems life cycle [Schm91].

The compilation group of the ARC-TEC project deals with suitable and efficient reasoning mechanisms for the represented knowledge. Taxonomies are handled using the terminological representation language TAXON [Baad91] This extension of a KL-ONE like language can be used to represent the hierarchical dependencies, e. g. a hierarchy of manufacturing tools. Constraints arising in the area can be tackled using the tool CONTAX, which couples a constraint propagation system with the taxonomies. The integrated forward/backward reasoning system FORWARD/RELFUN is suitable for reasoning about the preconditions of the skeletal plans.

9 Conclusion

The observation of human expert's problem-solving behavior resulted in a model of process planning which supports a knowlegde-based approach to CAPP. Based on technological/geometrical information about the workpiece, higher-level features are defined and associated with skeletal plans.

Special languages for a adequate representation of the necessary knowledge on the different abstraction levels were presented. The transformation and interpretation steps between the different languages have been implemented and form the planning system PIM. This system creates a production plan to a given workpiece by performing a sequence of abstraction, selection and refinement.

The approach pursued in the PIM system closely mimics the expert's problem solving behavior. The resulting system is especially tailored to a concrete manufacturing environment and uses the expert's knowledge optimally. This should lead to good quality of the produced plans and to high acceptance of the system. Positive comments of many domain experts support this claim.

10 References

[Baad91] Baader, F. and Hanschke, P.: A Scheme for Integrating Concrete Domains into Concept Languages. In *Proceedings of the 12th International Joint Conference on Artificial Intelligence*, 1991.

[Bern90] Bernardi, A., Klauck, C., and Legleitner, R.: *Abschlußbericht des Arbeitspaketes PROD.* Dokument, D-90-03 Deutsches Forschungszentrum für Künstliche Intelligenz GmbH, Postfach 20 80, D-6750 Kaiserslautern, september, 1990.

[Bern91a] Bernardi, A., Boley, H., Hanschke, P., Hinkelmann, K., Klauck, C., Kühn, O., Legleitner, R., Meyer, M., Richter, M.M., Schmalhofer, F., Schmidt, G., and Sommer, : ARC-TEC: Acquisition, Representation and Compilation of Technical Knowledge. In *Expert Sytems and their Applications:Tools, techniques and Methods*, 1991, pp. 133-145.

[Bern91b] Bernardi, A., Klauck, C., and Legleitner, R.: *TEC-REP: Repräsentation von Geometrie- und Technologieinformationen.* Dokument, D-91-07 Deutsches Forschungszentrum für Künstliche Intelligenz GmbH, Postfach 20 80, D-6750 Kaiserslautern, june, 1991.

[Chan90] Chang, T.C.: *Expert Process Planning for Manufacturing*, Addison-Wesley (1990).

[Espr89] Computer Integrated Manufacturing - Results and Progress of Esprit Projects in 1989. Esprit XIII/416/89, november, 1989.

[Frie85] Friedland, P.E. and Iwasaki, Y.: *The Concept and Implementation of Skeletal Plans.* Journal of Automated Reasoning (1)(october 1985), 161-208.

[Grab89] Grabowski, H., Anderl, R., Schilli, B. and Schmitt, M.: STEP - Entwicklung einer Schnittstelle zum Produktdatenaustausch. VDI-Z 131, september 1989, Nr. 9, pp. 68-76.

[Klau91a] Klauck, C., Bernardi, A., and Legleitner, R.: FEAT-REP: Representing Features in CAD/CAM. In *IV International Symposium on Artificial Intelligence:Applications in Informatics*, 1991.

[Klau91b] Klauck, C., Bernardi, A., and Legleitner, R.: *FEAT-REP: Representing Features in CAD/CAM*. Research Report, RR-91-20 Deutsches Forschungszentrum für Künstliche Intelligenz GmbH, Postfach 20 80D-6750 Kaiserslautern, june, 1991.

[Meer91] Meerkamm, H. and Weber, A.:Konstruktionssystem mfk - Integration von Bauteilsynthese und -analyse. In *Erfolgreiche Anwendung wissensbasierter Systeme in Entwicklung und Konstruktion*. VDI Berichte 903, 1991, pp. 231-249.

[Richt91] Richter, M. M., Bernardi, A. ,Klauck, C. and Legleitner, R. : Akquisition und Repräsentation von technischem Wissen für Planungsaufgaben im Bereich der Fertigungstechnik. In *Erfolgreiche Anwendung wissensbasierter Systeme in Entwicklung und Konstruktion*. VDI Berichte 903, 1991, pp. 125-145.

[Schm91] Schmalhofer, F. , Kühn, O., and Schmidt, G.: *Integrated knowledge acquisition from text, previously solved cases and expert memories* . Applied Artificial Intelligence (5)(1991), 311 - 337.

[Teng84] Tengvald, E.: *The Design of Expert Planning Systems: An Experimental Operations Planning System for Turning*, Ph.D. dissertation, Lingköping University, Department of Computer and Information ScienceLinköping UniversityS-58183 Linköping, Sweden, 1984.

Deutsches
Forschungszentrum
für Künstliche
Intelligenz GmbH

DFKI
-Bibliothek-
PF 2080
D-6750 Kaiserslautern
FRG

DFKI Publikationen

Die folgenden DFKI Veröffentlichungen sowie die aktuelle Liste von allen bisher erschienenen Publikationen können von der oben angegebenen Adresse bezogen werden.
Die Berichte werden, wenn nicht anders gekennzeichnet, kostenlos abgegeben.

DFKI Publications

The following DFKI publications or the list of all publisched papers so far can be ordered from the above address.
The reports are distributed free of charge except if otherwise indicated.

DFKI Research Reports

RR-91-08
Wolfgang Wahlster, Elisabeth André, Som Bandyopadhyay, Winfried Graf, Thomas Rist:
WIP: The Coordinated Generation of Multimodal Presentations from a Common Representation
23 pages

RR-91-09
Hans-Jürgen Bürckert, Jürgen Müller, Achim Schupeta: RATMAN and its Relation to Other Multi-Agent Testbeds
31 pages

RR-91-10
Franz Baader, Philipp Hanschke: A Scheme for Integrating Concrete Domains into Concept Languages
31 pages

RR-91-11
Bernhard Nebel: Belief Revision and Default Reasoning: Syntax-Based Approaches
37 pages

RR-91-12
J.Mark Gawron, John Nerbonne, Stanley Peters:
The Absorption Principle and E-Type Anaphora
33 pages

RR-91-13
Gert Smolka: Residuation and Guarded Rules for Constraint Logic Programming
17 pages

RR-91-14
Peter Breuer, Jürgen Müller: A Two Level Representation for Spatial Relations, Part I
27 pages

RR-91-15
Bernhard Nebel, Gert Smolka:
Attributive Description Formalisms ... and the Rest of the World
20 pages

RR-91-16
Stephan Busemann: Using Pattern-Action Rules for the Generation of GPSG Structures from Separate Semantic Representations
18 pages

RR-91-17
Andreas Dengel, Nelson M. Mattos:
The Use of Abstraction Concepts for Representing and Structuring Documents
17 pages

RR-91-18
John Nerbonne, Klaus Netter, Abdel Kader Diagne, Ludwig Dickmann, Judith Klein:
A Diagnostic Tool for German Syntax
20 pages

RR-91-19
Munindar P. Singh: On the Commitments and Precommitments of Limited Agents
15 pages

RR-91-20
Christoph Klauck, Ansgar Bernardi, Ralf Legleitner
FEAT-Rep: Representing Features in CAD/CAM
48 pages

RR-91-21
Klaus Netter: Clause Union and Verb Raising Phenomena in German
38 pages

RR-91-22
Andreas Dengel: Self-Adapting Structuring and Representation of Space
27 pages

RR-91-23
Michael Richter, Ansgar Bernardi, Christoph Klauck, Ralf Legleitner: Akquisition und Repräsentation von technischem Wissen für Planungsaufgaben im Bereich der Fertigungstechnik
24 Seiten

RR-91-24
Jochen Heinsohn: A Hybrid Approach for Modeling Uncertainty in Terminological Logics
22 pages

RR-91-25
Karin Harbusch, Wolfgang Finkler, Anne Schauder: Incremental Syntax Generation with Tree Adjoining Grammars
16 pages

RR-91-26
M. Bauer, S. Biundo, D. Dengler, M. Hecking, J. Koehler, G. Merziger:
Integrated Plan Generation and Recognition - A Logic-Based Approach -
17 pages

RR-91-27
A. Bernardi, H. Boley, Ph. Hanschke, K. Hinkelmann, Ch. Klauck, O. Kühn, R. Legleitner, M. Meyer, M. M. Richter, F. Schmalhofer, G. Schmidt, W. Sommer:
ARC-TEC: Acquisition, Representation and Compilation of Technical Knowledge
18 pages

RR-91-28
Rolf Backofen, Harald Trost, Hans Uszkoreit:
Linking Typed Feature Formalisms and Terminological Knowledge Representation Languages in Natural Language Front-Ends
11 pages

RR-91-29
Hans Uszkoreit: Strategies for Adding Control Information to Declarative Grammars
17 pages

RR-91-30
Dan Flickinger, John Nerbonne:
Inheritance and Complementation: A Case Study of *Easy* Adjectives and Related Nouns
39 pages

RR-91-31
H.-U. Krieger, J. Nerbonne:
Feature-Based Inheritance Networks for Computational Lexicons
11 pages

RR-91-32
Rolf Backofen, Lutz Euler, Günther Görz:
Towards the Integration of Functions, Relations and Types in an AI Programming Language
14 pages

RR-91-33
Franz Baader, Klaus Schulz:
Unification in the Union of Disjoint Equational Theories: Combining Decision Procedures
33 pages

RR-91-34
Bernhard Nebel, Christer Bäckström:
On the Computational Complexity of Temporal Projection and some related Problems
35 pages

RR-91-35
Winfried Graf, Wolfgang Maaß: Constraint-basierte Verarbeitung graphischen Wissens
14 Seiten

RR-92-01
Werner Nutt: Unification in Monoidal Theories is Solving Linear Equations over Semirings
57 pages

RR-92-02
Andreas Dengel, Rainer Bleisinger, Rainer Hoch, Frank Hönes, Frank Fein, Michael Malburg:
Π_{ODA}: The Paper Interface to ODA
53 pages

RR-92-03
Harold Boley:
Extended Logic-plus-Functional Programming
28 pages

RR-92-04
John Nerbonne: Feature-Based Lexicons: An Example and a Comparison to DATR
15 pages

RR-92-05
Ansgar Bernardi, Christoph Klauck, Ralf Legleitner, Michael Schulte, Rainer Stark:
Feature based Integration of CAD and CAPP
19 pages

RR-92-07
Michael Beetz:
Decision-theoretic Transformational Planning
22 pages

RR-92-08
Gabriele Merziger: Approaches to Abductive Reasoning - An Overview -
46 pages

RR-92-09
Winfried Graf, Markus A. Thies:
Perspektiven zur Kombination von automatischem Animationsdesign und planbasierter Hilfe
15 Seiten

RR-92-11
Susane Biundo, Dietmar Dengler, Jana Koehler:
Deductive Planning and Plan Reuse in a Command Language Environment
13 pages

RR-92-13
Markus A. Thies, Frank Berger:
Planbasierte graphische Hilfe in objektorientierten Benutzungsoberflächen
13 Seiten

RR-92-14
Intelligent User Support in Graphical User Interfaces:
1. InCome: A System to Navigate through Interactions and Plans
   Thomas Fehrle, Markus A. Thies
2. Plan-Based Graphical Help in Object-Oriented User Interfaces
   Markus A. Thies, Frank Berger

22 pages

RR-92-15
Winfried Graf: Constraint-Based Graphical Layout of Multimodal Presentations
23 pages

RR-92-17
Hassan Aït-Kaci, Andreas Podelski, Gert Smolka:
A Feature-based Constraint System for Logic Programming with Entailment
23 pages

RR-92-18
John Nerbonne: Constraint-Based Semantics
21 pages

RR-92-19
Ralf Legleitner, Ansgar Bernardi, Christoph Klauck
PIM: Planning In Manufacturing using Skeletal Plans and Features
17 pages

RR-92-20
John Nerbonne: Representing Grammar, Meaning and Knowledge
18 pages

DFKI Technical Memos

TM-91-05
Jay C. Weber, Andreas Dengel, Rainer Bleisinger:
Theoretical Consideration of Goal Recognition Aspects for Understanding Information in Business Letters
10 pages

TM-91-06
Johannes Stein: Aspects of Cooperating Agents
22 pages

TM-91-08
Munindar P. Singh: Social and Psychological Commitments in Multiagent Systems
11 pages

TM-91-09
Munindar P. Singh: On the Semantics of Protocols Among Distributed Intelligent Agents
18 pages

TM-91-10
Béla Buschauer, Peter Poller, Anne Schauder, Karin Harbusch: Tree Adjoining Grammars mit Unifikation
149 pages

TM-91-11
Peter Wazinski: Generating Spatial Descriptions for Cross-modal References
21 pages

TM-91-12
Klaus Becker, Christoph Klauck, Johannes Schwagereit: FEAT-PATR: Eine Erweiterung des D-PATR zur Feature-Erkennung in CAD/CAM
33 Seiten

TM-91-13
Knut Hinkelmann:
Forward Logic Evaluation: Developing a Compiler from a Partially Evaluated Meta Interpreter
16 pages

TM-91-14
Rainer Bleisinger, Rainer Hoch, Andreas Dengel:
ODA-based modeling for document analysis
14 pages

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

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

DFKI Documents

D-91-07
Ansgar Bernardi, Christoph Klauck, Ralf Legleitner
TEC-REP: Repräsentation von Geometrie- und Technologieinformationen
70 Seiten

D-91-08
Thomas Krause: Globale Datenflußanalyse und horizontale Compilation der relational-funktionalen Sprache RELFUN
137 Seiten

D-91-09
David Powers, Lary Reeker (Eds.):
Proceedings MLNLO'91 - Machine Learning of Natural Language and Ontology
211 pages
Note: This document is available only for a nominal charge of 25 DM (or 15 US-$).

D-91-10
Donald R. Steiner, Jürgen Müller (Eds.):
MAAMAW'91: Pre-Proceedings of the 3rd European Workshop on „Modeling Autonomous Agents and Multi-Agent Worlds"
246 pages
Note: This document is available only for a nominal charge of 25 DM (or 15 US-$).

D-91-11
*Thilo C. Horstmann:*Distributed Truth Maintenance
61 pages

D-91-12
Bernd Bachmann:
$H^{iera}C_{on}$ - a Knowledge Representation System with Typed Hierarchies and Constraints
75 pages

D-91-13
International Workshop on Terminological Logics
Organizers: Bernhard Nebel, Christof Peltason, Kai von Luck
131 pages

D-91-14
Erich Achilles, Bernhard Hollunder, Armin Laux, Jörg-Peter Mohren: *KRIS* : *K*nowledge *R*epresentation and *I*nference *S*ystem
- Benutzerhandbuch -
28 Seiten

D-91-15
Harold Boley, Philipp Hanschke, Martin Harm, Knut Hinkelmann, Thomas Labisch, Manfred Meyer, Jörg Müller, Thomas Oltzen, Michael Sintek, Werner Stein, Frank Steinle:
μCAD2NC: A Declarative Lathe-Worplanning Model Transforming CAD-like Geometries into Abstract NC Programs
100 pages

D-91-16
Jörg Thoben, Franz Schmalhofer, Thomas Reinartz: Wiederholungs-, Varianten- und Neuplanung bei der Fertigung rotationssymmetrischer Drehteile
134 Seiten

D-91-17
Andreas Becker:
Analyse der Planungsverfahren der KI im Hinblick auf ihre Eignung für die Abeitsplanung
86 Seiten

D-91-18
Thomas Reinartz: Definition von Problemklassen im Maschinenbau als eine Begriffsbildungsaufgabe
107 Seiten

D-91-19
Peter Wazinski: Objektlokalisation in graphischen Darstellungen
110 Seiten

D-92-01
Stefan Bussmann: Simulation Environment for Multi-Agent Worlds - Benutzeranleitung
50 Seiten

D-92-02
Wolfgang Maaß: Constraint-basierte Plazierung in multimodalen Dokumenten am Beispiel des Layout-Managers in WIP
111 Seiten

D-92-03
Wolfgan Maaß, Thomas Schiffmann, Dudung Soetopo, Winfried Graf: LAYLAB: Ein System zur automatischen Plazierung von Text-Bild-Kombinationen in multimodalen Dokumenten
41 Seiten

D-92-06
Hans Werner Höper: Systematik zur Beschreibung von Werkstücken in der Terminologie der Featuresprache
392 Seiten

D-92-08
Jochen Heinsohn, Bernhard Hollunder (Eds.):
DFKI Workshop on Taxonomic Reasoning Proceedings
56 pages

D-92-09
Gernod P. Laufkötter: Implementierungsmöglichkeiten der integrativen Wissensakquisitionsmethode des ARC-TEC-Projektes
86 Seiten

D-92-15
DFKI Wissenschaftlich-Technischer Jahresbericht 1991
130 Seiten

D-92-21
Anne Schauder: Incremental Syntactic Generation of Natural Language with Tree Adjoining Grammars
57 pages

PIM:
Planning In Manufacturing using Skeletal Plans and Features

Ansgar Bernardi, Christoph Klauck, Ralf Legleitner

RR-92-19
Research Report