

CABPLAN – Fallauswahl und fallbasierte Steuerung von nichtlinearen, hierarchischen Planungsprozessen am Beispiel der Arbeitsplanung*

Jürgen Paulokat und Stefan Weß
Fachbereich Informatik – SFB 314
Universität Kaiserslautern
W-6750 Kaiserslautern

Zusammenfassung

Die Verwendung von existierenden Planungsansätzen zur Lösung von realen Anwendungsproblemen führt meist schnell zur Erkenntnis, daß eine vorliegende Problemstellung im Prinzip zwar lösbar ist, der exponentiell anwachsende Suchraum jedoch nur die Behandlung relativ kleiner Aufgabenstellungen erlaubt. Beobachtet man jedoch menschliche Planungsexperten, so sind diese in der Lage bei komplexen Problemen den Suchraum durch Abstraktion und die Verwendung bekannter Fallbeispiele als Heuristiken, entscheidend zu verkleinern und so auch für schwierige Aufgabenstellungen zu einer akzeptablen Lösung zu gelangen. In dieser Arbeit wollen wir am Beispiel der Arbeitsplanung ein System vorstellen, das Abstraktion und fallbasierte Techniken zur Steuerung des Inferenzprozesses eines nichtlinearen, hierarchischen Planungssystems einsetzt und so die Komplexität der zu lösenden Gesamtaufgabe reduziert.

1 Einführung

Betrachten wir die aus der Literatur bekannten Planungsansätze, so sind zwei Grundströmungen zu beobachten. Einerseits existieren verschiedene suchraumorientierte Verfahren, die den Lösungsraum ausgehend von vorgegebenen *first-principles* mehr oder weniger systematisch durchsuchen und dabei Verfahren zur gezielten Suchraumbeschränkung einsetzen, z.B. STRIPS [FN71], NONLIN [Tat77] oder SIPE [Wil88]. Andererseits existiert auch eine Reihe von wissensintensiven Ansätzen, wie etwa Skelettpläne in Friedlands MOLGEN [FI85], Fallbeispiele in MEDIATOR [Sim85] oder CHEF [Ham89], die eine alternative, eher vorgehensorientierte Sichtweise verfolgen und sich daher mehr auf die Steuerungsaspekte von Planungsprozessen konzentrieren (vgl. auch [Wil83]). Beide Paradigmen wurden bisher jedoch mehr oder weniger streng getrennt betrachtet¹. Während suchintensive Verfahren an der Größe des Suchraums scheitern, haben wissensintensive Verfahren Schwierigkeiten, die Vollständigkeit des Lösungsraums (bzgl. der zugrundeliegenden Anwendung) sicherzustellen. Die Modifikation von Lösungen ist nicht oder nur mit ad-hoc Methoden, z.B. durch die *Modifikationsregeln* in CHEF, die eine sehr komprimierte Form von Wissen darstellen, möglich.

Für komplexe Aufgabenstellungen, wie etwa die Arbeitsplanung, sehen wir die Kombination beider Ansätze als die einzige Möglichkeit, trotz eines explodierenden Suchraums zu akzeptablen Lösungen zu gelangen. Die systematische Beschreibung des Suchraumes erfolgt dabei durch die aus der Sicht der Anwendung primitiven Operationen, die Navigation im Suchraum durch Planskelette bzw. Fallbeispiele. Fälle sind in diesem Szenario Heuristiken zur Suchraumbeschränkung mit klar definiertem Kontext, die ein Backtracking verhindern bzw. alternative Vorgehensweisen aufzeigen können. Fallbeispiele können daher zur Steuerung eines Planungsprozesses für ein aktuelles Problem eingesetzt werden und somit helfen, frühere „Fehler“ zu vermeiden und „bessere“ Lösungen zu generieren. Aus dieser abstrakten Sicht sind die Zielsetzungen beim Einsatz von Fallbeispielen eng verbunden mit den ursprünglich beim Meta-Planen [Ste81] verfolgten Zielen.

Im Verlauf dieser Arbeit geben wir zunächst einen Überblick über die Arbeitsplanung, aus der sich die Motivation für den Einsatz eines nichtlinearen, hierarchischen Planungssystems ergibt. Im Anschluß werden die dafür notwendigen Begriffe definiert, bevor die Vorgehensweise an einem ausführlichen Beispiel beschrieben wird. Aufbauend auf den Begriffen der nichtlinearen, hierarchischen Planung beschreiben wir im zweiten Teil einen Ansatz zur Fallauswahl und zur fallbasierten Steuerung der generativen Planung.

*Die hier vorgestellte Arbeit wurde zum Teil gefördert durch die Deutsche Forschungsgemeinschaft, SFB 314: „Künstliche Intelligenz - Wissensbasierte Systeme“, Projekt X9 - CABPLAN (in Zusammenarbeit mit dem CIM Centrum Kaiserslautern, Prof. Dr.-Ing. G. Warnecke).

¹Eine Ausnahme bildet G.J. Sussman [Sus75], der für sein System HACKER den Einsatz von Fallbeispielen schon 1975 propagierte.

2 Eine Anwendung: Arbeitsplanung

Ziel der Arbeitsplanung ist es, für ein gegebenes Werkstück eine Folge von Bearbeitungsoperationen zu bestimmen, die bei vorgegebenen Ressourcen, z.B. Maschinen, Werkzeuge und Spannmittel, die Fertigung des gewünschten Werkstückes aus einem Rohteil ermöglicht. Bei der Planung des Fertigungsablaufs spielen eine Reihe von Optimierungskriterien eine Rolle. So sollen z.B. die Werkzeugwechseloperationen, der Materialverschleiß bzw. die Maschinenkosten minimiert und gleichzeitig die Qualität der gefertigten Produkte maximiert werden. Dabei auftretende Zielkonflikte müssen während der Arbeitsplanung aufgelöst werden. Als prototypische Anwendung betrachten wir die Arbeitsplanung für rotationssymmetrische Drehteile in der Kleinserienfertigung (vgl. Abb. 1). Die im Rahmen dieser Arbeit zu lösende Aufgabenstellung kann aus der Sicht der Anwendung wie folgt beschrieben werden²:

- Gegeben:** Beschreibung des zu fertigenden Werkstücks
 Beschreibung des zur Verfügung stehenden Rohteils
 Beschreibung der vorhandenen Maschinen, Werkzeuge und Spannmittel
- Gesucht:** Möglichst optimale Folge von Bearbeitungsoperationen zur Fertigung des Werkstückes

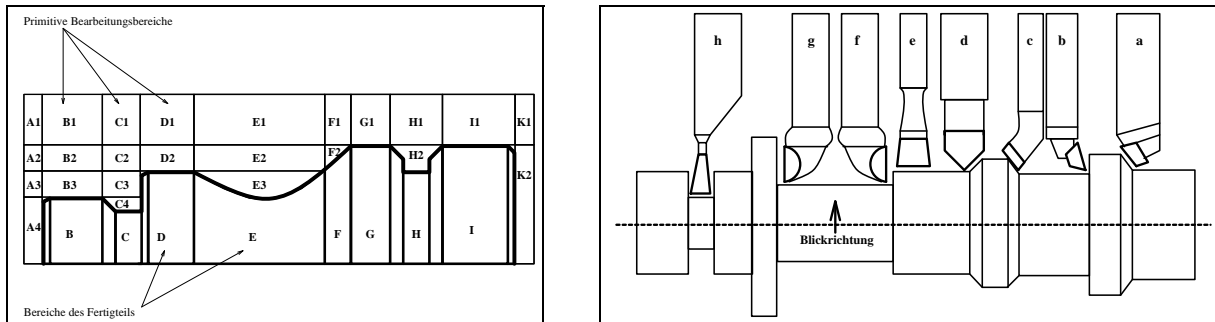


Abbildung 1: Werkstücke mit primitiven Bearbeitungsbereichen und Bearbeitungswerkzeugen

Werkstücke werden hierbei als Kombination von einzelnen Konturelementen aufgefaßt (vgl. Abb. 1). Basis der Repräsentation sind dabei eine Anzahl fest definierter *primitiver* Elemente (*Zylinder, Torroid, Kegelstumpf*). Aus primitiven Konturelementen können *komplexe* Komponenten bzw. Werkstücke konstruiert werden. Die Differenz zwischen Rohteil und Fertigteil, d.h. das zu zerspanende Materialvolumen (vgl. Abb. 1), kann aus dem Werkstück durch ein syntaktisches Kantenverlängerungsverfahren in sogenannte *primitive Bearbeitungsbereiche* aufgeteilt werden, die die Basis der Planung bilden. Ein Bearbeitungsbereich heißt *primitiv*, falls die Bearbeitung dieses Bereiches aus der Sicht der Arbeitsplanung eine atomare Fertigungsoperation ist³, d.h. der entsprechende Bereich wird während der Bearbeitung im allgemeinen nicht weiter zerteilt werden. Betrachten wir die Abbildung 1, so können wir damit aus der Sicht der Handlungsplanung die zu lösende Aufgabenstellung als eine Art erweitertes *Blockworld* Szenario verstehen. Dabei sind jedoch eine Reihe von Randbedingungen zu beachten, von denen die wichtigsten im folgenden kurz aufgeführt sind:

1. Ein primitiver Bearbeitungsbereich kann zerspannt werden, wenn er für das Werkzeug sowohl von oben als auch von links oder von rechts zugänglich ist.
2. Primitive Bearbeitungsbereiche können zu komplexen Bearbeitungsbereichen zusammengefaßt werden, die dann in einem Schritt bearbeitet werden können.
3. Bestimmte Formen von Bearbeitungsbereichen oder zusätzliche Anforderungen wie etwa die gewünschte Qualität der Oberfläche oder eine besondere Maß- oder Lagetoleranz erfordern zur Bearbeitung besondere Werkzeuge und Spannmittel.
4. Bevor ein Werkstück bearbeitet werden kann, muß es durch eine Aufspannung in der Maschine fixiert werden. Dabei werden Bearbeitungsbereiche verdeckt, die für eine Bearbeitung unter dieser Aufspannung dann nicht mehr zugänglich sind.

Bei der Lösung einer Arbeitsplanaufgabe greifen menschliche Arbeitsplanungsexperten aus Komplexitätsgründen sehr oft auf frühere Erfahrungen zurück. Ausgangspunkt der Planung sind dann Arbeitspläne von ähnlichen Teilen, die entsprechend den aktuellen Anforderungen geringfügig modifiziert werden. Fallbasiertes

²Es werden daher einige Aufgaben, die sich in realen Anwendungssituationen ergeben, wie z.B. die Bestimmung der genauen Schnittparameter bzw. die Erstellung eines entsprechenden NC-Programmes zur Steuerung der Werkzeugmaschine vernachlässigt, da bereits Lösungen zur Automatisierung dieser Prozesse existieren.

³Atomar im Sinne des hier zugrundeliegenden Szenarios. Bei der Fertigung auf einer CNC-Maschine werden diese Operationen durch eine Vielzahl von Schnittfolgen realisiert, die aber von der Maschinensteuerung selbst (lokal) ermittelt werden können.

Schließen ist also ein geeignetes Problemlösungsparadigma, um diese in der Praxis durchaus übliche Vorgehensweise zu realisieren. Ferner geht der Arbeitsplaner bei seiner Arbeit hierarchisch vor. Im ersten Schritt wird eine Folge von Aufspannungen festgelegt, die das Gesamtproblem in mehrere Teilprobleme, die weitgehend unabhängig voneinander behandelt werden, zerlegt. Hierbei wird die Menge der Alternativen durch technologische Constraints, wie Oberflächengüten, Maß- und Formlagetoleranzen eingeschränkt. Innerhalb der einzelnen Teilprobleme werden im nächsten Schritt Operationen zur Zerspannung der Bearbeitungsbereiche bestimmt, wobei auch eine Zusammenfassung zu komplexen Bearbeitungsbereichen durchgeführt wird. Nachdem alle Aktionen zur Zerspannung der Bearbeitungsbereiche festgelegt sind, kann die Fertigung zusätzlicher Features, wie Nuten oder Hinterschneidungen, im Detail geplant werden.

3 Hierarchische, nichtlineare Planung

Ein generatives Planungssystem für die Arbeitsplanung muß zur Reduktion der Gesamtkomplexität die hierarchische Vorgehensweise in der Domäne unterstützen. Jede Hierarchieebene repräsentiert einen weitgehend unabhängigen Spezialisten für ein Teilproblem (vgl. [KCTL91]). Zusätzlich ergeben sich aus der hierarchischen Strukturierung der Pläne neue Vorgehensweisen bei der Auswahl von Fallbeispielen und ihrer Verwendung bei der Steuerung des Planungsprozesses. Bei der hierarchischen Planung wird ein Planungsproblem in aufeinanderfolgenden Schritten auf verschiedenen Hierarchieebenen gelöst. Jede Hierarchieebene stellt eine Abstraktion von einem konkreten Problem dar und kann als eine eigene Planungswelt aufgefaßt werden.

Definition 1 (Planungswelt) *Eine Planungswelt ist ein Tripel $D = (S, T, O)$ über einer Sprache erster Ordnung, wobei S eine Menge von Aussagen zur Beschreibung von Situationen in der Planungswelt, T eine Theorie zur Herleitung weiterer Eigenschaften aus einer Situation und O eine Menge von Operatoren ist. Eine Situation ist eine Menge von konjunktiv verknüpften Aussagen aus S . Die Operatoren beschreiben die möglichen Übergänge zwischen Situationen.*

Ein Planungsproblem in einer Planungswelt D besteht aus einer Ausgangs- und einer Zielsituation. Die Lösung eines Planungsproblems ist ein Plan \mathcal{L} , der die Ausgangssituation in eine Situation überführt, aus der die Aussagen der Zielsituation gefolgert werden können.

Definition 2 (Planungsproblem) *Ein Planungsproblem $\mathcal{P} = (A, Z, D)$ ist definiert durch eine Ausgangssituation A , ein Zielsituation Z und eine Planungswelt D .*

Definition 3 (Lösung eines Planungsproblems) *Die Lösung eines Planungsproblems $\mathcal{P} = (A, Z, D)$ ist eine partiell geordnete Menge $\mathcal{L} = (\mathcal{O}, \prec)$ von Operatoren, die die Ausgangssituation A durch Anwendung der Operatoren aus \mathcal{O} unter Beachtung der Vorgängerrelation \prec in einen Zustand Z' überführen, so daß $Z', T \vdash Z$ gilt. \mathcal{L} wird als Plan bezeichnet.*

Eine abstrakte Planungswelt enthält abstrakte Operatoren, die ein eigenständiges Planungsproblem der darunterliegenden Ebene repräsentieren. Somit ist der Planungsprozeß zur Lösung eines Planungsproblems in der Planungswelt einer Hierarchieebene E_{i-1} von einem zerlegbaren Operator eines Plans der abstrakteren Hierarchieebene E_i abhängig. Daher müssen Entscheidungen, die auf einer abstrakten Ebene getroffen wurden, an die nachfolgende Ebene weitergegeben und dort als Restriktionen bei der Planung berücksichtigt werden.

Werden beim Übergang auf eine abstraktere Ebene nicht nur Details der konkreteren Ebene weggelassen sondern wird die Sprache zur Beschreibung von Situationen, Operatoren und Theorien gewechselt, so benötigt man eine zusätzliche Sprache, mit der das initiale Planungsproblem unabhängig von allen Abstraktionsebenen beschrieben werden kann. Das initiale Planungsproblem setzt sich aus einer Ausgangssituation und einer Zielsituation zusammen. Es wird in ein Planungsproblem der obersten Abstraktionsebene E_n transformiert, mit dem der hierarchische Planungsprozeß gestartet wird. Die Ausgangssituation der initialen Problembeschreibung besteht aus einem statischen und einem dynamischen Teil. In der Blockworld würde z.B. die unveränderliche Aussage $block(A)$ zu dem statischen und eine Aussage der Form $on(A B)$, die durch eine Operatoranwendung ungültig werden kann, zu dem dynamischen Teil gehören. Die Aussagen des statischen Teils sind in allen Situationen auf allen Hierarchieebenen gültig und müssen, sofern sie dort benötigt werden, aus der initialen Ausgangssituation in die Sprache der betrachteten Ebene transformiert werden.

Somit bestehen vielfältige Abhängigkeiten zwischen den Ebenen sowie zwischen einer Ebene und der initialen Problembeschreibung einer hierarchisch strukturierten Planungswelt, die durch Konkretisierungs- bzw. Abstraktionsabbildungen beschrieben werden können.

Definition 4 (Hierarchische Planungswelt) *Eine hierarchische Planungswelt $D^H = (S_0, \mathcal{D}, \mathcal{A}, \mathcal{K})$ ist definiert durch die Menge S_0 zur Beschreibung eines initialen Planungsproblems. \mathcal{D} ist ein Tupel von Weltbeschreibungen (D_1, \dots, D_n) , $D_i = (S_i, T_i, O_i)$, wobei D_i die Planungswelt für die Abstraktionsebene E_i definiert. \mathcal{A} ist ein Tupel von Abstraktionsabbildungen (f_1^A, \dots, f_n^A) mit $f_i^A : 2^{S_0} \rightarrow 2^{S_i}$. \mathcal{K} ist ein Tupel von Paaren von Konkretisierungsabbildungen $((k_2^A, k_2^Z), \dots, (k_n^A, k_n^Z))$, $k_i^A : O_i \rightarrow 2^{S_{i-1}}$ und $k_i^Z : O_i \rightarrow 2^{S_{i-1}}$, die einen zerlegbaren Operator der Ebene E_i auf eine Ausgangs- bzw. Zielsituation der Ebene E_{i-1} abbilden.*

Definition 5 (Hierarchisches Planungsproblem) Ein hierarchisches Planungsproblem $\mathcal{P}^H = (A_0, Z_0, D^H)$ ist definiert durch die Anfangssituation $A_0 = A_0^{stat} \cup A_0^{dyn}$, die Zielsituation Z und eine hierarchische Planungswelt D^H . Durch \mathcal{P}^H ist ein Planungsproblem $\mathcal{P}_n = (A_n, Z_n, D_n)$ der Abstraktionsebene E_n , mit $A_n = f_n^A(A_0)$, $Z_n = f_n^A(Z_0)$, definiert.

Der Planungsprozeß für eine Planungsaufgabe in einer hierarchischen Planungswelt beginnt immer auf der höchsten Ebene E_n . Für diese Planungsaufgabe der Ebene E_n , die durch eine Ausgangs- und eine Zielsituation gegeben ist, wird ein nichtlinearer Plan hergeleitet, der wie alle anderen Pläne der Ebenen E_2, \dots, E_n zerlegbare Operatoren enthalten kann. Jeder zerlegbare Operator o_{ij} einer Ebene E_i definiert ein neues Planungsproblem auf der nachfolgenden Ebene E_{i-1} . Diese Planungsprobleme werden unabhängig voneinander bearbeitet. Interaktionen zwischen den einzelnen Planungsproblemen auf einer Abstraktionsebene E_{i-1} müssen durch den nichtlinearen Plan der Ebene E_i vorweggenommen werden. Falls für ein Planungsproblem keine Lösung gefunden werden kann, ist der Plan der darüberliegenden Ebene E_i inkonsistent. Die Ursache für diese Inkonsistenz ist der Operator, der das unlösbare Planungsproblem der nachfolgenden Hierarchieebene generiert hat. Zur Beseitigung der Inkonsistenz muß auf der Ebene E_i Backtracking durchgeführt werden.

Definition 6 (Lösung eines hierarchischen Planungsproblems) Die Lösung eines hierarchischen Planungsproblems \mathcal{P}^H ist ein hierarchischer Plan $\mathcal{L}^H = \mathcal{L}_{n_0}^H$ für das durch \mathcal{P}^H definierte Planungsproblem $\mathcal{P}_n = \mathcal{P}_{n_0}$ der Abstraktionsebene E_n . Ein hierarchischer Plan \mathcal{L}_{ij}^H eines Planungsproblems \mathcal{P}_{ij} besteht aus einer Lösung \mathcal{L}_{ij} für \mathcal{P}_{ij} und je einem hierarchischen Plan $\mathcal{L}_{i-1k_j}^H$ für die hierarchischen Planungsprobleme \mathcal{P}_{i-1k_j} der Ebene E_{i-1} , die durch die abstrakten Operatoren o_{ijk} des Planes \mathcal{L}_{ij} der Ebene E_i gegeben sind. Ein abstrakter Operator o_{ijk} definiert ein hierarchisches Planungsproblem $\mathcal{P}_{i-1k_j}^H = (A_{i-1k_j}, Z_{i-1k_j}, D_{i-1})$ mit $A_{i-1k_j} = f_{i-1}^A(A_0^{stat}) \cup k_i^A(o_{ijk})$ und $Z_{i-1k_j} = k_i^Z(o_{ijk})$.

4 Ein Modell der generativen Arbeitsplanung

Der Ausgangszustand A_0 der initialen Problembeschreibung für das Werkstück aus Abb. 1 enthält für jedes Konturelement eine oder mehrere Aussagen, die die planungsrelevanten Eigenschaften festlegen. Dies sind: *stirnseite(A)*, *zylinder(B)*, *hinterschneidung(Hinter-C)*, *zylinder(D)*, *hinterschneidung(Hinter-E)*, *kegelstumpf(F)*, *zylinder(G)*, *hinterschneidung(Hinter-H)*, *zylinder(I)* und *stirnseite(K)*. Hinterschneidungen stellen in den meisten Fällen eine Aggregation von weiteren Konturelementen dar, so daß sie weiter unterteilt werden können: *komponente-von(C₁ C)*, *komponente-von(C₂ C)*, *kegelstumpf(C₁)*, *zylinder(C₂)*, *freiform(E)*, *komponente-von(H₁ H)*, *komponente-von(H₂ H)*, *komponente-von(H₃ H)*, *kegelstumpf(H₁)*, *zylinder(H₂)* und *zylinder(H₃)*. Zwischen den Darstellungsprimitiven bestehen Nachbarschaftsbeziehungen der Form: *links-von(A B)*, *rechts-von(B A)*, ..., *rechts-von(K I)*. Die Bearbeitungsbereiche *atom(A₄)*, ..., *atom(K₂)* haben ebenfalls linke und rechte Nachbarn. Hier ist jedoch auch relevant, daß Bearbeitungsbereiche an Konturelemente angrenzen und in Schichten übereinander liegen: *über(B₃ B)*, *unter(B₃ B₂)*, ..., *unter(K₂ K₁)*. Darüberhinaus müssen relevante Features der Form: *linke-fase(B)* oder *rechte-fase(I)* dargestellt werden.

Eine Lösung \mathcal{L}^H des hierarchischen Planungsproblems muß den Ausgangszustand A_0 in einen Zustand überführen, in dem die Aussage aus Abb. 2.a gültig ist. Aus dieser initialen Problembeschreibung muß im ersten Schritt ein

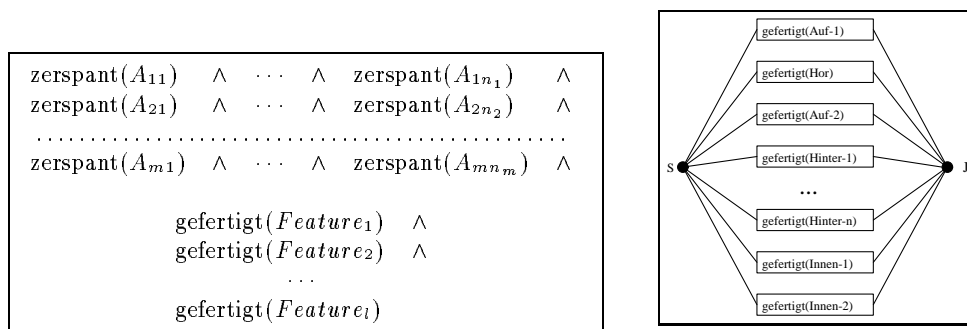


Abbildung 2: a.) Zielausdruck; b.) Initiale Ziele bei der Arbeitsplanung

Planungsproblem der höchsten Hierarchieebene hergeleitet werden. Bei der Arbeitsplanung wird auf dieser Ebene die Folge der Aufspannungen festgelegt. Die allgemeine Form dieser Problembeschreibung ist in Abb. 2.b dargestellt. Auf dieser Ebene wird ausschließlich die Grobkontur des Werkstücks betrachtet. Hierzu wird es in die beiden ansteigenden Bereiche an den Enden (A, \dots, F und K) und den dazwischenliegenden horizontalen Konturverlauf (G, \dots, I) unterteilt. Getrennt davon werden die Hinterschneidungen und die Innenbearbeitungen betrachtet.

4.1 Bestimmung der Aufspannung

Die Ziele zur Bestimmung der Aufspannfolge für das Beispiel aus Abb. 1 sind in Abb. 3.a dargestellt. Auf den Zielen ist bereits eine initiale Ordnung (gestrichelte Pfeile zwischen den Zielen) definiert. Sie ergibt sich aus der Domäne, da z.B. die Hinterschneidungen *Hinter-C* und *Hinter-E* erst gefertigt werden können, wenn die darüberliegenden Bereiche zerspannt sind, und führt somit zu einer Beschränkung des Suchraums.

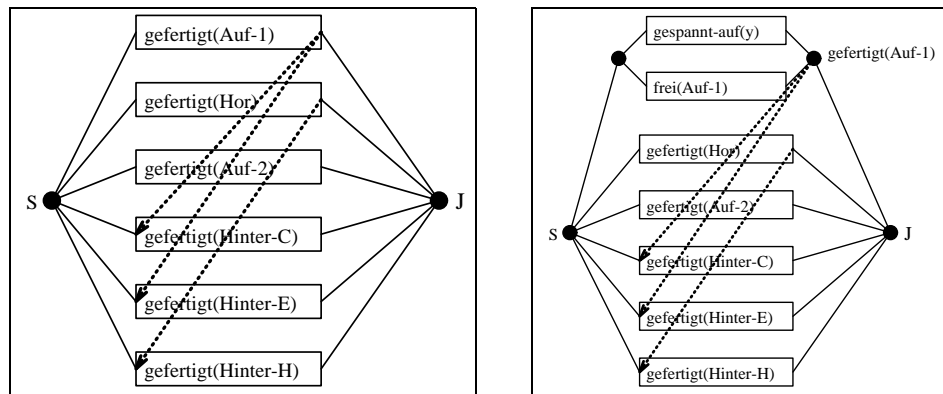


Abbildung 3: a.) Initiale Ziele für das Beispielproblem; b.) Plan nach erster Operatoranwendung.

Da auf der betrachteten Ebene nur die Aufspannungen und ihre Reihenfolge festgelegt werden, ist ein Bereich bearbeitet, wenn für ihn eine Aufspannung bestimmt ist, so daß er durch diese nicht verdeckt ist. Die Zerspanoperationen werden auf den nachfolgenden Hierarchieebenen bestimmt. Zur Fortführung des Planungsprozesses muß ein offenes Ziel ausgewählt und die darauf anwendbaren Operatoren bestimmt werden. Aus dem Operator-Schema in Abb. 4.a können die anwendbaren Operatoren durch Bindung der Variablen abgeleitet werden. Zur Bearbeitung des Ziels *gefertigt(Auf-1)* ergibt sich eine Menge von Operatoren, die durch die alternativen Belegungen für die Variable *y* bestimmt ist. Die Festlegung auf eine der möglichen Bindungen erfolgt zu einem möglichst späten Zeitpunkt (Least-Commitment), so daß ein partiell instantiiertes Operator-Schema zur Erfüllung des betrachteten Ziels verwendet wird. Im Plan wird das Ziel durch den Operator bzw. das Operatorschema ersetzt (Abb. 3.b).

Operator:	FERTIGE(<i>x</i>)	Operator:	SPANNE-AUF(<i>x</i>)
Vorbedingung:	frei(<i>x</i>)	Vorbedingung:	frei(<i>x</i>)
	gespannt-auf(<i>y</i>)		konturbereich(<i>x</i>)
	$x \neq y$	Effekte:	gespannt-auf(<i>x</i>)
Effekte:	gefertigt(<i>x</i>)		

Abbildung 4: Aufbau des a.) Fertige- und b.) Spanne-Auf-Operators

Im nächsten Schritt muß das Planungssystem zwischen sechs offenen Zielen auswählen. Das Ziel *frei(Auf-1)* ist bereits in der Ausgangssituation gültig. Es muß jedoch sichergestellt werden, daß es nicht durch die Anwendung eines Operators ungültig wird. Falls sich das System für das offene Ziel *gespannt-auf(y)* entscheidet, kann das Operator-Schema aus Abb. 4.b verwendet werden. Der hieraus resultierende partiell geordnete Plan ist in Abb. 5.a dargestellt. Die Ziele *konturbereich(y)* und *frei(y)* sind in der Ausgangssituation der betrachteten Abstraktionsebene gültig. Jedoch kann die Variable *y* an verschiedene Konstanten gebunden werden. Falls sich das System für die Bindung $y = \text{Auf-2}$ entscheidet, ergeben sich Seiteneffekte, die zur Erreichung des Ziels *gefertigt(Hor)* ausgenutzt werden können. Zunächst muß auf *gefertigt(Hor)* ebenfalls der *FERTIGE*-Operator angewendet werden. Hieraus resultieren die Ziele *gespannt-auf(z)* und *frei(Hor)*. Durch die Bindung von $z = \text{Auf-2}$ kann der bereits instantiierte Operator zur Erreichung von *gespannt-auf(y)* auch für das Ziel *gespannt-auf(z)* verwendet werden (Phantomisierung). Hierzu muß eine weitere Ordnungsrelation eingefügt werden, so daß *gespannt-auf(Auf-2)* gültig ist, wenn *Hor-2* gefertigt werden soll (Abb. 5.b). Die Ausnutzung solcher Abhängigkeiten ist entscheidend für die Güte eines Arbeitsplans, denn die Bearbeitung soll unter einer möglichst geringen Anzahl von Umspanoperationen durchgeführt werden.

4.2 Übergang zur nächsten Hierarchieebene

Beim Übergang zur nächst tieferen Hierarchieebene wird jeder abstrakte Operator zu einem eigenen Teilproblem expandiert. Diese Zerlegung wird im Beschreibungsteil *Unterproblem* eines Operators festgelegt (Abb. 6.a). Im Falle des Operators *FERTIGE(Auf-1)* resultiert ein Planungsproblem zur Zerspannung von zusammengesetzten Bearbeitungsbereichen (Abb. 6.b), die sich aus der horizontalen Zusammenfassung der atomaren Bearbeitungsbe-

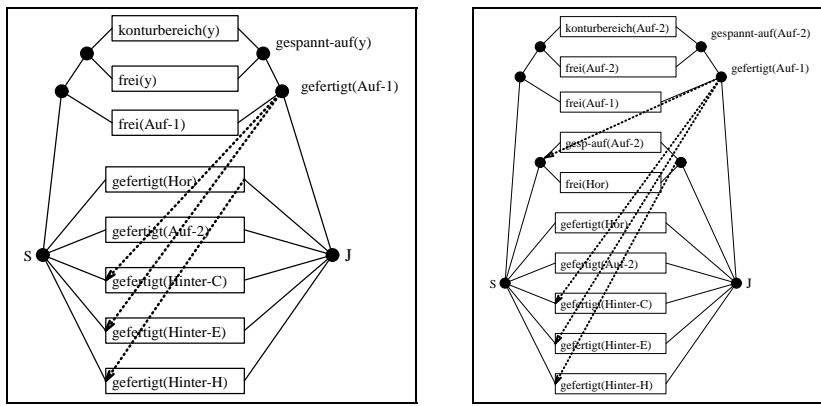


Abbildung 5: Expandierung der Ziele a.) *gespannt-auf(y)* und b.) *gefertigt(Hor)*

Operator:	FERTIGE(x)
Vorbedingung:	frei(x) gespannt-auf(y) $x \neq y$
Effekte:	gefertigt(x)
Unterproblem:	definiertNullpunkt zerpant(x_1), ..., zerpant(x_n) definiertNullpunkt < zerpant(x_1) ... definiertNullpunkt < zerpant(x_n)

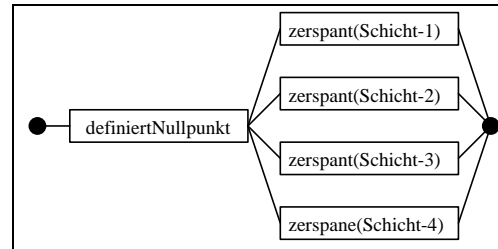


Abbildung 6: Aufbau des a.) Fertige- und b.) Spanne-Auf-Operators

reiche der linken aufsteigenden Kontur in Abb. 1 ergeben, so daß das neue Planungsproblem auf einen Ausschnitt der Gesamtkontur eingeschränkt ist. Dieser hierarchische Zerlegungsprozeß kann noch über drei Stufen fortgesetzt werden, so daß schließlich alle Bereiche zerspannt, die Feature und Schichtbereiche gefertigt und schließlich konkrete Werkzeuge für die einzelnen Bearbeitungsoperationen ausgewählt sind.

4.3 Konfliktmengen bei der generativen Planung

Bis zu diesem Punkt haben wir im wesentlichen beschrieben, welche Voraussetzungen erfüllt sein müssen, damit ein Operator angewendet oder eine Phantomisierung durchgeführt werden kann. Während des Planungsprozesses steht aber oftmals der nächste Schritt nicht eindeutig fest, vielmehr definieren die offenen Ziele, die Operatoren, die auf ein Ziel anwendbar sind, und die möglichen Linearisierungen Konfliktmengen von Alternativen, aus denen eine zur Fortführung der Planung ausgewählt werden muß. Jede Entscheidung für eine Alternativen hat zwei Effekte. Zum einen beeinflusst sie die Effektivität des Planungsprozesses, indem sie sich als eine richtige Entscheidung auf dem Weg zu einer Lösung herausstellt oder aber zu Berechnungen führt, die durch Backtracking überflüssig werden. Zum anderen hat jede konsistente Entscheidung Einfluß auf die Güte des resultierenden Planes. In unserem Beispiel sollen die Fertigungskosten möglichst gering gehalten werden.

Kann der Suchraum aufgrund seiner Größe nicht systematisch durchlaufen werden, muß ein Planungssystem Heuristiken enthalten, die aus Alternativen die am besten geeignete auswählen und dadurch die Suche steuern. Dieses Wissen kann in Form von Regeln repräsentiert werden. Regeln stellen eine Generalisierung der Vorgehensweise eines Experten in einer speziellen Situation dar, so daß eine Anwendung im falschen Kontext wahrscheinlich ist. Eine andere Möglichkeit der Steuerung stellen Fälle dar. Sie beinhalten eine Vielzahl von Kontrollentscheidungen, die aus globaler Sicht konsistent und bei einem „guten“ Fall eine Lösung produzieren, die optimal ist oder nahe bei einem Optimum liegt.

5 Fallbasiertes Schließen in der Arbeitsplanung

Das Ziel einer fallbasierten Unterstützung des Problemlöseprozesses ist es, bisherige Erfahrungen mit ähnlichen Problemen als Leitlinie für die Lösung neuer Probleme zu verwenden (vgl. auch [AWBS⁺92, WPA92]). Eine erneute, vollständige Durchsuchung des im allgemeinen exponentiell wachsenden Suchraumes soll somit vermieden werden. Ein in bekannten fallbasierten Systemen (vgl. z.B. MEDIATOR [Sim85], PERSUADER [Syc87], CHEF [Ham89]) verwendeter Ansatz zur fallbasierten Unterstützung von Planungsprozessen ist *Transformational Analogy* (TA) [Car83]. Eine gegebene Ausgangssituation A' wird in diesem Ansatz durch die Anwendung von

Operatoren O_R so modifiziert, daß sie einer bereits bekannten Ausgangssituation A entspricht. Die Bestimmung von A und die Transformation $A \mapsto A'$ wird als *Retrievalphase* bezeichnet. A kann dann mit dem bekannten Plan \mathcal{L} zur Lösung von A in die Zielsituation Z transformiert werden. Anschließend wird die so erreichte Zielsituation Z durch die Anwendung weiterer Operatoren O_M so modifiziert, daß die ursprünglich gewünschte Zielsituation Z' erreicht wird. Die Transformation $Z \mapsto Z'$ heißt *Modifikationsphase* und erfolgt durch die Anwendung von entsprechenden Modifikationsregeln. Sie wird durch die noch vorhandenen Differenzen zwischen Z' und Z (*Means-End-Analysis*) gesteuert. Die Lösung für ein gegebenes aktuelles Problem (A', Z') erhält man dann durch die Konkatenation der in der *Retrievalphase* O_R , in der *bekanntem Lösung* $O_{\mathcal{L}}$ und in der *Modifikationsphase* O_M angewendeten Operatoren (Regeln).

Fallbasierte Systeme, die einen Transformational Analogy Ansatz verfolgen sind im allgemeinen leicht zu realisieren. Unsere erste prototypische Implementierung des CABPLAN Systems [PPW92] verfolgte daher diesen Ansatz zur fallbasierten Unterstützung der Arbeitsplanung. Für die vorgestellte Anwendung ergaben sich dabei eine Reihe von Problemen, die in den folgenden Abschnitten ausgeführt werden sollen.

Keine Kriterien zur Auswahl von Fallbeispielen: Transformational Analogy liefert keine Kriterien für die Auswahl von geeigneten Fallbeispielen, d.h. für die Bestimmung von A , so daß zumeist auf ad-hoc Methoden, z.B. Tree-Matching, zurückgegriffen werden muß.

Trennung von Retrieval und Modifikationsphase: Transformational Analogy Prozesse werden in zwei Phasen aufgeteilt, die in existierenden Systemen zudem durch unterschiedliche Inferenzmechanismen und Wissensquellen realisiert werden. Die so verfolgte strenge Trennung in Retrieval- und Modifikationswissen ist in der Praxis aber oft nicht möglich, da für das erfolgreiche Retrieval eines geeigneten Fallbeispiels Wissen über die bei Verwendung des Fallbeispiels notwendigen Modifikationen unabdingbar ist.

Mangelnde Qualität der Lösungen: Fallbeispiele werden im Transformational Analogy Prozeß als Makros mit einem genau definierten Kontext aufgefaßt. Sie können damit nur als Ganzes angewendet werden. In der Modifikationsphase müssen somit unter Umständen in der Lösung \mathcal{L} des Fallbeispiels getroffene Entscheidungen und die damit verbundenen Effekte durch die Anwendung geeigneter Operatoren wieder rückgängig gemacht werden.

Steigerung der Komplexität: Die Verwendung von Fallbeispielen als Makros hat zudem den aus der Literatur bekannten Effekt, daß bei einer großen Zahl von Fallbeispielen keine Effizienzverbesserung, sondern eher eine Verschlechterung eintritt, da die Zahl der zu berücksichtigenden Entscheidungsalternativen anwächst.

Informationsverlust: Fallbeispiele besitzen als Makros keine innere Struktur. Entscheidungsalternativen, lokale und globale Begründungen für die Wahl einer bestimmten Alternative und während der Planung aufgetretene Fehlschläge die eine gezielte Modifikation der Lösung ermöglichen könnten, werden beim TA-Ansatz nicht im Fallbeispiel repräsentiert.

In den folgenden Abschnitten soll nun unser Ansatz, der die geschilderten Probleme beseitigt, vorgestellt werden. Basis der fallbasierten Steuerung der Arbeitsplanung ist der ebenfalls von Carbonell vorgeschlagene *Derivational Analogy* Ansatz [Car86]. Bei diesem Ansatz dient ein Fallbeispiel als Basis für ein am aktuellen Problem \mathcal{P}' orientiertes *Replay* der für das Fallbeispiel gefundenen Lösung \mathcal{L} . Das Replay kann, eine entsprechende Systemarchitektur vorausgesetzt, an zu Konflikten führenden Stellen gestoppt, an beliebigen Stellen wieder aufgenommen bzw. zurückgesetzt werden. Ein gefundenes Fallbeispiel übernimmt - soweit möglich und sinnvoll - in diesem Ansatz die Steuerung des Planungssystems für das aktuell zu lösende Problem, d.h. die Instantiierung einer Ordnung über den in Abschnitt 4.3 beschriebenen Konfliktmengen. Die Betrachtung von im Fallbeispiel nicht verfolgten Entscheidungsalternativen (gezielte Expansion des Suchraumes) und die Vermeidung von bereits in früheren Situationen aufgetretenen Fehlschlägen (Beschränkung des Suchraumes) wird somit ermöglicht.

Folgen wir der Auffassung von R.C. Schank "Retrieval is understanding" [Sch82], so sind Retrieval und Modifikation nicht streng getrennt, sondern im Gegenteil eng miteinander verzahnt. Diese Vorgehensweise kann durch die Erweiterung des Ansatzes auf nichtlineare, hierarchische Planungsprozesse, erreicht werden. Es können dann die Retrieval- und Modifikationsphase verschränkt ablaufen, da sich der Gesamttablauf jeweils an den definierten Abstraktionsebenen orientiert. Die in der Retrievalphase zur Beurteilung der Nützlichkeit eines Fallbeispiels durchgeführten Inferenzen, können dann - falls kein geeignetes Fallbeispiel in der Fallbasis gefunden wird - direkt zur generativen Neuplanung genutzt werden.

Definition 7 (Fallbeispiel) Ein Fallbeispiel \mathcal{F}_k beschreibt die Lösung \mathcal{L}^H des Planungsproblems \mathcal{P}^H auf unterschiedlichen Abstraktionsebenen E_i d.h. $\mathcal{F}_k = \{(\mathcal{P}_n, \mathcal{L}_n, \mathcal{R}_n), (\mathcal{P}_{n-1}, \mathcal{L}_{n-1}, \mathcal{R}_{n-1}), \dots, (\mathcal{P}_0, \mathcal{L}_0, \mathcal{R}_0)\}$. \mathcal{F}_{ki} beschreibt dabei das Fallbeispiel \mathcal{F}_k auf der Abstraktionsebene E_i . Ein Fallbeispiel \mathcal{F}_{ki} wird dabei durch ein 3-Tupel definiert, welches das zu lösende Planungsproblem \mathcal{P}_i , die Lösung \mathcal{L}_i von \mathcal{P}_i , sowie eine Rechtfertigung \mathcal{R}_i , die die Herleitung von \mathcal{L}_i , sowie die in \mathcal{L}_i vorhandenen Abhängigkeiten und Entscheidungsalternativen, beschreibt.

6 Auswahl eines geeigneten Fallbeispiels

Eine Reduzierung der Komplexität von Planungsprozessen durch den Einsatz von Fallbeispielen ist nur dann zu erwarten, wenn es gelingt ein für eine aktuelle Problemstellung geeignetes Fallbeispiel zur Steuerung des Inferenzprozesses so auszuwählen, daß die Komplexität des Auswahlprozesses und der Modifikationsphase im Mittel unter der Komplexität einer in der Qualität vergleichbaren generativen Neuplanung liegt (vgl. dazu auch [HW91]).

Aus einer abstrakten Sicht ist daher für die Auswahl eines Fallbeispiels die Frage nach der *Nützlichkeit* zentral. Ein Fallbeispiel ist dann nützlich für ein aktuelles Problem, wenn die gesuchte Lösung sich wenig von der bekannten Lösung unterscheidet, d.h. mit geringem Aufwand aus der Lösung des entsprechenden Fallbeispiels abgeleitet werden kann. In diesem Sinne sind z.B. die bei der Lösungsanpassung anfallenden Modifikationskosten ein Maß, das die Nützlichkeit eines Fallbeispiels in einer aktuellen Situation bewertet. Betrachten wir aber diese Vorgehensweise im Kontext des Problemlösens, so ist festzustellen, daß dies ein *a posteriori Kriterium* ist, denn erst nachdem wir versucht haben, das aktuelle Problem auf der Basis des gewählten Fallbeispiels zu lösen, können wir beurteilen, ob der eingeschlagene Weg zu einem Erfolg geführt hat oder nicht, bzw. welcher Aufwand noch zur Lösungsanpassung nötig war.

Das a posteriori Kriterium *Nützlichkeit* von Fallbeispielen wird daher, mangels anderer Alternativen, im fallbasierten Schließen auf den Begriff der *Ähnlichkeit* von Problemstellungen reduziert. Diese Vorstellung liegt in der Hoffnung begründet, daß die *Ähnlichkeit der Problemstellungen* die *Nützlichkeit für die Problemlösung* impliziert. Eine solche Vorgehensweise ist aber nur dann anwendbar, wenn es gelingt aus dem Unterschied zweier bekannter Problemstellungen eine Abschätzung der zu erwartenden Modifikationskosten für die entsprechenden Lösungen abzuleiten (vgl. auch [HSRW91]).

Ein naheliegender Ansatz zur Abschätzung von zu erwartenden Modifikationskosten ist die Verwendung von Abstraktionsabbildungen (vgl. auch [Hor92]). Zwei Probleme $\mathcal{P} = (A_0, Z_0, D^H)$ und $\mathcal{P}' = (A'_0, Z'_0, D^H)$ werden als ähnlich bzgl. der Abstraktionsebene E_i bezeichnet, falls sie auf dieser Ebene eine gemeinsame abstrakte Lösung besitzen, d.h. $\mathcal{L}_i = \mathcal{L}'_i$. Wird die verwendete Abstraktionsabbildung nun so gewählt, daß in der zu betrachtenden abstrakten Lösung \mathcal{L}_i die wesentlichen Punkte der Lösung \mathcal{L}^H festgelegt sind, so erhält man eine gute Abschätzung der noch zu erwartenden Modifikationssschritte (vgl. auch [BD92]).

Entscheidend für die praktische Durchführbarkeit eines solchen Ansatzes ist einerseits die Wahl einer geeigneten komplexitätsreduzierenden (domänenabhängigen) Abstraktionsabbildung und andererseits die Tatsache, daß um die Ähnlichkeit zweier Probleme $\mathcal{P}, \mathcal{P}'$ zu bestimmen, zunächst einmal die abstrakte Lösung \mathcal{L}'_i für das Problem \mathcal{P}' bestimmt werden muß.

Für die hier betrachtete Anwendung der Arbeitsplanung stellen beide Punkte kein Problem dar. Sinnvolle, d.h. komplexitätsreduzierende Abstraktionsabbildungen sind in der Anwendung gegeben (vgl. Abschnitt ttttt), die Bestimmung einer initialen abstrakten Lösung \mathcal{L}'_n (d.h. die Bestimmung der Aufspannung des Werkstückes) als Ausgangspunkt für einen Fallvergleich, der die zu erwartenden Modifikationskosten berücksichtigt, ist ohne großen Aufwand möglich.

6.1 Phase I: Bestimmung relevanter Eigenschaften der Problembeschreibung

Die Bestimmung der Ähnlichkeit eines gegebenen Fallbeispiels \mathcal{F} zu einer aktuellen Problemstellung \mathcal{P}^H können wir als einen zweiphasigen Prozeß verstehen. In der ersten Phase müssen aus dem Fallbeispiel und der aktuellen Situation relevante Eigenschaften extrahiert werden, die dann die Basis für den in der zweiten Phase stattfindenden Ähnlichkeitsvergleich bilden. Um den Aufwand zu verkürzen kann die erste Phase bereits bei der Aufnahme eines Fallbeispiels in die Fallbasis des Systems durchgeführt werden, während die zweite Phase zur Laufzeit ausgeführt werden muß. In diesem Abschnitt wollen wir zunächst den der ersten Phase zugrundeliegenden Algorithmus betrachten. Im nächsten Abschnitt werden wir dann den Algorithmus an einem Beispiel verdeutlichen.

Die im letzten Abschnitt vorgestellten Überlegungen führten zu dem Ergebnis, daß alle die Modifikationskosten beeinflussenden Eigenschaften einer Problemstellung \mathcal{P}^H bzw. der entsprechenden Lösung \mathcal{L}^H für einen Fallvergleich relevant sind. Wichtigster Punkt sind dabei die im gefundenen Plan \mathcal{L}^H vorhandenen Abhängigkeiten, z.B. die Teilzielinteraktionen. Sie müssen im Fallbeispiel explizit repräsentiert und in die Ähnlichkeitsbetrachtung einbezogen werden, da eine Verletzung von im Plan vorhandenen Abhängigkeiten bei der Modifikation einen nicht vorhersehbaren Anpassungsaufwand für das aktuelle Problem bedeuten kann. Im Gegensatz dazu können interaktionsfreie Teilpläne ohne Schwierigkeiten gelöscht, verändert bzw. hinzugefügt werden, d.h. der zu erbringende Modifikationsaufwand ist in diesen Situationen vorhersehbar. Da durch den zu generierenden Plan bestimmte Ziele erreicht werden sollen, sind Übereinstimmungen in der Zielsituation Z' zunächst relevanter als Übereinstimmungen in der Ausgangssituation A' . Die Übereinstimmungen der Ausgangssituationen müssen immer ausgehend von den zu erreichenden Zielen und den im bekannten Plan vorhandenen Abhängigkeiten bewertet werden. Die Bestimmung relevanter Eigenschaften erfolgt dabei ausgehend von der Abstraktionsbene E_n rekursiv bis zur Ebene E_0 .

Eingabe: Hierarchisches Planungsproblem $\mathcal{P}^H = (A, Z, D^H)$
 Nichtlinearer, hierarchischer Plan als Lösung \mathcal{L}^H von \mathcal{P}^H

Ausgabe: Partitionierung von Z_i in Mengen U_{ij} unabhängiger Teilziele bzgl. der Lösung \mathcal{L}_{ij}
 Relevante Eigenschaften \mathcal{M}_{ij} des Initialzustandes A_i bzgl. der im jeweiligen U_{ij} zu erreichenden Ziele

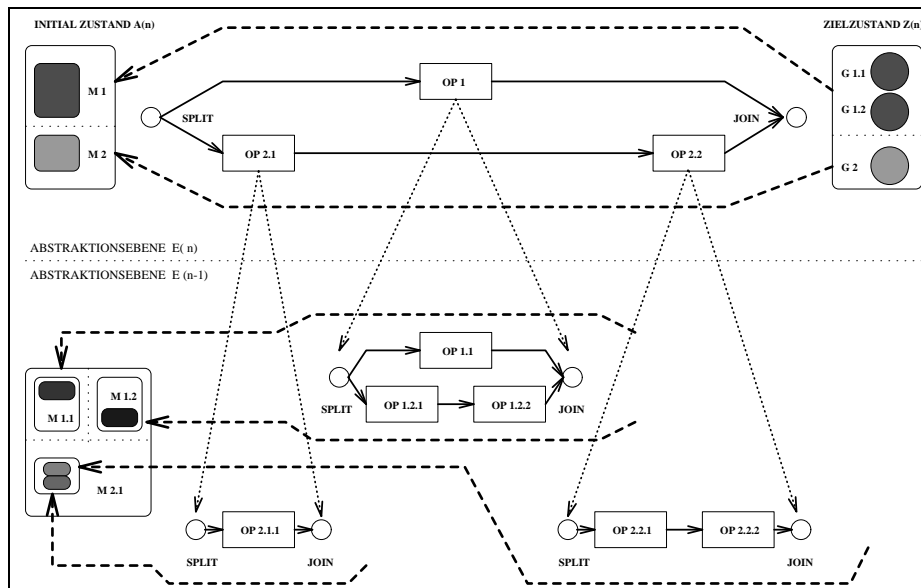
Starte mit der obersten Abstraktionsebene E_n , d.h. $i := n, j := 0$.

1. Bestimme die bzgl. \mathcal{L}_{ij} interagierenden Teilziele aus $Z_i = \{\mathcal{G}_1, \dots, \mathcal{G}_m\}$. Je zwei Ziele \mathcal{G} und \mathcal{G}' aus Z_i interagieren bzgl. \mathcal{L}_i , falls sie durch Operatoren o_s, o_t erfüllt werden, die in einem gemeinsamen Teilplan von \mathcal{L}_i liegen. Partitioniere die Zielsituation Z_i in minimale Mengen $U_{ij} = \{\mathcal{G}_{j1}, \dots, \mathcal{G}_{jk}\}$ die bzgl. \mathcal{L}_{ij} nicht interagieren.
2. Propagiere für jedes $\mathcal{G}_{jl} \in U_{ij}$ die Vorbedingungen der in \mathcal{L}_{ij} angewendeten Operatoren o_{l1}, \dots, o_{l1} bis zur initialen Situation A_i .
3. Bestimme Mengen von relevanten Vorbedingungen $\mathcal{M}_{ij} \subseteq A_i$, ausgehend von den in U_{ij} verzeichneten Zielen $\mathcal{G}_{j1}, \dots, \mathcal{G}_{jk}$
4. Generalisiere falls möglich die Vorbedingungen aus \mathcal{M}_{ij} unter Verwendung der Bereichstheorie \mathcal{T}_i .
5. Wechsle zur Abstraktionsebene E_{i-1} und wende das Verfahren rekursiv bis zur Ebene E_0 an.

Der vorgestellte Algorithmus erfüllt die oben formulierten Anforderungen. Er ist eine Erweiterung des *Goal-Driven Similarity Assessment*-Ansatzes (GDSA) [JWM92] auf nichtlineare, hierarchische Planungsprobleme, der auf die Arbeiten im Bereich von EBG [MKKC86] bzw. Barletta & Mark [BM88] und auf die *goal-regression* nach Waldinger [Wal77] aufsetzt. Der Algorithmus liefert für ein gelöstes Fallbeispiel auf unterschiedlichen Abstraktionsebenen E_n, \dots, E_0 , die nicht interagierenden Teilpläne \mathcal{L}_{ij} von \mathcal{L}_i , die Menge der Ziele U_{ji} , die mit diesen Teilplänen erreicht werden können, sowie die zur Anwendung der Teilpläne relevanten Eigenschaften \mathcal{M}_{ij} der Ausgangssituation A_i .

6.2 Beispiel

Der obige Algorithmus soll nun kurz an einem Beispiel verdeutlicht werden. Im folgenden Beispiel gehen wir davon aus, daß wir bereits ein hierarchisches Planungsproblem $\mathcal{P}^H = (A_0, Z_0, D^H)$ gelöst haben. Im Beispiel sind auf der obersten Abstraktionsebene E_n die Ziele $Z_n = \{\mathcal{G}_{1.1}, \mathcal{G}_{1.2}, \mathcal{G}_2\}$ zu erfüllen. Dazu existiert ein nichtlinearer Plan, der die Ausgangssituation A_n in die Zielsituation Z_n überführt. Der Plan besteht aus den drei Operatoren $Op_1, Op_{2.1}$ und $Op_{2.2}$. Die Ziele $\mathcal{G}_{1.1}, \mathcal{G}_{1.2}$ werden auf dieser Abstraktionsebene durch den Operator Op_1 , das Ziel \mathcal{G}_2 durch die Anwendung der Operatoren $Op_{2.1}$ und $Op_{2.2}$ erreicht. Betrachtet man den Plan ohne *Split* und *Join* Knoten, so kann das Ziel \mathcal{G}_2 unabhängig von den anderen Zielen erreicht werden, während die Ziele $\mathcal{G}_{1.1}, \mathcal{G}_{1.2}$ im gegebenen Plan interagieren und nur zusammen erreicht werden können. Ausgehend von dieser Partitionierung der zu erreichenden Ziele, können wir nun die zur Erreichung dieser Ziele mit dem vorgegebenen Plan notwendigen Eigenschaften der Ausgangssituation bestimmen. Dazu werden die Vorbedingungen der angegebenen Operatoren propagiert. Wir erhalten so die Mengen $\mathcal{M}_1 \subseteq A$ und $\mathcal{M}_2 \subseteq A$ die die notwendigen Eigenschaften von A_n repräsentieren, um mit dem (Teil)Plan \mathcal{L}_n die Ziele $\mathcal{G}_{1.1}$ bzw. $\mathcal{G}_{1.2}, \mathcal{G}_2$ zu erreichen.



Erfolgt nun ein Übergang auf die nächste Abstraktionsebene, d.h. auf die Ebene E_{n-1} , so werden während des Planungsprozesses durch die auf der Ebene E_n angewendeten Operatoren die auf der nächsten Ebene zu erfüllenden Ziele erzeugt und versucht, entsprechende Pläne auf der Ebene E_{n-1} zu finden, die diese Ziele erreichen.

Im Beispiel haben wir auf der Ebene E_{n-1} drei Planungsprobleme und die zugehörigen nichtlinearen Pläne. Durch die erneute Propagierung der Vorbedingungen auf der Ebene E_{n-1} erhalten wir eine weitere rekursive Strukturierung der Mengen \mathcal{M}_1 und \mathcal{M}_2 . In $\mathcal{M}_{2,1}$ benötigen wir sowohl die Eigenschaften aus $\mathcal{M}_{2,1,1}$ als auch die Eigenschaften aus $\mathcal{M}_{2,1,2}$, da auf der Ebene E_n die Operatoren $Op_{2,1}$ und $Op_{2,2}$ nicht unabhängig voneinander waren. In \mathcal{M}_1 können wir zwischen den Mengen $\mathcal{M}_{1,1}$ und $\mathcal{M}_{1,2}$ unterscheiden, da wir auf der Ebene E_{n-1} Ziele erhalten haben, die in dem auf der Ebene E_{n-1} erzeugten Plan nicht interagieren und somit unabhängig voneinander erreicht werden können. Der Algorithmus wird rekursiv fortgesetzt für alle Abstraktionsebenen E_n, \dots, E_0 die relevanten Eigenschaften \mathcal{M}_{ij} der Ausgangssituation A_i ermittelt wurden. Da sich die in einer Abstraktionsebene E_i vorhandenen Abhängigkeiten auf die folgenden Ebenen E_{i-1}, \dots, E_0 vererben, werden so bei einem Übergang auf eine tiefere Abstraktionsebene die Mengen \mathcal{M}_1 und \mathcal{M}_2 rekursiv immer feiner strukturiert.

6.3 Phase II: Bestimmung der Ähnlichkeit

In dieser Phase soll für ein gegebenes Planungsproblem $\mathcal{P}^{H'} = (A_0, Z_0, D^H)$ ein geeignetes Fallbeispiel $\mathcal{F} = (\mathcal{P}^H, \mathcal{L}^H, \mathcal{R}^H)$ ermittelt werden. Diese Aufgabenstellung wird in drei Schritten gelöst: Zunächst muß für das gegebene Problem \mathcal{P}^H auf der Abstraktionsebene E_n eine abstrakte Lösung \mathcal{L}_n erzeugt werden⁴. Im nächsten Schritt wird ausgehend von den im letzten Abschnitt ermittelten relevanten Anforderungen an eine Ausgangssituation A_i für jede der Abstraktionsebenen E_i rekursiv eine Menge von Fallbeispielen (potentiellen Kandidaten) ermittelt, mit deren Lösung \mathcal{L}_i Ziele \mathcal{G}_k des aktuellen Problems auf der Abstraktionsebene E_i erreicht werden können (vgl. auch [Bör92]). Die Menge der zu berücksichtigen Kandidaten wird so von Abstraktionsebene zu Abstraktionsebene weiter reduziert. Da dieser Schritt im allgemeinen nicht zur vollständigen Diskriminierung aller Fallbeispiele ausreicht, d.h. es existieren am Ende noch mehrere potentiell geeignete Kandidaten, wird im letzten Schritt eine Konfliktlösung durchgeführt, d.h. die Konfliktmenge wird auf die Fallbeispiele eingeschränkt, deren Teilpläne auf den unterscheidlichen Abstraktionsebenen beim *Replay* als Leitlinie zur Lösung von $\mathcal{P}^{H'}$ eingesetzt werden sollen.

Die Bestimmung der abstrakten Lösung \mathcal{L}_n (d.h. die Bestimmung der zur Dertigung notwendigen Aufspannungen) erfolgt mit analog zu der in Abschnitt ??? angegebenen Vorgehensweise. Wir wollen uns in diesem Abschnitt daher auf den im Kontext dieser Arbeit interessanten Aspekt der Bestimmung potentieller Lösungskandidaten konzentrieren. Die Beschreibung der Konfliktlösung bei bzgl. den ermittelten Anforderungen gleichwertigen Lösungskandidaten würde den dieser Arbeit gesetzten Rahmen sprengen und soll daher hier nicht näher behandelt werden. Wir verfolgen zur Konfliktlösung einen mehrstufigen Ansatz der sowohl domänenspezifische Heuristiken, wie auch eine weitergehende Analyse der Problemstellung und des Fallbeispiels beinhaltet.

<p>Eingabe: Zu lösendes hierarchisches Planungsproblem $\mathcal{P}' = (A'_0, Z'_0, D^H)$ gelöstes Fallbeispiel $\mathcal{F}_k = ((A_0, Z_0, D^H), \mathcal{L}, \mathcal{R})$</p> <p>Ausgabe: Menge von Zielen die für \mathcal{P}' mit \mathcal{L} erreicht werden können</p> <p>Starte mit der obersten Abstraktionsebene E_n, d.h. $i := n, j := 0$.</p> <ol style="list-style-type: none"> 1. Bestimme die Menge der gemeinsamen Ziele, d.h. $\mathcal{K} = \{G_l \mid G_l \in Z_i \cap Z'_i\}$ 2. Gilt $\mathcal{K} = \emptyset$, dann ist das vorliegende Fallbeispiel \mathcal{F}_k für das aktuelle Problem \mathcal{P}' auf der Abstraktionsebene E_i ungeeignet. Sind alle Fallbeispiele \mathcal{F} der Fallbasis FB zur Lösung von \mathcal{P}' auf der Ebene E_i ungeeignet, dann starte die generative Neuplanung ab der Abstraktionsebene E_i. 3. Betrachte für die Ziele aus \mathcal{K} die in \mathcal{L}_{ij} vorhandenen Interaktionen, d.h. bestimme die Menge der Partitionen $\mathcal{U}_k = \{U_{ij} \mid U_{ij} \text{ aus } \mathcal{F}_k, U_{ij} \subseteq Z_i \text{ mit } Z'_i \cap U_{ij} = U_{ij}\}$. In \mathcal{L}_{ij} interagierende Ziele $\mathcal{G}_l, \mathcal{G}_h$, müssen auch in der Zielsituation Z'_i des aktuellen Problems \mathcal{P} vorliegen. 4. Betrachte die notwendige Eigenschaften des Initialzustandes A'_i, d.h. bestimme Menge $\mathcal{M}_k = \{M_{ij} \mid U_{ij} \in \mathcal{U}_k \text{ und } A'_i \cap M_{ij} = M_{ij}\}$. Relevante Vorbedingungen aus A_i bzgl. der im Fallbeispiel gefundenen Lösung \mathcal{L}_{ij} für die Ziele aus U_{ij}, müssen auf jeden Fall auch in A'_i erfüllt sein. 5. Gilt $\mathcal{M}_k = \emptyset$, dann ist das vorliegende Fallbeispiel \mathcal{F}_k für das aktuelle Problem \mathcal{P}' auf der Ebene E_i ungeeignet. Sind alle Fallbeispiele \mathcal{F} der Fallbasis FB zur Lösung von \mathcal{P}' auf E_i ungeeignet, dann starte die generative Neuplanung ab der Abstraktionsebene E_i. 6. Gilt $\mathcal{M}_k \neq \emptyset$, dann sind auf der Abstraktionsebene E_i die Ziele $G_l \in \bigcup_{U_{ij} \in \mathcal{U}_k} U_{ij}$ des Planungsproblems \mathcal{P} mit der Lösung \mathcal{L}_i des Fallbeispiels \mathcal{F}_k erreichbar. D.h. bestimme die Vereinigung der Mengen der nicht interagierenden Ziele, deren Vorbedingungen in A'_i erfüllt sind. 7. Wechsle zur Abstraktionsebene E_{i-1} und setze das Verfahren rekursiv bis zur Abstraktionsebene E_0 fort.
--

Der vorgestellte Algorithmus liefert, sofern er für alle Fallbeispiele \mathcal{F}_k einer gegebenen Fallbasis durchgeführt wird, eine baumartig strukturierte Repräsentation aller Lösungen \mathcal{L}_{ij} die auf den unterschiedlichen Abstraktionsebe-

⁴Ergeben sich später Differenzen zu den in den Fallbeispielen verfolgten Lösungen, so deuten diese Differenzen auf im Fallbeispiel durchgeführte Backtrackingschritte hin, die dann beim *Replay* der Lösung entsprechend vorweggenommen werden können.

nen zur Erreichung von durch das aktuelle Problem $\mathcal{P}^{H'}$ definierten Ziele \mathcal{G}_k angewendet werden können. Nach Auswahl von einem oder auch mehreren Fallbeispielen und dem *Replay* der entsprechenden Lösung im Kontext von $\mathcal{P}^{H'}$ müssen die noch nicht erreichten Ziele bzw. weiter verfeinerbare Lösungen durch das generative Planungssystem bearbeitet werden. Bei dieser Vorgehensweise kann jedoch nicht ausgeschlossen werden, daß die durch das Fallbeispiel auf oberen Abstraktionsebenen vorgegebene Lösung auf tieferen Abstraktionsebenen doch noch zu Konflikten führt und so ein Backtracking und eine Änderung der vom Fallbeispiel propagierten Lösung notwendig wird.

7 Steuerung der Planung durch Fallbeispiele

Der zentrale Punkt bei der Steuerung des Planungsprozesses ist die Auswahl von Alternativen aus den einzelnen Konfliktmengen (Abs. 4.3), so daß ihre Kombination zu einem Gesamtergebnis führt, das aus globaler Sicht nahe bei einer optimalen Lösung liegt. Bei der Erstellung des Arbeitsplans, dessen Abarbeitung in Abb. 7 dargestellt ist, können die Aufspannungen auch in umgekehrter Reihenfolge durchgeführt werden. Die dargestellte Reihen-

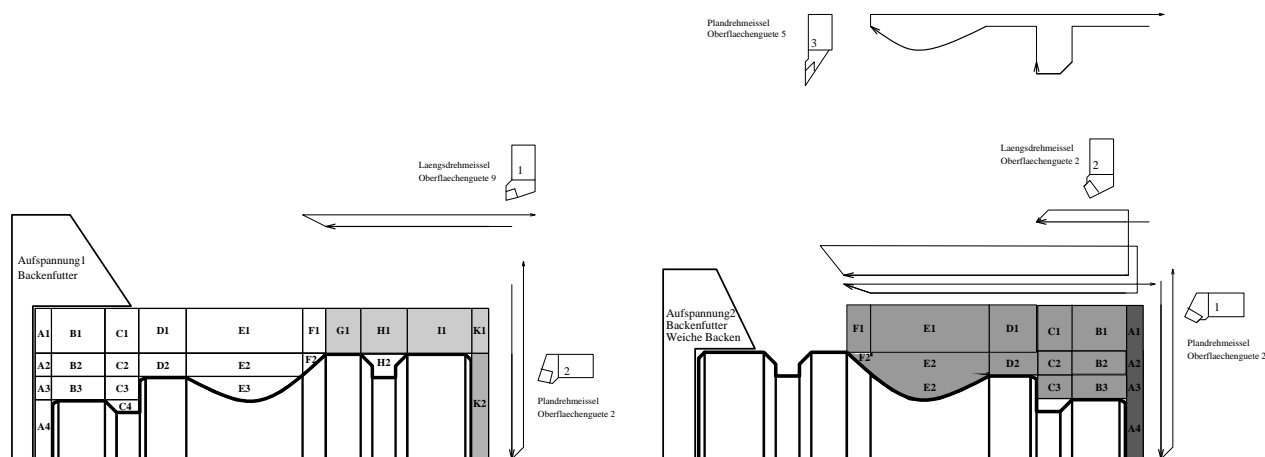


Abbildung 7: Bearbeitungsoperationen der 1. und 2. Aufspannung für das Beispielwerkstück

folge wurde ausgewählt, da das Konturelement *zylinder(B)* einen geringeren Durchmesser als das Konturelement *zylinder(I)* hat und da aufgrund der zu übertragenden Kräfte auf einem möglichst großen Durchmesser gespannt werden soll. Liegen im Fallbeispiel und in der aktuellen Problembeschreibung dieselben Eigenschaften vor, kann die Entscheidung direkt übernommen werden. Sollten darüberhinausgehende Abweichungen zwischen Fallbeispiel und aktuellem Problem bzgl. dieser Entscheidung zu einem späteren Zeitpunkt zu einer Inkonsistenz führen, muß Backtracking durchgeführt werden. Hierzu ist es notwendig, daß die Alternativen der Konfliktmenge weiterhin zur Verfügung stehen. Zur Realisierung eines solchen Ansatzes wird eine Systemarchitektur vorausgesetzt, die das *Replay* eines gefunden Fallbeispiels im Kontext einer aktuellen Problemstellung unterstützt und gleichzeitig Abhängigkeiten zwischen Eigenschaften und Entscheidungen speichert, die für die Fallauswahl notwendig sind.

In CABPlan wird ein modifiziertes TMS [Pau92, Pau93] verwendet, das die Konfliktmengen sowie die Begründungen für die Auswahl bzw. Zurückweisung von Alternativen verwaltet.

8 Diskussion

Der in dieser Arbeit vorgestellte Ansatz reduziert die im allgemeinen vorhandene Komplexität der Arbeitsplanungsaufgabe nicht. Die Integration von suchraumorientierten und wissensintensiven Planungsansätzen führt aber zu einer effektiveren Suche. Im Gegensatz zu den von Hammond u.a. vorgestellten Ansätze wird in CABPLAN ein klassisches nichtlineares Planungssystem zur Modifikation von Fallbeispielen eingesetzt.

Ein grundsätzliches Problem des Ansatzes ist die Festlegung von geeigneten Abstraktionsebenen. In der vorliegenden Domäne sind diese weitgehend vorgegeben. Eine denkbare Erweiterung ist die Kombination mit dem in [Ber92] beschriebenen Ansatz zur automatischen Bestimmung von Abstraktionsebenen.

Im Rahmen des PRODIGY Projektes (vgl. [CV88, VC91]) wurde inzwischen von Veloso [Vel92] ein vergleichbarer Ansatz zur Ähnlichkeitsbestimmung, *foot-printing* genannt, für die Aktionsplanung entwickelt. Veloso berücksichtigt ebenfalls Abhängigkeiten die sich aus Teilzielinteraktionen ergeben, kennt aber keine Abstraktionsebenen und generalisiert die Vorbedingungen nur indem Konstanten durch Variablen ersetzt werden.

Für die Konfiguration wurde in [GP91, Pfi92] ebenfalls ein Ansatz vorgeschlagen, bei dem relevante Anforderungen an Ausgangssituationen bestimmt werden, ohne jedoch die Abhängigkeiten zwischen diesen explizit zu berücksichtigen.

Ein TMS zur Unterstützung der Modifikationsphase wurde für fallbasierte Konfigurierungsaufgaben bereits im JULIA-System von Kolodner eingesetzt [Kol87, HK91].

9 Danksagung

An der Konzipierung und Implementierung des CABPLAN Systems sind Guido Derwand, Holger Hoch, Vera Kamp, Harald Meyer auf'm Hofe, Oliver Noack, Helmut Ritzer, Jürgen Wäsch und Frank Weberskirch beteiligt. Ein Teil der Abbildungen wurde aus der Studienarbeit von Bernd Gaede und Kenan Sabic entnommen. Für viele Diskussionen und Anregungen möchten sich die Autoren bei Reinhard Praeger, den Projektpartnern im CIM-Centrum Kaiserslautern Christian Schulz, Frank Filser und Mirko Radtke, der Arbeitsgruppe in Kaiserslautern, bei den Mitarbeitern des ARC-TEC Projektes im DFKI Kaiserslautern, bei Dietmar Janetzko, Kai Pfitzner, Charles Petrie und Ulrich Junker sowie insbesondere bei Alexander Horz bedanken.

Literatur

- [AWBS⁺92] K.-D. Althoff, Stefan Wess, B. Bartsch-Spörl, D. Janetzko, F. Maurer und A. Voss. *Fallbasiertes Schliessen in Expertensystemen: Welche Rolle spielen Fälle für wissensbasierte Systeme? KI – Künstliche Intelligenz*, 92(4), Dezember 1992.
- [AWBSJ92] K.-D. Althoff, S. Wess, B. Bartsch-Spörl und D. Janetzko, Hrsg. *Workshop: Ähnlichkeit von Fällen beim fallbasierten Schließen*, SEKI-Report, Universität Kaiserslautern, SFB 314, 25.-26. Juni, 1992.
- [BD92] Ralph Bergmann und Andreas Dannenmann. Abstrakte Fallrepräsentation für Abruf und Adaption von Planungsfällen auf verschiedenen Abstraktionsniveaus. In Althoff et al. [AWBSJ92].
- [Ber92] Ralph Bergmann. Learning plan abstractions. In H.J. Ohlbach, Hrsg, *Proc. german workshop on AI (GWAI-92)*. Springer, 1992.
- [BM88] R. Barletta und W. Mark. Explanation-Based Indexing of Cases. In Kolodner [Kol88], Seiten 50–61. Clearwater Beach, Florida, USA, May 10–13, 1988.
- [Bör92] Katy Börner. Komplexitätsreduktion von Konstruktionsaufgaben - Thesenpapier zum Workshop „Kontrolle von Problemlösungsverfahren“. In H.J. Ohlbach, Hrsg, *Proc. German Workshop on AI (GWAI-92)*. Springer, 1992.
- [Car83] J. G. Carbonell. Learning by Analogy: Formulating and Generalizing Plans from Past Experience. In R. Michalski, J. G. Carbonell und T. Mitchell, Hrsg, *Maschine Learning: An Artificial Intelligence Approach*, Band 1. Tioga, Palo Alto, California, 1983.
- [Car86] J. G. Carbonell. Derivational Analogy : A Theory of Reconstructive Problem Solving and Expertise Acquisition. In R. Michalski, J. G. Carbonell und T. Mitchell, Hrsg, *Maschine Learning: An Artificial Intelligence Approach*, Band 2. Morgan Kaufmann, Palo Alto, California, 1986.
- [CV88] Jaime Carbonell und Manuela Veloso. Integrating Derivational Analogy into a General Problem Solving Architecture. In Kolodner [Kol88]. Clearwater Beach, Florida, USA, May 10–13, 1988.
- [FI85] P.E. Friedland und I. Iwasaki. *The concept and Implementation of Skeletal Plans. Automated Reasoning*, 1(161), 1985.
- [FN71] R. Fikes und N. Nilson. *STRIPS: A New approach to Theorem Proving in Problem solving. Artificial Intelligence*, 2(189), 1971.
- [GP91] Andres Günter und Kai Pfitzner. Fallbasiertes Konstruieren mit Bibliothekslösungen. In R. Cunis, A. Günter und H. Strecker, Hrsg, *Das PLAKON Buch*, chapter 9. Springer Verlag, 1991.
- [Ham89] Kristian J. Hammond. *Case-Based Planning: Viewing Planning as a Memory Task*. Academic Press, Boston, Massachusetts, 1989.
- [HK91] Thomas R. Hinrichs und Janet L. Kolodner. The Roles of Adaptation in Case-Based Design. In *Proceedings of the Annual National Conference on Artificial Intelligence AAAI-91*. AAAI, Morgan Kaufmann Publishers, 1991. Anaheim, California, USA.
- [Hor92] Alexander Horz. Ähnlichkeit von Fällen beim fallbasierten Planen. In Althoff et al. [AWBSJ92].
- [HSRW91] B. Humm, C. Schulz, M. Radtke und G. Warnecke. A System for Case-Based Process Planning. In *Proc. of the 1st CIRP-Workshop on Learning in Intelligent Manufacturing Systems*, 1991.
- [HW91] Steven Hanks und Daniel Weld. Systematic adaptation for case-based planning: Algorithm description, formal properties and empirical results. Technical Report tr-91-10-03, University of Washington, Seattle, Dezember 1991.
- [JWM92] Dietmar Janetzko, Stefan Wess und Erica Melis. Goal-Driven Similarity Assessment. In Hans-Jürgen Ohlbach, Hrsg, *Proc. German Workshop on AI (GWAI'92)*. Springer Verlag, 1992. (in Vorbereitung).
- [KCTL91] S. Kamphampati, M. Cutkosky, M. Tannenbaum und S.H. Lee. Combining specialized reasoners and general purpose planners: A case study. In *Proc. AAAI-91*, Seiten 199–205, 1991.
- [Kol87] Janet L. Kolodner. Extending Problem Solver Capabilities through Case-Based Inference. In *Proceedings of the 4th International Workshop on Machine Learning*. ML, Morgan Kaufmann Publishers, 1987. Also in [Kol88].
- [Kol88] Janet L. Kolodner, Hrsg. *Proceedings Case-Based Reasoning Workshop*, San Mateo, California, 1988. DARPA, Morgan Kaufmann Publishers. Clearwater Beach, Florida, USA, May 10–13, 1988.
- [MKKC86] T. M. Mitchell, R. M. Kellar und S. T. Kedar-Cabelli. *Explanation-Based Generalization: A Unifying View. Machine Learning*, 1, 1986.
- [Pau92] Jürgen Paulokat. (A)TMS in Expert Systems. SEKI Report SR-92-06, University of Kaiserslautern, 1992.

- [Pau93] Jürgen Paulokat. Unterstützung der Kontrolle bei der Konfigurierung durch ein problemklassenspezifisches TMS. In *Proceedings der 2. Deutschen Expertensystemtagung (XPS-93)*, Informatik Fachberichte. Springer Verlag, 1993.
- [Pfi92] Kai Pfitzner. Zur Auswahl von Konfigurierungsfällen. In Althoff et al. [AWBSJ92].
- [PPW92] Jürgen Paulokat, Reinhard Präger und Stefan Wess. CABPLAN - fallbasierte Arbeitsplanung. In A. Winklhofer T. Messer, Hrsg, *Beiträge zum 6. Workshop „Planen und Konfigurieren“*. Forwiss München, Maerz 1992.
- [Sch82] Roger C. Schank. *Dynamic Memory: A Theory of Learning in Computers and People*. Cambridge University Press, New York, 1982.
- [Sim85] R. L. Simpson. *A Computer Model of Case-Based Reasoning in Problem Solving: An Investigation in the Domain of Dispute Mediation*. PhD thesis GIT-ICS-85/18, School of Information and Computer Science, Georgia Institute of Technology, Atlanta, Georgia, 1985.
- [Ste81] M. Stefik. *PLanning and Meta-Planning (MOLGEN: Part II)*. *Artificial Intelligence*, 15(111), 1981.
- [Sus75] Gerald J. Sussman. *A Computer Model of Skill Acquisition*. American Elsevier, New York, 1975.
- [Syc87] E. P. Sycara. *Resolving Adversarial Conflicts: An Approach to Integrating Case-Based and Analytic Methods*. PhD thesis GIT-ICS-87/26, School of Information and Computer Science, Georgia Institute of Technology, Atlanta, Georgia, 1987.
- [Tat77] Austin Tate. Generating project networks. In *Proc. IJCAI*, 1977.
- [VC91] M. M. Veloso und J. G. Carbonell. Learning by Analogical Replay in PRODIGY: First Results. In Y. Kodratoff, Hrsg, *Proceedings of the European Working Session on Learning EWSL-91*. Springer Verlag, 1991.
- [Vel92] Maluela Veloso. *Learning by Analogical Reasoning in General Problem Solving*. Phd thesis cmu-cs-92-174, Carnegie Mellon University, August 1992.
- [Wal77] R. Waldinger. *Achieving Several Goals Simultaneously*. *Machine Intelligence*, 8:94–136, 1977.
- [Wil83] R. Wilenski. *Planning and Understanding - A Computational Approach to Human Reasoning*. Addison Wesley, 1983.
- [Wil88] David E. Wilkins. *Practical Planning - Extending the classical AI Planning Paradigm*. Morgan Kaufmann, 1988.
- [WPA92] Stefan Wess, Jürgen Paulokat und Klaus-Dieter Althoff. Fallbasiertes Schließen - Ein Überblick. Technical Report, Universität Kaiserslautern, Fachbereich Informatik, Januar 1992.