

# Repräsentation und Auswahl von Ablaufplanungsverfahren durch Heuristiken<sup>1</sup>

Jürgen Sauer  
Universität Oldenburg, FB Informatik  
Postfach 2503, 2900 Oldenburg

## Zusammenfassung

Ein wesentliches Problem für die Erstellung "guter" Ablaufpläne stellt die Auswahl des "richtigen" Planungsverfahrens dar. Dabei spielen sowohl die zu erfüllenden Zielsetzungen als auch die auftretenden Ereignisse, die zu Störungen der Planung führen können, eine Rolle. Ein Ansatz wird vorgestellt, der es erlaubt, Ablaufplanungswissen und das für die Auswahl von Planungsverfahren nötige Meta-Wissen explizit und transparent darzustellen. Damit wird es z.B. an einem Leitstand möglich, ziel- oder ereignisbezogen geeignete Ablaufplanungsverfahren aus einer Sammlung möglicher auszuwählen und anzuwenden.

## 1. Ablaufplanung

Die **Ablaufplanung** als zeitliche Zuordnung von gegebenen Aufträgen zur Herstellung von Produkten auf vorhandenen Ressourcen, wobei bestimmte vorgegebene Bedingungen erfüllt sein müssen, stellt einen wichtigen Aufgabenbereich innerhalb der Produktionsplanung und -steuerung (PPS) dar. Ablaufplanung umfaßt die Bereiche:

- **Planerstellung:** Neu-Erstellung eines Ablaufplanes, der den Produktionsablauf längerfristig im voraus festlegt. Dieser Bereich ist eher der Produktionsplanung zuzuordnen.
- **Plankorrektur (Replanning):** aktuelle, reaktive Anpassung eines bestehenden Ablaufplans bei sich ergebenden Änderungen im Umfeld der Planung und Produktion. Plankorrektur wird i.a. durch ein **Ereignis** nötig, womit hier eine externe (z.B. Auftragsausfall, Eilauftrag) oder interne (z.B. Maschinen-, Ressourcenausfall) Störung mit Einfluß auf die Planung oder den bereits erstellten Plan gemeint ist. Plankorrektur stellt eine der wesentlichen Aufgaben innerhalb der Produktionssteuerung dar.

### 1.1 Modellierung von Ablaufplanungsproblemen

Ein Ablaufplanungsproblem kann durch ein Schema (A, P, R, HC, SC) beschrieben werden, mit der Bedeutung:

- $A = \{A_1, \dots, A_0\}$  bezeichnet eine Menge von Aufträgen zur Herstellung von bestimmten Mengen von Produkten zu gewünschten Terminen,
- $P = \{P_1, \dots, P_p\}$  bezeichnet eine Menge von herstellbaren Produkten mit Informationen über Varianten, Operationen, deren Dauer und den verwendbaren Ressourcen,

---

<sup>1</sup> erscheint in: Proceedings zur GI-Jahrestagung, Karlsruhe, 1992

- $\mathbf{R} = \{R_1, \dots, R_r\}$  bezeichnet eine Menge von Ressourcen, z.B. Maschinen,
- $\mathbf{HC} = \{H_1, \dots, H_h\}$  bezeichnet eine Menge von Hard Constraints, die bei der Planung eingehalten werden müssen (z.B. technische Produktionsvorschriften),
- $\mathbf{SC} = \{S_1, \dots, S_s\}$  bezeichnet eine Menge von Soft Constraints, die eingehalten werden sollten, aber in gewissem Umfang auch verletzt werden können, um überhaupt Pläne zu finden (z.B. die Einhaltung von Fertigstellungsterminen).

$\mathbf{HC} \cup \mathbf{SC}$  wird auch als Menge der **Constraints** bezeichnet.

Gesucht wird ein **Plan**, der die zeitliche Zuordnung der einzelnen Operationen zu den zugehörigen Ressourcen darstellt und die vorgegebenen Constraints erfüllt. Zu beachten sind dabei vor allem auch die verfolgten **Ziele**, die die planerischen Aufgaben beschreiben, die mit dem erstellten Plan erreicht werden sollen, u.a. globale Ziele wie Termineinhaltung, oder lokale wie Auslastung einer speziellen Maschine oder Reduzierung von Reinigungsaufwand.

Der Problemraum der Ablaufplanung - unter Berücksichtigung von alternativen Produktionsvarianten und Maschinen - läßt sich als **heterogener Und/Oder-Baum** darstellen.

Das Finden eines Ablaufplans entspricht dem Finden einer Lösung für den Und/Oder-Baum für die die Constraints erfüllt sind. Eine Lösung (siehe Abb. 1) ist ein Teilbaum ST, für den gilt:

- die Wurzel gehört zu ST
- gehört ein UND-Knoten zu ST, so auch alle direkten Nachfolgerknoten (falls sie existieren)
- gehört ein ODER-Knoten zu ST, so auch genau ein direkter Nachfolgerknoten (falls Nachfolgerknoten existieren).

Ein Ablaufplan heißt gültig, wenn alle Hard Constraints erfüllt sind, und konsistent, wenn er gültig ist und alle (evtl. abgeschwächten) Soft Constraints erfüllt sind.

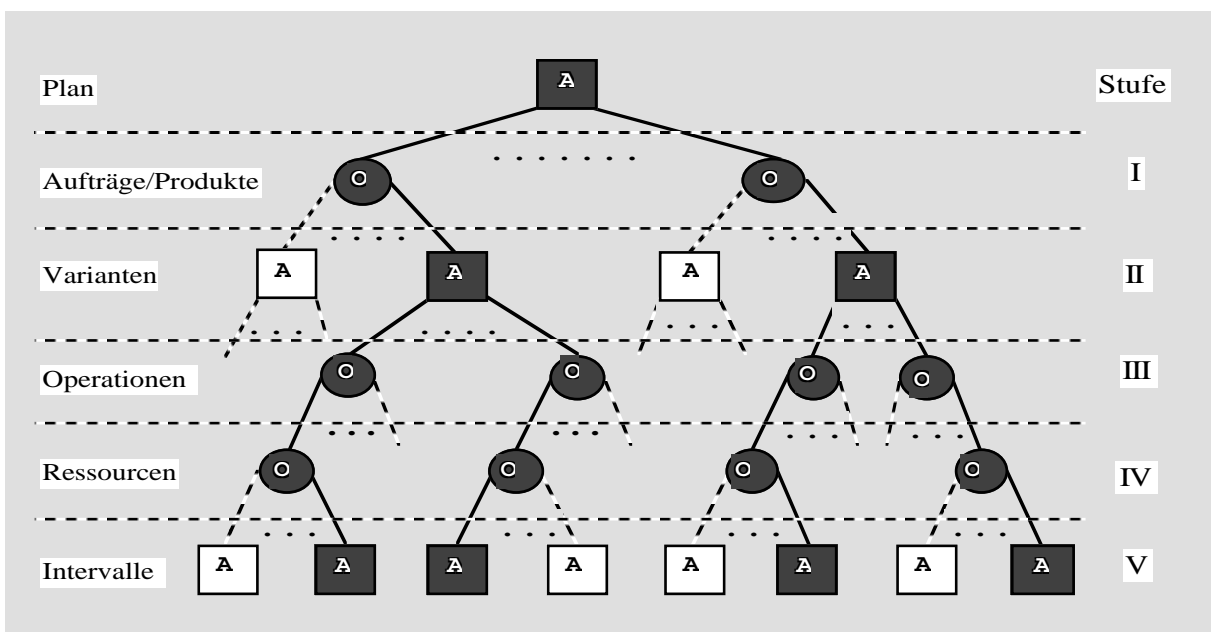


Abb. 1: Und/Oder-Baum der Ablaufplanung

## 1.2 Wissensbasierte Lösungsansätze für Ablaufplanungsprobleme

Optimale Lösungen sind wegen der kombinatorischen Komplexität des Suchraums (er wird durch alle möglichen Lösungen und Teillösungen für den Und/Oder-Baum gebildet) i.a. nicht zu ermitteln. Damit ergibt sich als eine wesentliche Aufgabe der Ablaufplanung die Ermittlung konsistenter, "guter" Lösungen ohne den gesamten Problemraum betrachten zu müssen. Die Bewertung der Lösungen kann dabei quantitativ über Bewertungsfunktionen, die einem Plan einen numerischen Wert zuordnen, oder auch qualitativ durch den Benutzer erfolgen.

Wissensbasierte Ansätze verwenden verschiedene Methoden, vor allem auch das Erfahrungswissen eines Planers, um den Problemraum zu begrenzen und dabei annehmbare Lösungen zu erzielen, u.a.:

- **Heuristische Suche**  
die meisten der beschriebenen Systeme verwenden heuristische Suchverfahren oder heuristische Regeln zur Reduzierung des Suchraumes; heuristische Verfahren des OR, die Prioritätsregeln verwenden [Huber 90], zählen ebenso zu dieser Klasse wie die Constraint gesteuerte Suche bei ISIS [Fox 87] oder der PROTOS-Ansatz [Sauer 90].
- **Problemzerlegung**  
um den Suchraum zu verkleinern wird das Ablaufplanungsproblem in eine Reihe einfacherer Planungsprobleme zerlegt, deren Lösungen die Gesamtlösung ausmachen; praktisch angewandte Zerlegungen sind der auftragsbasierte Ansatz, z.B. [Fox 87], [Sauer 91], der operationsbasierte Ansatz, z.B. [Keng 88], oder der ressourcenbasierte Ansatz, z.B. [Liu 88].
- **Opportunistische Planung**  
in diesem Ansatz werden die erfolgversprechenden Aktionen zuerst betrachtet, z.B. diejenigen, für die die meisten Constraints vorliegen, oder die die Engpässe beseitigen [Smith 90].
- **Kooperative Planung**  
das Planungsproblem wird durch eine Gruppe kooperierender Problemlösungsagenten gelöst, dabei ist ein Agent z.B. für eine bestimmte Ressource zuständig [Ow 88, Sycara 91].

Da eine Vielzahl von Ablaufplanungsalgorithmen existieren, stellt sich als wichtige Aufgabe die Auswahl und Anwendung des in der gegebenen Planungssituation unter den aktuellen Planungszielen am besten geeigneten Algorithmus. Diese Aufgabe soll mit dem im folgenden vorgestellten Ansatz gelöst werden. Der Ansatz basiert auf der expliziten Repräsentation von Zielen und Ereignissen als Kontrollwissen und "dynamischem" Ablaufplanungswissen zur Integration und Auswahl geeigneter Ablaufplanungsverfahren. Damit lassen sich folgende Zielsetzungen erreichen:

- Verbesserung der Planqualität durch
  - zielorientierte Auswahl von geeigneten Planungsalgorithmen
  - angemessene Reaktion auf Ereignisse durch Auswahl geeigneter Plankorrekturverfahren
- Flexibilität in der Anwendung von Planungsalgorithmen
- Integration unterschiedlicher (auch "neuer") Planungsalgorithmen.

### 1.3 Dynamisches Ablaufplanungswissen

Ablaufplanungswissen lässt sich in verschiedene Kategorien unterteilen. Besonders wichtig dabei sind das dynamische Ablaufplanungswissen bestehend aus kombinierbaren Skeletten und Regeln und das Kontrollwissen über die Anwendbarkeit von Ablaufplanungswissen. Im einzelnen sind zu betrachten:

- **Statisches Planungswissen**

dazu gehören Fakten über den Planungsbereich, z.B. Beschreibungen der Produkte und Maschinen und fest implementierte Planungsalgorithmen.

- **Dynamisches Planungswissen**

dies besteht aus den Fakten, die dynamisch während der Planung erzeugt werden, wie der aktuelle Plan oder aktuelle Kapazitäten, und dem dynamischen Ablaufplanungswissen, das aus Planskeletten und Regeln zusammengesetzt wird. Die Unterteilung in Planskelette und Regeln resultiert aus folgender Beobachtung: Viele der Ablaufplanungsansätze können als Algorithmus zur Konstruktion einer Lösung für den Und/Oder-Baum beschrieben werden. Durch den Algorithmus wird ein Durchlauf durch den Baum bzw. die Abarbeitung der einzelnen Stufen des Baumes beschrieben. Dabei werden unterschiedliche Regeln zur Auswahl der nächsten zu betrachtenden Knoten verwendet, z.B. Prioritätsregeln zur Auswahl des nächsten Auftrags. Als Basiskomponenten für einen solchen Ablaufplanungsalgorithmus lassen sich damit identifizieren:

- die zugrundeliegende Durchlaufart (wie man den Baum bzw. die einzelnen Stufen des Baumes durchlaufen muß), die auch als **Planskelett** (nach [Friedland 85]) bezeichnet wird. Durchlaufarten sind z.B. ein Tiefendurchlauf durch den Baum oder die auftragsbasierte, ressourcenbasierte oder operationsbasierte Abarbeitung des Baumes (das in Abb. 2 dargestellte Verfahren stellt das Schema des PROTOS-Verfahrens dar und ist auftragsbasiert).
- die Regeln zur Auswahl unter alternativen Knoten (z.B. welcher Auftrag als nächstes betrachtet werden muß). Regeln können empirisch (z.B. EDD oder SPT), benutzerdefiniert (siehe Abb. 2, Regel 3) oder kombiniert (siehe Abb. 2, Regel 1) sein.

Durch die Kombination von Planskelett und Auswahlregeln ergeben sich eine Vielzahl "neuer" Ablaufplanungsalgorithmen. Abb. 2 zeigt ein Beispiel für ein Planskelett und zugehörige Regeln.

```

WHILE Aufträge zu planen
wähle Auftrag/ Produkt
Regel Kombination von
        (Engpass-Ressourcen, EDD, minimaler Slack, Benutzerpriorität)

wähle Intervall für Auftrag
Regel beginne mit frühest möglichem Start, wenn nötig Verschiebung in die Zukunft

wähle Variante
Regel 'Stammvariante' zuerst, dann Alternativen

WHILE Operationen einer Variante zu planen
    wähle Operation
    Regel in absteigender Reihenfolge

    wähle Apparat
    Regel 'Stammapparat' zuerst, dann Alternativen
END
END

```

Abb. 2: Skelett eines Ablaufplanungsalgorithmus mit möglichen Regeln

- **Kontrollwissen (Meta-Ablaufplanungswissen)**

Das Kontrollwissen beschreibt mit Hilfe der Ziele und Ereignisse die Anwendbarkeit der Algorithmen bzw. Regeln. Bzgl. bestimmter Zielsetzungen sind einige Ablaufplanungsalgorithmen sowie die darin verwendeten Konfliktlösungsstrategien oder Auswahlregeln besser geeignet als andere, dieses Kontrollwissen soll darstellbar und anwendbar sein.

Gleiches gilt auch für Algorithmen zur Plankorrektur. Wichtig ist hier die Angabe der Ereignisse bzw. Störungen, die mit den entsprechenden Verfahren beseitigt werden können.

## 2. Repräsentation von Ablaufplanungswissen durch Heuristiken

Heuristiken können als Basis für die beschriebenen Aufgaben verwendet werden. Eine **Heuristik** (heuristische Regel) der Form "IF situation THEN action" beschreibt, in welchen Situationen welche Aktionen geeignet sind, um ein bestimmtes Ziel zu erreichen [Lenat 83]. Dieser Formalismus wird erweitert zu dem Schema:

<b>HEURISTIC</b>	<name>	eindeutiger Name der Heuristik
<b>IF</b>	<b>SITUATION</b>	<situation> Situation, die zur Anwendung der Heuristik vorliegen muß
	<b>GOAL</b>	<goal> Ziele, die mit der Heuristik erreicht werden können
	<b>EVENT</b>	<event> Ereignisse, die mit der Heuristik verarbeitet werden können
<b>THEN</b>		
	<actions>	Aktionen, die diese Heuristik ausmachen
<b>END HEURISTIC.</b>		

Im IF-Teil werden das Kontrollwissen zur Anwendung der Heuristik durch die explizite Darstellung von Zielen (<goal>) und Ereignissen (<event>) und die Situation, die bei Anwendung vorliegen muß, beschrieben. Die Situation (<situation>) wird durch eine mit AND verkettete Folge von einfachen (Retrieval-) Operationen und Prolog-Literalen beschrieben.

Der THEN-Teil einer Heuristik (<actions>) erlaubt die Repräsentation von Ablaufplanungsverfahren, die Syntax dafür lautet:

```

<actions> ::= <action> | <action> AND <actions>
<action> ::= WHILE <w_condition> DO <actions> END WHILE |
           OR [ <action>, <act_list> ] | <simple_action>
<act_list> ::= <simple_action> | <simple_action>, <act_list>
<simple_action> ::= HEURISTIC-CALL(<heuristic_name>, <goals>, <events>) |
                 <rulecall> | <operation> | <prolog_literal>.

```

Damit besteht der THEN-Teil entweder nur aus dem Aufruf eines vorgegebenen, fest implementierten Verfahrens, z.B.

```

HEURISTIC protos_algorithm
  IF SITUATION
    GOAL [scheduling:[meet_due_dates]]
    EVENT []
  THEN
    call(protos_basic_algorithm)
  END HEURISTIC.
/* Falls das Ziel der Planerstellung Termineinhaltung ist, dann ist der PROTOS-Basis-Algorithmus
geeignet. */

```

oder aus der Beschreibung des Verfahrens (z.B. zeigt Abb. 3 den auftragsbasierten PROTOS-Basis-Algorithmus als Heuristik) mit Hilfe der Konstrukte:

- AND:** einzelne Aktionen werden mit AND konkateniert
- WHILE:** Schleifenkonstrukt, um z.B. alle direkten Nachfolger eines Und-Knotens abarbeiten zu können, dabei bezeichnet <condition> eine Bedingung, die zum Durchlaufen der While-Schleife erfüllt sein muß
- OR:** Alternativenbetrachtung, eine Liste alternativ anwendbarer Aktionen kann angegeben werden; OR unterstützt z.B. das Springen im Baum, um alternative Pfade oder Teilbäume zu betrachten.

Als "einfache" Aktionen (<action>) sind möglich:

- **HEURISTIC-CALL:** Heuristiken können explizit durch ihren Namen (<heuristic\_name>) aufgerufen werden, oder durch Angabe von gewünschten Zielen (<goal>) und Ereignissen (<event>) können gewünschte Heuristiken beschrieben werden; dabei sind erlaubte Aufrufbeziehungen zu beachten
- Regelaufrufe (<rulecall>), z.B. Aufruf einer Regel zur Auswahl einer Maschine
- einfache Operationen (<operation>) auf den Planungsfakten (retrieve, insert, delete), z.B. ein Retrieval von Informationen zu einem bestimmten Objekt einer Klasse von Objekten
- Prolog-Literale (<prolog\_literal>), z.B. Sortierroutinen oder Vergleichsoperationen.

```
HEURISTIC plan_order_based_PROTOS
  IF SITUATION no_plan
    GOAL      [scheduling:[meet_due_dates]]
    EVENT     []
  THEN
    create_orderlist
  AND
  WHILE orders_to_plan DO
    select_order_fifo
  AND
    select_interval_earliest_start
  AND
    select_variant_stem_first
  AND
  WHILE steps_to_plan DO
    select_step_fifo
  AND
    select_app_stem_first
  AND
    OR [plan_step, HEURISTIC-CALL(solve_overlap_basic, [], [])]
  END WHILE
  END WHILE
END HEURISTIC.
```

Abb. 3: Darstellung des PROTOS-Planungsverfahrens als Heuristik

Regeln und die zugrundeliegenden Planungsobjekte werden in ähnlicher Art repräsentiert, Regeln in der Form:

```
RULE <rule_name>                eindeutiger Name der Regel
  IF SITUATION <rule_situation>  Situation, die zur Anwendung der Regel vorliegen muß
  THEN <rule_actions>            Aktionen, die diese Regel ausmachen
END RULE.
```

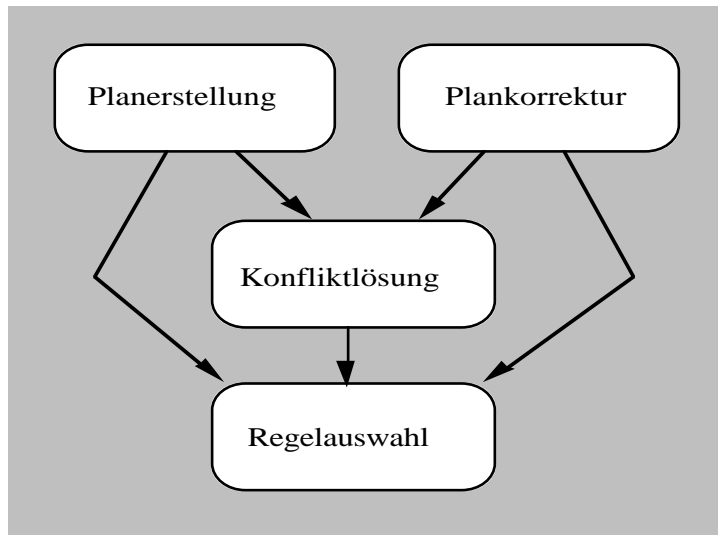
Erlaubte Aktionen in einer Regel sind Operationen, Regelaufrufe oder Prolog-Literale.

Ziele und Ereignisse werden durch eindeutige Namen repräsentiert:

```
GOAL <goalname>.                z.B. GOAL termineinhaltung.
EVENT <eventname>.              z.B. EVENT auftragsausfall.
```

Heuristiken erlauben damit die Repräsentation des Planungs- und Kontrollwissens, das für eine flexible Ablaufplanung nötig ist. Für die Repräsentation von dynamischem Ablaufplanungswissen ergeben sich vier Gruppen von Heuristiken, die alle durch den vorgestellten Formalismus darstellbar sind:

- **Heuristiken zur Planerstellung:**  
beschreiben Verfahren oder Skelette zur Generierung eines Plans und die Anwendbarkeit bezogen auf bestimmte Ziele und Situationen, z.B. das in Abb. 3 dargestellte Verfahren, das bei Planungsziel Termineinhaltung (*meet\_due\_dates*) verwendet werden kann und eine auftragsbasierte Abarbeitung des Baumes beschreibt, bei der auf den einzelnen Stufen bestimmte Regeln zur Auswahl von Knoten aufgerufen werden. Läßt sich eine getroffene Auswahl von Operation, Maschine und Zeit nicht einplanen, so wird eine spezielle Heuristik zur Konfliktlösung aufgerufen.
- **Heuristiken zur Plankorrektur:**  
beschreiben Verfahren oder Skelette zur Korrektur eines Plans und deren Anwendbarkeit bzgl. aufgetretener Ereignisse, Ziele und Situationen, z.B. ein Verfahren, bei dem möglichst wenige Teile des bestehenden Plans verändert werden.
- **Heuristiken zur Auswahl von Regeln:**  
beschreiben die Anwendbarkeit bestimmter Regeln zur Auswahl von Knoten auf den Stufen des Baumes, z.B. verschiedene Prioritätsregeln zur Auswahl von Aufträgen.
- **Konfliktlösungsheuristiken:**  
beschreiben Strategien zur Lösung von Konflikten unter Beachtung bestimmter Situationen, Ziele und Ereignisse. Konfliktlösung ist i.a. die Auflösung einer Überlappung (gleichzeitige Belegung von Maschinen durch mehrere Operationen), aber auch die Lösung von Problemen wie Nichtverfügbarkeit von Ressourcen (Material, Personal).



**Abb. 4: Kategorien von Heuristiken**

Heuristiken zur Auswahl von Regeln und zur Lösung von Konflikten werden innerhalb der Heuristiken zur Planerstellung und Plankorrektur verwendet. Abb. 4 zeigt die erlaubten Aufrufbeziehungen zwischen den verschiedenen Heuristiken.

### **3. Auswahl und Anwendung geeigneter Heuristiken**

Auswahl und Anwendung geeigneter Ablaufplanungsstrategien basieren auf dem in Abb. 5 dargestellten Verarbeitungsmodell, das folgendermaßen beschrieben werden kann:

Aufgrund von vorgegebenen Zielen und Ereignissen werden geeignete Heuristiken - Heuristiken, die die gegebenen Ereignisse und Ziele erfüllen - gewählt und angewendet. Die "am besten geeignete" dieser Heuristiken ist die, die die meisten Ziele abdeckt. Sie wird zuerst versucht. Führt eine Heuristik zum Erfolg, d.h. ein konsistenter Plan wurde erzeugt, so ist die Aufgabe gelöst, falls nicht, so muß eine nächste Heuristik gewählt und angewendet werden.

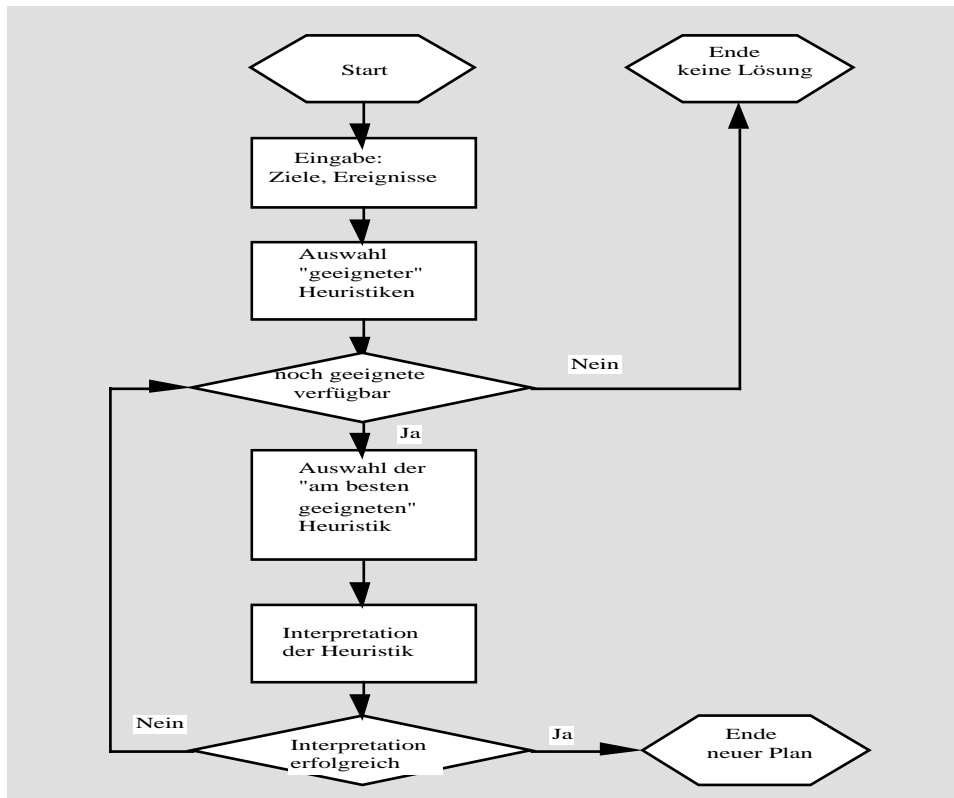


Abb. 5: Verarbeitungsmodell für Heuristiken

Führt die gewählte Heuristik zum Erfolg, so wird ein neuer Plan erzeugt und die Ablaufplanung war erfolgreich. Schlägt sie fehl, so wird die "nächste" Heuristik entsprechend der Auswahlregeln gewählt. Ist keine Heuristik mehr vorhanden, so schlägt die gesamte Planung fehl und es wird kein neuer Plan erzeugt.

Wird eine Heuristik angewendet ("Interpretation einer Heuristik" in Abb. 5), so werden zunächst anhand des Situationsteils die Anwendungsvoraussetzungen geprüft, sowie benötigte Informationen zusammengestellt. Dies geschieht durch Interpretation der durch AND verknüpften Prolog-Literale und Operationen. Dabei ist Backtracking wie in Prolog möglich, um auch Fragen nach der Existenz von Fakten in der Wissensbasis beantworten zu können. Ist der Situationsteil erfolgreich abgearbeitet, so wird der THEN-Teil der Heuristik interpretiert. Schlägt der Situationsteil fehl, so ist die Heuristik nicht anwendbar.

Der Aktionsteil (THEN-Teil) ist erfolgreich, wenn alle mit AND verknüpften Aktionen erfolgreich abgearbeitet werden können. Für die einzelnen Arten von Aktionen gelten folgende Regeln:

- eine OR-Aktion ist erfolgreich, **falls** eine der einfachen Aktionen erfolgreich ist (falls alle Aktionen fehlschlagen, dann schlägt auch die OR-Aktion fehl)
- für eine WHILE-Aktion gilt: solange die While-Bedingung erfüllt ist, werden die durch AND verknüpften Aktionen des Rumpfs interpretiert (falls die Bedingung schon beim ersten Aufruf nicht erfüllt ist, dann wird der Rumpf nicht interpretiert; falls eine der Aktionen im Rumpf fehlschlägt, dann schlägt die gesamte WHILE-Aktion fehl)
- ein Heuristik-Aufruf ist erfolgreich, **wenn** gilt: wird der Name der Heuristik angegeben, **falls** die benannte Heuristik erfolgreich ist, oder werden Ziele und/ oder Ereignisse spezifiziert, **falls** der Verarbeitungszyklus für die gegebenen Ziele/ Ereignisse erfolgreich ist
- ein Regelaufruf ist erfolgreich, **falls** die entsprechende Regel erfolgreich ist

- eine einfache Operation auf Planungsobjekten oder ein Prolog-Literal ist erfolgreich, **falls** die Interpretation durch das Prolog-System erfolgreich ist.

Falls eine der Aktionen fehlschlägt, so schlägt der THEN-Teil der Heuristik und damit die Heuristik selbst fehl.

Eine Regel ist erfolgreich, falls Situationsteil und Aktionsteil der Regel erfolgreich sind. Die Abarbeitung des Situationsteils entspricht der bei Heuristiken, für den Aktionsteil der Regel bestehend aus Regelaufrufen, Operationen oder Literalen gelten die entsprechenden Regeln der Abarbeitung von Heuristiken.

## 4. Realisierung

Das System METAPLAN integriert die Repräsentation unterschiedlicher Ablaufplanungsverfahren sowie deren situations- und zielgerichtete Auswahl und Anwendung. Abb. 6 zeigt die Architekturskizze des Systems.

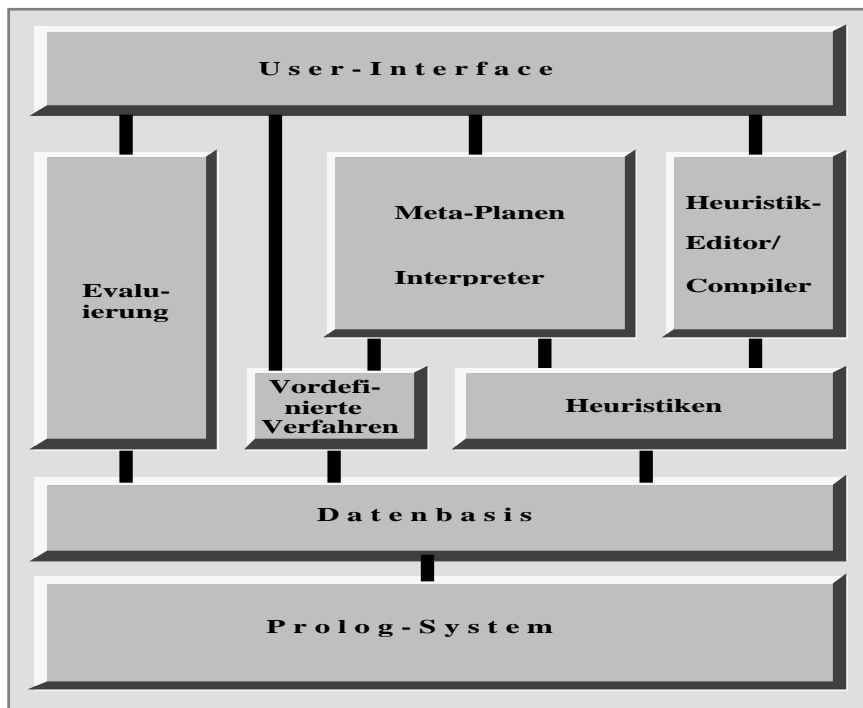


Abb. 6: Architekturskizze METAPLAN

Das gesamte System basiert auf einem kommerziellen Prolog-System (Quintus-Prolog mit Prowindows). Die "Datenbasis" enthält das statische und dynamische Faktenwissen über den Planungsbereich. Die Verknüpfung zu vordefinierten Verfahren wird über die entsprechende Komponente realisiert. "Heuristiken" enthält das dynamische Planungswissen in Form von Heuristiken und Regeln und damit die Realisierung der Heuristik-Sprache. Die "Evaluierung" dient zur Auswertung und damit zum Vergleich verschiedener Pläne auf Basis unterschiedlicher Bewertungsfunktionen wie Summe der Terminüberschreitungen oder Verzug einzelner Aufträge. "Meta-Planen" realisiert die Auswahl und Anwendung der geeigneten Ablaufplanungsheuristiken. Der "Heuristik-Editor" dient zur Integration neuer Heuristiken in das System. Das "User-Interface" umfaßt die Funktionen zur Kommunikation mit dem Benutzer sowie die Darstellung der Ergebnisse als Gantt-Diagramm.

Mit Hilfe des Systems ist es möglich, unterschiedliche Ablaufplanungsverfahren zu integrieren oder Verfahren zu beschreiben und neue zu kombinieren. Damit ist es möglich, verschiedene Verfahren zu testen und zu vergleichen. Neu konstruierte und getestete Verfahren können in Produktionsplanungs- und -steuerungssysteme übernommen werden. Eine Integration des Systems METAPLAN in ein Produktionsplanungs- und -steuerungssystem würde zum einen die Auswahl unter verschiedenen Verfahren unterstützen und zum anderen dem Benutzer selbst die Möglichkeit zur Gestaltung neuer Verfahren geben, z.B. durch Hinzufügen oder Ändern von Auswahlregeln.

## Literatur

- [Fox 87] Fox, M.: "Constraint Directed Search: A Case Study of Job-Shop Scheduling", Pitman Publishers, London, 1987.
- [Huber 90] Huber, A.: "Wissensbasierte Planung und Überwachung in der Fertigung", Erich Schmidt Verlag, 1990.
- [Friedland 85] Friedland, P.E., Iwasaki, Y.: "The Concept and Implementation of Skeletal Plans", in: Journal of Automated Reasoning, No. 1, 1985.
- [Keng 88] Keng, N.P., Yun, D.Y., Rossi, M.: "Interaction sensitive planning system for job-shop-scheduling", in: Oliff, D.M.: "Expert Systems and Intelligent Manufacturing", Elsevier, 1988.
- [Lenat 83] Lenat, D.B.: "Toward a Theory of Heuristics", in: Groner, R., Groner, M., Bischof, W.: "Methods of Heuristics", Lawrence Erlbaum, Hillsdale, 1983.
- [Liu 88] Liu, B.: "A Reinforcement Approach to Scheduling", Proc. 8th ECAI, München, 1988.
- [Ow 88] Ow, P.S., Smith, S.F., Howie, R.: "A Cooperative Scheduling System", in: Oliff, D.M.: "Expert Systems and Intelligent Manufacturing", Elsevier, 1988.
- [Sauer 90] Sauer, J., Appelrath, H.-J.: "Knowledge-Based Production Planning and Scheduling", in: Carnevale, M. et.al.: "Modelling the Innovation: Communications, Automation and Information Systems", IFIP TC7 Conference, North Holland, 1990.
- [Sauer 91] Sauer, J.: "Knowledge Based Scheduling in PROTOS", in: Vichnevetsky, R.: "Proc. of IMACS 91", Trinity College, Dublin, 1991.
- [Smith 90] Smith, S.F., Ow, P.S., Matthys, D.C., Potvin, J.-Y.: "OPIS: An Opportunistic Factory Scheduling System", in: Proc. of 3rd International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems, IEA/AIE, Charleston, USA, 1990..
- [Sycara 91] Sycara, K.P., Roth, S.F., Sadeh, N., Fox, M.S.: "Resource Allocation in Distributed Factory Scheduling", in: IEEE Expert, Feb. 1991.