

## Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>3</b>
1.1	Problemrahmen	3
1.2	Aufgabenstellung der Studienarbeit	4
<b>2</b>	<b>Theoretischer Hintergrund</b>	<b>5</b>
2.1	Zuordnungsprobleme	5
2.2	Vorschläge und Vertauschen	6
<b>3</b>	<b>Problemdefinition „Dienstplanerstellung“</b>	<b>7</b>
<b>4</b>	<b>Implementierung in COKE</b>	<b>10</b>
4.1	Arbeitsweise von COKE	10
4.2	Klassen- und Objektstruktur	11
4.3	Strategien	12
4.4	Vorschläge	13
4.5	Restriktionen	13
4.6	Korrekturen	16
<b>5</b>	<b>PEPSI-Handbuch</b>	<b>17</b>
5.1	Schritte zum Start	17
5.2	Einleitung einer Planung	17
5.3	Optimierung und Benutzereingriff	18
<b>6</b>	<b>Analyse</b>	<b>23</b>
6.1	Evaluierung	23
6.2	Eigenschaften von PEPSI	24
<b>7</b>	<b>Ausblicke und Anregungen</b>	<b>25</b>
7.1	Erweiterungen der Funktionalität	26
7.2	Auswahl der Schichten eines Dienstplanes	26
7.3	Dienstleister	27
<b>8</b>	<b>Anforderungen an eine graphische Repräsentation</b>	<b>27</b>
<b>9</b>	<b>Fazit</b>	<b>29</b>

PEPSI ein Tool zur  
interaktiven Dienstplanung des BORD-Personals der  
Deutschen Bahn AG

Studienarbeit im Fach Informatik

Betreuer

Prof.Dr.Frank Puppe  
Dipl.-Inform. Ciske Busch

Lehrstuhl für Künstliche Intelligenz  
und Angewandte Informatik  
Universität Würzburg

## 1 Einleitung

Die Personaleinsatzplanung ist eine Aufgabe, bei der viele Randbedingungen beachtet werden müssen. Die Fülle an Regelungen ist für den Menschen sehr schwer zu überblicken. Es liegt also nahe, in diesem Bereich ein rechnergestütztes System zu verwenden, das Verletzungen solcher Regeln bereits bei der Planung anzeigt. Darüberhinaus wäre ein automatisiertes oder teilautomatisiertes Planen durch das System wünschenswert. Die Schwierigkeit liegt in der Komplexität der Aufgabe und der Tatsache, daß es oft schwer ist, alle Randbedingungen zu erfassen. Ein Ansatz zur Lösung dieser beiden Probleme ist, das Wissen des menschlichen Planers einzubeziehen.

Im Rahmen dieser Studienarbeit sollte deshalb untersucht werden, inwieweit sich die Dienstplanerstellung des fahrenden Personals bei der Deutschen Bundesbahn mittels der Zuordnungs-Shell COKE unterstützen läßt. COKE verwendet die in [1][Seite 149f] beschriebene „Vorschlägen und Vertauscher“-Strategie zur Lösung von Zuordnungsproblemen. Diese Strategie ermöglicht es einerseits, erfasstes Wissen zur automatisierten Planung zu verwenden, als auch den menschlichen Planer interaktiv am Prozeß der Lösungsfindung mitwirken zu lassen.

Nach einer kurzen Beschreibung des Problemrahmens, aus dem die Aufgabe der Studienarbeit gegriffen ist, folgt eine Spezifikation der Aufgabenstellung in 1.2. Zuvor noch eine Bemerkung zum Sprachgebrauch: Der Begriff *Regel*, der schon in der Einleitung verwendet wurde, wird bei der Bahn verwendet und ist im folgenden synonym zu *Randbedingung*, *Constraint* und *Restriktion* zu verstehen. Ebenfalls synonym sind die Bezeichnungen *Disponent* und *Planer*. Auf weitere Begriffsäquivalenzen wird an entsprechender Stelle in Fußnoten hingewiesen.

### 1.1 Problemrahmen

Die Einsatzplanung des fahrenden Personals bei der Deutschen Bahn AG wird derzeit noch weitgehend von Hand gemacht und es wird wenig mit rechnergestützten Methoden gearbeitet. Die Firma sdkm entwickelt dazu im Rahmen des Projektes BORD ein Assistenzsystem, das dem Disponenten eine graphisch veranschaulichte Planung am Computer ermöglicht und einzuhaltende Regeln in Echtzeit überprüft. Das Gesamtproblem besteht darin, daß „alle Zugläufe auf Zeitintervalle des Personals abgebildet werden“ [3] sollen. Diese Aufgabe läßt sich in mehrere Teilprobleme zerlegen:

1. Aufteilung der Zugläufe in Planungseinheiten
2. Erstellen von Schichtmengen aus Zugläufen
3. Generierung von Dienstplänen aus Schichtmengen
4. Der konkrete Dienstausteiler

Die Studienarbeit wird sich nur mit Teil 3 beschäftigen, zum Verständnis der Problemstellung gebe ich hier einen kurzen und vereinfachten Überblick über jeden der vier Punkte.

### Zu 1. Aufteilung der Zugläufe in Planungseinheiten

In einem ersten Arbeitsschritt werden die Zugläufe zu Planungseinheiten zusammengenommen. Die Zerlegung in Planungseinheiten vereinfacht die folgende Phase der Schichtgenerierung. Um sinnvoll Schichten bilden zu können, sollten Planungseinheiten so gewählt werden, daß sie beispielsweise Zugläufe gleicher Wochentage und möglichst gleicher Fahrtstrecken zusammenfassen.

### Zu 2. Erstellen von Schichtmengen aus Zugläufen

Innerhalb dieser Planungseinheiten werden z.B. mit Weg-Zeit-Diagrammen aus den Zugläufen bzw. den Teilstrecken dieser Läufe Schichten erzeugt. Der Start- und Endort jeder Schicht muß gleich sein, und die Umstanzzeiten (erkennbar im Diagramm) müssen ausreißend lang sein. Die entstandene Menge der Schichtmengen wird nun nach Einsatzort gegliedert und in geeignete Schichtmengen für Gruppen von ca 6-12 Personen unterteilt.

### Zu 3. Erstellen von Dienstplänen aus Schichtmengen

Aus diesen Mengen sollen nun Dienstpläne für die jeweiligen Gruppen zusammengestellt werden. Ein Dienstplan umfaßt einen Planungszeitraum von mehreren Monaten und sorgt dafür, daß an allen Tagen für jede Schicht ein Mitarbeiter eingeteilt ist, unter der Maßgabe, daß die gesetzlichen Rahmenbedingungen nicht verletzt werden und die Dienstzeiten möglichst sozial gerecht verteilt sind.

### Zu 4. Der konkrete Dienstausteiler

Der Einsatz der einzelnen Mitarbeiter wird unter Berücksichtigung von Feiertagen und Urlaub im „Dienstaussteiler“ geregelt. Ihm zugrundegelegt wird der Dienstplan der Gruppe, wobei jedes Gruppenmitglied eine andere Startwoche bekommt und von da an den Plan zyklisch bis zum Ende der Fahrplamperiode durchläuft. In diesen Ausgangsplan werden dann Umplanungen aufgrund von Feiertagsschichten und Urlaubswünschen gemacht. Weiterhin gibt es noch „Springer“, die bei Krankheitsfällen oder anderen Planungseingüssen eingesetzt werden können. In dieser Planungsphase können auch noch personenbezogene Unterschiede berücksichtigt werden.

### 1.2 Aufgabenstellung der Studienarbeit

Wie noch genauer ausgeführt werden wird, kann man das Problem aus Punkt 3 des vorhergehenden Abschnitts als Zuordnung von Dienstplanteilen zu Schichten sehen. Zuordnungsprobleme ähnlicher Art wurden bereits erfolgreich mit REST, einer interaktiven Schichtstundenplanung, und mit WIZARD, einem Programm zur Ressourcenbelegungsplanung, bearbeitet. Beide Programme setzen an COKE, einer allgemein gehaltenen Zuordnungsschale, an.

Basierend auf COKE sollte in dieser Studienarbeit ein Prototyp zur „Erstellung von Dienstplänen aus Schichtmengen“ entwickelt und anhand von Fallstudien erprobt werden. Danach galt es die aufgetretenen Erfolge und Probleme zu beschreiben und anhand des konkreten Problems spezielle und vielleicht allgemeingültige Anforderungen an eine graphische Repräsentation zu erarbeiten.

## 2 Theoretischer Hintergrund

In Kapitel 3 wird verdeutlicht werden, daß es sich bei der Dienstplannerstellung um ein Zuordnungsproblem handelt und sich deshalb ein Werkzeug wie die Zuordnungsshell COKE zur Lösung anbietet. Dieses Kapitel beginnt vorbereitend mit einer Definition des Begriffes „Zuordnungsproblem“ und einem Beispiel in Abschnitt 2.1. Danach folgt eine Beschreibung der in COKE verwendeten Lösungsstrategie, dem „Vorschlägen und Vertauschen“-Algorithmus.

### 2.1 Zuordnungsprobleme

Zuordnungsprobleme zeichnen sich durch disjunkte Mengen von Anbietern und Nachfragern aus. Gesucht ist eine Abbildung der Menge der Nachfrager auf die Menge der Anbieter.

Um diese sehr mathematische Formulierung zu veranschaulichen, wird das von Beck [9] verwendete Beispiel der Oskarverleihung benutzt:

Die Menge der Oskars der verschiedenen Kategorien ist die Nachfragemenge. Die Menge aller nominierten Personen ist die Anbietermenge. Gesucht ist eine Abbildung der Oskarmenge auf die Menge der Anbieter.

Wir können hier schon einige Eigenschaften von Zuordnungsproblemen erkennen. Jeder Oskar muß einer Nominierten/einem Nominierten zugeordnet werden. Eine Person kann mehrere Oskars erhalten (in diesem Falle spricht man von einer 1-N-Zuordnung). Ein Oskar kann aber nicht mehreren Personen zugeteilt werden.

Ein weiterer Bestandteil von Zuordnungsproblemen sind Randbedingungen, die für die Zuordnungen gelten.

In unserem Beispiel wären das z. B. Regeln folgender Art:

- Oskars für eine Kategorie dürfen nur Personen zugeordnet werden, die dafür nominiert sind.
- Oskars für weibliche Hauptrollen dürfen nur an weibliche Darsteller gehen.

Die Randbedingungen der Zuordnungsprobleme sind gewichtet. Die Summe über die Gewichte aller verletzten Constraints ist ein Maß für die Qualität einer Lösung.

Es ist z. B. wichtiger, daß die Kategorie des Oskars mit der Kategorie der Nominierung übereinstimmt, als daß der Film ein Kassenschlager war.

Diese Beschreibung zielt mehr darauf, anschaulich als exakt zu sein. Mehr zu Zuordnungsproblemen findet der Interessierte in [1],[5] und [2].

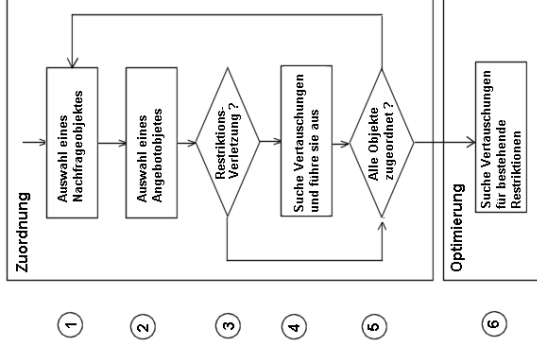


Abbildung 1: Illustration aus [5] zum „Vorschlägen und Vertauschen“-Algorithmus.

### 2.2 Vorschlägen und Vertauschen

Da der Lösungsraum bei solchen Zuordnungsproblemen sehr groß ist, sind Suchverfahren mit heuristischem Wissen zur Lösung von Zuordnungsproblemen besonders interessant. Eine dieser Strategien ist die des „Vorschlägen und Vertauschens“, die auch in dem Zuordnungs-Shell-Baukasten COKE implementiert ist.

Zur Erklärung der „Vorschlägen und Vertauschen“-Strategie wird hier die von Busch [5] verwendete Beschreibung zitiert, da sie sich an der Vorgehensweise von COKE orientiert. (Vergleiche Abbildung 1)

„Die Problemlösung beginnt mit der Auswahl des nächsten zuzuordnenden Nachfrageobjektes aus der vom Benutzer vorgegebenen Startmenge durch den Algorithmus. Diese Auswahl kann zufällig geschehen, aufgrund einer vorbestimmten statistischen Reihenfolge oder nach speziellen Auswahlstrategien, z. B. nach der Zahl der Freiheitsgrade.“ [5]

„Im zweiten Schritt wird ein geeignetes, d. h. passendes und freies Angebotsobjekt ausgesucht und zugeordnet. Auch hier kann die Auswahl automatisch oder durch den Benutzer erfolgen. Im Spezialfall kann der Benutzer ein einzelnes Angebotsobjekt gezielt mit einem Element aus der Nachfragemenge belegen lassen.“

Für die erfolgte Zuordnung erfolgt in Schritt 3 eine Überprüfung aller zugehöriger Randbedingungen. Würden Randbedingungen verletzt,

so sucht der Algorithmus im vierten Schritt nach Tauschmöglichkeiten. Dazu werden zunächst alle Zuordnungen, bei denen Verletzungen aufgetreten sind, zurückgezogen und dann Alternativmöglichkeiten ausprobiert. Dieser Tauschvorgang wird durch eine zeitliche Schranke begrenzt, nach deren Überschreitung die Alternativensuche abbricht und die beste der bis dato gefundenen Zuordnungen ausgewählt wird. Über die Vorgabe der Zeitschranke kann der Benutzer in diesem Schritt die Laufzeit des Algorithmus und die Qualität der Lösung steuern. Falls keine Verbesserung erzielt wurde und wesentliche Randbedingungen nicht berücksichtigt werden konnten, kann der Benutzer später gezielt versuchen, diese aufzulösen.<sup>[5]</sup>

„Eine gefundene vollständige Zuordnung kann schließlich durch weitere Vertauschungen noch optimiert werden, indem weitere Zeit in die Resolvierung verletzter Randbedingungen investiert wird“<sup>[5]</sup>

### 3 Problemdefinition „Dienstplanerstellung“

Ausgehend von einer Beschreibung der Problembestandteile wie zum Beispiel Schichten, Plantage und Einsatzort möchte ich in diesem Abschnitt eine möglichst exakte Definition des Problems geben.

Beim Erstellen der Dienstpläne liegt dem Disponenten eine Menge von Schichten vor, die in einem bestimmten Zeitraum von einer Zugpersonengruppe abgeleitet werden müssen.

#### Schichten

Die Schichten dieser Menge sind an einem oder mehreren Wochentagen gültig. Der Dienstplan dient der langfristigen Planung und berücksichtigt keine kurzfristigen Fahrplan- (und deshalb Schicht-)änderungen. Jede Schicht hat eine eindeutige Bezeichnung, mit der sich alle dazugehörige Daten in Erfahrung bringen lassen. Für den Planungsprozess interessieren insbesondere die Dienstzeit (Beginn und Ende einer Schicht) und die angerechnete Arbeitszeit. Bei längeren Pausen innerhalb einer Schicht kann sich die angerechnete von der tatsächlichen Arbeitszeit unterscheiden.

#### Einsatzort und Qualifikation

Der Einsatzort einer Gruppe ist der, an dem alle von ihr übernommenen Schichten beginnen und auch enden. Eine Ausnahme sind die später beschriebenen Doppelschichten, aber auch hier kehrt das Zugpersonal am zweiten Tag wieder an den Einsatzort zurück. Eine Gruppe sollte möglichst homogen bezüglich der Qualifikationen sein, denn alle Schichten müssen von jedem Mitglied übernommen werden können. Aus diesem Grund wurde bei der Generierung einer Schichtmenge auch auf „Produktivität“ geachtet, d. h., daß z.B. überwiegend Schichten mit einem Zugtyp (ICE, RE o.ä.) zusammengefaßt wurden.

#### Zeitraum - Zyklisches Durchlaufen

Ein Dienstplan umfaßt soviele Wochen, wie sich Mitglieder in einer Gruppe befinden. Im Beispielformular Tabelle 1 wären dies also 6 Wochen. Jedes Mitglied einer Gruppe setzt in diesem Dienstplan mit jeweils einer anderen Startwoche ein und durchläuft von da an zyklisch alle Wochen des Dienstplanes bis zum Ende

Woche	Mo	Di	Mi	Do	Fr	Sa	So	Ruhetage						AZ
								35	45	55	65	75		
1								W						
2								S						
3								W						
4								W						
5								S						
6								W						

Tabelle 1: Typisches Formular zur Einsatzplanung

Sa		So		Ruhetag	
	6	W	36	48	
Ruhe	6	S			
1+2	7,06	W			
3+4+5	11,12+13	S			
6+7+8	14,15+16	W			
Ruhe	Ruhe	S			

Abbildung 2: Ein ausgefüllter Ausschnitt eines Dienstplanes

einer Dienstplanperiode, die üblicherweise ein halbes Jahr dauert, betrachtet man das Dienstplanformular (Tabelle 1) hat man folgende Anschauungsmöglichkeiten: „Zeile 2 ist die zweite Woche des Zugbegleiters Nr. 1“ oder „Zeile 2 ist die erste Woche des Zugbegleiters Nr. 2“. Die Menge der Schichten ist - je nach Anschauung - die, die eine Gruppe in einer Woche ableisten muß, oder die, die ein Zugbegleiter in 6 Wochen ableisten muß. Jeder dieser Schichten muß ein Feld im Dienstplanformular zugeteilt werden (jede Schicht muß in eines der Felder eingetragen werden). Abbildung 2 zeigt in einem vergrößerten Ausschnitt, wie ein solches Formular ausgefüllt aussehen kann.

#### Ruhen, Ruhetage und Übergänge

Nicht belegte Felder<sup>1</sup> oder Sequenzen von unbelegten Feldern im fertigen Dienstplan sind „Ruhe“. Eine „Ruhe“ ist definiert als der Zeitraum zwischen zwei Schichten. Ihre Dauer errechnet sich aus der Zeitdifferenz zwischen dem Ende der letzten Schicht und dem Beginn der darauffolgenden. Ruhephasen von weniger als 36 Stunden heißen Übergänge und Ruhen von mehr als 36 Stunden heißen Ruhetage. Eine Ruhephase von Schichtende Montag 20:00 Uhr bis zum Schichtbeginn am darauffolgenden Mittwoch 8:00 Uhr hat z.B. 36 Stunden und ist damit ein Ruhetag. Ruhetage werden kategorisiert in folgende Typen

- Ruhetage von 36-48 Stunden
- Ruhetage von 48-56 Stunden
- Ruhetage von 56-72 Stunden
- Ruhetage von mehr als 72 Stunden

<sup>1</sup>Ich nenne sie ab jetzt Plantage

Weiter werden sie zusätzlich dazu noch danach unterschieden, ob die Ruhetage einem Wochenende zugeordnet werden können. Für die Dauer und Häufigkeit der Ruhen gibt es gesetzliche Regelungen und tarifrechtliche Vereinbarungen, die eingehalten werden müssen. Der rechte Formularbereich (siehe Tabelle 1) dient als Strichliste für die Ruhephasen und zum Eintragen der Wochenarbeitszeit.

#### Doppelschichten

In jedem Plantage, also in jedem Feld des Dienstplanes, kann normalerweise nur eine Schicht eingetragen werden. Ausnahmen bilden Kurzschichten, die zu einer Doppelschicht zusammengefasst werden dürfen und damit auch einen erlaubten Tagedienst bilden. Ein zweiter Typ von Doppelschicht muß bei der Planung besonders berücksichtigt werden: Schichten mit Auswärtsübernachtungen. Die beiden Tagestappen dieser Schichten dürfen nicht wahrfrei in Plantage eingetragen werden, sondern müssen zwei aufeinanderfolgende Plantage im Zyklus jedes Zugbegleiters belegen.

#### Regeln

Es gibt eine Reihe fester Vorgaben, die beim Planungsprozess berücksichtigt werden müssen. Sie ergeben sich größtenteils aus der Dienstauverschrift (DDV) der Deutschen Bundesbahn. Diese und einige weiche Nebenbedingungen, die im Idealfall auch eingehalten werden sollen, werden ausführlich in [6] beschrieben. Hier nur einige typische Beispiele:

- Die maximale Arbeitszeit, die in einem 7-Tage-Zeitraum geleistet werden darf, ist beschränkt (55 Stunden).
- Jeder Angestellte benötigt eine Mindestzahl von Ruhetagen, die hochgerechnet auf einen vorgegebenen Zeitraum (ein Jahr) eingehalten werden müssen. Ruhetage müssen in festgelegten Mindestabständen (höchstens 144 Stunden) eingelegt werden.
- Die Folge der Schichten (mit Ausnahme der Doppelschichten) muß bestimmte Mindestruhezeiten gewährleisten.
- Die Arbeitszeiten sollten möglichst gleichmäßig auf die Wochen verteilt sein (unter Berücksichtigung der Ruhezeiten)

Die beim Planungsprozess beachteten Regeln bzw. Constraints lassen sich außer der Unterscheidung in harte und weiche Regeln noch weiter klassifizieren.

#### • Harte Vorgaben begründet in der Logik

Diese Constraints müssen unbedingt eingehalten werden, um einen gültigen Dienstplan zu erhalten. Sie sind nicht explizit in irgendeinem Regelkatalog gefordert, sondern intuitiv klar. Ein Beispiel wäre ein Constraint, das dafür sorgt, daß der Wochentag einer Schicht gleich der des zugeordneten Plantages ist, um z. B. keine Dienstagsschicht am Mittwoch einzuplanen.

#### • Harte Vorgaben aus „Regelir“

Diese Constraints sind ebenfalls verbindlich. Ihr Ursprung liegt in arbeitsrechtlichen Regelungen, die auch in [6] zusammengefaßt werden. Ein Beispiel ist der maximale Zeitabstand, der zwischen zwei Ruhetagen liegen darf.

#### • Weiche Constraints als „Güteindikator“

Solche Randbedingungen sind Soll- nicht Mußbedingungen. Sie dienen dazu festzustellen, welcher Dienstplan im Rahmen der zulässigen Dienstpläne besser, d. h. sozial verträglicher oder wirtschaftlicher, ist als andere. Solche Constraints sind teilweise als Soll-Regeln in der DDV festgeschrieben oder entspringen den Vorstellungen der Disponenten von einem angenehmen Dienst. Ein solches Constraint wäre z. B., Übergänge möglichst lange zu halten.

#### • Weiche Constraints begründet in der Strategie

Solche Randbedingungen lenken die Wahl der Plantage aus strategischen Gründen. Ein Beispiel für solch eine Strategie wäre: „Halte die Übergänge zwischen zwei aufeinanderfolgenden Schichten so kurz wie möglich, um nicht im späteren Verlauf der Planung in einen Engpaß zu kommen“.

#### Problemdefinition

Das in dieser Planungsphase zu lösende Problem ist das Finden einer Zuordnung von Plantagen zu Schichten, die alle harten Regeln einhält und die Qualitätsbedingungen möglichst optimal erfüllt.

Da eventuell auch zwei Schichten an einem Tag eingeplant werden können, ist die Zuordnung nicht eindeutig. Man nennt solche Probleme auch 1-n-Zuordnungsprobleme.

## 4 Implementierung in COKE

Kapitel 3 hat gezeigt, daß es sich bei der betrachteten Problemstellung um ein Zuordnungsproblem handelt. Im folgenden erkläre ich zunächst die Funktionsweise von COKE und darauf aufbauend die konkreten Schritte zur Umsetzung der Problemdomäne in COKE. Gemäß der Tradition Projekte, bzw. Programme nach Anfangsbuchstaben einer sinnigen Beschreibung zu bezeichnen, nenne ich die Personal-Einsatz-Planungs-Implementierung PEPSI.

### 4.1 Arbeitsweise von COKE

Die Arbeitsweise von COKE wird in diesem Abschnitt nur sehr knapp beschrieben, gerade soweit wie zum Verständnis des Kapitels nötig ist. Der interessierte Leser findet in [8] eine genauere Beschreibung. Nachfrager und Anbieter einer Zuordnung werden in COKE zunächst im Klasseneditor als Benutzerklasse definiert und dann mit Hilfe des Objekteditors eingegeben. Der „Vorschlagen und Vertauscher“-Algorithmus benötigt dann zur sinnvollen Ausführung noch folgende Teile:

- Vorschläge
- Restriktionen
- Start-Korrekturen

Attribut-Name	Wertebereich
angerechnete_arbeitszeit	Dauer
plantag	Ein Objekt aus Klasse Plantag
reihungs_id	Zahl
schiebtbeginn	Dauer
schiebtbezeichnung	Bei_Text
schiebtende	Dauer
schiebtplannummer	Zahl
schiebt_id	Zahl
wochentag	Zahl zwischen 1 und 7

Abbildung 3: Die Attribute der Klasse Schicht

Zu Beginn der Zuordnung reiht COKE die Nachfragerobjekte in eine Liste ein. Diese wird nun Element für Element abgearbeitet, d. h. einem Anbieter zugeordnet, so daß am Ende für jeden Nachfrager ein Anbieter vorhanden ist. Zu einem Nachfragerobjekt ermittelt COKE mögliche Anbieter mit **Vorschlägen**. Es wird dann der Anbieter gewählt, der die geringste Strafe durch verletzte **Restriktionen** erzeugt. Bei einer Restriktionsverletzung ermitteln **Korrekturen** konkurrierende Nachfrager, die COKE dann wegzutauschen versucht.

#### 4.2 Klassen- und Objektstruktur

Im wesentlichen gibt es zwei Benutzerklassen, die von Bedeutung sind. Das sind die Klassen **Schicht** und **Plantag**.

Ein reales Schichtobjekt hat einen Gültigkeitszeitraum (z. B. vom 1. April bis zum 31. November 1999, ausgenommen an Feiertagen) und einen oder mehrere Verkehrstage (z. B. Schicht 601 wird an jedem Werktag gefahren). Im Planungsszenario von PEPSI wird der Gültigkeitszeitraum nicht betrachtet, da in der Ausgangsmenge nach Vereinbarung nur Schichten sind, die für den gewünschten Planungszeitraum relevant und Ausnahmen durch Feiertage erst Teil einer späteren Planungsphase sind. Hat eine Schicht mehrere Verkehrstage, muß man für die Zuordnung mehrere COKE-Schichtobjekte daraus generieren, um jedes einzeln zuordnen zu können. Eine Übersicht der Attribute einer Schicht zeigt Bild 3.

Die Plantage stellen die einzelnen Tage im Dienstplan dar. Sie sind die Platzhalter (genauer: Anbieter) für die zu verteilenden Schichten. Ein Plantag ist ein Wochentag einer bestimmten Woche. Ein Attribut kennzeichnet, ob es bevorzugt für eine Ruhephase genutzt werden soll (es gibt Constraints, die davon abhängig sind). Die Anzahl der Plantage ist abhängig von der Größe der Gruppe, die als Eingangsparameter gegeben sein muß. Vor der Zuordnung müssen also aus diesem Parameter die Plantag-Objekte generiert werden. Die genaue Definition der Klasse Plantag zeigt Bild 4.

Eine dritte, allerdings weniger wichtige COKE-Benutzerklasse, ist die Klasse **Woche**. Eine Woche enthält jeweils 7 Plantage und hat eine Vorgänger- und eine Nachfolgerwoche. Über die Wochen wird auch die Verkettenng zu einem zyklischen Dienstplan realisiert. Die Wochenobjekte sind nicht in den Zuord-

Attribut-Name	Wertebereich
woche	Ein Objekt aus Klasse Woche
absolutzeit	Dauer
ruhetag	Wahr oder Falsch
schichten	Mehrere Objekte aus Klasse Schicht
wochentag	Zahl zwischen 1 und 7

Abbildung 4: Die Attribute der Klasse Plantag

mungsprozeß eingebunden, sie dienen lediglich intern zur einfacheren Berechnung der Kennzahlen.

#### 4.3 Strategien

Die Vorgehensweisen und Strategien der Disponenten bei der Erstellung eines Dienstplanes sind unterschiedlich und oft von der speziellen vorgegebenen Schichtmenge abhängig. Ich werde im folgenden einige Strategideen diskutieren und auf ihre Umsetzung in der COKE-Implementation eingehen.

##### Ruhephasenschema

Eine Möglichkeit ist, zunächst ein bewährtes Schema für die Anordnung der Ruhephasen zuzugrundelegen und dann die Schichten - soweit sinnvoll - einzupassen. Erst wenn dies nicht mehr möglich ist, darf das Ruheschema gebrochen werden. Ist die Anzahl der Wochen und Wochenschichten bekannt, kann man ein geeignetes Ruheschema generieren lassen. Ein naheliegendes ist z. B. , ein freies Wochenende im Wechsel mit einem Arbeitswochenende stattfinden zu lassen.

Diese Strategie ist durch ein Constraint implementiert, das testet, ob ein Angebotsobjekt (also ein Plantag) ein Ruhetag-Flag besitzt und in diesem Falle eine leichte Verletzung liefert.

##### Blockbildung

Eine weitere Heuristik ist es, Blöcke von Früh- und Spätschichten zu bilden. Dies verhindert den häufigen Wechsel von Früh- auf Spätschicht, der lange Übergangszeiten und damit eine Verringerung der effektiven Arbeitszeit bewirkt. Im späteren Planungsverlauf kann das kritisch für die Unterbringung aller gegebenen Schichten im Planungszeitraum werden. Manche Disponenten gehen deshalb so vor, daß sie zunächst Schichten für einen Tagesschichtentblock von etwa 4 Tagen aneinanderreihen. Darauf folgt entweder ein Übergang zu einem (kürzeren) Nachtschichtentblock oder zu einer Ruhephase.

In die COKE-Umsetzung des Problems geht diese Strategie dadurch ein, daß ein Constraint Schichtfolgen mit kürzeren Übergängen begünstigt. Ein weiteres sorgt dafür, daß die gesetzlichen Mindestlängen für solche Übergänge eingehalten werden. Man kann diese Methode also als „so kurz wie möglich, aber so lang wie nötig“ bezeichnen.

##### Verteilen von Problemschichten

Ein dritter Ansatz besteht darin, Problemschichten (z. B. Doppelschichten mit Auswärtsübernachtung) herauszusuchen und diese am Anfang, wenn noch genügend freie Plantage im Einsatzplan vorhanden sind, gleichmässig über die

Wochen zu verteilen. Spitzenbelastungen werden somit vermieden.

Da die bisherige Prototyp-Implementation keine Doppelschichten kennt, ist eine Strategie nach diesem Ansatz noch nicht verwirklicht. Eine Realisierung durch eine Vorsortierung der Nachfragerliste ist aber theoretisch mit COKE möglich.

#### Wochenende vor Arbeitswoche planen

Grundsätzlich scheint es vorteilhaft zu sein, zunächst die Wochenendschichten zu planen, da sie durch Constraints stärker eingengt sind, und sich somit weniger kombinatorische Möglichkeiten für ihre Anordnung bieten. Diese Strategie ist implementiert und wird durch eine Vorsortierung der Nachfrager vor Beginn der Zuordnung bewirkt.

#### Wochenweise Planung

Eine wochenweise Planung unterstützt das Teilziel, die Wochenarbeitszeit Woche für Woche möglichst gut auszufüllen. Die Wochenarbeitszeit ist nach oben hin durch ein hartes Constraint begrenzt und die Vorgehensweise, die ersten Wochen innerhalb des erlaubten Spielraums möglichst dicht zu packen, läßt mehr Luft für die Endphase der Planung.

#### 4.4 Vorschläge

Die Vorschlagsfunktionen sind einfach gehalten. Der Vorschlag *V. Wochentag* schlägt zu einer Schicht alle Plantage mit dem richtigen Wochentag vor. Der Geschwindigkeitszuwachs zur naiven Vorschlagsfunktion (*V. Naiv*), einfach alle Plantage vorzuschlagen, ist enorm, da der Suchraum hierdurch schon sehr eingeschränkt wird. Vertauschungen beim „Vorschlagen und Vertauschen“ finden also nur noch innerhalb einer Wochentagspalte im Dienstplan statt. Ein weiterer Versuch (*V. Block*) war, die Vorschlagsliste vorzusortieren, so daß Schichten vorzugsweise an Plantagen eingereiht werden, an denen sie an Schichtblöcke mit kurzen Übergangszeiten angrenzen. Ein spürbarer Gewinn konnte damit nicht gezeigt werden.

#### 4.5 Restriktionen

PEPSI enthält alle Restriktionen aus [6], ausgenommen den Regelungen zu Schichten mit Auswärtsübernahmen und solchen, die unter den vorgegebenen Daten unveränderlich sind. Doppelschichten mit Auswärtsübernahmen wurden in dem mit `sd&am` vereinbarten Dateiformat bisher noch nicht explizit gekennzeichnet. Entsprechende Constraints zu ihrer Berücksichtigung wurden deshalb aufgrund der fehlenden Testmöglichkeit zwar erstellt, aber noch nicht eingebunden. „Unter den vorgegebenen Daten unveränderlich“ sind Kennzahlen, zu denen zwar Regeln in Vorschriften gegeben sind, die aber bereits bei der Übergabe der Dienstplanschichten feststehen. Beispiele sind die Regeln zu den maximal erlaubten Schichten pro Woche (durchschnittlich) und der maximal zulässigen Nachdienstfähigkeit. Diese sind allein von der anfänglichen Schichtmenge abhängig und werden nicht von der Zuordnung beeinflusst. Sie müssen deshalb auch in PEPSI nicht berücksichtigt werden. Zur Information werden diese Kennzahlen jedoch auf Wunsch berechnet und angezeigt. Eine Liste der umgesetzten Constraints mit ihren Gewichtungen und den dazugehörigen Startkorrekturen zeigt Abbildung 5.

Name	Aktiv	Gewichtung	Startkorrektur
C_Wochenarbeitszeit	<input checked="" type="checkbox"/>	Sehr schwere Verletzung	Default-Startkorrektur
C_Verlängerte_Wochenende	<input type="checkbox"/>	Mittlere Verletzung	Lange Regeln Mehrern
C_Nachtschichten	<input checked="" type="checkbox"/>	Sehr schwere Verletzung	Default-Startkorrektur
C_Selber_Wochentag	<input checked="" type="checkbox"/>	Konstanzverletzung	Test
C_Zwei_Schichten_Pro_Tag	<input checked="" type="checkbox"/>	Konstanzverletzung	Default-Startkorrektur
C_Übergänge_In_Haare	<input checked="" type="checkbox"/>	Sehr schwere Verletzung	Default-Startkorrektur
C_Abstand_Der_Ruhen	<input checked="" type="checkbox"/>	Sehr schwere Verletzung	Default-Startkorrektur
C_Ruhen_Lang_Ok	<input type="checkbox"/>	Etwas schwerere Verletzung	Lange Regeln Mehrern
C_Laenge_Arbeitsblock	<input checked="" type="checkbox"/>	Sehr schwere Verletzung	Default-Startkorrektur
C_Ruhen_Kurz_Ok	<input type="checkbox"/>	Etwas schwerere Verletzung	Kurze Regeln Mehrern
C_Schichtüberlappung	<input checked="" type="checkbox"/>	Konstanzverletzung	Default-Startkorrektur
C_Eine_Schicht_Pro_Tag	<input checked="" type="checkbox"/>	Mittlere Verletzung	Default-Startkorrektur
C_Ruheflag_Test	<input checked="" type="checkbox"/>	Mittlere Verletzung	Default-Startkorrektur
C_Kurzüberlappung	<input checked="" type="checkbox"/>	Leichte Verletzung	Default-Startkorrektur

Abbildung 5: Die Restriktionen und ihre (Standard-)Gewichtung

#### Konsistenzverletzungen

Am schwersten gewichtet sind Constraints, die logisch unmögliche Dienstpläne hervorbringen. Diese sind

- **C\_Selber\_Wochentag**  
Diese Restriktion verhindert, daß eine Schicht an einem Wochentag eingeplant wird, an dem sie gar nicht gefahren werden muß.
- **C\_Schichteneberlappung**  
Diese Restriktion führt bei Überlappung von zwei Schichten ebenfalls zu einer Konsistenzverletzung, da jeder Diensttuende nur eine Schicht zur gleichen Zeit übernehmen kann
- **C\_Zwei\_Schichten\_Pro\_Tag**  
Es können maximal zwei Schichten an einem Tag eingeplant werden. Mehr sind nicht einmal zur Überbrückung lokaler Minimas sinnvoll.

#### Schwere Verletzungen

Die nächste Gruppe von Constraints erzeugt schwere und schwerere Verletzungen. Um einen gültigen Dienstplan zu erhalten, darf ebenfalls keines von ihnen verletzt sein. Sie sind meist Repräsentationen von Regeln der DDV.

- **C\_Wochenarbeitszeit**

Die Wochenarbeitszeit für jeden beliebigen 7-Tage-Zeitraum des Dienstplanes darf einen gewissen Wert - zur Zeit 55 Stunden - nicht überschreiten. Die möglicherweise veränderlichen Parameter zu diesem wie auch zu allen anderen Constraints sind nur als Defaultwerte vorgegeben und leicht zu ändern.

- **C\_Nachtschichten**  
Nachtschichten dürfen an nicht mehr als 4 aufeinanderfolgenden Schichten ohne Unterbrechung durch einen Ruhetag eingeplant werden.
- **C\_Uebergaeenge\_In\_Heimat**  
Dies beinhaltet zwei Constraints. Zum einen wird überprüft, ob die Übergänge zwischen den Schichten mindestens 9 Stunden betragen, zum anderen, ob die Regelung zum Anteil an der Dauer der angrenzenden Schichten eingehalten wird. (Siehe dazu [6][3.6.5])
- **C\_Abstand\_Der\_Ruhen**  
Dieses Constraint testet, ob die Zeiträume zwischen den Ruhetagen 144 Stunden nicht überschreiten.
- **C\_Max\_Kurzeuebergaeenge**  
Es darf keine Sequenz von mehr als zwei Übergängen geben, die kürzer als 10 Stunden sind.

Ebenfalls auf Regelungen der DDV beruht folgende Gruppe von Constraints. Sie unterscheiden sich von den vorangegangenen dadurch, daß sie nicht eine Verletzung speziell für eine einzelne Zuordnung, sondern ein Kriterium für die Gültigkeit bzw. Güte des ganzen Zuordnungsprozesses sind. Sie erfordern damit eine grundsätzlich andere Behandlung. Betrachten wir das Beispiel einer Regel, die gerechnet auf das Jahr mindestens 26 Ruhephasen von 48 Stunden Länge erfordert. Bei der Zuordnung der ersten Schicht in einen leeren Dienstplan macht es noch keinen Sinn, diese Regel zu beachten. Erst wenn alle Schichten zugeordnet sind, lohnt es sich, in den Optimierungsläufen dieses Constraint zu berücksichtigen. Da auch nicht mehr eine bestimmte Schicht der Verursacher der Verletzung ist, sondern andere Verursacher gefunden werden müssen, haben die Ruhetage-Constraints auch, jeweils andere Startkorrekturen.

- **C\_Kurze\_Ruhen\_Ok**  
In der Jahreshochrechnung müssen mindestens 26 kurze Ruhen im Dienstplan sein. Ruhen über 72 Stunden können wahlweise als 2 kurze Ruhen genommen werden. Dieses Constraint ist sehr rechenzeitintensiv und auch erst bei der globalen Optimierung nützlich.
- **C\_Lange\_Ruhen\_Ok**  
In der Jahreshochrechnung müssen weiterhin auch mindestens 26 lange Ruhen im Dienstplan sein. Lange Ruhen sind Ruhen über 48 Stunden. Die Gewichtung dieses Constraints ist, wie bei allen dieser Gruppe, auch von der Anzahl der schon bestehenden Ruhen abhängig, um eine schrittweise Verbesserung durch die Korrekturen zu ermöglichen.
- **C\_Verlaengerte\_Wochenenden**  
Die Hälfte der (geforderten) langen Ruhen muß als Wochenendruhe gewährt werden. Dieses Constraint ist oft der Engpass bei Schichtmengen mit vielen Wochenendschichten.

#### Mittlere und leichte Verletzungen

Die letzte Constraintgruppe, die leichte und mittlere Verletzungen erzeugt, bringt heuristisches Wissen von Disponenten der Deutschen Bahn ein. Diese Restriktionen haben keine direkten Auswirkungen auf die Gültigkeit des Dienstplanes.

Ein Plan, der am Ende der Zuteilung nur diese Constraints verletzt, kann gestrost als „guter Dienstplan“ bezeichnet werden. Sie werden dennoch benötigt, um „Richtungsweisungen“ im Anfangsstadium zu geben, in dem noch wenige Zuordnungen stattgefunden haben und deshalb wenige harte oder härtere Restriktionen verletzt werden.

- **C\_Ruheflag\_Test**  
Verwirklicht eine bevorzugte Einhaltung eines vorgegebenen Ruheschemas. Kann jedoch durch eine Verletzung dieses Constraints ein stärkeres Optimalitätskriterium aus den vorherigen Constraintgruppen erfüllt werden, wird das Ruheschema gebrochen.
- **C\_Kurze\_Uebergaeenge**  
Zu lange Übergänge zwischen Schichten in einer frühen Planungsphase verringern die Spielräume für die Ruhen am Ende der Zuteilung. Dieses „weichere“ Constraint bevorzugt also, wenn sonst nichts dagegen spricht, kurze Übergänge. Die Schwere der Strafe bei Verletzung des Constraints ist außerdem abhängig von der Länge des Übergangs.
- **C\_Eine\_Schicht\_Pro\_Tag**  
Zwei Schichten am Tag sind nur in Ausnahmefällen üblich. In jedem Fall soll dieses Constraint eine leichte Verletzung erzeugen, so daß der menschliche Planer noch einmal darauf aufmerksam gemacht wird.

#### 4.6 Korrekturen

Wie bereits im letzten Abschnitt angedeutet, hat jede Gruppe von Constraints ihre spezielle Startkorrektur. Bei Verletzungen, die mit einer konkreten Zuordnung in Zusammenhang stehen, werden alle bereits zugeordneten Schichten desselben Wochentags als mögliche Tauschpartner genommen. Dies macht die *Default-Startkorrektur*<sup>2</sup>.

Die Korrektur *Lange Ruhen Mehrer*<sup>3</sup> wird auf Verletzung des Constraints *C\_Lange\_Ruhen\_Ok* ausgeführt und schlägt alle Schichten zur Neuzuteilung vor, die nach ihrer Entfernung aus dem Dienstplan eine kurze Ruhe in eine lange Ruhe verwandeln.

Ebenso gibt die Korrektur *Kurze Ruhen Mehrer*<sup>4</sup> alle Schichten frei, die nach ihrer Entfernung aus dem Schichtplan eine kurze Ruhe in einem Arbeitsblock erzeugen.

Die Korrektur *Wochenendruhen Mehrer*<sup>5</sup> versucht zu freien Sonntagen auch die störenden Samstags- und Montagsschichten zu finden. Diese werden dann zur Neuordnung in Betracht gezogen, um eine bessere Bewertung zu erreichen.

Bei Zuordnungen von Schichten zu falschen Wochentagen versagt die *Default-Korrektur*. Für diesen Fall gibt es also eine spezielle Korrektur *Wochentag Korrigieren*.

<sup>2</sup>in der Wissensbasis als *S\_Default*

<sup>3</sup>in der Wissensbasis auch *S\_Lange\_Ruhen\_Mehrer*

<sup>4</sup>in der Wissensbasis als *S\_Kurze\_Ruhen\_Mehrer*

<sup>5</sup>in der Wissensbasis als *S\_Wochenendruhen\_Mehrer*



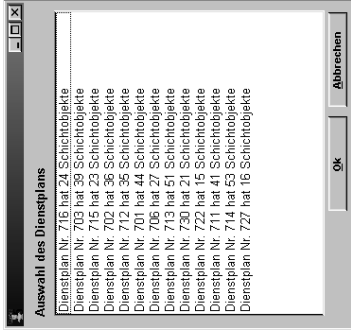


Abbildung 6: Dienstplanauswahlbox

## 5 PEPSI-Handbuch

PEPSI generiert Dienstpläne aus einer Schichtmenge. Der Erfolg hängt dabei wesentlich von der Unterstützung des Bedieners ab, der jedoch bei seinen Tätigkeiten vom System beraten wird.

Das Handbuch beschreibt die Möglichkeiten von PEPSI und führt in die Bedienung ein. Im ersten Abschnitt werden die nötigen Schritte zum Starten und Kompilieren des Programmes beschrieben. Der zweite Abschnitt beschreibt die Einleitung des Zuordnungsprozesses, der dritte die Möglichkeiten zu Beurteilung und Optimierung des Planes.

### 5.1 Schritte zum Start

COKE und PEPSI wurden mit der Programmiersprache LISP entwickelt. Die aktuelle Version bei Erstellung dieser Ausarbeitung lief unter Allegro Common Lisp Version 5.0 unter Windows 95/98 und Windows NT 4.0. In der zu dieser Arbeit gehörigen CD befindet sich im Wurzelverzeichnis die Datei „pepsi.bat“. Ein Ausführen dieser Startdatei leitet alle nötigen Schritte zum Laden der COKE- und PEPSI-Komponenten, Wissensbasen und des Programmcodes ein.

### 5.2 Einleitung einer Planung

Zunächst muß man die Schichtdaten für die zu erstellenden Dienstpläne laden. Dies geschieht über den Menüpunkt „PEPSI/Schichten einlesen“. Eine Datei mit Beispieldaten befindet sich im Ordner „data“. Sie enthält Schichten zu 13 Dienstplänen. Insgesamt handelt es sich um etwa 450 Schichten, die im Format einer speziellen Markup-Sprache vorliegen. Das Einlesen kann also etwas Zeit benötigen, während dieser Zeit sieht man jedoch einen Fortschrittsbalken. Sind alle Schichtdaten eingeladen, fragt PEPSI welchen Dienstplan man bearbeiten möchte (siehe Abbildung 6).

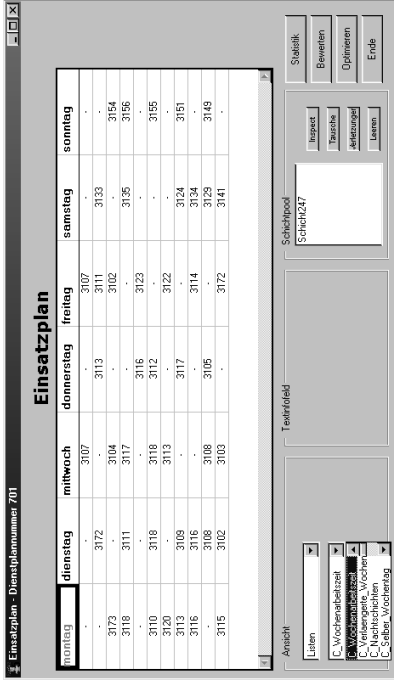


Abbildung 7: Ein optimierter Einsatzplan

Nach dem Einlesen kann man den Zuordnungsvorgang starten. Dies geschieht mit dem Menüpunkt „Zuordnung starten“ des Menüs „Zuordnung“. Das System schlägt, falls nicht schon Planwochen angelegt sind, jetzt eine aus der Arbeitszeit der Schichten berechnete Wochenanzahl vor. Diese geht von einer Arbeitszeit von 38 Stunden pro Woche aus und bietet somit einen guten Anhaltspunkt. In manchen Fällen, besonders bei Schichtmengen mit sehr kurzen Arbeitszeiten, muß man diesen Wert gegebenenfalls erhöhen.

In Abschnitt 4.3 wurde bereits die Bedeutung der Ruheschemata erklärt. PEPSI verwendet standardmäßig ein Ruheschema, das das Wochenende jeder zweiten Woche zur Ruhe vorsteht. Durch einen Aufruf der Funktion `gen_ruheschema` können alternative Ruheschemata gewählt werden.

PEPSI beginnt nun mit der Planung. Anhand eines Fortschrittsbalkens kann man den Vorgang verfolgen. Die Planung wird im allgemeinen „schnell“ starten, und gegen Ende langsamer werden, da die den Fortschrittsbalken bestimmende Größe die Anzahl der noch zuzuteilenden Schichten ist. Bei den letzten zuzuteilenden Schichten werden jedoch i. a. mehr Constraints verletzt und dadurch mehr Vertauschungsschritte nötig sein.

### 5.3 Optimierung und Benutzereingriff

Nach der Zuordnung mit lokaler Optimierung wird ein Fenster mit der Darstellung des Ergebnisses - einem Einsatzplan - geöffnet. Ein Beispiel zeigt Abbildung 7. Dies muß nicht als endgültiges Ergebnis gesehen werden. Vom Benutzer können hier weitere Aufrufe zur globalen Optimierung aufgerufen und Schichten per Mausclick verschoben werden. Vor einer weiteren Optimierung kann er die Rahmenbedingungen verändern, indem er z. B. die Bewertungen von Constraints verändert und so den gewünschten Ergebnissen vielleicht näherkommt. Ein „Spielen mit dem System“ wird also ermöglicht.

Einsetzplan - Dienstplannummer 716

### Einsetzplan

montag	dienstag	mittwoch	donnerstag	freitag	samstag	sonntag
Ruhe	20:14:00	15:30:00	14:16:00	12:57:00	06:34:00	
Ruhe	03:51:00	20:14:00	Ruhe	Ruhe	Ruhe	15:00:00
14:16:00	07:14:00	03:51:00	03:51:00	Ruhe	14:57:00	Übergang
03:51:00	03:51:00	03:51:00	03:51:00	Ruhe	Ruhe	Ruhe
11:25:00	11:25:00	11:25:00	11:25:00	11:25:00	Ruhe	Ruhe

Arbeitsfeld

Arbeitsplan: 30h  
48h  
56h  
davon 72h : 20

Wochenstunden: 20

Schichtpool

Substanz

Bewerten

Optimieren

Ende

Arbeitsplan

Arbeitsplan: 30h  
48h  
56h  
davon 72h : 20

Wochenstunden: 20

Schichtpool

Substanz

Bewerten

Optimieren

Ende

Abbildung 8: Dargestellt sind im Gitter des Wochenplans die Startzeiten der Schichten. Sie wurden über die oberste Listbox im Ansichtsfeld selektiert.

### Ansichten

Um sinnvoll in den Planungsprozess eingreifen zu können, benötigt der Anwender viele Informationen. Es wurde versucht, alle dem System bekannten Informationen auch dem Planer über die Oberfläche zugänglich zu machen. Aufgrund ihrer Menge musste eine Aufteilung auf verschiedene selektierbare Ansichten vorgenommen werden. Die Möglichkeiten von PEPSI zur Darstellung sind zwar recht vielfältig, ihre Form und Handhabung ist allerdings noch nicht völlig konsistent und in mancher Weise verbesserungsfähig. In Kapitel 7.1 werden Ausbau-, Verbesserungs- und Erweiterungsmöglichkeiten ausführlich beschrieben werden. Alle für die Planung relevanten *Attribute der Schichten* können anstelle der Schichtbezeichnung in den Wochenplan eingeblendet werden. Dies geschieht über die Auswahl des entsprechenden Attributs in den Listenboxen des „Ansichtsfeldes“. Ein Beispiel ist die Ansicht der Schichtstartzeiten (siehe Abbildung 8) oder der angerechneten Arbeitszeit.

Zum Anzeigen von *Constraintverletzungen* bietet PEPSI drei Möglichkeiten.

- wahlweise als Gesamtaufstellung der aufgetretenen Verletzungen in einer Liste, oder
- es bestimmt die Verletzungen zu einzelnen Schichten nach deren Selektion, oder
- es stellt zu den Restriktionen eine Übersicht der Verletzungsverursacher im Wochenplan dar.

Gleichzeitig mit dem Öffnen des Einsatzplanfensters erscheint auch ein Fenster mit einer Übersicht der Constraintverletzungen. Sie enthält im Gegensatz zu den beiden anderen Anzeigemöglichkeiten alle Verletzungen, die durch den gegenwärtigen Plan erzeugt werden. Idealerweise wäre dieses Fenster leer, in der Regel werden jedoch zumindest einige Verletzungen weicher Restriktionen

Konkade Optimierung vom 24.3.1999 16:53 Uhr

Stärke	Schwere	Begründung	Objekt
153.33	38.33	C_Jurzebelangeange	Schicht102 planlag Woche02_Mitwoch
153.33	38.33	C_Jurzebelangeange	Schicht170 planlag Woche02_Dienstag
128.0	1.0	C_rufeflag_leist	Schicht104 planlag Woche01_Samstag
128.0	1.0	C_rufeflag_leist	Schicht85 planlag Woche03_Samstag
128.0	1.0	C_rufeflag_leist	Schicht87 planlag Woche01_Sonntag
97.33	24.33	C_Jurzebelangeange	Schicht85 planlag Woche03_Samstag
97.33	24.33	C_Jurzebelangeange	Schicht86 planlag Woche04_Montag

Abbildung 9: Fenster mit Darstellung aller verletzten Constraints. Die Tabelle ist nach der Schwere der Strafe sortiert, so daß der Disponent die wichtigsten Verletzungen als erste bemerkt.

angezeigt. Abbildung 9 zeigt ein solches Fenster. Bei Neuordnungen von Schichten durch den Benutzer wird dieses Fenster nicht aktualisiert. In diesem Falle oder nach vorherigem Schließen kann man es durch Klicken auf die „Bewerten“-Schaltfläche erneut öffnen oder aktualisieren.

Mit einem Mausklick der linken Taste werden Schichten aus dem Gitter des Wochenplans in den Schichtpool übernommen (siehe Abbildung 10, hier befinden sich bereits zwei Schichten im Schichtpool). Dort kann man sie selektieren und sich anschließend mit dem „Verletzung“-Button die diese Schicht betreffenden Constraintverletzungen anzeigen lassen. Es erscheint ein Fenster, das ähnlich dem aus Abbildung 11 ist.

Die dritte Möglichkeit der Anzeige von Verletzungen ist eine Anzeige abhängig von einzelnen Restriktionen. Selektiert man in der Constraint-Listbox ein Constraint, wird der „Ort“ von Restriktionsverletzungen dieses Typs im Wochenplan angezeigt. Felder, die verletzungserzeugende Schichten enthalten, werden dann mit einem „x“ markiert.

Die meisten *Kennzahlen*, die als Eingangsparameter für Constraints dienen, können je nach Typ, im Wochenplan oder in einem Informationskasten dargestellt werden. Diese Ansichten werden ebenfalls über die Listboxen im „Ansichtsfeld“ aufgerufen. Abbildung 10 zeigt als Beispiel die Zeit zur nächsten Ruhe.

In folgender Liste werden die Kategorien der Informationen, die der Disponent erfragen kann, noch einmal zur Übersicht zusammengestellt:

- Attribute der Schichten
- Constraintverletzungen
- berechnete Kennzahlen

### Benutzereingriff

Nach der ersten Zuordnung kann man mit der Schaltfläche „Optimieren“ Optimierungsläufe anstoßen. COKE versucht hier, durch weitere Vertauschungen

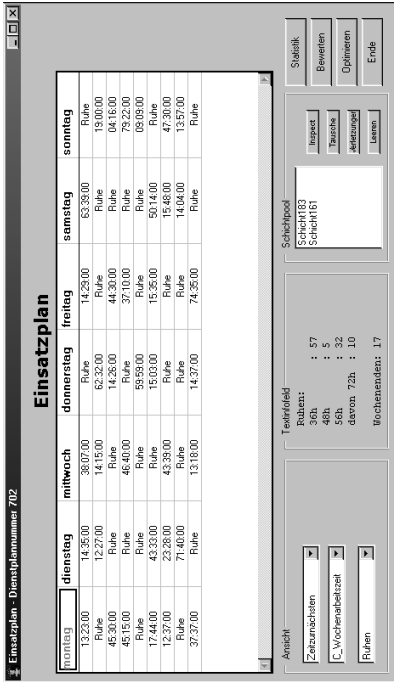


Abbildung 10: Kennzahlenanzeige im Wochenplan. Angezeigt ist die Zeit zur nächsten Ruhe, Äquivalent dargestellt werden z.B. die Wochenarbeitszeit ab einem Wochentag oder die Zeit zur nächsten Schicht.

das Ergebnis noch zu verbessern. Nach einem oder zwei Optimierungsläufen verändert sich der Dienstplan meist nicht mehr. In diesem Falle ist ein (lokal) optimaler Dienstplan gefunden.

Im Menüpunkt „Wissensverb/Restriktionen/Aktivierung der Restriktionen“ kann der Benutzer vor einer weiteren Optimierung das Gewicht der Constraints verändern, um seine Vorstellungen in den weiteren automatischen Planungsverlauf einzubringen. Den Dialog dazu zeigt Abbildung 5.

Dem Benutzer steht es auch frei, selbstständig beliebige Vertauschungen vorzunehmen. Ein Drücken der linken Maustaste selektiert eine Schicht, ein weiterer Druck auf die rechte Maustaste gibt das Ziel an. Werden durch diese Neuordnung Restriktionen verletzt, wird der Planer darauf hingewiesen. Das System erklärt ihm, welche Verletzungen mit welchem Schweregrad er gerade ausgelöst hat. Er kann seine Entscheidung daraufhin noch einmal überdenken und die Aktion nötigenfalls abbrechen (siehe Abbildung 11).

Mit einer Option des Kontextmenüs der Schichten kann man einzelne Schichten vor weiteren Optimierungsläufen festsetzen (siehe Abbildung 12). Sie werden dann bei Umplanungen des Systems nicht mehr weggetauscht. Das System erreicht damit eine stärkere Benutzerkontrolle, denn der Disponent kann Hintergrundwissen einbringen, auch wenn es der programmierten Wissensbasis widerspricht.

Eine letzte Eingriffsmöglichkeit ist die Möglichkeit zur manuellen Veränderung des Ruheschemas. Mit der Ansicht „Ruheschema“ aus Ansichtsauswahl kann man sich einen Überblick über die gerade gesetzten Ruheflugs verschaffen (Abbildung 12). Über Kontextmenüs kann man dann die Ruheflugs der Plantage setzen oder zurückziehen. In der weiteren Zuordnung oder Optimierung werden diese Veränderungen der Rahmenbedingungen berücksichtigt.

In folgender Liste werden die Möglichkeiten des lenkenden Eingriffs durch den

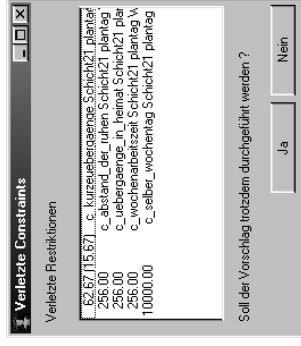


Abbildung 11: Anzeige von Verletzungen nach einem Benutzereingriff. Gezeigt werden die Verletzungen zu einer Schicht bzw. zu einer Zuordnung

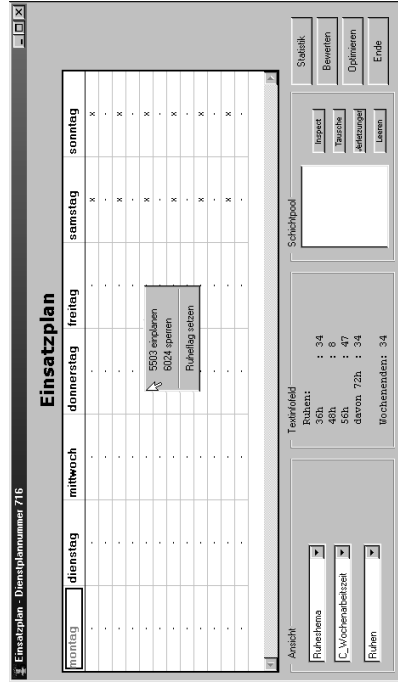


Abbildung 12: Dargestellt ist ein Ruheschema und die Möglichkeit, dieses zu editieren. In einem Kontextmenü werden Optionen geboten, wie Ruheflugs zu setzen oder einzelne Schichten zu sperren (d.h. ihren Wegtausch zu verhindern). Die Tabellenfelder, die ein „x“ enthalten, sind Plantage, die zur Ruhe vorgesehen sind.

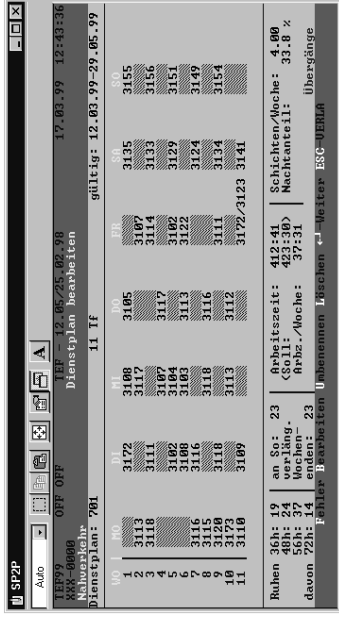


Abbildung 13: Das Dienstplanprogramm TEF. Dieses Programm ermöglicht die Eingabe von Dienstplänen. Es berechnet zu berücksichtigende Kennzahlen und gibt Warnungen bei Regelverletzungen.

Planer noch einmal zusammengefasst. Mit ihnen und der Bereitstellung aller nötigen Informationen hat er die Grundlage für eine kooperative Dienstplanerstellung.

- Erneuter Anstoß zum Vertauschen (Optimieren)
- Ungewichtung der Constraints
- Ändern von Zuordnungen (Vertauschen)
- Festsetzen von Schichten
- Modifizieren des Ruheschemas

## 6 Analyse

Dieses Kapitel fikt die erreichten Ziele und gewonnen Erfahrungen zusammen. Es zeigt die Eignung der „Zuordnen und Vertauschen“-Methode und spezieller die des entworfenen Prototypen für die gegebene Problemstellung. Der Abschnitt 6.1 handelt zunächst von der Evaluierung und den Ergebnissen, die mit Testdaten erzielt wurden. Abschnitt 6.2 beschreibt und bewertet verschiedene Aspekte der Verwendung von PEPSI für die Dienstplanerstellung.

### 6.1 Evaluierung

Die Daten für das Testscenario stammen aus dem derzeit bei der Deutschen Bundesbahn eingesetztem Dienstplanstellungsprogramm TEF (siehe Abbildung 13). Es lagen sowohl Schichtmengen als auch die daraus von Disponenten erstellten Dienstpläne zum Vergleich vor. Bei allen getesteten Schichtmengen gelang es mit PEPSI, in kurzer Zeit Einsatzpläne zu erstellen, die allen harten Constraints gerecht wurden. Die benötigte Wochenzahl glich dabei immer

der, die auch im dazugehörigen TEF-Plan verwendet wurde. In einigen Fällen genügte bereits ein Zuordnungslauf oder ein Zuordnungslauf mit nachfolgender globaler Optimierung. Andere benötigten Hilfestellung durch den Benutzer in Form von Neugewichtung der Constraints oder manuellen Schichtumsetzungen. Im Beispiel von Dienstplan Nr. 701 sind in den Abbildungen 7 und 13 der vorgegebene und der neu erstellte Dienstplan gegenübergestellt.

Zusammen mit einem Mitarbeiter der Deutschen Bundesbahn wurden einige Beispielfälle durchgespielt. Trotz anfänglicher Skepsis zeigte er Gefallen an den Möglichkeiten, die sich mit diesem Ansatz eröffnen. Die Strategien wurden hierbei noch einmal bestätigt und der Disponent hatte zu den mit den Beispieldaten erzeugten Einsatzplänen keine Einwände, ausgenommen der fehlenden Berücksichtigung der Doppelschichten.

Zu bemerken ist, daß die Aussagekraft der Evaluierung hier naturgemäß noch sehr gering ist, da das System im derzeitigen Status noch nicht in einem realen Umfeld getestet wurde und nur die Meinung eines Bereichsexperten zur Brauchbarkeit erfragt wurde. Desweiteren ist ein direkter Vergleich des Zeitaufwands zur Erstellung von Dienstplänen problematisch, da bei der manuellen Planung die Arbeitsschritte „Zusammenstellung der Schichten“ und „Dienstplanerstellung“ ineinandergreifen, die für den Zuordnungsvorgang voneinander getrennt wurden.

## 6.2 Eigenschaften von PEPSI

### Aspekte manueller Umplanung

Durch manuelles Umsetzen einzelner Schichten kann man dem „Vorschlagen und Vertauschen“-Algorithmus aus lokalen Minima<sup>6</sup> heraushelfen. Ist die Änderung zu gering (zu wenige Vertauschungen), wird sie eventuell beim nächsten Optimierungslauf wieder zurückgenommen und man landet im selben lokalen Optimum. Anders beim Festsetzen der benutzergesteuerten Schicht, wie es in Kapitel 5.3 beschrieben wird. Eine solche Fixierung ist stärker als jede Konsistenzverletzung und kann somit in manchen Fällen zu unnötig hohen Strafen durch Regelverletzungen führen. Das typische Szenario für *blokes Umsetzen* ist „ich habe eine Idee, die das System nicht hatte“ und für *Umsetzen mit Festsetzen* „ich habe Wissen von Fakten, die das System nicht hat“. Beide Tauschvarianten laden jedoch zum spielerischen Verbessern durch Ausprobieren ein und können dabei gleichermaßen zum Erfolg führen.

### Aspekt der Modifizierbarkeit

Ein Kriterium zur Beurteilung eines Programms für die Personaleinsatzplanung ist auch die Wartbarkeit. Dies ist vielfach ein Mangel in bisherigen Systemen, bei denen es sich um reine Eingabehilfen handelt, die Eingaben des Disponenten auf Übereinstimmung mit den Regeln der Dienstvorschriften hin überprüfen. Solche Systeme sind schnell veraltet und zwingen den Benutzer zu „Tricks“, um es dennoch benutzen zu können. Ein Vorteil der COKE/PEPSI-Lösung ist, daß Änderungen in den Vorschriften oder eine Berücksichtigung neuer Regeln keinen großen Pflegeaufwand erfordern. Der versierte Benutzer hat die Möglichkeit zum einfachen Hinzufügen oder Ändern von Restriktionen. Die programmier-technisch ausdrucksvolleren Teile der Restriktionen sind in abstrakten Hilfs-

<sup>6</sup>Das Bewertungskriterium ist, die Strafe durch Constraintverletzungen möglichst gering zu halten, dehalb „Minimum“

funktionen (z.B. einer Funktion zur Berechnung der Zeit von einer Schicht zur nächsten) verborgen. Die Constraints selbst bestehen meist aus wenigen Zeilen, die auf diese Funktionen zurückgreifen (z.B. ein einfacher Vergleich einer Zeit mit einer Maximalzeit). Auf diese Weise sind die Constraints sehr aussagekräftig und können auch von einem Anwender mit wenig Programmiererfahrung verändert werden.

### Konfliktziele

Das übergeordnete schwer definitorisch fassbare Ziel einen *guten* Dienstplan zu erstellen, hat viele Teilziele, die teilweise auch miteinander in Konflikt stehen. Ein guter Dienstplan berücksichtigt Interessen bezüglich der wirtschaftlichen Effektivität als auch der sozialen Verträglichkeit. Regeln müssen eingehalten werden, doch auch innerhalb dieser Regeln gibt es gewisse Freiräume.

Ein Beispiel für einen solchen Konflikt ist die Länge der Ruhetage und Übergänge. Hält man die Übergänge zwischen den Schichten möglichst kurz, bleibt mehr Zeit für lange Ruhetage. Gömt man dem Zugpersonal allerdings längere Übergänge zwischen den Dienstschtichten, so verbleibt weniger Spielraum für die Ruhetage.

Die Abwägung zwischen diesen Zielen kann der Benutzer vornehmen und über verschiedene Constraintgewichtungen Einfluß auf die Planung durch das System nehmen.

### Nachträgliche Änderungen

Ein Kriterium für die Praxistauglichkeit der rechnergestützten Planung ist auch, wie sich kleinere Änderungen in den Vorgaben auf schon bestehende Dienstpläne auswirken. Eine nachträgliche Verschiebung der Startzeit einer Schicht, das Hinzufügen einer weiteren oder das Tauschen einer Schicht mit einer neuen ändern die Bewertung und, falls sie Verletzungen erzeugen, werden bei einem anschließenden Optimierungslauf die vorhandenen Strategien zur Aufhebung der Konflikte eingesetzt.

Die Constraints sind alle derart, daß ein Entfernen einer Schicht keine wirklich verbindliche Randbedingung verletzt. Nach dem Löschen der Schicht aus dem Dienstplan kann es aber ein neues lokales Minimum im Lösungssuchraum geben. Der Tausch einer Schicht läßt sich im wesentlichen als ein Entfernen und das Hinzufügen einer neuen Schicht beschreiben. Es genügt also diesen Fall zu betrachten. Der „Vorschlägen und Vertauscher“-Algorithmus in seiner Zuordnungsphase fügt auch Schritt für Schritt eine weitere Schicht hinzu. Ein nächst mögliches Einfügen einer Schicht unterscheidet sich nur dann, wenn es durch eine Vorsortierung der Nachfrager (erwähnt in Kapitel 4.3) zuerst eingeteilt worden wäre. In diesem Falle lohnt sich die Neuplanung.

Änderungen in den Daten zu einer Schicht können verschiedenste Auswirkungen auf den bisherigen Dienstplan haben. Manche verändern die Bewertung der Constraints nicht, andere dagegen können eine Folge von Korrekturen nach sich ziehen.

## 7 Ausblicke und Anregungen

Dieses Kapitel stellt Erweiterungsmöglichkeiten für PEPSI vor. Diese Erweiterungen lassen sich in zwei Typen einteilen. *Erweiterungen der Funktionalität*,

die in Abschnitt 7.1 aufgelistet werden, und *Erweiterungen der Problemstellung*, die Thema der Abschnitte 7.2 und 7.3 sind.

### 7.1 Erweiterungen der Funktionalität

Zunächst geht es um einige Erweiterungen, die die Erstellung eines Dienstplanes im Rahmen der hier behandelten Problemstellung erleichtern bzw. verbessern können. „erleichtern“ im Bezug auf die Bedienung und „verbessern“ im Bezug darauf, ein möglichst gutes Ergebnis zu erhalten.

- Verbalisierung der Verletzungen  
Bisher werden Beschreibungen von Constraintverletzungen nach einem einheitlichen Schema angegeben. Neuere COKE-Versionen als die, die bei der Programmierung dieses Prototypen verwendet wurde, unterstützen individuelle Verbalisierungsfunktionen zu den Constraints. Eine Verletzung wird dann z.B. nicht mehr als „C\_Schichtueberlappung, Schicht43, Woche01\_Mittwoch“ sondern verständlicher als „Schicht43 überschneidet sich mit Schicht55 am Mittwoch der Woche!“ formuliert.
- Rubricschemateditor  
Rubricschemas können von Anwender über die PEPSI-Benutzeroberfläche verändert werden, die Auswahl vorgegebener Rubricschemata über die Benutzeroberfläche ist bisher nicht möglich. Ideal wäre eine Möglichkeit zur Speicherung bearbeiteter Rubricschemata in der Wissensbasis.
- Speicherung von Dienstplanzwischenständen  
Die Zwischenspeicherung der Bearbeitungsstände funktioniert bisher nur als COKE-Wissensbasis. Es gibt zwar sowohl eine Import- als auch eine Exportfunktion für das BORD-Format, beim Import wird jedoch der alte Planungsstand nicht wieder hergestellt.
- Vorschlagsgenerator - Step-By-Step-Planung  
Benutzereingriffe sind bisher nur zwischen den einzelnen Optimierungphasen möglich. Vollständigere Benutzerkontrolle könnte man z.B. dadurch erreichen, daß man jede Zuordnung mit seinen dazugehörigen Korrekturen vor seiner Ausführung visualisiert. Diesen „Vorschlag“ kann der Planer dann annehmen oder alternativ seine eigene Idee einbringen.
- Viele Erweiterungsmöglichkeiten bieten sich in der graphischen Benutzeroberfläche an. Diesen wird jedoch ein eigenes Kapitel (Kapitel 8) gewidmet.

### 7.2 Auswahl der Schichten eines Dienstplanes

Die Zusammenstellung der Schichten für einen Dienstplan wird noch immer von Hand gemacht. Die Schichten werden nach Erfahrungswerten zusammengefaßt und durch mehrfaches Probieren werden sie zu geeigneten Ausgangsmengen für Schichtpläne. Später kann der Disponent auch noch Schichten zwischen den einzelnen Dienstplänen einer Einsatzstelle tauschen, um die Bewertung der Pläne zu verbessern. Der tatsächliche Vorgang ist also noch um einiges komplexer

## 8 Anforderungen an eine graphische Repräsentation

Einer der Vorteile der „Vorschlagen und Vertauschen“-Methode ist die Möglichkeit, den Benutzer und sein Wissen in den Zuordnungsprozess einzubinden. Diese Interaktion stellt jedoch auch hohe Ansprüche an die Benutzeroberfläche. Da im Rahmen dieser Studiearbeit nicht alle Anforderungen umgesetzt werden konnten, wird ihnen an dieser Stelle ein eigenes Kapitel gewidmet.

### Schneller Wechsel zwischen den Ansichten

Der Planer benötigt für eigene Entscheidungen bei interaktiven Eingriffen detaillierte Informationen, sowohl über die Objekte, mit denen er umgeht, als auch über die Auswirkungen seiner Handlungen. Ein Darstellen aller Informationen (Kennzahlen, Constraintverletzungen, Zeichnungen,...) innerhalb des Schichtplanes ist aufgrund ihrer Fülle nicht möglich. Ein schneller Wechsel zwischen den verschiedenen Informationsansichten muß deshalb ermöglicht werden. Sowohl die Darstellung als auch die Navigation sollte konsistent, verständlich und leicht erlernbar sein.

### Kontextmenüs

Die bei interaktiven Planung angesprochenen Objekte - Schichten und Plantage - haben alle ihre graphische Repräsentation im Schichtplan. Ein schneller Zugriff z. B. Selektieren oder Verschieben ist deshalb möglich. Ebenso ist es wichtig, den Zugriff auf alle objektbezogenen Funktionen effektiv zu gestalten. Eine Aufnahme dieser Funktionen in ein Kontextmenü zu jedem Objekttyp würde diese Forderung erfüllen. Die Kontextmenüs sollten folgende Funktionen umfassen:

#### Kontextmenü zu „Schicht“

- Schicht festsetzen (kein Umplanen durch das System)
- Schicht zurückziehen
- Vorschlag für Tausch geben
- Schichtparameter darstellen
- Verletzte Constraints anzeigen

#### Kontextmenü zu „Plantage“

- Ruheflag setzen

### Weg von Slot-, hin zu Fließdarstellung

Informationen über die Lage der Schichten, über Startzeit und Länge, oder über Überlappungen können qualitativ in die graphische Darstellung der Schichten im Schichtplan einfließen. Der Slot eines Plantages bekommt eine zeitliche Skalierung in x-Richtung, die Schichten wären dann verschiedenen lange Blöcke, die abhängig von der Startzeit in diesen Plantagen aufzeichnet werden. Festgesetzte Schichten können graphisch gekennzeichnet werden. Der Dienstplandirektor des BORD-Projekts, der derzeit noch in der Entwicklung ist, soll eine solche Darstellung bieten. Abbildung 15 zeigt an einem Bildausschnitt der Benutzeroberfläche, wie dies geplant ist.

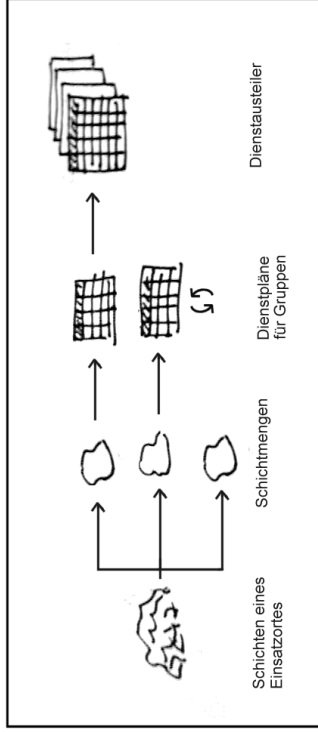


Abbildung 14: Diagramm zum Arbeitsablauf. Alle Schichten eines Einsatzortes werden in Schichtmengen für Gruppen zerlegt. Diese dienen als Basis für die Dienstplanung. Zur Optimierung kann der Disponent auch während des Planungsvorganges Schichten unter den Dienstplänen tauschen. In einem weiteren Arbeitsschritt werden aus den Dienstplänen für die Gruppen Diensttausteiler für die einzelnen Mitarbeiter gemacht.

als der bis jetzt von PEPSI unterstützte. Ob und wie sich dies als Zuordnungsproblem formulieren und implementieren läßt, ist eine mögliche Frage zur Weiterführung dieser Arbeit.

Die Ausgangsmenge wäre in diesem Falle ein großer Schichtpool mit allen Schichten einer Einsatzstelle, das Ziel eine vorher unbestimmte Menge an Dienstplänen. Man muß in diesem Falle eine dienstplanübergreifende Planung ermöglichen.

Möglicher Vorteil wäre dabei nicht nur die Arbeitserleichterung beim Gruppieren der Schichten, sondern die Aussicht durch Ausprobieren mehrerer Aufteilungs-möglichkeiten bessere Dienstpläne als Ergebnis zu erhalten.

### 7.3 Diensttausteiler

Da der Dienstplan keine Feiertagsschichten berücksichtigt und nur als Grundlage für die kurzfristige Diensterteilung dient, stellt sich die Frage, ob ähnliche Methoden auch zur Generierung der Diensttausteiler eingesetzt werden könnten. Dazu müßte man die Schichtmenge erweitern und ausgehend von einem Dienstplan mehrere Diensttausteiler (sowie wie Mitglieder in einer Gruppe) für einen vorher festgelegten Zeitraum erstellen. Diese Diensttausteiler würden soweit möglich nach dem Muster des Dienstplanes vorbelegt und dann der Zuordnungsprozess mit den abweichenden Schichten gestartet. Vertauschungen finden nicht innerhalb einer Spalte des Anstellers statt, sondern zwischen den Diensttausteilern.

Ausschnitt aus Dienstplanelditor						
Woche	Montag	Dienstag	Mittwoch	Donnerstag	Freitag	Sonntag
1	4711 14:28 - 21:43 06:15	4713 ..... 14:28 - 21:43 06:15	RUHE [3828]	4711 14:28 - 21:43 06:15	4711 14:28 - 21:43 06:15	RUHE [7528]
2		4712 22:30 - 04:10 05:40	4712 22:30 - 04:10 05:40	4711 14:28 - 21:43 06:15	4713 14:28 - 21:43 06:15	4712 22:30 - 04:10 05:40

Abbildung 15: Ein Ausschnitt aus der geplanten Oberfläche des BORD-Dienstplanelditors. Dargestellt ist ein Wochenplan mit zugewiesenen Schichten. Die Schichten werden durch schwarze Balken symbolisiert, deren Länge variabel ist und von der zeitlichen Länge der Schicht abhängt. Die Lage der Schichten verdeutlicht ihre Start- und Endzeiten.

### Drag&Drop

Ein komfortables Drag&Drop zum manuellen Umsetzen von Schichten ist eine weitere Anforderung an die Benutzerschnittstelle. Der Einfachheit wegen wurde bisher nur ein Verschieben über einen abwechselnden Links-Rechts-Mausklick implementiert.

### Verletzungsanzeige

Neben den bisherigen Möglichkeiten zur Verletzungsanzeige sollte schon in der Standardansicht mit deutlichen farblichen Signalen auf die Verursacher von Verletzungen aufmerksam gemacht werden. Verletzungen, die keinen speziellen Verursacher haben (z.B. Verletzung des Constraints  $C_{\text{Verlängerte\_Wochenenden}}$ ), sollten getrennt davon angezeigt werden, z. B. mit verschiedenen Warnleuchten an einer festen Stelle der Benutzeroberfläche. Bei Verletzungen, die eindeutig einer Schicht oder einer Gruppe von Schichten zugeordnet werden können, liegt es dagegen nahe, die Verletzungsmarkierung an den Objekten im Wochenplan anzubringen.

Die Möglichkeit, Verletzungen mit schwachen Strafen auszublenden ist hilfreich, um den Disponenten nicht vom wesentlichen abzulenken.

Beim Drag&Drop könnten nach dem Aufheben einer Schicht, Planträge, die als gewählter Anbieter eine Verletzung erzeugen würden, schraffiert dargestellt werden.

### Graphisch gestützte interaktive Planung

Eine Ausbaumöglichkeit, um den Benutzer zu unterstützen, ist der schon in Kapitel 7.1 vorgestellte Vorschlagsgenerator. Das System schlägt dabei zu einer Schicht, oder auch allgemein zur Menge aller Schichten, die beste Tauschsituation vor, die mit seinen Strategien errechnen kann. Eine solche Funktionalität ließe sich mit wenig Aufwand in COKE erstellen. Eine graphische Darstellung der Vorschläge und Korrekturen kann man mit verschiedenfarbigen Pfeilen im Wochenplan realisieren. Abbildung 16 veranschaulicht meine Vorstellungen dazu.

ig	dienstag	mittwoch	donne
3014	8004	8003	80
8004	6014	8004	80
8023	8023	8001	
8005	8003	.	80
8001	8002	8002	80
8003	8001	8004	80
8002	8005	8005	60
.	.	.	80

Abbildung 16: Mögliche Darstellung eines Vorschlags. Der dunkle Pfeil stellt den Vorschlagsstart vor (ausgehend von der Schicht, zu der ein Vorschlag vom Benutzer angefordert wurde, oder ausgehend von der Schicht, die die schwerste Constraintverletzung erzeugt). Der helle Pfeil zeigt alle Tauschs an, die als Konsequenz daraus nötig werden.

## 9 Fazit

Im Gegensatz zu reinen Constraintüberwachungssystemen, die den Disponenten nur auf Verletzungen hinweisen, kann PEPSI selbst - basierend auf heuristischen Wissen - Zuordnungsschritte vornehmen. Bei der Erstellung von Dienstplänen aus den betrachteten Falldaten zeigte sich der Prototyp schon als brauchbares Werkzeug, um selbstständig einen guten Ausgangsplan für eine weitere operative Planung zu erstellen. Durch die Möglichkeiten des Benutzerergriffs und weiterer Optimierungen durch das System wird es zu einer Lösung, die die Arbeit der Disponenten wesentlich vereinfachen kann. Der Pflegeaufwand, der bei Änderungen in Vorschriften und einer gewünschten Berücksichtigung weiterer Regeln entsteht, ist sehr gering. Desweiteren können Konfliktziele oder Präferenzen verschiedener Dienststellen sehr einfach über Constraintgewichtungen berücksichtigt und nachträgliche Dienstplanänderungen an schon geplanten Dienstplänen vorgenommen werden.

Der Einsatz eines solchen Systems kann in Zukunft nicht nur den Planungsvorgang vereinfachen, sondern auch aufgrund der Möglichkeiten, mehr Regeln zu berücksichtigen als dies in einem nicht unterstützenden System der Fall wäre, die Qualität der Planung erhöhen. Es liegt am Benutzer, zwischen den sozialen und wirtschaftlichen Kriterien abzuwägen und diese zu berücksichtigen.

## Literatur

- [1] Frank Puppe, „Problemlösungsmethoden in Expertensystemen“: *Springer Verlag* (1990)
- [2] Karsten Poedk, „Konfigurierbare Problemlösungsmethoden am Beispiel der Problemlklassen Zuordnung und Diagnostik“: *Dissertation, Universität Würzburg* (1995)
- [3] Stefan Klingenbeck, Frank Puppe, „Personaleinsatzplanung von Zugpersonal als mehrstufiges Zuordnungsproblem“: *interner Bericht, Universität Würzburg*
- [4] Ciske Busch, Frank Puppe, „Interaktive Dienstplanung des BORD-Personals der Deutschen Bahn AG“: *interner Bericht, Universität Würzburg*
- [5] Ciske Busch, „Spezifikation eines kooperativen Werkzeugs zur Lösung komplexer Zuordnungsprobleme“: *Dissertation, Universität Würzburg*
- [6] Stefan Klingenbeck, „Projektbeschreibung BORD“: *interner Bericht, sd&in*
- [7] Wolfgang Franzki, Thilo Hermann, „Projekt BORD - Fensterbeschreibung - DP Dienstplaneditor“: *interner Bericht, sd&in*
- [8] Christian Hestermann, „COKE und WIZARD - Technische Referenz“: *interner Bericht, Universität Würzburg*
- [9] Johannes Beck, „Oskar-Verleihung - Erste Schritte in COKE“ (1998) *interner Bericht, Universität Würzburg*