

Stefan Meyer

Planausführung und Konfliktverwaltung

Am Beispiel einer digitalen Modelleisenbahn

Studienarbeit

BEI GRIN MACHT SICH IHR WISSEN BEZAHLT



- Wir veröffentlichen Ihre Hausarbeit, Bachelor- und Masterarbeit
- Ihr eigenes eBook und Buch - weltweit in allen wichtigen Shops
- Verdienen Sie an jedem Verkauf

Jetzt bei www.GRIN.com hochladen
und kostenlos publizieren



Friedrich-Alexander-Universität Erlangen-Nürnberg

Institut für Informatik 8

Lehrstuhl für Künstliche Intelligenz

Planausführung und Konfliktverwaltung

Studienarbeit im Fach Informatik

vorgelegt von

Stefan Meyer

Betreuer:

Prof. Dr. Günther Görz

Dipl. Inf. Bernd Ludwig

Beginn der Arbeit: 11.11.03

Ende der Arbeit: 31.03.03

Ich versichere, daß ich die Arbeit ohne fremde Hilfe und ohne Benutzung anderer als der angegebenen Quellen angefertigt habe und daß die Arbeit in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegen hat und von dieser als Teil einer Prüfungsleistung angenommen wurde. Alle Ausführungen, die wörtlich oder sinngemäß übernommen wurden, sind als solche gekennzeichnet.

Erlangen/Nürnberg, den 31.3.2004 _____

Inhaltsverzeichnis

Inhaltsverzeichnis.....	i
Abbildungsverzeichnis	iii
1 Einleitung	1
2 Der momentane Stand des Eisenbahnprojekts	3
2.1 Überblick über das vollständige Eisenbahnsystem	3
2.2 Das Eisenbahnkontrollsystem und seine Agenten	5
2.2.1 Multiagentensysteme und der KQML-Standard	5
2.2.2 Die verschiedenen Agenten der Eisenbahn	7
2.3 Kommunikation zwischen den verschiedenen Agenten	10
2.3.1 Die Initialisierung der Agenten	10
2.3.2 Das Schalten einer Weiche	11
2.3.3 Die Bewegung des Zuges	11
2.3.4 Das Ändern der Zuggeschwindigkeit	12
2.3.5 Die Reservierung eines Weges	13
3 Planen und daraus entstehende Konflikte	15
3.1 Grundlagen des Planens	15
3.2 Planungsprobleme	16
3.3 Verschiedene Planungsverfahren	17
3.3.1 STRIPS an Hand eines Beispiels	17
3.3.2 Vergleich von STRIPS mit anderen gängigen Verfahren	19
3.4 Konflikte	20
3.4.1 Klassifikation von Konflikten	20
3.4.2 Grundlagen des Konfliktlösend	23
4 Implementierung der Konfliktlösung	27
4.1 Zusammentragen der benötigten Informationen	27
4.1.1 Statische Informationen an "theBrain"	27
4.1.2 Dynamische Informationen an "theBrain"	28
4.2 Abspeichern der Informationen und Konsistenzproblem	31
4.2.1 Situationsberechnung nach jeder Nachricht mit Racer	32
4.2.2 Situationsberechnung zum Zeitpunkt des Konfliktes mit Racer	34
4.2.3 Situationsberechnung nach jeder Nachricht ohne Racer	36
4.3 Die Konfliktlösung beim Auftritt eines Konfliktes	38
4.3.1 Die einfache Konfliktlösung	38
4.3.2 Die komplexe Konfliktlösung mittels Lparse und Smodels	39
4.3.2.1 Lparse, Smodels und die Logikprogrammierung	39
4.3.2.2 Das Erstellen der nötigen Dateien	41
4.3.2.3 Die von Lparse benötigten Constraints	43
4.3.2.4 Die von Lparse und Smodels berechneten Lösungen	46

4.4	Das Auswählen und die Durchführung der besten Lösung	47
4.4.1	Die Bewertung der Lösungen	47
4.4.2	Das Umsetzen einer Lösung.....	49
5	Beispiel einer Konfliktlösung	51
5.1	Das Starten der benötigten Komponenten	51
5.2	Beispiel eines Konfliktes zwischen zwei Zügen	52
6	Zusammenfassung und Ausblick	57
	Literatur.....	59

Abbildungsverzeichnis

Abb. 2.1	Die Architektur des Eisenbahnsystems	4
Abb. 2.2	Die Architektur des Eisenbahnkontrollsystems	7
Abb. 2.3	Screenshot der Steuertafel der Züge.....	8
Abb. 2.4	Screenshot der Visualisierung	9
Abb. 3.1	Start- und gewünschte Zielsituation der Klötzchen [DaWe].....	18
Abb. 3.2	Alle mögliche Zustände im Klötzchen-Beispiel [DaWe].....	19
Abb. 3.3	Hierarchie der Konflikte und zugehörige Löseverfahren [HaBe],[MKle]	23
Abb. 4.1	Screenshot der bei “theBrain” ankommenden Nachrichten	30
Abb. 4.2	Der Teil der T-Box, der das Wissen über Züge und Weichen definiert.....	32
Abb. 4.3	Der Algorithmus für den Konsistenztest nach jeder ankommenden Nachricht ..	33
Abb. 4.4	Die Performance des Konsistenztest-Algorithmus der Abb. 4.3.....	34
Abb. 4.5	Der Algorithmus für den Konsistenztest über eine History-Liste	35
Abb. 4.6	Die Berechnungsdauer beim Auftreten eines Konfliktes mit der Historyliste	36
Abb. 4.7	Der Ablauf ohne Benutzung des Racer-Tools.....	37
Abb. 4.8	Die Funktionsweise von Lparse und Smodels (aus [ToSy])	41
Abb. 4.9	Der Aufruf von Lparse und Smodels von der Kommandozeile [ToSy].....	42
Abb. 4.10	Beispiel der Dateien, die Lparse übergeben werden.	43
Abb. 4.11	Die Datei “constraints.lp”	44
Abb. 4.12	Die berechnete Ausgabe von Smodels	46
Abb. 5.1	Screenshot des Embassi-Servers	51
Abb. 5.2	Screenshot der Visualisierung vor dem Konflikt (mit Beschriftung).....	52
Abb. 5.3	Screenshot der aktuellen Situation und der berechneten Lösungen	53
Abb. 5.4	Screenshot der Bewertung der Lösungen	54
Abb. 5.5	Screenshot der Visualisierung nach der Auflösung des Konfliktes	54