

Lösen von Scheduling-Konflikten durch Verhandlungen zwischen Agenten

Karl-Heinz Krempels
krempels@i4.informatik.rwth-aachen.de

RWTH Aachen
Lehrstuhl für Informatik IV
Ahornstraße 55
D-52056 Aachen

Zusammenfassung Die Terminplanung im medizinischen Akutbereich ist gekennzeichnet durch eine hohe Komplexität sowohl der Arbeitsabläufe als auch der beteiligten Personalstrukturen. Notfälle und unvorhersehbare Behandlungsabläufe verlangen zusätzlich eine hohe Flexibilität der Planung. Diese Kombination flexibler Planungsprozesse mit komplexen Organisationsstrukturen führt zum Scheitern mathematischer Optimierungsansätze. Die vorliegende Arbeit beschreibt einen Ansatz zur Lösung von Scheduling-Konflikten und stellt einen Teil eines Terminplanungssystems im Krankenhausbereich dar, das auf intelligenten, präferenzbasierten Agenten (Policy-Agenten) als Vertreter reeller Personen basiert. Im Mittelpunkt des verfolgten Ansatzes steht die Integration der Präferenzstrukturen aller beteiligten Personen und der gegebenen hierarchischen Strukturen der Organisationseinheiten in den Schedulingprozeß sowie die Lösung von Scheduling-Konflikten mittels Verhandlungen zwischen Agenten.

1 Einleitung

Die Terminplanung im Akutkrankenhaus ist ein wesentliches Element des Klinikmanagements. Aufgrund der Ungewissheit hinsichtlich des Kapazitätsangebotes und der tatsächlichen Nachfrage ist die Terminplanung im Akutbereich durch ein hohes Maß an planerischer Flexibilität gekennzeichnet, welche von der Terminplanungssoftware gehandhabt werden muss [3]. Aus Akzeptanzgründen sollte eine weitgehend automatisierte Terminplanung auch die Interessen und Präferenzen der individuellen Akteure berücksichtigen. Dies bedingt, dass ein Terminplanungssystem diesen Faktoren in ähnlicher Weise Rechnung tragen muss, wie es bei der derzeitigen Praxis manueller Planung möglich ist und begünstigt den Einsatz von Agentensystemen.

2 Beschreibung des Lösungsansatzes

Die Integration der Präferenzstrukturen aller beteiligten Personen und der gegebenen hierarchischen Strukturen der Organisationseinheiten eines Krankenhauses in den Schedulingprozess bedingt deren Modellierung und formale Spezifikation. Die Idee, einen Scheduling-Konflikt durch eine Verhandlung zwischen Agenten zu lösen, begünstigt die Einführung einer Verhandlungsmasse, aufgrund welcher eine Verhandlung oder ein Tausch ermöglicht wird. Dies muss bei der Modellierung der Präferenz- und Hierarchiestrukturen berücksichtigt werden. Die Modellierung von hierarchischen Strukturen kann einfach vorgenommen werden, indem die Akteure mit einer Verhandlungsmasse ausgestattet werden, die ihrer Rolle in der Hierarchiestruktur entspricht. So erhält ein Chefarzt zum Beispiel mehr Verhandlungsmasse als ein Stationsarzt. Als aufwendiger erweist sich die Modellierung der Präferenzen und die Verteilung der Verhandlungsmasse auf diese. In den folgenden Abschnitten wird hierauf näher eingegangen.

Das Scheduling von Aktionen und Ressourcen wird in zwei Phasen vorgenommen:

- in der ersten Phase erstellt der Scheduler einen vorläufigen Plan ohne die Präferenzen der Akteure zu berücksichtigen. Hierfür können bekannte Ansätze [9] verwendet werden.
- In der zweiten Phase versucht der Scheduler den vorläufigen Plan durch die Berücksichtigung der Präferenzen der Akteure zu verbessern.

Die von dem Scheduler erkannten konfigurierenden Präferenzen bezüglich einer Ressource werden an das Agentensystem zwecks Lösung durch Verhandlung weitergegeben, wobei hier zu Gunsten genau eines Agenten entschieden wird.

2.1 Die Präferenzen

Eine Präferenz [12] ist die Angabe einer Reihenfolge für zwei beliebige Alternativen a, b in der Form, dass $a \prec b, a \succ b \Leftrightarrow b \prec a$ oder $a \sim b$. So kann ein Akteur in einem Krankenhausszenario zum

Beispiel angeben, dass er lieber mit Schwester A in OP 1 als mit Schwester B in OP 2 operiert. Ein Akteur muss seine Präferenzen der Einfachheit halber in eine totale Ordnung bringen, die die Wichtigkeit der Präferenzen in aufsteigender Reihenfolge darstellt: $p_1 \prec p_2 \prec \dots \prec p_k$.

2.2 Die Gewichtsfunktion

Um die gesamte Gewichtsmasse M so auf die k Präferenzen eines Agenten zu verteilen, dass die gegebene Präferenzordnung seines Prinzipals erhalten bleibt, ergeben sich die folgenden Anforderungen an die Gewichtsfunktion W , wobei die erste Präferenz p_i als die unwichtigste angenommen wird und dementsprechend mit dem kleinsten Gewicht belegt wird:

- das Gewicht der Präferenz p_i soll grösser sein als die Gewichte der einzelnen Präferenzen p_j für alle $i > j$ und $1 \leq i, j \leq k$:

$$w_i > w_j, \forall i > j \wedge 1 \leq i, j \leq k. \quad (1)$$

Dies bedeutet, dass die Gewichte der ersten beiden Präferenzen vorgegeben werden müssen

$$w_1 = m, w_2 = qm, \quad (2)$$

wobei entweder das Gewicht der ersten Präferenz m oder das Verhältnis zwischen der zweiten und der ersten Präferenz q angegeben werden kann. Hier werden die Wertebereiche für m und q der Einfachheit halber eingeschränkt: $m \geq 1$ und $q > 1$.

- das Gewicht der Präferenz p_i an der Position i der gegebenen Präferenzordnung muß gleich der Summe der Gewichte der Präferenzen p_j mit $0 < j < i$ und $i > 2$ sein (also derjenigen Präferenzen die weniger wichtig sind als die Präferenz p_i). Im Folgenden werden die Gewichte der Präferenzen $W(p_i)$ der Einfachheit halber mit w_i bezeichnet:

$$w_i = \sum_{j=1}^{i-1} w_j. \quad (3)$$

- die gesamte Gewichtsmasse M soll auf die Präferenzen verteilt werden:

$$M = \sum_{i=1}^k w_i. \quad (4)$$

Die Gewichte der restlichen Präferenzen w_k lassen sich unter Berücksichtigung der in den Gleichungen 1 bis 4 spezifizierten Anforderungen bestimmen: $w_k = 2^{k-3}m(q+1)$, wobei die maximale Anzahl der gewichtbaren Präferenzen auf $k_{max} = \lfloor 1 + \log_2 M \rfloor$ beschränkt ist.

2.3 Die Verhandlung

Die an der Verhandlung teilnehmenden Agenten werden von einem ServiceAgent mit den Präferenzen ihrer Prinzipale und deren Gewichtsmasse initialisiert und verteilen anschließend die Gewichtsmasse entsprechend der vorgestellten Gewichtsfunktion auf ihre Präferenzen. Der ServiceAgent fragt alle Agenten nach dem Gewicht der konfigurierenden Präferenz und wählt den Agenten mit der größten Gewichtung als Gewinner der Verhandlung aus. Dieser Agent verteilt die Gewichtsmasse der konfigurierenden Präferenz auf die restlichen beteiligten Agenten nach einer vorgegebenen Strategie, wobei diese auf ihre Präferenz verzichten. Die Verhandlung wird mit der Übergabe der aktuellen Gewichtsmasse und der Präferenzliste eines jeden beteiligten Agenten an den ServiceAgenten beendet.

2.4 Die Nutzenfunktion

Ein Agent muss bestimmen können, wann er während einer Verhandlung besser oder schlechter gestellt ist. Hierfür wird als Maß die gesamte jeweils aktuelle Gewichtsmasse des Agenten verwendet. Die Nutzenfunktion $N(t)$ ist damit gleich der Summe über alle Gewichte der Präferenzen w_i zum Zeitpunkt t : $N(t) = \sum_{i=1}^k w_i$. Eine Besserstellung des Agenten liegt vor, wenn $N(t) < N(t+1)$ ist.

3 Umsetzung des Lösungsansatzes

Auf die Anwendung des beschriebenen Lösungsansatzes auf das geplante System wird anhand einer

Beschreibung des in Abbildung 1 dargestellten Ablaufes eingegangen. Die relevanten Eigenschaften der existierenden Ressourcen sowie der aktuellen Anforderungen werden in einer Wissensbasis abgelegt (1) und von einem Expertensystem für das Scheduling (2) verwendet. Wenn von dem Scheduler ein Konflikt gefunden wird, dann werden die davon betroffenen Akteure (personelle Ressourcen) ermittelt und für jeden als Vertreter ein Agent auf einem Agentensystem gestartet (3). Anschließend initialisiert der Scheduler die Agenten mit den Informationen ihrer Prinzipale und der

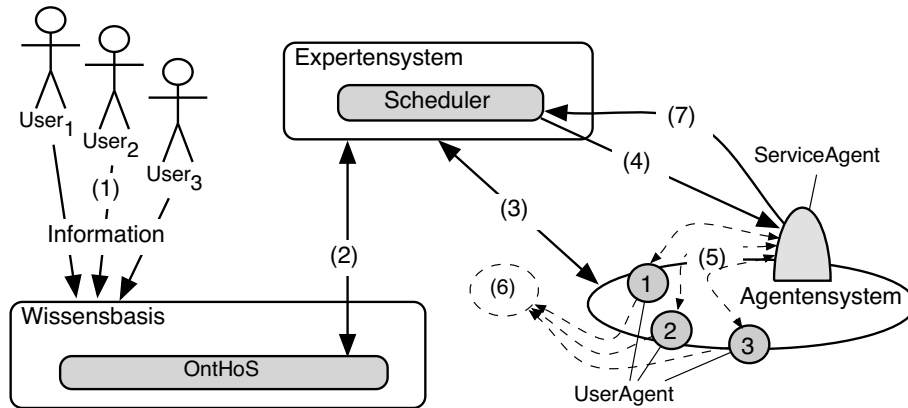


Abbildung 1: Ablauf einer Konfliktlösung durch Verhandlungen von Agenten

konfigurierenden Präferenz über einen Dienst des Agentensystems (4,5) und startet eine Verhandlung zwischen den Agenten (6). Das Ergebnis dieser Verhandlung, das genau ein gewinnender Agent ist, wird dem Scheduler über den Dienst des Agentensystems mitgeteilt (7), so dass dieser nun eine konfliktfreie Zuweisung der Aktion zu einer Ressource vornehmen kann. Im Folgenden werden die verwendeten Systemkomponenten näher beschrieben, wobei insbesondere auf ihre Funktionalität und Ihren Aufbau eingegangen wird.

Die Wissensbasis Um eine Lösung des Problems losgelöst von dem realen Umfeld eines Krankenhauses zu ermöglichen, wurde das Domänenwissen mit Hilfe der Domänenontologie OntHoS¹ modelliert und in einer Wissensbasis abgelegt. Hierdrauf können verschiedene Ansätze getestet und ihre Ergebnisse miteinander verglichen werden. Für jeden Ansatz kann eine eigene Task-Ontologie verwendet werden, die allerdings einen Teil der Domänenontologie beinhalten muss, um das benötigte Wissen aus der Wissensbasis zu verwenden. Innerhalb der Domänenontologie OntHoS sind die Anforderungen an eine Aktion spezifiziert. Als Beispiel soll hier die Blinddarmoperation betrachtet werden. Die Klasse *appendectomy* ist eine Spezialisierung der Klasse *operation* und erbt von dieser sämtliche Slots². Die Slots werden weiter spezialisiert, so dass die Instanzierung der Klasse *appendectomy* für den Slot *required resources* konkrete Instanzen der Klassen *operating room*, *nurse*, *nurse*, *microscope* zugewiesen werden müssen. Durch die Spezialisierung der Klasse *operation* konnte zusätzliches Wissen in die Domänenontologie OntHoS eingebracht werden, welches bei der Instanzierung eines Terms der spezialisierten Klasse *appendectomy* als *hard constraint* berücksichtigt werden muss. Die Pflege der Ontologie wird mit dem Werkzeug Protégé [11] vorgenommen, wobei verschiedene Erweiterungen (BeanGenerator³ [2], JessTab⁴ [6]) hierfür zum Einsatz kommen.

Das Expertensystem Die Zuweisung der konkreten Instanzen unter Berücksichtigung der an die Slots gehefteten Facets erfolgt durch einen Scheduler, der in JESS⁵ [5] implementiert wird. Die JESS Komponente spielt eine zentrale Rolle in dem System, da hier die Zuweisungen der Aktionen zu den Ressourcen vorgenommen werden und eventuelle Konflikte erkannt werden, die dann anschließend zur Lösung an das Agentensystem weitergereicht werden. JESS verfügt über eine Schnittstelle zu Protégé, über die die gesamte Ontologie und die existierenden Instanzen der ontologischen Terme importiert werden können.

¹Die Domänenontologie OntHoS wurde innerhalb des DFG Schwerpunktprogramms 1083 "Intelligente Softwareagenten und betriebswirtschaftliche Anwendungsszenarien entwickelt [1, 10]"

²Hier werden nur die Klassentypen als Facets der Slots betrachtet.

³Das BeanGenerator Plugin erweitert Protégé um die Fähigkeit, eine Ontologie in Java Quellcode zu exportieren.

⁴Das JessTab Plugin stellt die Schnittstelle zwischen Protégé und dem Expertensystem JESS bereit

⁵Java Expert System Shell.

Das Agentensystem Als Agentensystem wird das FIPA⁶-konforme System JADE⁷ [4, 7] verwendet, das über Schnittstellen zu JESS und Protégé verfügt. Über die Schnittstelle zu Protégé können die von den Agenten benötigten Task-Ontologien importiert werden, während die Schnittstelle zu JESS für die gesamte Steuerung des Agentensystems genutzt werden kann. Hierdurch können Agenten und Agentendienste auf einer Agentenplattform gestartet und gesteuert werden, wobei die Kommunikation zwischen JESS und einem Agenten immer über einen Agentendienst vorgenommen werden muß. Das von den Agenten beigetragene Wissen zur Lösung einer Aufgabe wird an die Wissensbasis übertragen.

4 Bewertung und Ausblick

Messungen des Datenaufkommens und der Dauer einer Verhandlung an dem Prototypen des umgesetzten Lösungsansatzes haben ergeben, dass dieser im Krankenhausbereich zur Lösung von Schedulingkonflikten eingesetzt werden kann. Der Schwerpunkt der Weiterentwicklung des Prototypen wird auf der Umsetzung von mehreren Verhandlungsstrategien für die Agenten sowie auf der Implementation des vollständigen Schedulers liegen, so daß dieser in einem realen Umfeld eingesetzt werden kann.

Literatur

- [1] Becker, M., Heine, C., Herrler, R., Krempels, K.-H.: OntHoS - an Ontology for Hospital Scenarios. 2002
- [2] BeanGenerator Home Page, <http://www.swi.psy.uva.nl/usr/aart/beangenerator/>
- [3] Fitzpatrick, K.E., Baker, J.E., Dave, D.S.: An application of computer simulation to improve scheduling of hospital operating room facilities in the United States. In: International Journal of Computer Applications in Technology, Vol. 6, No. 4, p. 215-224, 1993.
- [4] Jade Home Page, <http://jade.csel.it>
- [5] Java Expert System Shell, <http://herzberg.ca.sandia.gov/jess/>
- [6] JessTab: Integrating Protégé and Jess, <http://www.ida.liu.se/her/JessTab/>
- [7] Foundation for Intelligent Physical Agents, <http://www.fipa.org>
- [8] Fritsch, M., Wein, T., Ewers, H.-J.: Marktversagen und Wirtschaftspolitik - Mikroökonomische Grundlagen staatlichen Handelns, München 1999. S.30-32.
- [9] Nareyek, A.: Constraint-Based Agents. Lecture Notes in Artificial Intelligence, Vol. 2062. Springer-Verlag, Berlin Heidelberg New York (2001)
- [10] OntHoS Home Page, <http://l6-spike-dos.informatik.uni-wuerzburg.de/Ontologie>
- [11] Protégé Home Page, <http://protege.stanford.edu>
- [12] Schneeweiß, C.: Planung 1, Systemanalytische und entscheidungstheoretische Grundlagen, Springer-Verlag, Berlin Heidelberg New York (1991)

⁶Foundation for Intelligent Physical Agents.

⁷Java Agent Development Environment.