Agent Based Scheduling of Operation Theaters

Marc Becker[1], Karl-Heinz Krempels[2], Marti Navarro[3], Andriy Panchenko[2]

[1] University of Trier, Dept. of Business Information Systems 1
54286 Trier, Germany
mb@wiinfo.uni-trier.de

[2] Aachen University of Technology, Computer Science Dept.,
Communication and Distributed Systems,
52056 Aachen, Germany
{krempels,panchenko}@i4.informatik.rwth-aachen.de

[3] Politechnic University of Valencia,
Dept. of Information Systems and Computation,
46022 Valencia, Spain
mnavarro@dsic.upv.es

Abstract. [4]
Medical operation planning is a substantial element of hospital management. It is characterized by high complexity, which is caused by the uncertainty between the offered capacity and the true demand. As emergency cases occur the planning requirements will change. Therefore, the use of a dialog-based system is preferred against full manual or automated systems, because of the inability of the latter to recognize the changes in a heigh dynamic environment. To make it possible to add new tasks in the planning process "on the fly" and to adequate the planning to new situations we involve a human planner in the scheduling process. The planner acts as a "sensor" for occurring changes and integrates his knowledge in the planning process.
In order for a created schedule to be accepted by involved personnel resources, it should include the interests and preferences of all the human actors. Existing planning systems do not take this into account and suffer therefore of in-acceptance of their resulting schedules. This leads to manual planning in hospital environments and binds a very specialized person to this task. It is required that the person should know the set of available resources, both material and human, requirements to medical operations and interdependencies among them. Nowadays, the medical director realizes this task. He is one of the most expensive resources in the area and his specialty is not the planning, but the surgery.
The presented work overcomes the drawbacks of existing solutions and includes the interests and preferences of each actor representing him/her through an intelligent, preference based agent. It also provides the mechanisms to solve the conflicts among them through a negotiation among agents.

[4] This work is supported by the German Research Foundation (DFG) as a part of the priority research program "Multi-agent systems and their business applications" (SPP 1083) and the European Agentcities.NET Project (IST-2000-28384).

I. Rudomín, J. Vázquez-Salceda, J. Díaz de León (Eds.). e-Health: Application of Computing Science in medicine and Health

1 Introduction

Scheduling in hospitals is done in two phases, in the large and the small. On the long run, patients get an approximate date of operation, which in some cases - if it is not vitally necessary - might be several months ahead. In Germany, as in other European countries, the refunding system of hospitals by public health insurances is highly regulated and limited on a yearly basis, yielding to a backlog of wished operations. This backlog of several months essentially is caused by the limiting effects of available yearly budget, which yields to a steady number of operations per week independent of actual need.

A few days before the fixed date for the operation, patients enter the hospital to be prepared. Public health insurances are very eager to shorten the time of patient stay at hospitals before operation and will not refund any overtime. Thus, after admission of a patient, there is a definite need to do the planned surgery operation as soon as possible.

In the short run, scheduling of patients will happen from one day to the next. In the following, we will concentrate on this short run scheduling of operation theaters. Short term scheduling is well known being a process with an outcome that is highly dependent on situational variables:

- The duration of an operation in some cases cannot be determined beforehand. Thus, there is uncertainty of needed time.
- Also the specific tasks to be performed during an operation may depend on situations not observable during diagnosis (planning phase).
- The daily schedule often will be interrupted by incoming emergency cases. The frequency of emergencies depends on the medical department. For example, schedules of orthopedic departments seem to be very stable, whereas schedules of surgical of neurosurgical departments in general have to take incoming emergencies into account.

If a patient is scheduled for operation today, it may happen that by incoming emergencies, he suddenly will find himself in a position to be rescheduled for tomorrow. In the analyzed hospital this rescheduling results from a contract between nurses and hospital management that limits regular operating time from 8.00 o'clock in the morning to 4.00 o'clock in the afternoon at the latest. Management had been forced to sign this contract, since dissatisfaction of nurses because of working overtime on a regular basis had been overwhelming. On the other hand, any rescheduling of an operation of one day to the next is a major source of dissatisfaction of patients, physicians, and management. It essentially contributes to the increase of the backlog and causes possible shortcuts of refunding, since public insurances will not pay overtime in front of surgery operations.

2 Description of the Approach

The scheduling problem presented here leads to different requirements for a multi-agent system. First, human interaction should be reduced to a minimum, thus reducing the needed time to resolve incompatibilities by phone. Second, the planning procedure should be done simultaneously, thus allowing for more flexible solutions, which for example utilize the available operating rooms more intensively. Third, a multi-agent system should take care of the individual interests of the involved personnel whenever possible. This will give strong evidence for the acceptance of the planning system and will allow a better degree of satisfaction of staff.

As representative for each individual we use software agents that contain the preferences of their principal. The scheduling of actions and resources is made in two stages:

- In the first stage the scheduler creates a preliminary plan without respect to preferences. Therefore we can use known scheduling approaches and algorithms. The scheduler interacts with the planner via dialogs and offers him subplans for modification, reordering, or to place them into a Gantt-chart. Subplans consist of a set of actions, selected with respect to the constraints of the concepts of the used ontology OntHoS[5] [4] from the resource database.
- In the second stage the scheduler improves the preliminary plan with respect to the preferences of each individual. When a conflict between the specified preferences of two or more individuals is found, the representative agents of these individuals should negotiate together with the goal to obtain a commitment. This means that only one agent obtains the conflicting resource that is specified in the preferences of all these agents while the others obtain adequate wages from the winning agent.

For the implementation of this approach some mathematical assumptions and restrictions regarding the preference ordering and importance have to be made. One restriction is that we suppose a linear ordering of all preferences specified by one individual. The used approach will work as well for partial orderings but it will become more sophisticated because of the additional transformation steps from one ordering to the other one. An other assumption that we make is, that the attributes of a preference can be randomly chosen from the set of all possible attributes.

Usually a negotiation supposes at least two parties, which have conflicting goals. These goals will be negotiated among the different agents by trading a bargaining-mass. Under the assumption that also different status levels in hospital hierarchy have to be taken into consideration this bargaining-mass will be weighted according to this status levels, e.g. a doctor gets more bargaining-mass than an operating room nurse.

2.1 Preferences

To describe motives and behaviors of persons there should be made a distinction between preferences and desires. Desires form a simpler and more basic notion than preferences. A desire for something involves only one object and refers

[5] OntHoS - An Ontology for Hospital Scenarios.

to a pro-attitude toward this object, whereas a preference for one object over one or more other objects involves at least two objects and indicates that the decision making person assigns a higher priority to his pro-attitude toward the first object than towards the others. Preferences therefore not only indicate the priority a person assigns towards his various desires, but indicate also the relative importance, which the decision maker assigns to his objects of desire. Because of this, the utility functions needed for the decision making component of the software agents can only be defined in terms of preferences, rather than desires. The set of all preferences $P := O^m$, $m \in N$ over the set of all preferred objects $O := \{o_1, \cdots, o_n\}$, $n \in N$ makes together with the relation $\prec \subseteq P \times P$, which represents the order of principal's preferences, its preference structure. The relation P is *irreflexive* (for no $a \in P$ is $a \prec a$ valid), *transitive* (if from $a \prec b$ and $b \prec c$, follows also $a \prec c$) and *comparative* (for all $a, b \in P$ is $a \prec b$ or $a \equiv b$ or $b \prec a$).

2.2 The Weight Function

The bargaining-mass is distributed over the specified preferences with respect to their importance. We assume that the importance of the preference at range i is higher than or equal to the sum of importances of all preferences with a range lower than i in the specified ordering.

The distribution of a bargaining-mass M over an ordering of preferences is made in the following way: the most important preference has the highest order in the specified ordering (i.e. if we have to deal with k preferences the highest order is k and the lowest is 1). This means that the preference with the highest order should obtain the highest weight of bargaining-mass.

This assumption leads us to the following requirements for the distribution function W of the available bargaining-mass (assuming p_1 is the least important preference) :

- The weight $w_i = W(p_i)$ of the preference p_i in the given order must be equal to the sum of the weights of the preferences p_j $(3 < j < i)$, which are less important than p_i.
- The weight w_i of the preference p_i should be greater than the weight of the single preferences p_j for all $i > j$ and $1 \leq i, j \leq k$. That means that the weight of the first two preferences, w_1 and w_2, has to be specified: $w_1 = m$, $w_2 = qm$. The range of values of m and q to simplify matters is restricted: $m \geq 1$ and $q \geq 1$.

The weight w_i of the other preferences can be calculated [2] as expected with a negative exponential function :

$$w_i = \frac{M}{2^{(k+1)-i}}, \text{for } 3 \leq i \leq k. \qquad (1)$$

Due to the range values restriction of m and q is it possible to define only a limited number of preferences with a positive weight. The maximum number is

$k_{max} = \lfloor 1 + \log_2 M \rfloor$. In case of a greater number of preferences k than k_{max} of an actor, only the k_{max} most important preferences p_i (with $k - k_{max} \leq i \leq k$) are defined with a weight $w_i > 0$ and the other preferences p_i with the weight $w_i = 0$.

2.3 The Utility Function

During the negotiation an agent must be capable to distinguish whether the proposed changes will improve or impair the current state. The utility function $N(t)$ is the sum of weights of all preferences w_i at the time instance t:

$$N(t) = \sum_{i=1}^{k} w_i. \tag{2}$$

An agent will improve its state in a negotiation if $N(t) < N(t+1)$ will be true.

2.4 Preference Conflict Detection

In a common context the preferences specified by the actors can contain conflicts, because of the different and possible opposite goals that they should represent. Preference conflicts have to be solved and therefore, at first all the preferences involved in a conflict have to be detected. For this problem there exist many solutions based on graph-theory or resolution. The used approach is based on graph theory and works as follows: all the preferences $p_{j,i}$ (preference i of actor j) are inserted in a graph G as vertices and their ordering dependencies as directed edges $< p_{j,i}, p_{j,i+1} >$. If the insertion of a new edge forms a cycle with the existing edges of the graph, then the preferences represented by vertices on this cycle contain a conflict.

2.5 Preference Conflict Solving

The representative agents of the actors with conflicting preferences negotiate together with the goal to achieve an agreement. Therefore, the agents are initialized with the preferences and the weight volume of their actors. The weight volume is then distributed by the agents among their preferences. Afterwards, the scheduler asks each of the agents involved in the conflict for the weight of the conflicting preference. It selects the agent with the highest weight as the winner for the conflicting preference. The winner distributes then the weight of his preference to the other concerned agents with the help of a selected strategy, as those abandon one preference. The negotiation process ends with the returning of new weight volumes to the scheduler.

3 Implementation of the Approach

The described approach is implemented as a prototype in the framework for multi agent systems *Agent.Hospital* [5, 7]. The analyzed application scenario was modeled with Protégé[6] with the help of the domain ontology OntHoS (see Figure 1, step 1). The ontology (T-Box) together with its instances (A-Box) were exported into the expert system JESS[7] as facts and rules (step 2). Further the A-Box, T-Box and all the problem solving methods as scheduling heuristics, conflict detection, negotiation strategies are implemented in JESS (step 3), and exported into the used JessAgents[8] (step 4). Further, all the agents are started and the schedulling process initiated by the PAScheduler agent. The user interface for interaction with the planner as well as the subplans generated based on ontological constraints are provided by this agent.

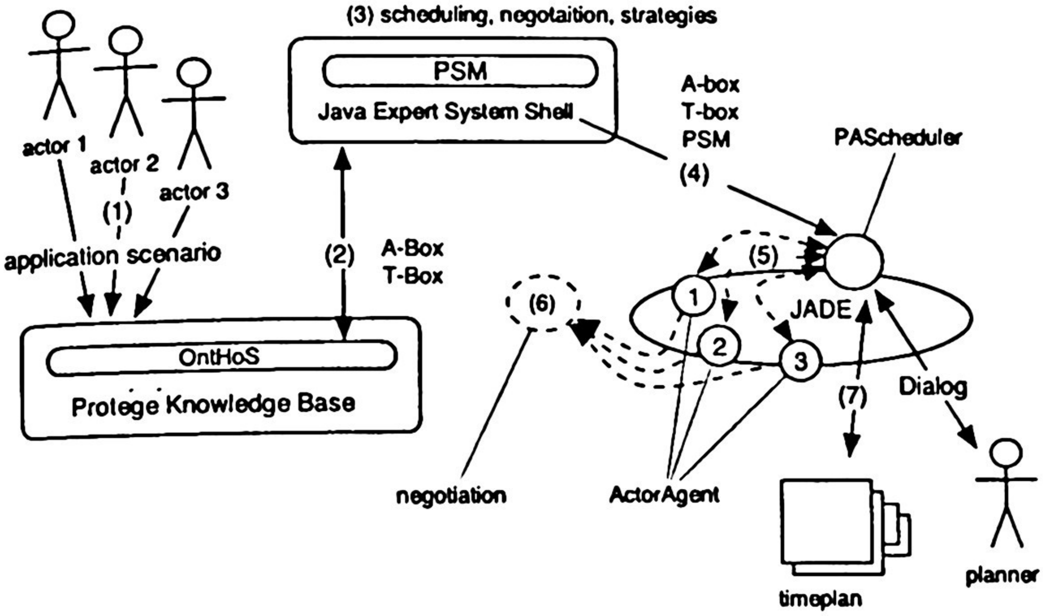

Fig. 1. Development Process Workflow

The agents are all initialized with the preferences of their principals and the weight volume with the FIPA[9] Request Interaction Protocol [9] (step 5). All the conflict solving negotiations are made with the FIPA English Auction Interaction Protocol [10] (step 6). Each solved conflict improves the preliminary plan (step 7) generated by the PAScheduler agent in interaction with the planner. This prototype was implemented with the help of the FIPA-compliant agent system JADE[10] [3]. The deployed agents are JessAgents (see Figure 2), developed to support a faster development process in agent technology. Problem solving methods and the behavior of a JessAgent can be written at a higher level

[6] Protégé Home Page. http://protege.stanford.edu/ .
[7] Java Expert System Shell. http://herzberg.ca.sandia.gov/jess/ .
[8] JessAgent Home Page. http://www-i4.informatik.rwth-aachen.de/agentcities/ .
[9] Foundation for Intelligent Physical Agents, http://www.fipa.org/ .
[10] Java Agent Development Environment, http://jade.cselt.it/ .

than Java programming without compilation of source code. Task ontolgies and problem solving methods can be loaded at runtime as well as a new fact base.

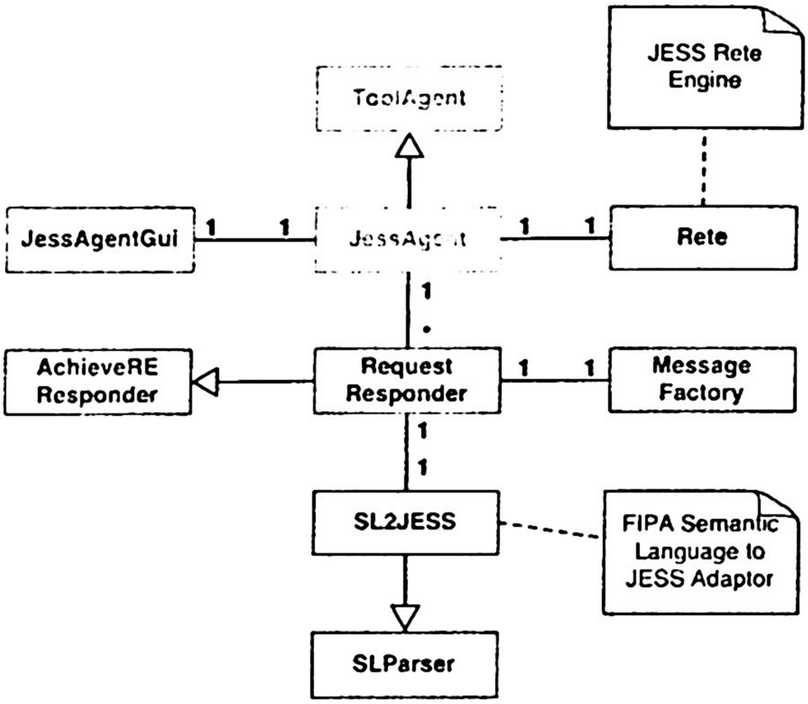

Fig. 2. Diagram of the JessAgent

Inside of a JessAgent all the received messages are translated by the SL2JESS adaptor to JESS functions, evaluated in JESS and the answer is automatically generated by the MessageFactory object with respect to the used interaction protocol, speechact, content language and the agent's fact base. All the rules, facts, and functions of an agent can be accessed by the developer through the agents GUI.

4 Outline and Conclusion

The presented work is work in progress. The selected scheduling heuristics as well as the implementation of the prototype give the possibility to test the presented approach in the lab. It remains to analyze the distribution of the weight volume among the agents during the continuous period of time, and to investigate different negotiation strategies. That is why the planning interface should be expanded in order to monitor planning actions taken by the scheduler. Time synchronization problems within the Agent.Hospital Framework should be solved with the help of an additional synchronization service [8] for agents. Further, the deployment of an ontology and a problem solving method repository [6] is planned and the test of the prototype inside of the agent.

References

1. 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.
2. Krempels, K.-H.: Lösen von Scheduling-Konflikten durch Verhandlungen zwischen Agenten. GI Workshop: Planen, Scheduling und Konfigurieren, Entwerfen 2002, Freiburg, Germany, October 2002.
3. Hennig, K. J.:Diplomarbeit - Verhandlungen für Agenten. Lehrstuhl für Informatik IV, Rheinisch-Westfälische Technische Hochschule Aachen, September 2002.
4. Becker, M.; Heine, C.; Herrler, R.; Krempels, K.-H.: OntHoS - an Ontology for Hospital Scenarios. First International Workshop on Agent Applications in Health Care, Barcelona, February 2003.
5. Kirn, S.; Heine, C.; Herrler, R.; Krempels, K.-H.: Agent.Hospital - agent-based open framework for clinical applications. IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises WETICE 2003, Linz, June 2003.
6. Krempels, K.-H.; Nimis, J.; Braubach, L.; Herrler, R.; Pokahr, A.: How words can tell what actions are doing. Challenges in Open Agent Systems '03 Workshop, Melbourne, Australia, July 2003.
7. Kirn, S.; Heine, C.; Herrler, R.; Krempels, K.-H.: Agent.Hospital - Agentenbasiertes offenes Framework für klinische Anwendungen. Wirtschaftsinformatik 2003, Dresden, September 2003.
8. Braubach, L.; Pokahr, A.; Krempels, K.-H.; Woelk, P.-O.; Lamersdorf, W.: Time Synchronization for Process Flows in Multi-Agent Systems. EBR 2003, Agent-oriented Technologies for E-Business, Rio de Janeiro, Brazil, November 2003.
9. Foundation of Intelligent and Physical Agents: FIPA Request Interaction Protocol Specification. http://www.fipa.org/specs/fipa00026/SC00026H.pdf. October, 2003.
10. Foundation of Intelligent and Physical Agents: FIPA English Auction Interaction Protocol Specification. http://www.fipa.org/specs/fipa00031/XC00031E.pdf . October, 2003.