

BDI Agents in a Patient Monitoring Scenario

Andrzej Borowczyk

Faculty of Mathematics and Information Science
Warsaw University of Technology
Email: andrzejborowczyk@o2.pl

Maciej Gawinecki, Marcin Paprzycki

Systems Research Institute
Polish Academy of Science
Email: {maciej.gawinecki,marcin.paprzycki}@ibspan.waw.pl

Abstract—One of interesting issues in modern health care is possibility of automating rudimentary tasks involving monitoring patients' state and providing them with an immediate help in emergency situations. This note discusses how application of BDI-based software agents can automate some tasks as well as improve service received by the patient. An implemented system skeleton which utilizes Jadex BDI agents is discussed.

I. INTRODUCTION

With the constant increase of capabilities of sensing devices there is a growing number of patients whose state should and can be permanently monitored. Here, there are at least two general scenarios to be considered. First, when the status of the monitored patient “changes slowly” and some “low-level” medication should be provided to respond to such changes. For instance, a slow but persistent increase of heartbeat may be remedied by prescribing a small dose of a medication (for the purpose of this paper we name it *valium*). The second scenario involves rapid change in patients condition, or lack of response to the already provided medication. For instance, lack of response to a stimulant (let us name it *adrenaline*), or a sudden decrease in heartbeat rate may require injection of a strong stimulant (here, named *super-adrenaline*) and very likely also notification of a nurse, a physician or an emergency response team (the latter is the case of a patient living at home). Note also, that the first scenario is in line with the modern practice of adjusting dosage of the medicine to the specific condition of the patient. This practice is further supported by the fact that constant monitoring of patient's status allows precise assessment of her condition (e.g. data mining techniques can be applied to distinguish between a heartbeat raising due to the waking up in the morning and heartbeat raising because of a heart condition). Furthermore, utilization of sensing devices combined with intelligent behaviors allows first, to move some patients from hospitals to outpatient care, e.g. back home with their families (which is highly desirable for their overall well-being), and second, to provide better care for all patients; especially for the elderly (whose health should be monitored without them being restricted in their activities). Let us also observe that life-threatening situations, require instant reaction (e.g. immediate personnel response). Thus, an agent monitoring patient's state indicators (e.g. heartbeat and/or blood pressure) can both inform the personnel and, in special circumstances, request action of an agent responsible for injecting life saving drug(s). The latter task would be performed by the personnel

after some time (needed to reach the patient) and the resulting time delay could prevent their actions from being successful.

Separately, there exist tasks that should not waste medical personnel's time and that do not actually require human intervention. The most trivial example is setting the proper temperature (adjusted to the patient's will and needs). It suffices to use an air conditioning agent that would know patient's desired temperature, monitor the current temperature in the room and adjust it accordingly. Obviously, such a task could be done by a patient pressing a button, but there exist a number of scenarios in which patients may be able to indicate their wishes only through specific means of communication. In this case the AC-controlling agent can remain unchanged, while the agent-patient interface would need to be adjusted (and this is one of the often mentioned advantages of utilization of software agents, see for instance [7]).

The aim of this note is to show how BDI agents can be utilized to develop a system that supports patient status monitoring and reacts to specific conditions. The reason for utilizing the BDI model ([1], [2]) is that it matches very well the needs of the above described scenarios. Furthermore, the BDI supporting Jadex framework [3] was recently introduced (note that current release of this software is version 0.96, so the first complete release – version 1.0 – is not completed yet). Thus our goal was to assess its actual usability in designing and implementing a system. This being the case our aim was *not* to build a realistic and complete representation of any actual medical condition(s), but rather to develop a backbone of the system with a few basic agents; provide them with a set of beliefs, desires and intentions and observe their behavior when actually implemented (using Jadex) and run.

II. SYSTEM OVERVIEW

As specified above, the current design of the system is very simplistic and consists of the following agents: *Air Conditioning Agent*, *Assistant Agent*, *Heartbeat Measure Agent*, *Injection Agent*, and *Emergency Agent*. In Figure 1 we depict their basic interactions.

Let us now describe in more detail content of the use case diagram. The *Air Conditioning Agent* (ACA), has beliefs about the current room temperature (originating from a sensor), desired (target) temperature, as well as two temperatures: high and low, defining a range within which the agent will not undertake any action. For instance, if the current room temperature is 24, while the target is 23, and the high and

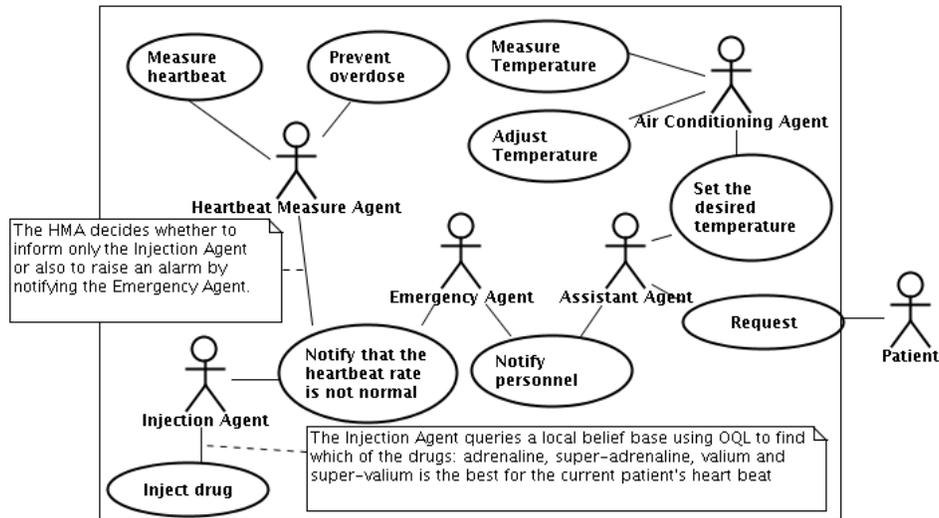


Fig. 1. Use Case diagram of the system skeleton

low values are 25 and 23, respectively, then the ACA will do nothing. However, was the temperature in the room 27, the agent would attempt at decreasing it to 23. Hence, the *main goal* of this agent is to keep the proper temperature (Jadex *maintain goal*). Separately, to be able to act within the system this, and other agents, have to be registered in the *Directory Facilitator (DF)* service (this is the only way for an agent to be known to other agents in the system and thus to be able to communicate with them). Therefore all agents have this as one of their Jadex *achieve goals*; and this goal has to be achieved once they are created. Furthermore, in the case when any of the agents in the system is associated with a GUI interface (for instance to display its current status and actions undertaken in the past), updates of the interface consist of local events of an agent. Similarly, events associated with each agent are messages obtained from various sources; e.g. information about the desired temperature received from the patient. To complete the description of the ACA, let us depict the way that the goal to maintain temperature is represented within the Jadex in the Agent Definition File (ADF); here `<` and `>` are named entities standing respectively for the `<` and the `>` characters in XML:

```
<maintaingoal name="keep_temperature"
  exclude="never" retry="true">
  <maintaincondition>
    $beliefbase.room_temperature
    &lt;= $beliefbase.temperature_high
    &amp;
    $beliefbase.room_temperature
    &gt;= $beliefbase.temperature_low
  </maintaincondition>
  <targetcondition>
    $beliefbase.room_temperature ==
    $beliefbase.target_temperature
  </targetcondition>
</maintaingoal>
```

In this statement we can see that the *keep_temperature* maintain goal will come into process only when the current

room_temperature defined in agent's belief base is higher or lower than defined constants, i.e. when the maintain condition is not satisfied. In such situation all possible plans responsible for realizing this goal will be retried (*retry* flag) until the *room_temperatures* reaches the defined *target_temperature*, i.e. the target condition becomes satisfied.

The *Assistant Agent (AA)* is responsible for assisting the patient (it could be viewed as an instance of a medically-skilled personal agent). Thus it possesses beliefs about the desired temperature and about the agent identifier of the correct *Air Conditioning Agent* (note that in a hospital there may exist separate ACAs managing different rooms, or room-clusters), so it can send to it messages specifying the currently desired temperature. Thus one of the goals of the AA is to find in the *DF* the identifier of the ACA. Obtaining an identifier of an agent is achieved by sending to the *DF* a request that specifies a certain registered service being searched for. The main event of the AA is associated with a message about the desired temperature sent to the ACA. Obviously, in a larger-scale system, additional needs of the patient would also be represented as beliefs and desires of the AA. The same comment applies to other agents, as their extended roles could be easily envisioned.

The *Emergency Agent (EA)* is responsible for alarming the personnel in a dangerous situation. Its plan of action is triggered by an incoming message from the *Heartbeat Measure Agent* about the heartbeat going out of the "safe range." Among its goals is to know at any time which emergency personnel and in what way can be contacted (e.g. at 2:34 AM nurse Smith should be contacted by sending an SMS to her cell phone).

The *Heartbeat Measure Agent HMA* has beliefs about the heartbeat, the high and low values that decide whether it is within normal, safe or dangerous range, and about identifiers

of the *Injection Agent* and the *EA*. If the heartbeat goes out of normal range, the *Injection Agent* is informed; if the heartbeat goes out of safe range both the *Injection Agent* and *Emergency Agent* are notified.

The *Injection Agent (IA)* has a set of beliefs related to a collection of drugs that are in its disposal (in our case we utilize four drugs named *valium*, *super-valium*, *adrenaline* and *super-adrenaline*) and can be applied in various circumstances. It also knows the interrelations between specific health conditions and drugs that are to be used. Its main task is to analyze the current patient’s condition and inject an appropriate dosage of a drug. When the *IA* receives a message about the patient’s heartbeat, it checks the belief database using the Jadex query language (similar to the Object Query Language) in order to select the appropriate medication. For instance, upon receiving information that the heartbeat is below the normal range, but still in safe range (*\$heart_beat* belief variable equals to '*too_low*') it would issue the following query (note that each agent has a direct access to its knowledge base):

```
select one $heart_beat_drug_pair.get(1)
from Tuple $heart_beat_drug_pair
  in $beliefbase.heart_beat_drugs
where $heart_beat_drug_pair.get(0).
  equals($heart_beat)
```

where *heart_beat_drugs* is a set of beliefs structured in the following way (as a dictionary of possible symptoms and appropriate medications):

```
<beliefs>
  <beliefset name="heart_beat_drugs"
    class="Tuple">
    <fact>new Tuple("too_high",
      "valium")</fact>
    <fact>new Tuple("much_too_high",
      "super_valium")</fact>
    <fact>new Tuple("too_low",
      "adrenaline")</fact>
    <fact>new Tuple("much_too_low",
      "super_adrenaline")</fact>
  </beliefset>
</beliefs>
```

More precisely, in the considered case the query can be understood as follows: select a single medication from a tuple (*heart_beat_drug_pair*) in the dictionary *heart_beat_drugs*, where a possible symptom pointed by the first element of this tuple matches the current value of *heart_beat*. Therefore the response to this query would result in administering *adrenaline*.

A. Application of goal deliberation

Let us note that the BDI approach to agent system design allows us to effectively utilize goal deliberation, i.e. handling situation with competitive goals. For this purpose Jadex utilizes the easy deliberation strategy [6] which we can illustrate in the case of actions necessary when dealing with the problem of potential medicine overdose. Here, we have two competing goals, one of them is to inject a given medicine – based on patient’s current state (heartbeat in our case). The second goal is to prevent overdosing the drug by keeping count of applied doses and to wait with the next one until

a certain time-limit has been reached (where the time-limit is drug-dependent). The overdose prevention goal, when in process (this is also a technical term used in Jadex) inhibits the injection goal. So the overdose prevention goal blocks the drug injection goal for a specific time after which the agent decides (based on the current status of the patient) what action to take at this stage, i.e. the drug injection goal decides which, and if any, drug to apply. It has to be stressed that in this situation the overdose protection is “more important” than drug injection. This is based on the fact that it takes some time for a drug to take effect and applying it several times one dose after another does more harm than good. Thus, we have to allow previous doses some time to take effect and in many cases the result will be that the state of the patient will improve without the need of injecting the blocked dose in the first place. It is easy to envision how this approach can be utilized in multiple scenarios similar to the one described here.

III. RUNNING THE SYSTEM

The system’s agents are autonomous and associated with certain medical devices or persons responsible for taking an appropriate action. They react accordingly to a given situation by deliberating goals, realizing appropriate plans and querying their belief databases. Let us see a simple scenario in action. In Figure 2 we present a screen-shot of the system that is running. We can see the Jadex console and Jadex Control Center GUI. Furthermore, GUI interfaces of all agents running in the system are visible.

The patient’s heart beat has been simulated as sinusoid-type function of time and based on the age of the patient (*in no way this simulation is claimed to be realistic*, it is used only to see the system running and to test it).

In the top right corner we can see the *Assistant Agent*, which receives data from the patient, and forwards her desires to the *Air Conditioning Agent*. In response to these desires and to the changing room temperature the *ACA* acts to adjust the room temperature accordingly.

Now, to see the “medical procedures” in action, let us use as an example the moment when the screenshot in Figure 2 was taken. Here, one can observe the scenario triggered when the patient’s heartbeat was 121. Such a heartbeat stands out of the “allowed safe range” and thus the *Heartbeat Measure Agent* creates *HeartBeatMeasurePlan*, which informs about it the *Injection Agent* and the *Emergency Agent* — this happens at 21:34:19. This can be seen on the GUI of the *HMA*. After a minimal delay, the *Emergency Agent* received this information and notified the personnel (see the GUI of the *EA*) within *EmergencyPlan*. Three seconds later, the *Injection Agent*, having received the information from the *HMA*, decided to inject *super-valium* (see the GUI of the *IA*).

IV. CONCLUDING REMARKS

The BDI model had been proposed by Bratman, as a philosophical approach to explaining human behavior [1]. Later its utility has been observed in the context of information technology (see for instance [4], [5]). Unfortunately, most of

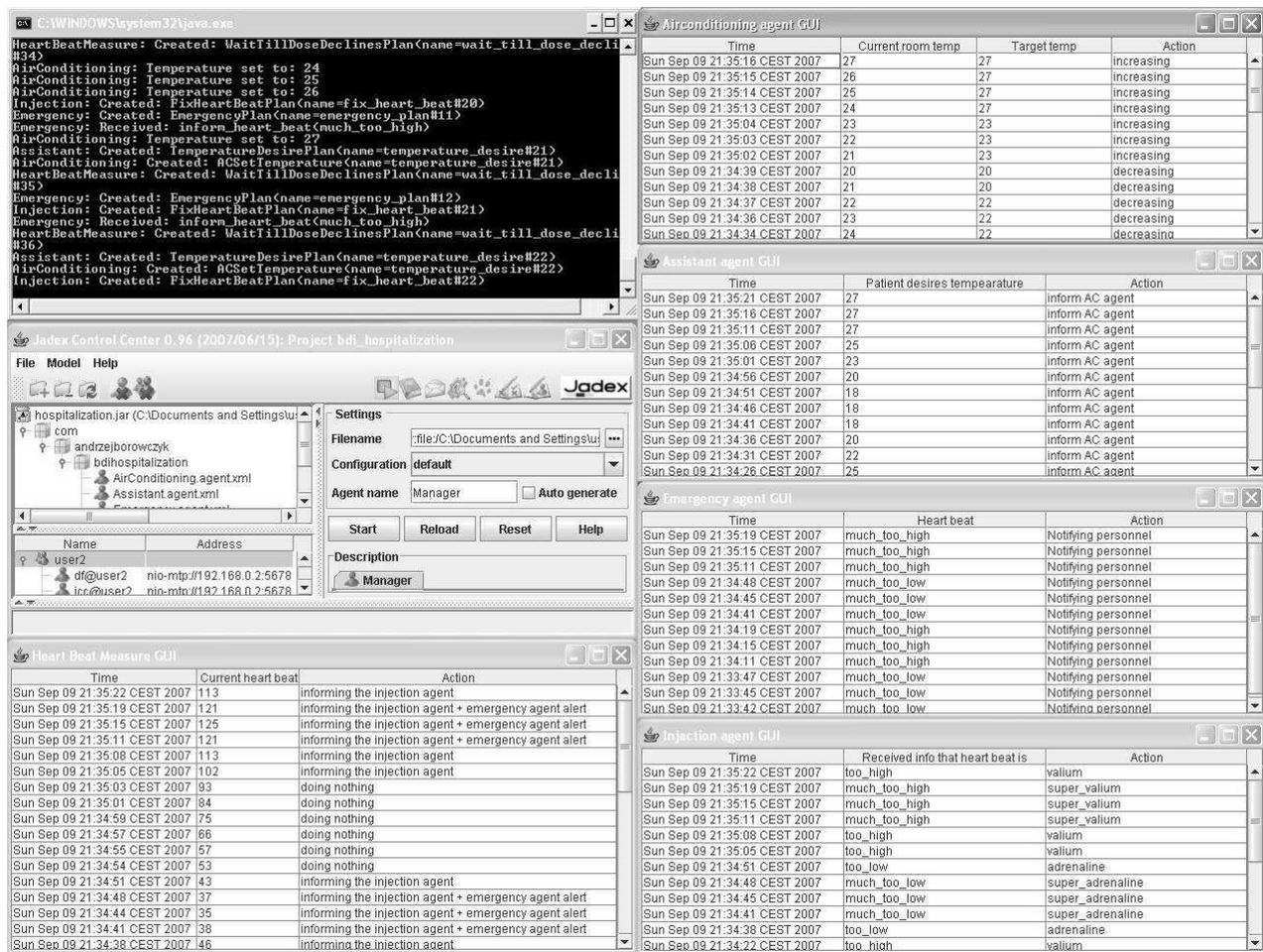


Fig. 2. Running the system: Jadex console, Jadex Control Center GUI and GUIs' of agents in the system.

the systems developed using the BDI approach were created using their individual home-made agents.

Note that the advantage of using BDI software agents is natural. Actors in a system like the one described above are usually associated with some device (e.g. a sensor) and thus, have limited resources (e.g. belief base). They need both some scope of autonomy and solutions for cooperations, e.g. one of the agent's proper functionality depends on some other agent playing a role of a sensor. For instance, the *Injection Agent* relies on the information being sent from the *Heartbeat Measure Agent*. On the other hand this part of the system is completely independent from the *Air Conditioning Part*. Once the role and scope of autonomy of each agent are defined, there is also a need to handle competitive goals inside of a single agent. Since the patient life is crucial, there is a need to define dependencies between such goals precisely. Here the BDI approach naturally provides intuitive abstraction, together with a set of formal tools, like goal deliberation strategies.

The aim of our work was to utilize an existing and extensively developed BDI agent framework (Jadex) in the health-care context. In a limited scenario we have shown how the BDI approach naturally fits with the context of patient

status monitoring. We have also indicated how natural it would be to extend the proposed system to make it more realistic. We have found that Jadex is a very useful and promising tool for the development of BDI agents and plan to continue developing the proposed system with the goal of making it more realistic and thus complex. This will allow us to assess some aspects of scalability of Jadex.

REFERENCES

- [1] M. Bratman. *Intention, Plans, and Practical Reason*. Harvard University Press. Cambridge, MA, USA. 1987
- [2] M. E. Bratman, D. J. Isreal, and M. E. Pollack. Plans and resource-bounded practical reasoning. *Computational Intelligence*, 4(4), 1988
- [3] Jadex website <http://vvis-ww.informatik.uni-hamburg.de/projects/jadex/>
- [4] A. Rao, M. Georgeff, BDI Agents from Theory to Practice, Technical Note 56, AAIL, April 1995
- [5] M. Ljungberg and A. Lucas. The oasis air-traffic management system. In *Proceedings of the Second Pacific Rim International Conference on Artificial Intelligence, PRICAI '92*, Seoul, Korea, 1992
- [6] A. Pokahr, L. Braubach, and W. Lamersdorf. A Goal Deliberation Strategy for BDI Agent Systems. T. Eymann, F. Klugl, W. Lamersdorf, M. Klusch, and M. Huhns. In *Proceedings of the third German conference on Multi-Agent System Technology (MATES-2005)*. Springer-Verlag. Berlin Heidelberg New York. 2005
- [7] N. R. Jennings, M. J. Wooldridge, *Agent technology*, Springer-Verlag, New York, NY, 1998