Using *Jason* to Implement a Team of Gold Miners (a preliminary design)

Rafael H. Bordini[1], Jomi F. Hübner[2], and Daniel M. Tralamazza[3]

[1] University of Durham, UK
[2] FURB, Brazil
[3] EPFL, Switzerland

Jason [2, 3] is an agent platform based on an extension of an agent-oriented programming language called AgentSpeak [5]. The language is inspired by the BDI architecture [6], hence based on notions such as goals, plans, beliefs, intentions, etc. One of the existing software engineering methodologies which is appropriate fore this type of agents is the Prometheus methodology [4]. Below, we describe the ongoing work on designing a ***Jason*** team for the CLIMA-VII Contest (the Gold Miners scenario), using preliminary diagrams generated using the Prometheus methodology.

The overall strategy of the ***Jason*** team is as follows. Each of the four agents will be responsible for one quadrant of the environment. If the depot is in a position within a quadrant which will make it awkward for all other three agents to take any gold they find to the depot (say, near the top left corner of the environment), then the agent responsible for that quadrant will also play the role of *courier* besides of a miner agent (the "normal behaviour" described next). When idle (in the sense of not having any other commitments) each miner agent searches systematically (rather than randomly) *within* their own quadrant and *starting* from the position in the grid (within that quadrant) which is *closest* to the depot position (obviously, considered distances are always the "Manhattan distance" between two positions of the grid). When an agent has to take some found gold to the depot and there is a courier agent, the miner tries to agree with the courier a suitable position (for both) where they will meet so that the miner can hand in the gold to the courier. The courier can refuse to meet to collect gold if it is too busy, but if it becomes idle then the courier itself takes the initiative of contacting agents for which it turned down a request for meeting (for gold collection). The "handing in" of gold is done only in case there are no agents of the opposing team around; i.e., sufficient care is taken to avoid other agents stealing the gold to be passed on to the courier.

The main goal structure of the gold mining agents is very simple. The Goal Overview and the System Overview diagrams can be seen in Figures 1 and 2. One of the miner agents can further play the role of a "courier" (recall, that's on top of being a normal miner). The overview diagrams for miners and the courier agent can be seen in Figures 3 and 5, respectively; Figure 4 shows the overview diagram for one of the capabilities that appear in Figure 3. The overview diagram for the courier agent is only used here to show the interaction with miner agents (the plans for the main percepts are as for the miner agents). The diagrams are fairly clear so we do not explain them in detail. Instead, we next proceed to describe the behaviour of the agents of the ***Jason*** team.

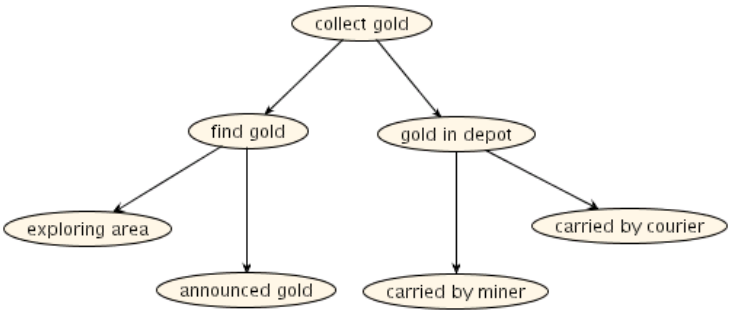

Fig. 1. Goal Overview Diagram

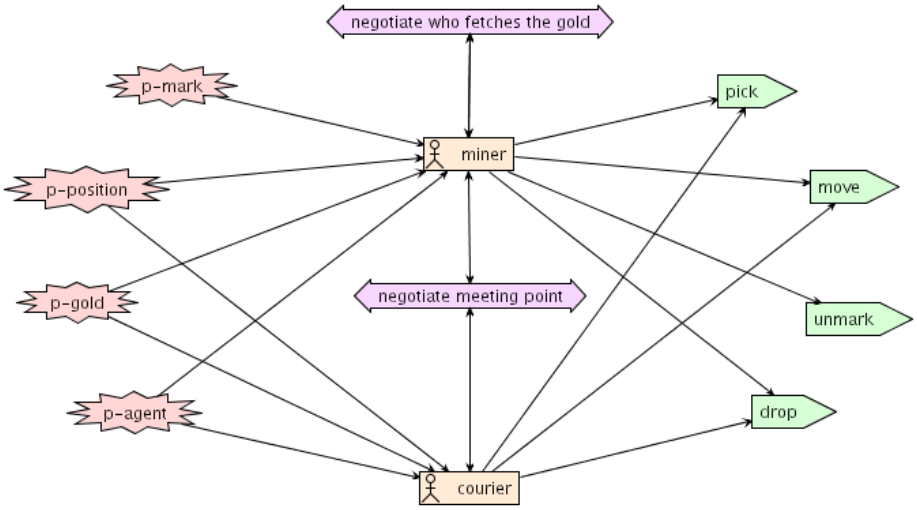

Fig. 2. System Overview Diagram

The behaviour of the miner agents can be described as follows:

1. Whenever the agent has currently no other intention, it adopts the goal of exploring its own quadrant. As mentioned above, the plan for achieving this goal determines that the agent should start in the last place where it last stopped searching for gold (systematically, and within its quadrant), or start from the position in its quadrant which is closest to the depot.
2. Whenever the agent finds gold and it is “free” (meaning it is not carrying some other piece of gold nor committed to doing something else, e.g., collecting another piece of gold due to a previous negotiation, as explained below), the agent picks up the gold and adopts a new goal of getting the gold to the depot. The agent thus tries to arrange a meeting point with the courier. This meeting point needs to be

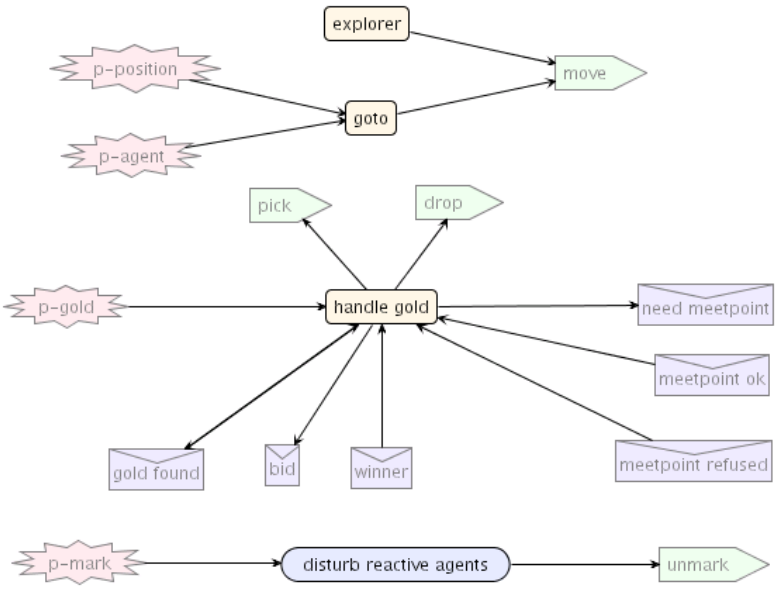

Fig. 3. Miner Agent Overview Diagram

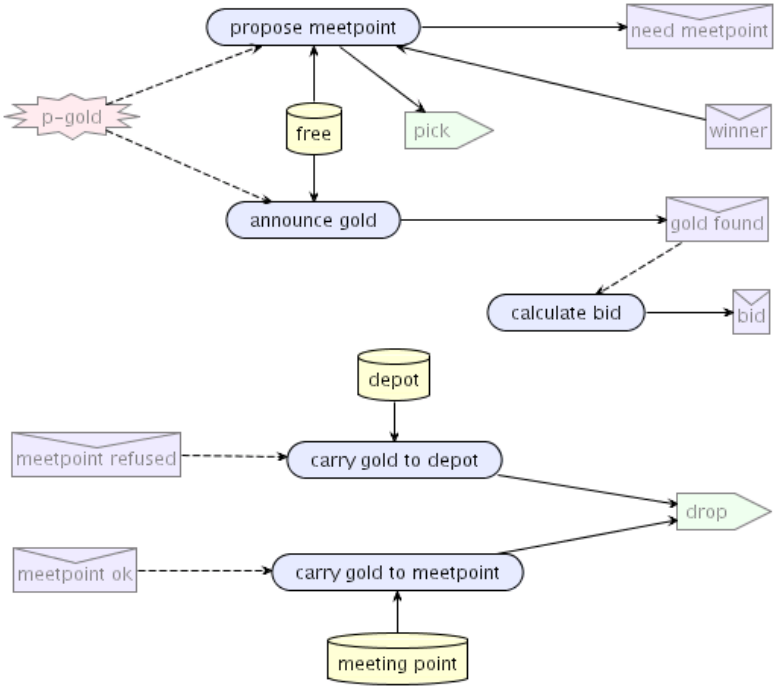

Fig. 4. Miner Agent's handle gold Capability

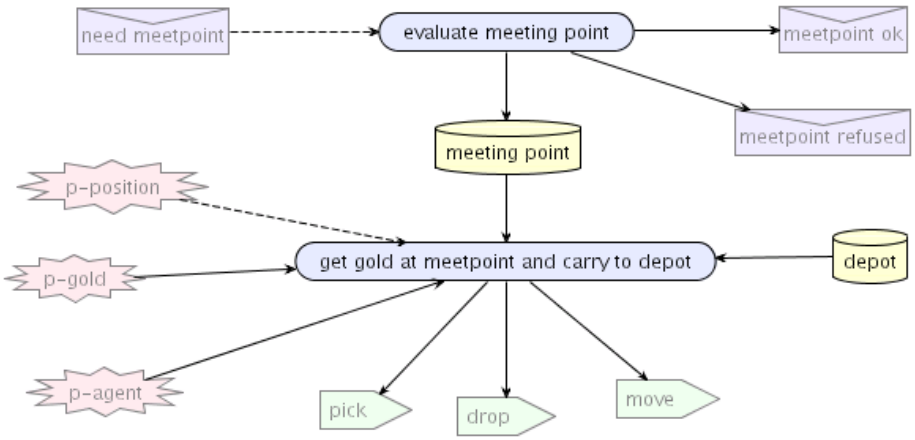

Fig. 5. Courier Agent Overview Diagram

negotiated because it will depend on the courier's commitments and its current location. The courier is, of course, autonomous, and may refuse to arrange the meeting point where the miner could pass on the gold to the courier. In this case, the miner assumes it will have to carry the gold all the way to the depot itself. As mentioned above, the courier takes the initiative to arrange a new meeting point if it becomes free and that miner happens not to have arrived at the depot yet.

3. If a miner agent finds new gold whilst carrying gold or when committed to go and fetch another piece of gold, it broadcasts to all team members the fact that there is gold available in that position where it was perceived. If the agent needs to go past that position again (e.g., in its way back from delivering gold at the depot), it actively checks for gold again to confirm the accuracy of its pervious perception and broadcasts an update on that gold if necessary (e.g., to say there is no gold there as previously thought, as it might have been faulty perception, or the gold might not be there anymore). When new gold is found by an agent that is not free, a negotiation process starts among all miners currently not carrying gold. Even a miner that is already committed to fetch another piece of gold enters this negotiation. If an agent becomes free (e.g., by handing in gold to the courier) and there are found pieces of gold which no agent is committed to collect, the agents engage again in such negotiation process.
4. If an agent is assigned the task of fetching a piece of gold as a result of the negotiation process, that becomes one of its intention. Indeed, such intention will have priority over all others.
5. If a miner perceives a mark in one position of the grid, it takes the time to remove the mark. Our agents are not very friendly towards reactive agents. Think of them as naughty children who mess ants' paths and nests out of sheer cruelty.

For finding the best paths from one location to another, we are considering the use of a search algorithm such as A*. We are not as yet sure exactly how the task allocation process we described above as "negotiation" will take place. Instead of a negotiation technique, we may use a DCOP algorithm to find an optimal allocation of known pieces of gold to available agents, and then having the agents accept whatever recommendation the DCOP algorithm provides. Whatever techniques we decide to use, it will be very easy to invoke them from the AgentSpeak code, specially if there are Java implementations for them, as ***Jason*** provides a very simple mechanism for this.

Clearly, this design is still preliminary and needs to be much more carefully considered (including the Prometheus diagrams) before the actual competition. The various supporting techniques we might need to use within the BDI agents (e.g., a DCOP algorithm) are, as mentioned above, not completely decided yet. Only after we experiment with the ***Jason*** team in CLIMA Contest simulations we will be able to make final decisions about which techniques will work best. The general strategy outlined above may need to be changed too as we start experimenting with the ***Jason*** team.

References

1. R. H. Bordini, M. Dastani, J. Dix, and A. El Fallah Seghrouchni, editors. *Multi-Agent Programming: Languages, Platforms and Applications*. Number 15 in Multiagent Systems, Artificial Societies, and Simulated Organizations. Springer-Verlag, 2005.
2. R. H. Bordini, J. F. Hübner, et al. ***Jason****: A Java-based agentSpeak interpreter used with saci for multi-agent distribution over the net*, manual, first release edition, Jan 2004. `http://jason.sourceforge.net/`.
3. R. H. Bordini, J. F. Hübner, and R. Vieira. ***Jason*** and the golden fleece of agent-oriented programming. In Bordini et al. [1], chapter 1, pages 3–37.
4. L. Padgham and M. Winikoff. *Developing Intelligent Agent Systems: A Practical Guide*. John Wiley and Sons, 2004.
5. A. S. Rao. AgentSpeak(L): BDI agents speak out in a logical computable language. In W. Van de Velde and J. Perram, editors, *Proceedings of the Seventh Workshop on Modelling Autonomous Agents in a Multi-Agent World (MAAMAW'96), 22–25 January, Eindhoven, The Netherlands*, number 1038 in Lecture Notes in Artificial Intelligence, pages 42–55, London, 1996. Springer-Verlag.
6. A. S. Rao and M. P. Georgeff. BDI agents: From theory to practice. In V. Lesser and L. Gasser, editors, *Proceedings of the First International Conference on Multi-Agent Systems (ICMAS'95), 12–14 June, San Francisco, CA*, pages 312–319, Menlo Park, CA, 1995. AAAI Press / MIT Press.