Multiagent Systems and Societies of Agents, as presented by Shabbir Syed

Cory Lueninghoener

April 16, 2002

Lin Xu began our discussion on multiagent systems with an introduction and a description of agent communications. This discussion was documented by Daniel Buettner and has been publicly distributed.

Shabbir picked up the discussion with agent interaction protocols. Here he pointed out that there are two important states a set of agents can be in: one in which the agents have conflicting goals, or one in which the agents have similar goals. Since the agents need to communicate to get their goals accomplished, they need a protocol to govern this communication. Remember, communication is very important between agents. When agents have similar goals, it is useful for them to work together. Thus, we want to "maintain globally coherent performance without violating autonomous behavior of agents." This can be done with communication through known protocols.

Seven interaction protocols were listed by Shabbir: coordination protocols, cooperation protocols, contract nets, blackboard systems, negotiation, multi-agent belief maintenance, and market mechanisms. He then proceeded to discuss them each in turn.

Coordination protocols are used to maintain dependencies between actions and meet global constraints. The messages passed using these protocols help share information between various agents and contribute to the idea of distributed AI among the group. Unfortunately, this isn't perfect: since not everybody hears all messages, the knowledge base becomes fragmented among all of the agents and each agent only has a partial view of the whole KB. This makes for uncertainty in actions confused global behavior. Here Shabbir pointed out his own thoughts that peer-to-peer communications might work better.

The above mentioned knowledge and actions can be represented as an and/or

graph in which the leaves represent the goals. Thus, the activities that need coordination are defining the goal graph, assigning work to agents, controlling access of the graph, traversing the graph, and ensuring that traversals are reported.

Of course, bandwidth is not limitless, so having a convention in place to control communication is good. We discussed the convention found on page 98 in the text that is useful for informing other agents of dropped/satisfied commitments. We also discussed a protocol convention for use when two agents decide to work together. This too is covered in detail on page 98/99 of the paper, and thus is not repeated here.

Shabbir next discussed cooperation protocols with us. These protocols are used for dividing work among several less capable agents so they can all work together. Several methods for distributing tasks among agents were discussed, including spatial decomposition and functional decomposition. The bottom half of figure 2.5 from the paper brought up some doubting confusion from both Dan and Shabbir, but it was eventually concluded that the idea was right even if the diagram made no sense.

We then moved on to discussing contract nets, which Dr. Choueiry pointed out are a very early multiagent system. Contract nets are based on the idea that one agent declares itself "manager" and sends out a task that needs to be performed to all of the other agents. These agents then evaluate their ability to do the task and reply to the manager. Finally, the manager chooses one of the "bids" it heard back and tells that agent to do its work. This idea tends to work fairly well, but has some drawbacks: the perfect agent for the job may already be busy, agents may bid on a task and never hear that they didn't get it, no agents are able to take the task, etc. Thus, a more restricted form of the contract net method was discussed in which specifications are set on who must talk to whom when a request is made.

Next Shabbir talked about blackboard systems. Some characteristics of blackboard systems are independence of expertise, diversity of problem-solving techniques, flexible representation of information, a common interaction language, event based activation, need for control, and incremental solution generation. The general idea behind a blackboard system is that there is a central repository of knowledge that is updated by the agents as they feel the ability to contribute work and knowledge. Each agent tells the system its abilities, and the blackboard lets them know when they can help. This makes for incremental solution generation.

Negotiation was the next top of discussion. This is a joint decision reached by two ore more agents. The negotiation process has a number of attributes mentioned by Shabbir. These were efficiency, stability, simplicity, distribution, and symmetry. There are also two types of systems for negotiation: environment cen-

tered and agent centered. To get negotiation done requires a unified negotiation protocol. This is based on the idea that agents will make deals with each other where each agent wants to maximize its utility of each deal. Thus, they talk about the negotiation set, which holds all of the deals that have positive utility for all of the agents. Shabbir pointed out that there are three situations that can arise with respect to the negotiation set: conflict, in which the set is empty; compromise, in which agents decide to work together because they have to; and cooperative, where everybody is happy working together.

Next Shabbir talked about Truth Maintenance Systems (TMS). Specifically, with relation to multiagent systems he talked about justification based TMS. A general TMS is used to keep an agent's knowledge stable, well-founded, and logically consistent. In a JTMS, each datum has justifications and a status of INTERNAL, EXTERNAL, or OUT. When the justification status of a datum is to be changed, a series of actions must be performed on several agents (assuming the datum is shared). First, the newly justified datum and its consequences are unlabeled. Next, new labels for the unlabeled shared data are chosen. Last, the data inside each agent are relabeled. If an agent fails, it needs to backtrack.

Shabbir next talked about market mechanisms. This is the name for a method of keeping agents busy and working through a market-like model. It works by modeling jobs that need to be done and work that is done as goods that are traded between the agents. This allows the consumers to maximize their utility according to their budget constraints and producers to maximize their profits among the consumers' capabilities.

The last topic covered was societies of agents. Here Shabbir talked about how intelligent agents work well in groups, which also happens to work well with distributed systems and peer-to-peer conversation. He also pointed out how social dependence works: `social_dependence(x, y, a, p)` means that agent `x` depends on agent `y` with regard to act `a` for realizing state `p`, where `p` is a goal of `x` and `x` is unable to realize `p` while `y` is able to do so. There are three types of dependencies that can arise: voluntary, compound, and reciprocal. The agents form a cooperative team when all agents share a common goal, each agent is required to do its share to achieve the common goal, and/or each agent adopts a request to do its share.

At this point we began the full-class discussion, which brought up the following points:

Dr. Choueiry : In societies of agents, point-to-point is better than client-server, but in game theory they say "Convergence cannot be had with P2P." Thus,

it depends on the task at hand.

Answer : You are the professor, so you must know what you are talking about.

Tibor : Why do we bother considering facts that are out?

Answer : We must have all in's in and all out's out.

Cory : Can a manager ever be its own contractor?

Answer : It isn't needed - the agent would figure out that it can do things on its own before needing to send out a request for help.

Dan : In this Market protocol, what are the goods and currency being exchanged? It sounds kind of fake.

Answer : Limited resources and information are what are exchanged.

Rob : Can agents starve?

Answer : Yeah, that's bad. If they don't send in a response, they will never get any jobs in a manager-worker setup. Thus, they must send in responses.

Rob : But still, they may be too weak to get any work at all done.

Rob : In the negotiation protocol, symmetry weakens autonomy.

Answer : Correct you are.

Lin Xu : What if an agent bids, and then gets a better offer from a different agent before the response on its original bid comes back?

Answer : There needs to be a bid timeout to help with that.

Amy : How do you decided if a statement is true? Backtrack search?

Answer : Yes. Er, no. Well, the point is to avoid backtrack search. You have all of the facts and just need to switch them on and off.