Minutes from "Multiagent Systems and Societies of Agents"

Daniel Buettner
Lecture by Lin Xu
CSCE 976 Advanced Artificial Intelligence

April 8, 2002

1 Introduction

In the lecture from 3 April, we covered material from the second chapter of *Multiagent Systems*. This chapter of the text was written by Michael Huhns and Larry Stephens.

In this lecture, Lin discussed the first two sections of the chapter leaving the remaining two sections to be covered by Shabbir Syed at a future time.

2 Lin's lecture

The previous chapter introduced us to the idea of agents, so this chapter jumps right in and starts talking about agents working together. Lin's talk was chiefly introducing us to mechanics of agent communication.

Communication protocols are used to enable agents to exchange and understand messages. There are several messages that can pass between two agents:

- Propose a course of action
- Accept a of course of action
- Reject a course of action
- Retract a course of action
- Disagree with a proposed course of action
- Counterpropose a course of action

Interaction protocols (based on lower level communication protocols) are used to enable agents to have conversations. Conversations can be seen as structured exchanges of messages. An example of agent interaction is the case of two agents negotiating a course of action.

In order to motivate the study of multiagent systems, the author begins by stating:

> But why should we be interested in distributed systems of agents? Indeed, centralized solutions are generally more efficient: anything that can be computed in a distributed system can be moved to a single computer and optimized to be at least as efficient.

This bold assertion caused an outcry to rise up from the class. If this were indeed the case, we would have no need to ever take a parallel algorithms course. After some discussion it was decided that the authors did not mean this in a practical sense, but in a purely theoretical sense. That is to say, the authors were expressing the same sentiment as Archimedes when he famously said "Give me a lever long enough and a fulcrum on which to place it, and I shall move the world.".

At any rate, the authors go on to say that distributed systems can be easier to understand when the problem itself is distributed. Another feature in favour of distributed systems is that there are times when a centralized approach is impossible. Additionally, a distributed approach allows for a very natural mechanism to respect privacy: agents are allowed to decide which information they wish to make public.

Multiagent environments provide an infrastructure specifying communication and interaction protocols. They are typically open, meaning that agents can be freely added or removed from the environment without great impact on performance. They typically have no centralized design, that is to say that there is seldom a "supervisor" agent. Multiagent environments contain autonomous and distributed agents that may be self-interested or cooperative.

2.1 Agent Communications

This chapter defines an agent as an active object with the ability to perceive, reason, and act. An agent has explicitly represented knowledge and a method for drawing inferences from this knowledge. Agents have the ability to communicate with each other by sending a receiving messages.

Coordination is a property of a system of agents performing an activity in a shared environment. Coordination entails avoidance of extraneous activity,

avoidance of livelock and deadlock, as well as maintenance of necessary safety conditions. Coordination can be achieved by either cooperation among non-antagonistic agents or by negotiation among competitive agents.

Coherence is how well a system behaves as a unit. The question is, can a system maintain global coherence without explicit global control. To help ensure global control, we want to ensure that these properties hold:

- agents should be able to determine on their own which goals they share with other agents
- agents should be able to determine common tasks
- unnecessary conflicts should be avoided
- knowledge and evidence should be pooled

Some amount of organization among the agents can help ensure these properties.

There are three key aspects to the formal study of communications.

syntax how the symbols of communication are structured

semantics what the symbols denote

pragmatics how the symbols are interpreted

Meaning is a combination of semantics and pragmatics.

There are several dimensions of meaning associated with communication:

Descriptive vs. Prescriptive whether a message describes a phenomena or prescribes a behavior

Personal vs. Conventional Meaning a meaning for a message understood by a particular agent but at odds with the understanding of the message by the other agents

Subjective vs. Objective Meaning if a message has an explicit effect on the environment, it is possible for an agent to subjectively expect a different outcome from that which can be objectively measured

Speaker's vs. Hearer's vs. Society's Perspective independent of the objective meaning of a message, it can be expressed according to the viewpoint of the speaker, hearer, *&c.*

Semantics vs. Pragmatics the pragmatics of a communication are concerned with how the communicators use the communication. Semantics are strictly an issue of understanding.

Contextuality messages cannot be understood in isolation; they must be understood in the context of a conversation

Coverage a language must be large enough and expressive enough that an agent can convey the meaning it intends

Identity the understanding of a message is dependant on the parties involved in it

Cardinality a private message would be understood differently than a broadcast message

There are two basic message types – assertions and queries. A passive agent can accept a query and answer it with an assertion. An active agent can issue queries and make assertions. All agents should be able to accept assertions. Two agents are peers if they take turns playing the active and passive rôles.

Speech act theory is a popular basis for analyzing human communication. Consequently, spoken human communication is used as the model for communication among computational agents. A speech act has three aspects

locution the physical utterance by the speaker

illocution the intended meaning of the utterance

perlocution the action that results from the locution

For example, the statement "I am cold" could have the intended meaning of "hand me my sweater" but instead result in the action of turning up the thermostat. The concept of illucutionary force constrains the semantics of a communication action. The sender's intended communication act is clearly defined, and the receiver has no doubt as to the type of message sent.

At this point, the paper goes into a rather in depth discussion of the KQML knowledge query and manipulation language. KQML is a fairly rigidly defined language for enabling inter-agent communication. Each communication specifies the sender, receiver, language of the attached message, the ontology delineating the semantics of the message, and the content of the

attached message itself. KQML provides mechanisms for basic query performatives, generic informational performatives, networking performatives *&c.*

Some of the issues involved with KQML are that the sender and receiver must understand the agent communication language. Additionally, the intended ontology must be accessible to all agents involved in a communication. KQML must operate within a communication infrastructure that allows agents to locate each other, since the language provides for directly addressing specific agents. KQML is somewhat old now, and newer languages like FIPA have been defined to take its place.

The knowledge interchange format (KIF) is a logic language used as a standard to describe facts in expert systems, databases, intelligent agents and so forth. It was specifically designed to serve as the Esperanto of computer communications by providing an *interlingua* in the translation of other languages. KIF was designed around first order predicate calculus and was developed in LISP at Stanford.

In order to communicate, agents need to share the same ontology. An ontology specifies objects, concepts, and the relationships in some area of interest. This is similar to the entity relationship diagrams used in databases.

Once the communication protocols have been defined, higher level protocols can be defined to allow agents to begin interacting.

3 Student interaction

TIBOR:	Can passive agents accept new information?
A:	All agents should be able to accept assertions.
TIBOR:	Why must agents be able to locate each other?
A:	This just means that agents must be able to get in contact with each other, not necessarily physically locate each other.
AMY:	Centralized computing is always faster? What about clustered super computing?
A:	In *theory* centralized computing is faster, but this requires an infinitely powerful machine.
AMY:	These two chapters don't seem to agree on the definition of an agent.
A:	This chapter is chiefly concerned with communication and doesn't focus on the nature of agents.
CORY:	They should XML for information interchange!
A:	Or maybe even WML!
CORY:	I also took issue with the statements about single computers vs. distributed computing.
A:	That's been discussed in this very document.
ROB:	The claim that "Multiagent systems are the best way to characterize or design distributed computing systems." seems to be rather brash.
A:	Yes, it certainly does!
DANIEL:	There is no discussion of limiting communication to prevent saturating the network.
A:	That's true. But neither do they mention how much bandwidth is available – those are lower level details.
SHABBIR:	How does centralized control figure in?
A:	Centralized control can be used to ensure that agents compete to solve problems, cooperate to solve problems, or are given subtasks to solve independently of each other.
PRAVEEN:	What is meant by "push and pull" communication on page 82 of the text?
A:	Just like with the web, you can either push information from a server to a client, or you can pull information from a server to a client.