RULES FOR COLLABORATIONS AMONG CYBER-AGENTS

Sarosh Talukdar
Carnegie Mellon University
talukdar@cmu.edu

Invited Talk
Improving Decision Trees Using Tabu Search
Informs National Meeting
Montreal, Quebec
April 26-29, 1998

CLAIM

A few simple rules are sufficient for effective collaborations among software agents engaged in off-line problem-solving

(an off-line problem has no hard constraints on solution-time, rather both solution-quality and solution-time are objectives)

MOTIVATION

- Computer networks make it possible to interconnect very large numbers of software agents and thereby, to amass enormous amounts of raw problem-solving capability
- If this raw capability is to be useful, it must be organized (focused, coordinated). That is, the agents must collaborate effectively.
- Organizing software agents is very expensive
- Solution: reduce the amount of organizing necessary for effective collaboration.

WHAT IS AN AGENT?

- An agent is a bundle of skills:
  - problem-solving skills (what the agent can do)
  - social skills (what the agent chooses to do)
  - learning skills (how the agent transforms experience into new skills)
- An agent’s environment is the set of all things that it can affect or by which it is affected.

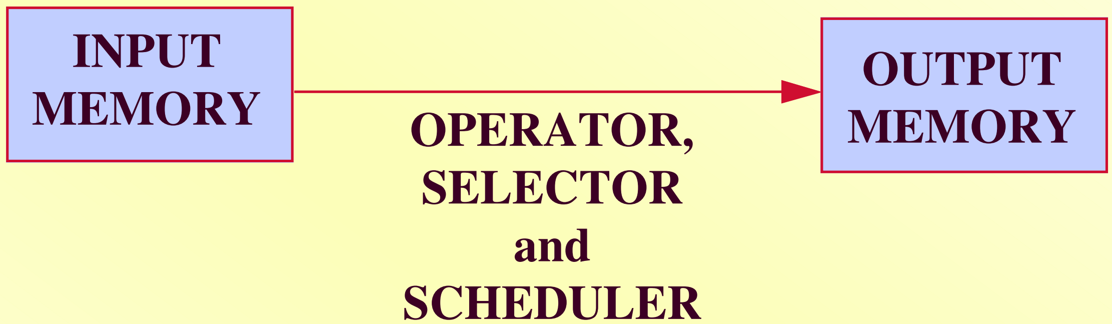
WHAT IS A CYBER-AGENT?

A cyber-agent is an agent whose environment is computer-mediated.

The two main types of cyber-agents are:

- Software agents (programs)
- Humans (while they are engaged in computer-mediated work)

A MODEL OF AN AUTONOMOUS CYBER-AGENT:



SELECTOR: picks objects from input memory (space of all things the agent can perceive)

OPERATOR: reads these objects, modifies them and places results in output memory (space of all things the agent can affect)

SCHEDULER: determines when selector and operator will work.

Note:

- The agent is autonomous because its selector and scheduler are self-contained
- Input and output memories may overlap

COLLABORATION

- Two cyber-agents collaborate whenever they exchange data, whether the exchange is productive or not.
- Collaboration is effective if it results in high solution-quality and low solution-time.

SOFTWARE AGENTS

Existing software agents tend to have:

- useful problem-solving skills
- little or no social skills
- little or no learning skills

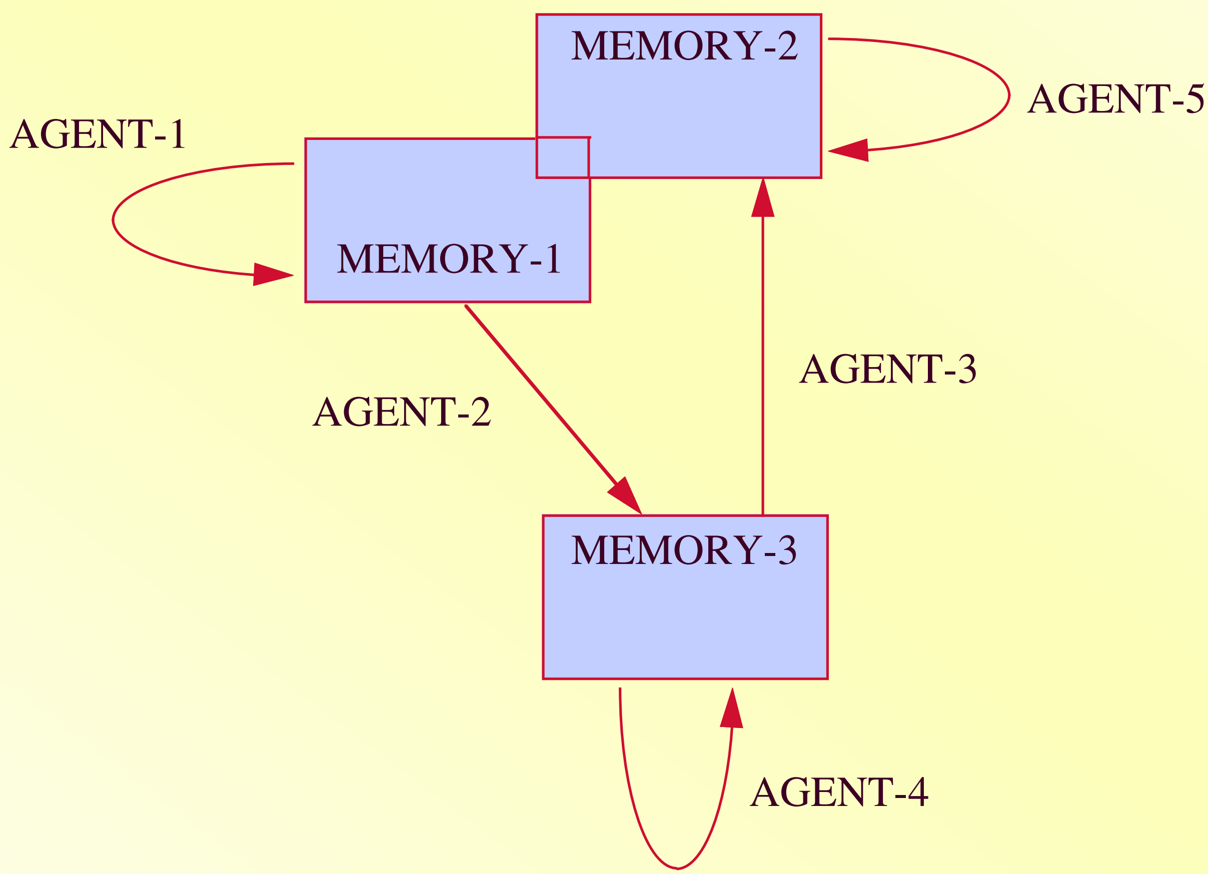
Question: what minimal set of social skills need to be added to each software agent to make it effective in its collaborations with other agents?

An answer: the A-Teams (Asynchronous Teams) approach.

THE A-TEAMS APPROACH

- Agents are made autonomous
- Agents are connected to form a strongly cyclic network, so their results can circulate
- When not circulating, results accumulate in stores (memories) to form populations of results
- Agents use QBS selection

EXAMPLE OF A SIMPLE A-TEAM



TWO TYPES OF AGENTS

Constructors:
- select and copy a result from a memory
- modify the result
- put the modified result back in a memory

Destroyers:
- select and erase a result from a memory

QUALITY BASED SELECTION (QBS)

A scheme by which agents select results to modify (if they are constructors) or erase (if they are destroyers) is:

For constructors: choose a result randomly from a result-population, with a bias towards higher quality results

For destroyers: choose a result randomly from a result-population, with a bias towards lower quality results

OR

Use a Tabu list to identify results for destruction

(This scheme assumes that a scalar quality measure has been designed for all the results in each population)

A FINDING

Both empirical evidence and theory suggest that:

If A-Teams and QBS are used and the agents are allowed to work concurrently

Then

- the results will converge
- the quality of the final results will improve with the addition of agents
- the speed of convergence will improve with the addition of computers (for the agents to use)

WHERE HAVE A-TEAMS AND QBS BEEN SUCCESSFULLY APPLIED?

- Traveling Salesman Problems
- High-Rise Building Design
- Reconfigurable Robot Design
- Power System Diagnosis
- Railroad Scheduling
- Job Shop Scheduling
- Paper Mill Scheduling
- Steel Mill Scheduling
- Path Planning for Welding Robots
- Spatial Realization
- A Prototype for Contingency Planning

WHERE DID THE IDEAS FOR A-TEAMS AND QBS COME FROM?

- Insect Societies
- Cellular Communities
- Genetic Algorithms
- Blackboards
- Simulated Annealing
- Tabu Search
- The Brainstorming Protocol

Theory Suggests, Though Experimental Conformation Remains to Be Obtained, That:

- Construction and destruction are duals: adept destruction agents can compensate for inept construction agents, and vice-versa.

- There are no restrictions on agent complexity or phyla: in an A-Team, simple agents can be combined with complex agents, software agents can be combined with humans.

SUMMARY: RULES FOR EFFECTIVE COLLABORATION

Organizations that are scale-effective (solution-quality improves with the addition of agents, solution-speed improves with the addition of computers) and open (agents and computers are easy to add) are likely to be obtained if:

- the agents are autonomous
- the agents are allowed to collaborate only by modifying one another's results
- the agents use quality-based selection
- the agents are connected to form a strongly cyclic network
- construction and destruction are "balanced" (inept construction agents are compensated by adept destruction agents, and vice-versa)
- humans are used to provide skills not available in software