SELF-ORGANIZING SOFTWARE AGENTS

Sarosh Talukdar
Carnegie Mellon University
talukdar@ece.cmu.edu

Software Engineering Working Group
Winter Power Meeting
Tampa, Florida
February 1998

MOTIVATION

- Computer networks make it possible to interconnect very large numbers of software agents (computer programs) 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)
- Organizing software agents is very expensive
- Solution: make software agents self-organizing

CLAIM

It is easy to make software agents self-organizing, or very nearly so (much easier than trying to organize large numbers of software agents)

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.

COLLABORATION AND ORGANIZATIONS

- Two information processing agents collaborate whenever they exchange data, whether the exchange is productive or not.
- An organization is:
  - a set of agents
  - a prescription for their collaboration
- If the prescription for collaboration is built-in to the agent, that is, if the prescription is part of its social skills, then the agent is self-organizing

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 self-organizing?

An answer: ACS (Asynchronous Collaboration Scheme)

THE ASYNCHRONOUS COLLABORATION SCHEME (ACS)

- 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
- Two types of agent-work cycles:

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

  Destroyers: - select and erase a result from a store

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 result randomly from a result-population, with a bias towards higher quality results

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

(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: ACS 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)
- construction and destruction are duals - adept destruction agents can compensate for inept construction agents, and vice-versa

WHERE DID THE IDEAS FOR ACS AND QBS COME FROM?

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

WHERE HAVE ACS 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

CONCLUSIONS

- QBS schemes are easy to implement and are the only social skills needed by software agents to collaborate effectively, provided the agents are in an ACS, which is also easy to implement.
- In other words, effective collaboration can be achieved by a hybrid organization scheme:
  - self organizing by QBS, plus
  - some external organizing by ACS.
- Potential is very great, for instance, for distributed, real-time control of power system dynamics and power system markets.