Organizational self-design in semi-dynamic environments

Sachin Kamboj and Keith S. Decker

Dept. of Computer and Information Science, University of Delaware, Newark, DE

Motivation

What do we want to do?

To organize agents and address the following issues:

- The number of agents needed to solve the problem
- Allocation of subtasks and resources to the agents
- Coordination of inter-agent activities

So what's the problem?

- There is no best way of organizing and all ways of organizing are not equally effective:
  - The optimal organizational structure depends on the underlying problem being solved and the environmental conditions.
- Environmental conditions or problem structure may change.
  - Precludes the use of a static (compile-time) organization.
- All problem instances and environmental conditions are not unique.
  - Precludes the use of a new, bespoke organizational structure for every problem instance.

Our Answer:

We propose a dynamic run-time approach to organization based on Organizational-Self Design (OSD)

Our Contributions:

- Extend existing OSD work (2) to use TÆMS as its underlying problem representation, which allows us to:
  - Perform quantitative reasoning over the task structure
  - Represent cost/quality tradeoffs
  - Soft relationships like facilitates and hinders
- Incorporate coordination mechanisms into OSD

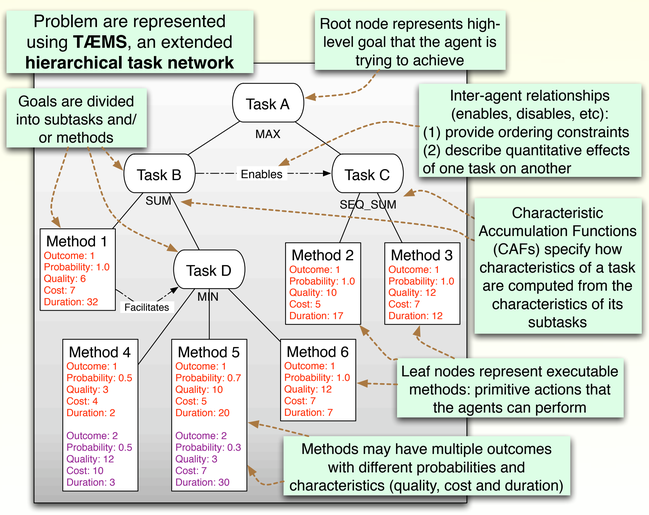
Basic Premise

- Problem solving requests arrive continuously at varying rates and with varying deadlines

Approach

- Start off with an initial organization
  - consisting of a single agent, solely responsible for all activities
- Each agent in the organization checks to see if:
  - It is overloaded:
    - It spawns off a new agent to handle part of its load
  - It is free (underloaded):
    - It combines with another agent to save resources

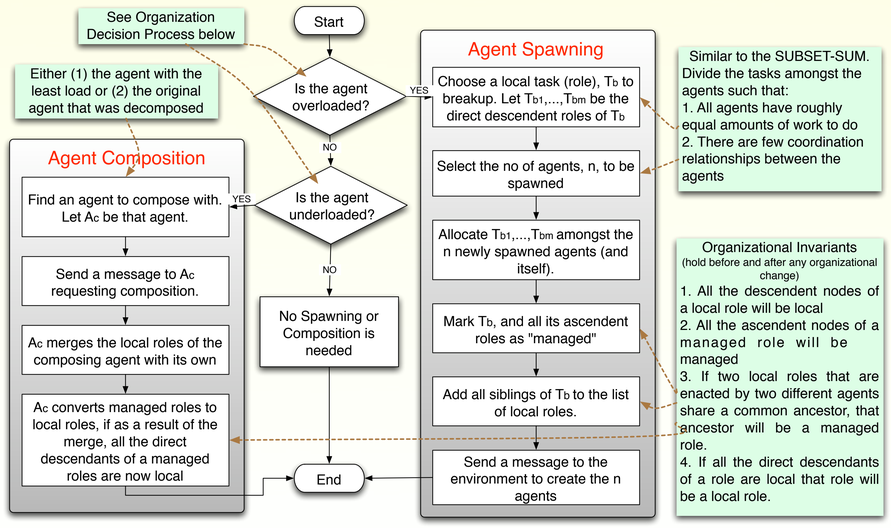
Problem Representation



Organizational Structure: Roles and Relationships

- Organizations are characterized by an organizational structure, which
  - consist of roles and relationships between the roles
  - is contingent on the TÆMS task structure
- A role is defined as a TÆMS subtree rooted at a particular node
  - By definition, a role may consist of one or more sub-roles
  - A role may be enacted by one or more agents and one or more roles may be enacted by every agent
- Relationships constrain the communication between the agents

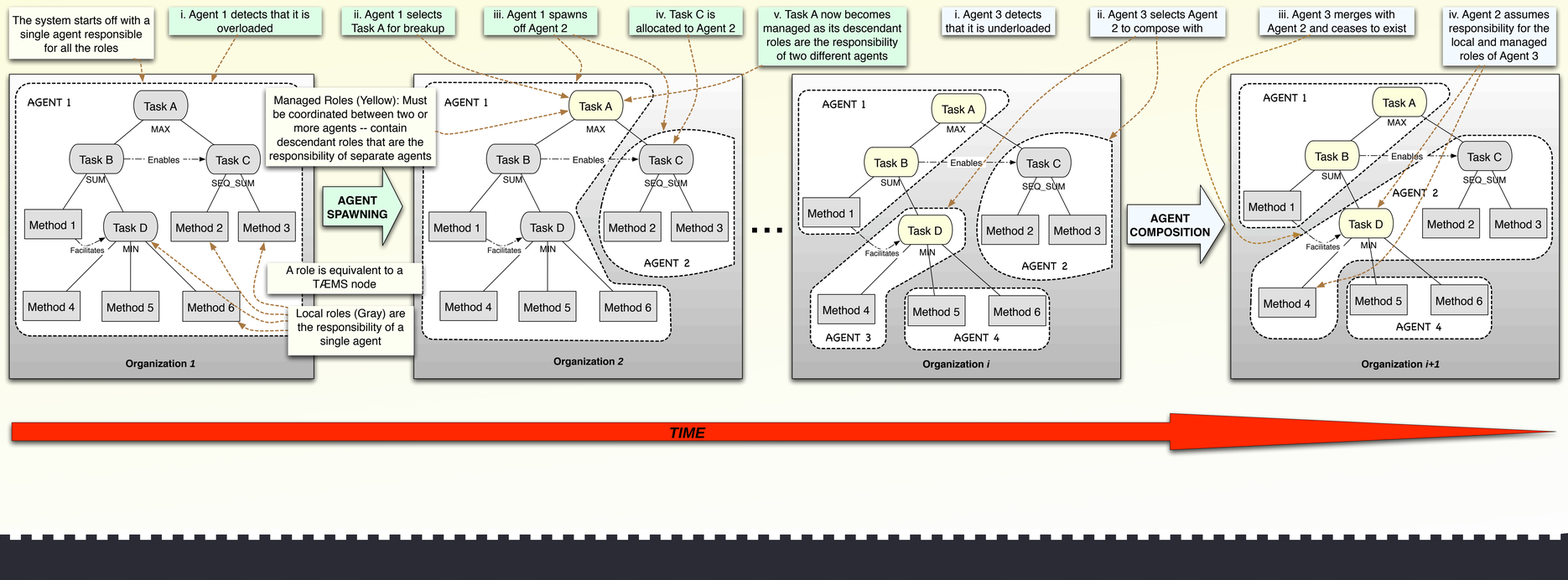
Organization Formation/Adaptation Process



Organization Decision Process

- Probability of organizational change should be inversely proportional to time since last organizational change
  - We use simulated annealing to vary this probability.
  - $p = e^{-\frac{\Delta E}{kT}}$, where
    - p = probability of keeping an existing organizational structure
    - T = time since the last organizational change
    - ΔE = amount of overload/underload
  - p is capped at $p_{threshold}$ to prevent the agents from being unresponsive to environmental change
- Computation of ΔE:
  - Agent Spawning
    - Sort the outstanding tasks by their deadlines
    - Iterate over the sorted tasks and compute:
      - min = absolute minimum time needed to have a non-zero probability of task completion
      - expt = expected time needed to complete the task
      - g_min = minimum time needed to guarantee task completion in the absence of failures
      - if sum(min) > Deadline
        p = 0, breakup the agent immediately
        else
        $\Delta E = \frac{1}{\alpha(Deadline - Current\_Time - \sum expt) + (1-\alpha)(\sum g\_min - \sum min)]}$
  - Agent Composition
    - If the agent, is idle, compute p using the simulated annealing equation with
      $\Delta E = \beta * Idle\_Time$

Example Illustrating OSD



Evaluation

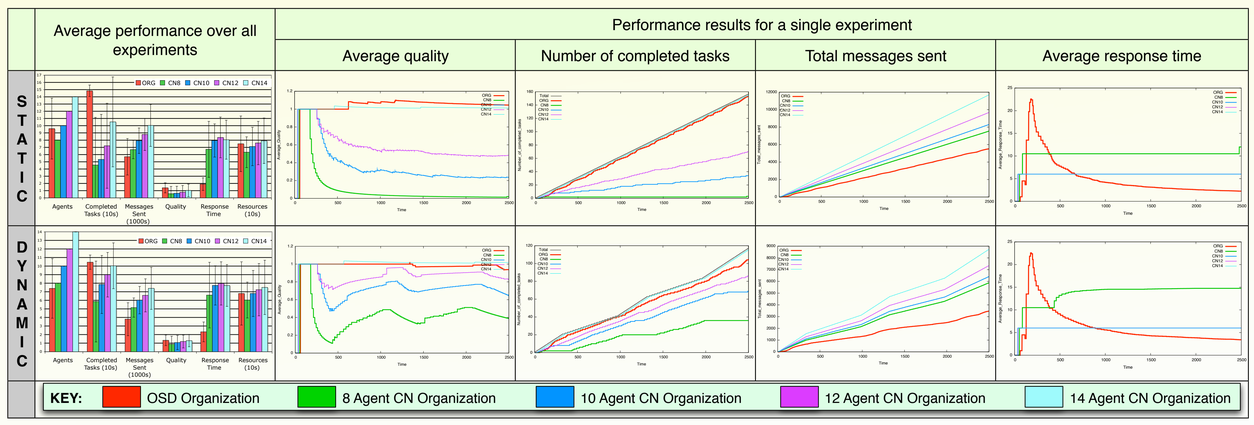
Experimental Setup

- Compared to the Contract Net Protocol (CNP)
- Ran 40 Experiments
  - each experiment was repeated 5 times
    - our OSD approach was used the first time
    - the CNP approach, with 8, 10, 12 and 14 agents, was used in the 4 subsequent runs
  - 20 experiments had a static environment
    - Fixed task arrival rate every 15 cycles
    - Fixed deadline of 20 cycles
  - 20 experiments had a dynamic environment
    - Arrival rate changed from 15 to 30 cycles and back to 15 cycles after every 20 tasks
    - Deadline of 20 cycles
  - Random task structure was generated for each experiment

Hypothesis Tested

Hypotheses Tested using Wilcoxon Matched-Pair Signed-Rank Tests	Static Environments	Dynamic Environments
The OSD organizations require **fewer agents** to **complete an equal or larger number** of tasks when compared to the Contract Net organizations Null Hypothesis: ~~The CN agents complete more tasks than the OSD agents~~ **REJECTED**	$p < 0.0003$	$p < 0.03$
The OSD organizations achieve an **equal or greater average quality** than the Contract Net organizations Null Hypothesis: ~~The CN agents achieve greater quality~~ **REJECTED**	$p < 0.01$	$p < 0.05$
The OSD agents have a **lower average response time** as compared to the Contract Net agents Null Hypothesis: ~~The OSD agents have the same or greater response time compared to the CN agents~~ **REJECTED**	$p < 0.0002$	$p < 0.0004$
The OSD agents **send less messages** than the Contract Net Agents Null Hypothesis: ~~The OSD agents send the same or more messages than the CN agents~~ **REJECTED**	$p < 0.0003$	$p < 0.02$

Performance Graphs



Future Work

- Investigate how changes in the task structure affect the organizational structure
- Incorporate a larger set of non-local effects
- Incorporate robustness into the organization
  - Formalize and evaluate alternative robustness schemes to determine the best one for OSD
- Incorporate communication delays
- Apply OSD to a real-world application

References

1. K. M. Carley and L. Gasser: *Computational organization theory.* In G.Wiess, editor, Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence, pages 299 -- 330, MIT Press, 1999.
2. T. Ishida, L. Gasser, and M. Yokoo: *Organization self-design of distributed production systems.* IEEE Transactions on Knowledge and Data Engineering, 4(2):123--134, April 1992.
3. V. Lesser, K. Decker, T. Wagner, et. al.: *Evolution of the GPGP/TAEMS Domain-Independent Coordination Framework*. AAMAS '02 (Plenary Lecture/Extended Abstract)}, pages 1--2, 2002.