

Principles in Multi-Agent Systems

DCSC Colloquium

Rudy Negenborn

<http://www.dcsc.tudelft.nl/~rnegenborn/>

June 23, 2004

1/22

Overview

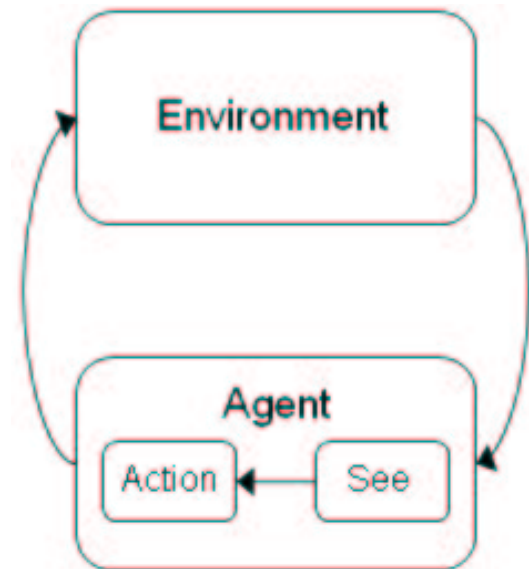
- **Introduction**
- **Multi-Agent Systems**
- **Application Areas**
- **Principles**
- **Real-Time Control**
- **Conclusions**

Introduction

- 1980s: Distributed Artificial Intelligence:
Construct systems of intelligent entities
that interact productively with one another
Distribution and coordination
of knowledge and actions
- Main directions:
Distributed Problem Solving
Multi-Agent Systems

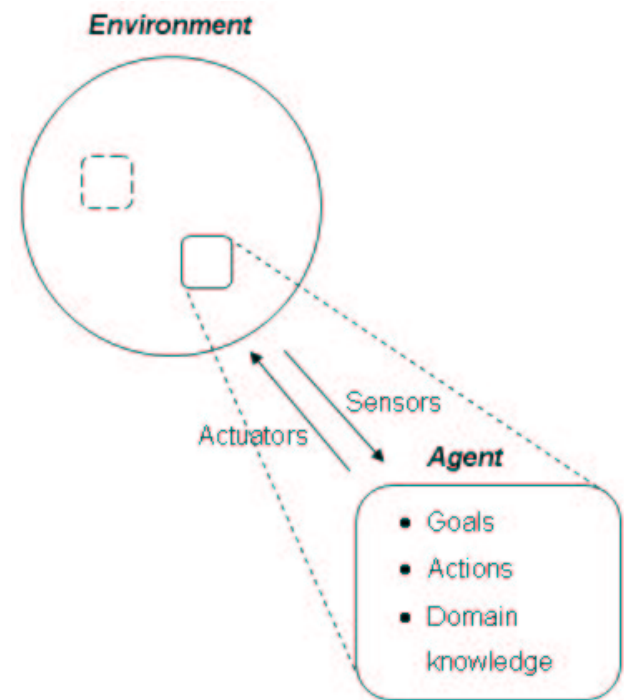
What is an Agent?

- An entity
in an environment,
capable of acting,
capable of sensing,
with goals and knowledge
- Examples:
Controllers, humans,
robots, software, ...
- Limited knowledge, computing re-
sources, perspective



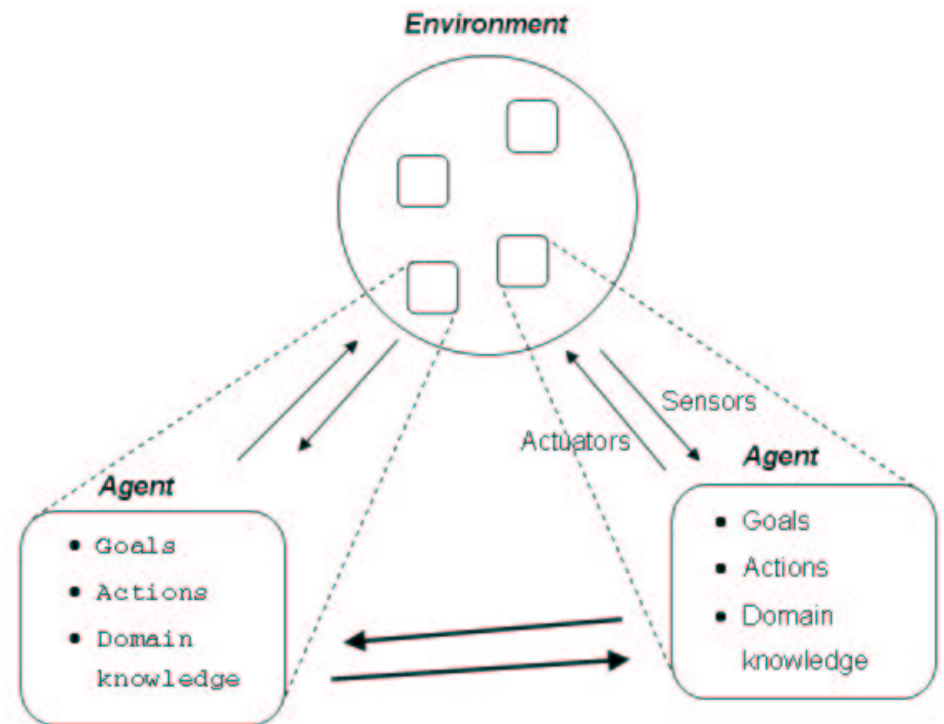
Single-Agent Systems

- One agent
- In general, the agent models:
itself,
environment,
interactions with env.
- Other agents not recognized
- Dynamics environment only affected by central agent



Multi-Agent Systems

- Multiple agents
- In general, agents model:
themselves, environment, env. interactions,
each other's goals/
actions, ...
- Dynamics environment
affected by all agents



Application Areas

- Open environments
 - Dynamic structure
 - Unknown components
 - Heterogeneous comp.
- Applications that are
 - Modular, decentralized,
 - changeable, ill-structured,
 - and complex
- Examples:
 - Internet, (air) traffic, power, social sciences, ...



Advantages

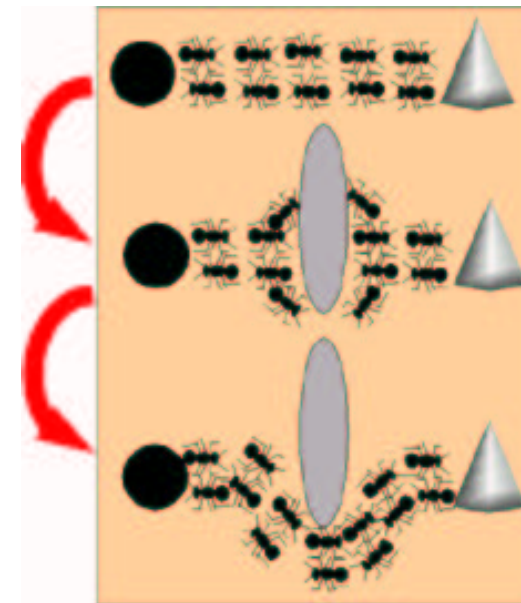
- Computational efficiency
- Reliability
- Scalability
- Robustness
- Maintainability
- Responsiveness
- Flexibility
- Reuse

Challenge

- Obtain coherence
 - No chaos, unpredictable behavior
 - Efficiency, quality, consistency global solution
- Difficulties
 - No global perspective
 - No global control
 - No global data
- How can coherence be increased?

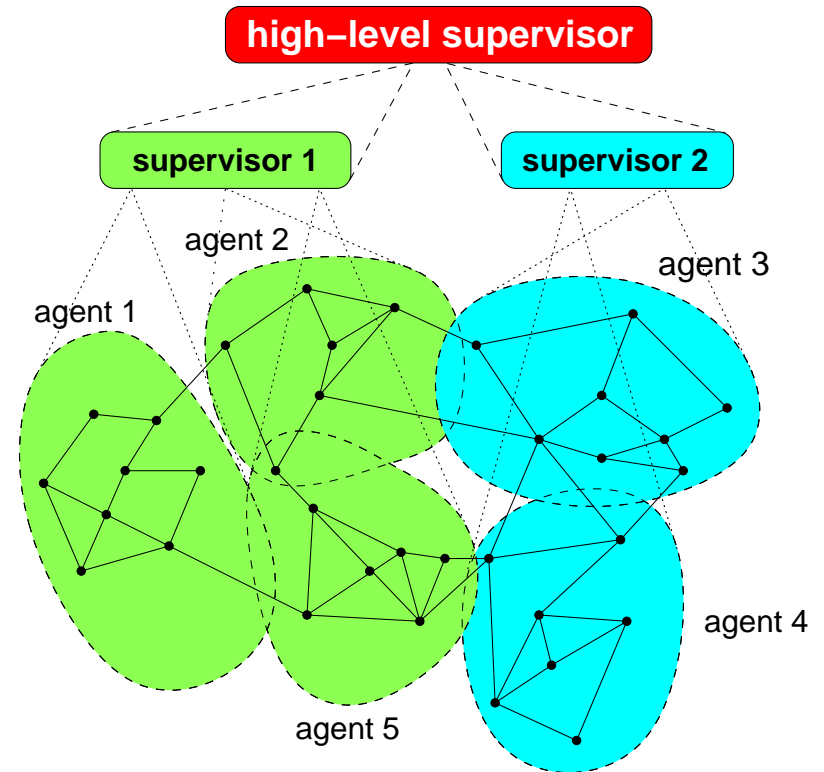
Individual Reasoning

- Deliberative
 - Symbolic reasoning
 - Reason about nonlocal effects, other agents, ...
- Reactive
 - React directly to input
 - Robust, fault-tolerant, fast
 - Possibly unpredictable/
unstable behavior
- Mix
 - Multiple layers
 - Reactive deliberation



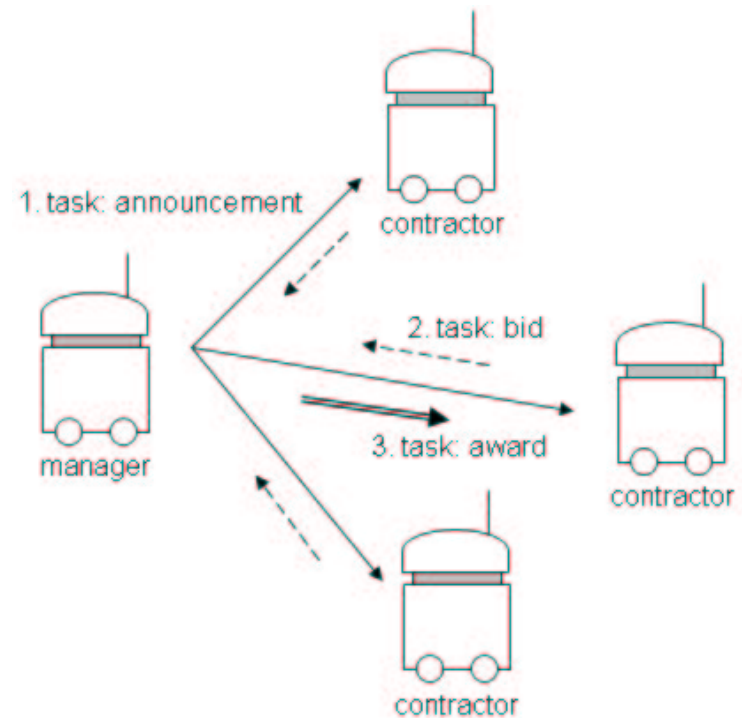
Organizations

- Roles, behavior expectations, authority relations
- Middle agents
- Organization theory
 - Hierarchy
 - Community of experts
 - Market
 - Scientific community
- Adaptivity important



Task Allocation

- Minimize task interdependencies
 - Improves problem solving efficiency
 - Improves solution consistency
- Static allocation
 - Nonadaptive
 - Limited & inflexible
- Dynamic allocation
 - Contract Net Protocol



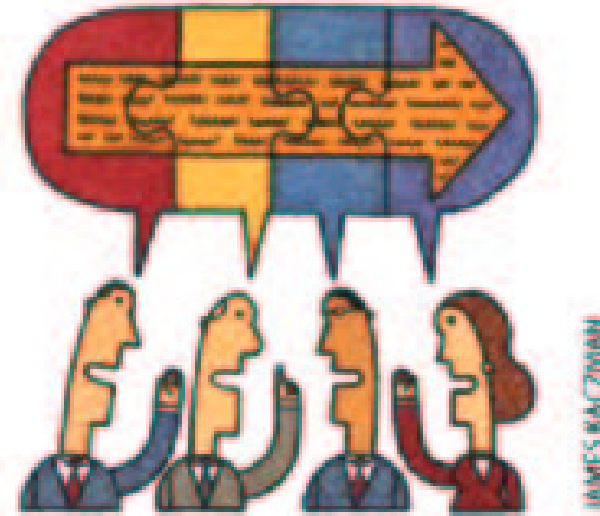
Planning

- Sequence of actions considering
 - own goals, capabilities, environmental constraints
 - other agents actions, commitments, unpredictability
- Individual-based
 - Complete planning before action
 - Partial Global Planning
- Team-based
 - Joint-intentions framework

Recognizing & Resolving Conflicts

- Disparities/inconsistencies in goals, plans, knowledge, beliefs, ...
- Approaches

Omniscient agent
Evidential reasoning
Constraint relaxation
Social norms
Negotiation



Modeling Other Agents

- Knowledge about or ability to model others

- Advantages

Future conflicts, actions, interpretations

Info about knowledge, beliefs, goals, ...

Flexibility increases

Unobservable behavior inferable

Managing Communication

- Plan content, amount, type, timing
- Abstractions decrease overhead
- Agent interoperability

Finding agents: Middle agents, matchmakers, ...

Communicating: KQML, Ontology problem

```
AGENT A
(ask-if
 :language      KIF
 :ontology      electronic
 :reply-with    q1
 :content (> (size chip1) (size chip2)))
```

```
AGENT B
(tell
 :language      KIF
 :ontology      electronic
 :in-reply-to   q1
 :content       true)
```

Managing Resources

- Effective allocation necessary
- Approaches
 - Operations research-based
 - Economics-based
 - Cloning-based
- Self-interested agents
 - Overuse
 - Untruthful



Adaptation & Learning

- Self-improvement of future behavior
- Necessary in complex environments
- Learning in MAS
 - Agents can influence one another
 - Agents can learn together
- Credit-assignment problem



Real-Time Control

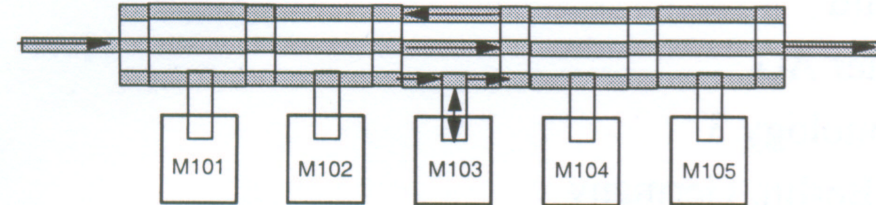
- Real-time control systems
 - Fast operation, semantically constraint
 - Real-time response required
- PID loop
 - Agent with 3 functions
 - Communication through shared environment

- Examples:

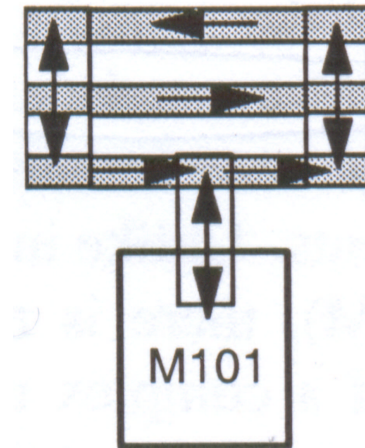
AMROSE	Robotic shipbuilding control	Links in segmented arm
MBCC	Building climate control	Thermostats, air duct dampers
Zone Logic	Handling/machining mech. parts	Transfer-line mechanisms
West	Manufacturing control	Individual devices, pieces

Real-Time Control: West

- Dynamic workpiece allocation
- Decentralized approach
 - Workpiece agent
 - Machine agent
 - Switch agent

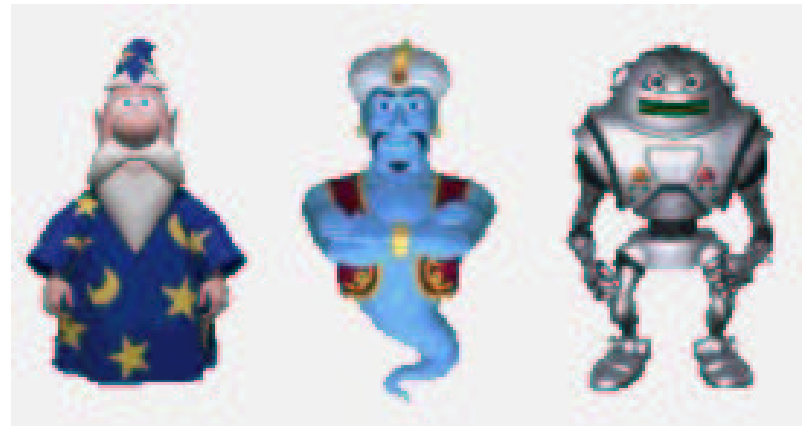


- Coordination
 - Workpiece-machine negotiation
 - Workpiece-switch negotiation
- Performance: 99.7%
of theoretical optimum



Conclusions

- Many challenging issues to solve
- What is needed are
 - Systematic methodologies
 - High-level MAS toolkits
- Public must trust agents



References

- G. Weiss, editor. *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*. MIT Press, USA, 1999.
- K.P. Sycara. Multiagent Systems. *Artificial Intelligence*, 19(2):79–92, 1998.
- P. Stone and M. Veloso. Multiagent Systems: A Survey from a Machine Learning Perspective. *Autonomous Robots* 8:345–383, 2000.
- S. Bussmann and K. Schild. Self-Organizing Manufacturing Control: An Industrial Application of Agent Technology. *Proc. of the 4th Intl. Conf. on Multi-Agent Systems*, USA, 2000.
- More on request.