The National Workshop on
Intelligent Systems and Their Applications
November 5-6, 2008

Intelligent Multi-Agent Systems

Prof. Hani Mahdi
Computer And Systems Engineering Department
Faculty of Engineering, Ain Shams University
Cairo, Egypt

Outline

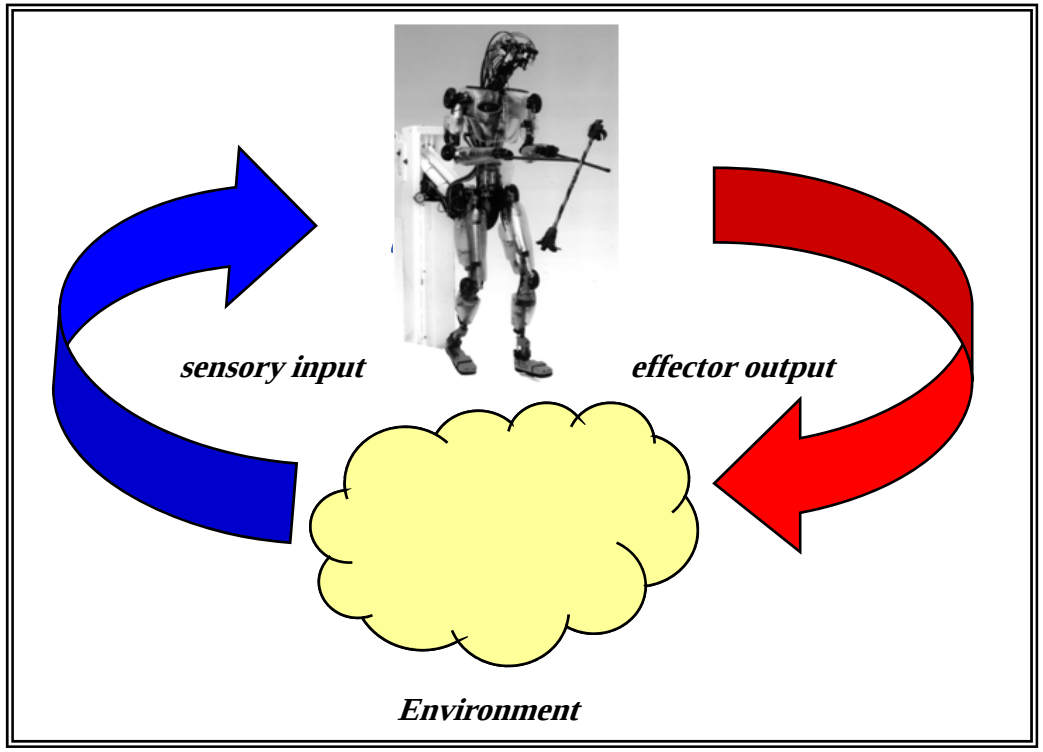
- Agents and Environments
- Foundation for Intelligent Physical Agents
- Multi-Agent Systems
- Challenging Issues

Background

- Mid 1990s, the agent concept has become important in both artificial intelligence (AI) and mainstream computer science.
- In 2000s agent-based and multi-agent systems (MASs) gain attentions beyond traditional computer science and artificial intelligence (AI).
- In 2002, Agent-Oriented Software Engineering (AOSE).

Agent Sense and Respond

- An **Agent** is anything that can be viewed as ***perceiving*** its **Environment** through ***sensors*** and ***acting*** upon that environment through ***actuators***



Some Agent Definitions

Merriam-Webster Online Dictionary:

1: one that acts or exerts power

2: something that produces or is capable of producing an effect **:** an active or efficient cause

3: a means or instrument by which a guiding intelligence achieves a result

4 : one who is authorized to act for or in the place of another:

→ Software Agent

Russel and Norvig: ”An agent is anything that can be viewed as perceiving its environment through sensors and acting upon that environment through effectors.”

Wooldridge and Jennings: “An agent is a hardware and/or software-based computer system displaying the properties of autonomy, social adeptness, reactivity, and proactivity

Other Agent Definitions

Coin: "Software agents are programs that engage in dialogs and negotiate and coordinate the transfer of information."

IBM: "Intelligent agents are software entities that carry out some set of operations on behalf of a user or another program with some degree of independence or autonomy, and in doing so, employ some knowledge or representations of the user's goals or desires."

Maes, Pattie: "Autonomous Agents are computational systems that inhabit some complex dynamic environment, sense and act autonomously in this environment, and by doing so realize a set of goals or tasks for which they are designed."

Agent Key Feature

There is a consensus that Autonomy, the ability to act without the intervention of humans or other systems, is a key feature of an agent.

Beyond that, different attributes take on different importance based on the domain of the agent.

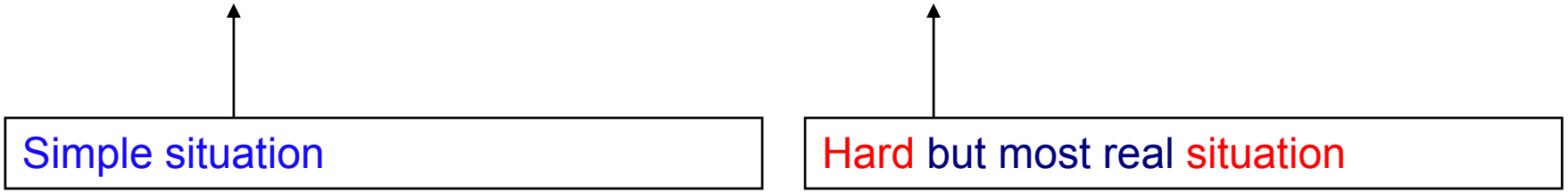
Stanford Racing Team Cars

2005

2007

Simple and Hard Environment

- Fully Observable (vs. Partially Observable)
- Deterministic (vs. Stochastic)
- Episodic (vs. Sequential)
- Static (vs. Dynamic)
- Discrete (vs. Continuous)
- Single agent (vs. Multiagent)



A Viewpoint

- A program deserves to be called a *Software Agent* only if it *exhibits a human agent behavior in at least one of properties of Hard Environment* (including Non-deterministic) properties and may go beyond human performance

Environments

To design a Rational Agent, we must specify its task environment.

PEAS – way to describe the Task Environment

- Performance Measure
- Environment
- Actuators
- Sensors

Example: Fully Automated Taxi

PEAS description of the environment:

- Performance Measure
  - Safety, destination, profits, legality, comfort
- Environment
  - Streets/freeways, other traffic, pedestrians, weather,, …
- Actuators
  - Steering, accelerating, brake, horn, speaker/display,…
- Sensors
  - Video, sonar, speedometer, engine sensors, keyboard, GPS, …

Main Agent Properties

- Ongoing, Non-terminating Software
- Internal Processes remain Encapsulated
- Capable of Autonomous action
- Guided by its own Goals

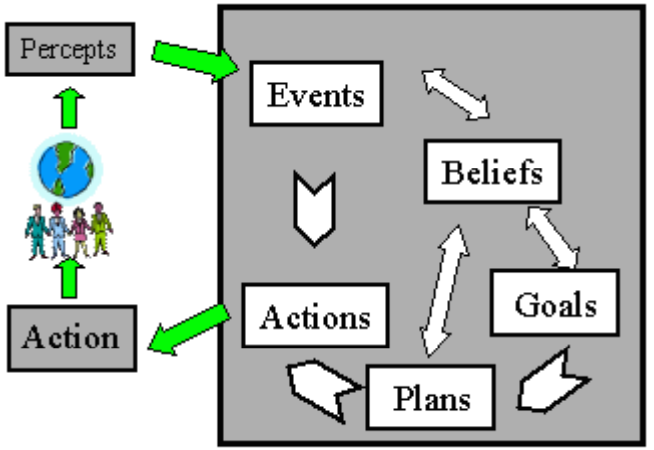
Main Intelligent Agent Properties

An Intelligent Software Agent is capable of operating in a non-deterministic environment.

- Responds in a Timely Manner
- No External Dictation
- May be Proactive - take the Initiative
- May have to possess Social ability
- May be able to Learn

Basic Agent Concepts

- **Actions** and **Percepts** interfacing with the environment.
- **Events** are for reactivity
- **Beliefs** are for achieving goals
- **Goals** are for proactivity
- **Planning** are for changing environments



The Structure of Agents

- The job of AI is to design the **Agent Program** that implements the Agent Function mapping percepts to actions.
  The **Agent Program** will run on some sort of computing device with physical sensors and actuators-we call this the **architecture**

- Agent = Architecture + **Program** .

Agent's Function and Program

- The difference between them
  the Agent Program takes the current percept as input, and
  the Agent Function takes the entire percept history.
- If the agent's actions depend on the entire percept sequence, the agent will have to remember the percepts.

Analyzing Tool!

- The Concept (Notion) of an Agent is meant to be a Tool for Analyzing Systems, NOT an absolute characterization that divides the world into Agents and Non-Agents.

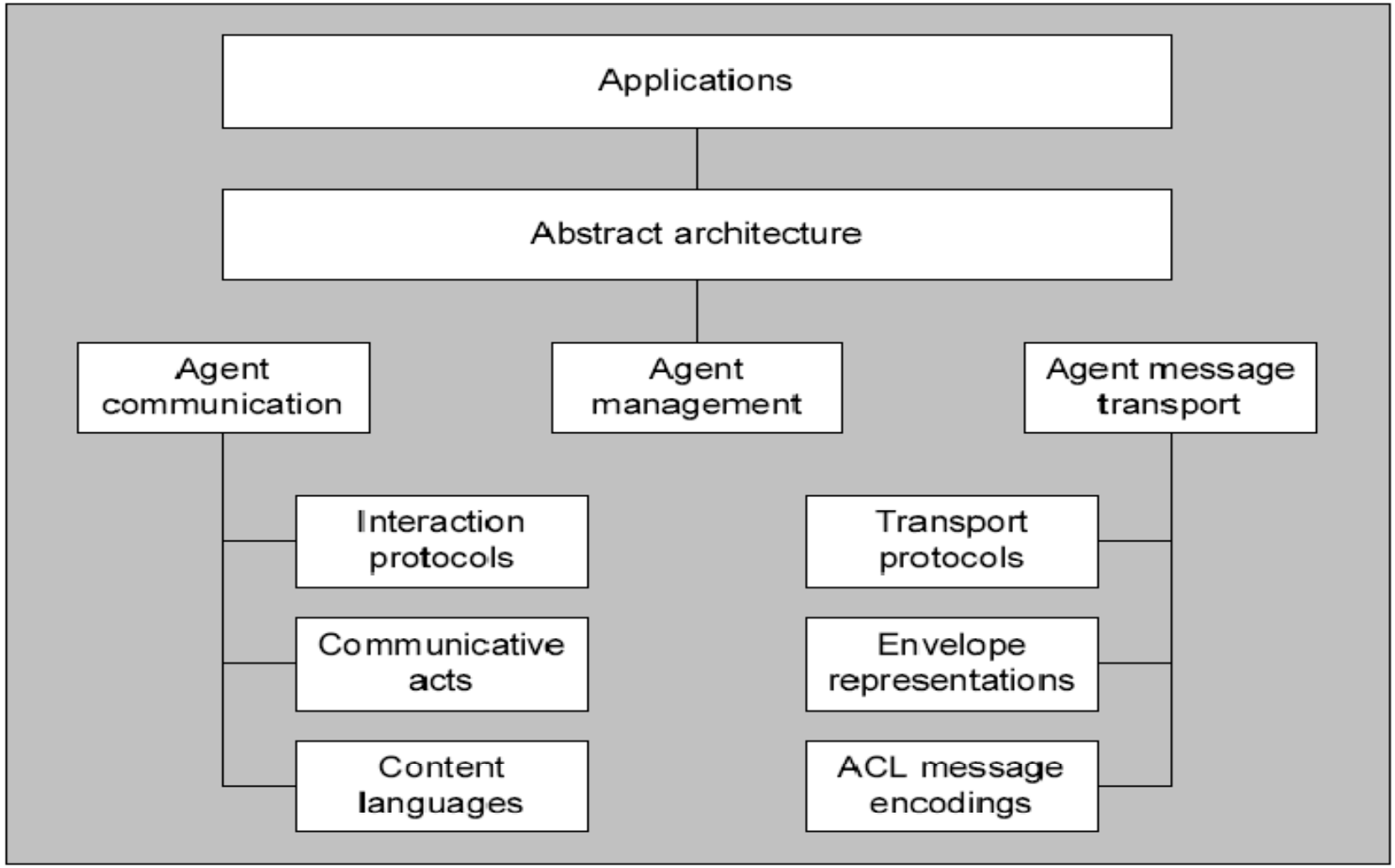
Misusing/Overusing the Name

- Search Agents, Report Agents, Presentation Agents, Navigation Agents, Management Agents, Help Agents, Smart Agents, User Agents, Collaborative Agents, Interface Agents, Mobile Agents, Information Agents, Hybrid Agents, Reactive Agents, …
  to name a few

FIPA Standards Overview

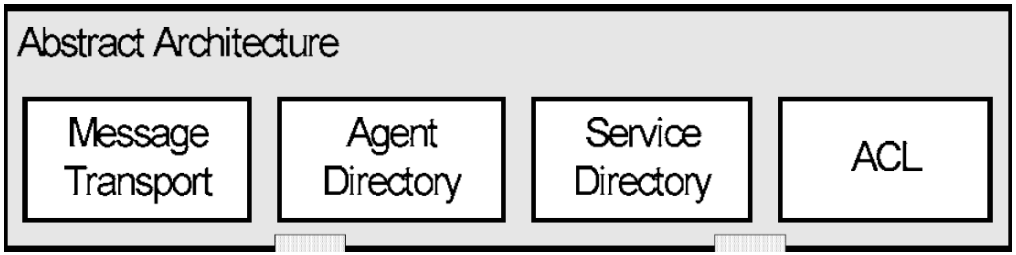
- Foundation for Intelligent Physical Agents
- FIPA - the Standards Organization for Agents and Multi-agent Systems (1996)
- Developing Specifications supporting inter-operability among agents and agent-based applications.
- FIPA was officially accepted by the IEEE as its eleventh standards committee on 2005.

Overview of the FIPA Standards



FIPA Abstract Architecture Scope

- A model of Services and discovery of services available to agents
- Message Transport interoperability
- Supporting various forms of ACL (Agent Communication Language) representations and Content Languages
- Supporting the representations of multiple Directory Services



Abstract and Concrete Specs

- From Abstract to Concrete Specifications
- Abstract Architecture cannot be implemented
- Abstract Architecture forms the basis for the development of Concrete Architectural Specifications.

Concrete Specifications

- Concrete Specifications describe how to construct an agent system, including the Agents and the Services that they rely upon, in terms of Concrete Software Artefacts, such as
  - Programming Languages,
  - Applications Programming Interfaces,
  - Network Protocols,
  - Operating System Services,
  - ...

FIPA Compliant

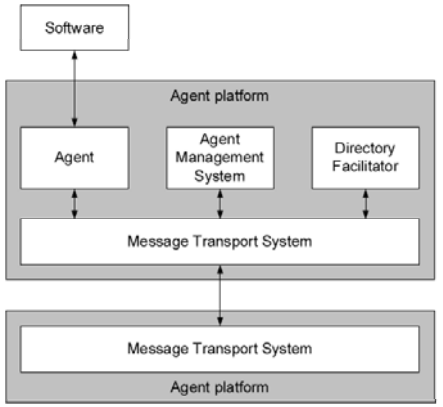
- A Concrete Architectural Specification must have certain Properties to be FIPA Compliant.
- The Concrete Architectural must include Mechanisms for
  Agent Registration
  Agent Discovery and
  Inter-Agent Message Transfer.
- These Services must be explicitly described in terms of the corresponding elements of FIPA Abstract Architecture.

Minimum Required Elements

- FIPA Abstract Architecture does not *prohibit* the introduction of elements useful to make a good agent system, it merely sets out the *Minimum* Required Elements.

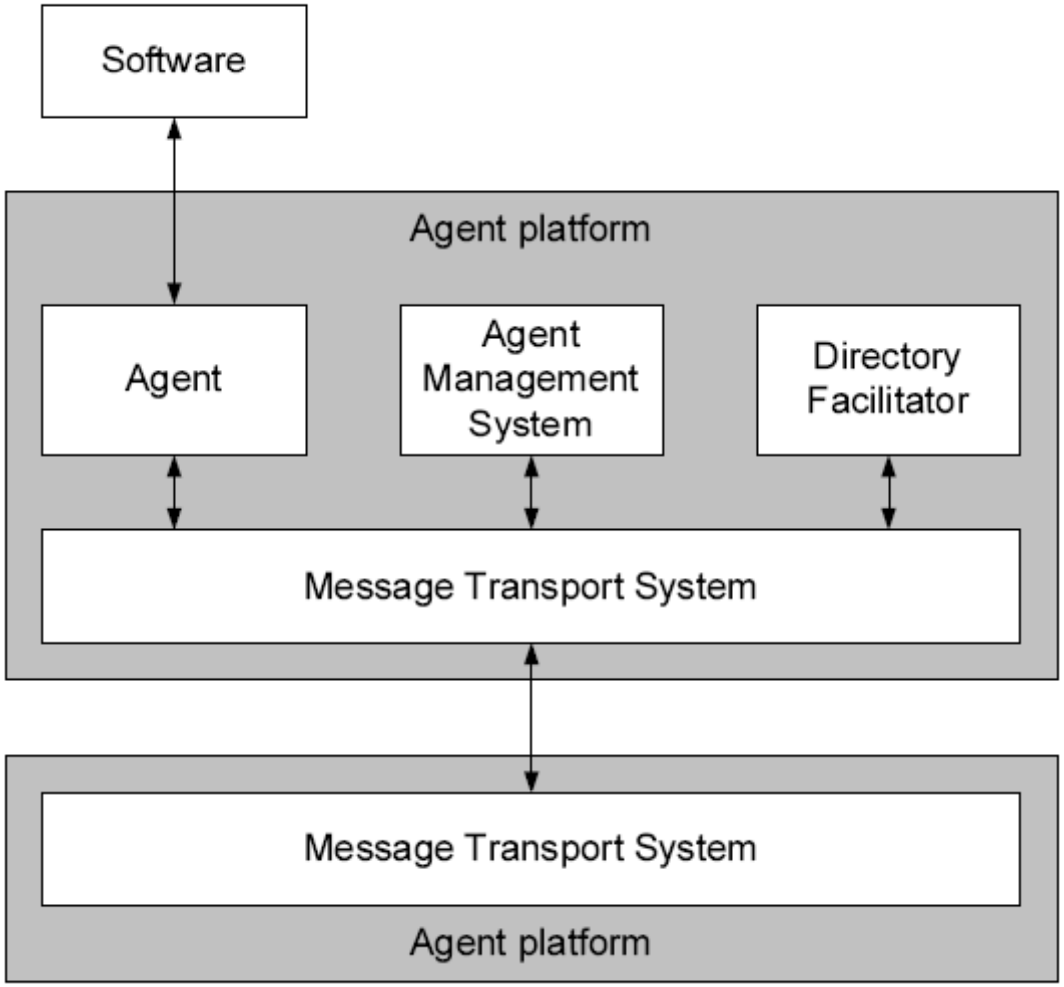
The Reference Model

The Agent Management Reference Model consists of the following logical components



- An Agent
- An Agent Platform (AP)
- A Directory Facilitator (DF)
- An Agent Management System (AMS)
- A Message Transport Service (MTS)

Management Reference Model



Agent Communications Language

Two popular and acknowledge ACLs :

- FIPA-ACL (by the Foundation for Intelligent Physical Agents, a standardization consortium)
- KQML (Knowledge Query and Manipulation Language) a language and protocol for communication among software agents and knowledge-based systems

Multi-Agent Systems (MASs)

Informal definitions:

- An Agent is a computer program and an architecture those capable of flexible and autonomous action in a Dynamic Environment, usually an Environment Containing Other Agents.

  The Society in which they operate is called a Multi-Agent System.

MASSIVE

Software

Agents on film…

“Fellowship of the Ring” battle scenes by Weta digital.

Motivations for MAS construction

In a MAS, the agents co-operate to perform some task that a single agent can't do on its own because

- A Single Agent doesn't have all the Capabilities or Knowledge required to perform the task
- A Single Agent would be Too Slow

Multi-Agent System - Viewpoint

- Like the notion of an ‘Agent’,
  a ‘Multi-Agent System’ is an Analysis Tool:

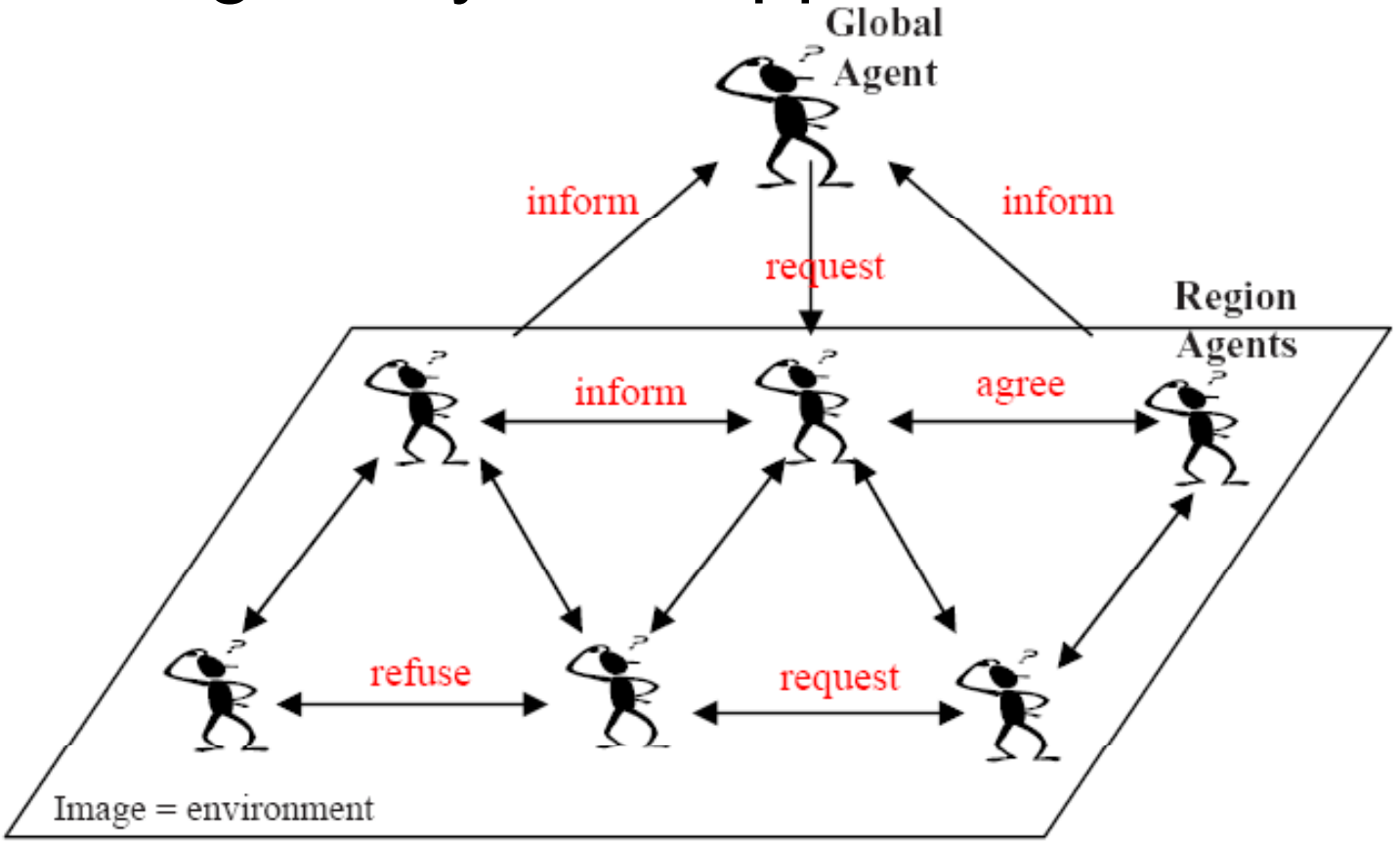
- It is worthless trying to identify precisely which systems are really Multi-Agent Systems
- the Key Point is whether we gain by looking at a System as a Multi-Agent System.

Perspectives of MAS

- **Sociological** Perspective
  Members of an Agent Society
- **AI** Perspective
  Intelligent and Autonomous Behavior
- **Economic** Perspective
  Economically Rational behavior
- **Application** Perspective
  Examples*:* Mechanical Engineering, Robotics, Production Planning, Control, …

An Example

- Medical Image Segmentation by a Multi-Agent System Approach



MAS and DAI

- Distributed Artificial Intelligence (DAI) is a subfield of Artificial Intelligence (AI) which is concerned with a society of problem solvers or agents interacting in order to solve a common problem

Three Broad Areas which fall under DAI,

- Multi-Agent System (MAS)
- Distributed Problem Solving (DPS), and
- Parallel AI (PAI).

Key Properties of a MAS

- The Key Properties of MAS are

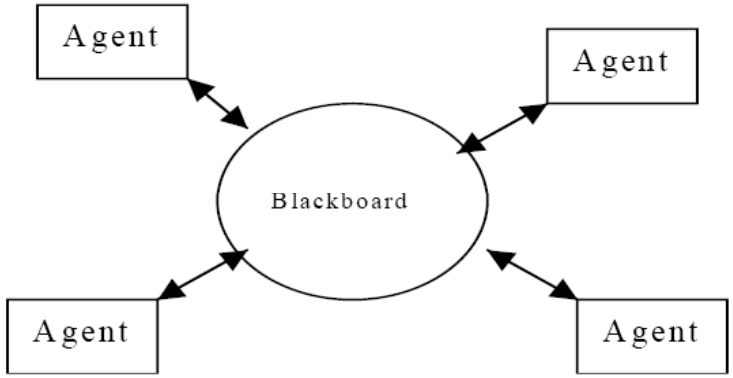
  Communication and Interaction
  Structures and Roles

Agent Communication

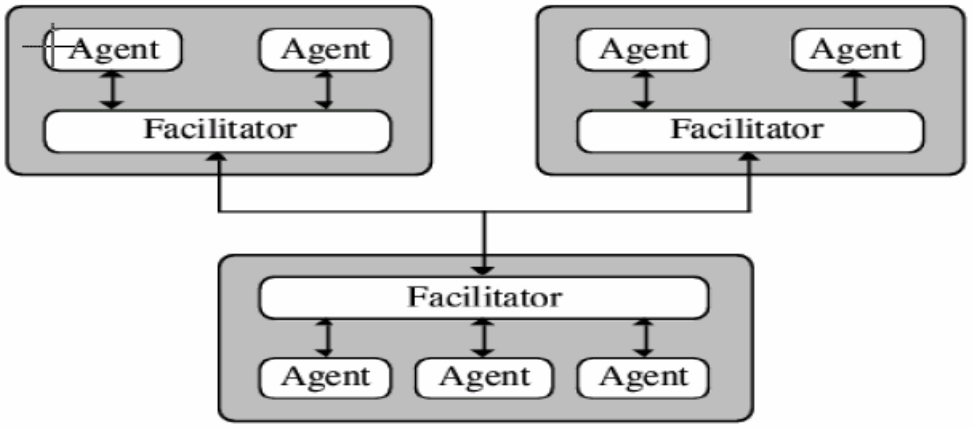
- Agent Communication
  encourages autonomy and
  the existence of societies
  avoids the regard of other agent's internal structure
- Different Agents (interface, programmers, developed language, platform, …) a Common Communication Language for the Information Exchange.

Communication Architectures

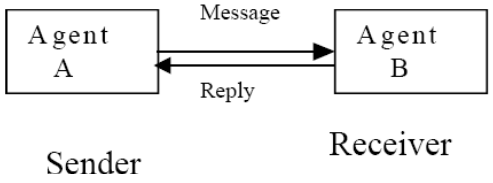
Three Communication Architectures are commonly used



Blackboard



Facilitation Architecture

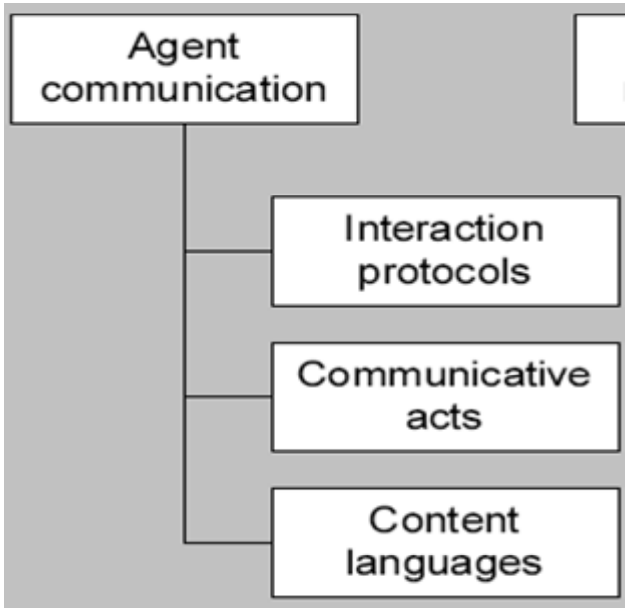


Message Architecture

Communication Sub-Sections

Three Sub-Sections

- Interaction Protocol: Intended High Level Strategy; such as negotiation schemes, auctions, contracts, etc.
- Communication Language: Used to exchange the contents of the messages.
- Transport protocol: the transportation mechanism; such as TCP, IP, SMTP, HTTP.



MAS Architectures

- Several Architectural Styles have been used in the development of Multi-agent Systems.

Four Known Organizations:

- Hierarchical Architecture
- Flat Architecture
- Subsumption Architecture
- Modular Architecture

Hierarchical Architectures

- agents communicate according to a hierarchical structure, such as a tree.
- The root is known as a broker or facilitator agent.
- Disadvantage: Reduce
  the Autonomy of The Individual Agents.
- Advantage: Reduce
  the amount of Required Communications
  the Complexity and Reasoning capabilities

Flat Architecture

- any agent may contact any of the others.
- Disadvantage: Increase
  Communications between Agents
  Needs the locations of their partners or
  Agent Location Mechanisms such as
  yellow pages services
- Advantage: Increase
  Agent Autonomy
- Suitable for Small Multi-agent Systems

Subsumption Architecture

- Agents are Themselves made up of other Agents.
- the subsumed agents are completely controlled by the containing agents.
- Disadvantage: Reduce
  the Flexibility of the system
- Advantage: Increase
  The Efficiency of the system

Modular Architecture

- A Modular Multi-Agent System is comprised of a Number of Modules.
- Each module normally employs a Flat Structure, while inter-module communications is relatively limited.
- It is Advantageous for Specific Applications where the agents can be grouped according to their inter-communications

Challenging Issues - 1

Challenging Issues in MASs:

- How to Decompose a Problem, Allocate Subtasks to agents, and Synthesize Partial Results.
- How to handle the Distributed Perceptual Information. How to enable agents to maintain consistent shared models of the world.
- How to Implement Decentralized Control and build Efficient Coordination Mechanisms among agents.

Challenging Issues - 2

- How to Design Efficient Multiagent Planning and Learning Algorithms.
- How to Represent Knowledge. How to enable agents to Reason About the Actions, Plans, and Knowledge of other Agents.
- How to Enable Agents to Communicate. What Communication Languages and Protocols to use. What, when, and with whom Should an Agent Communicate.

Challenging Issues - 3

- How to Enable Agents to Negotiate and Resolve Conflicts.
- How to enable agents to Form Organizational Structures like teams or coalitions. How to Assign Roles to agents.
- How to Ensure Coherent and Stable System Behavior.

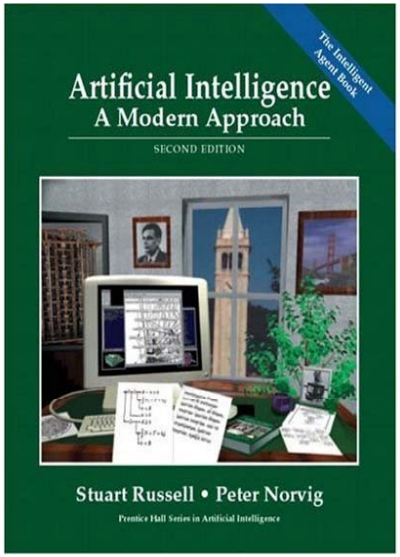
Traditional Textbook

- Artificial Intelligence: A Modern Approach (AIMA), Second Edition, Stuart Russell & Peter Norvig, Pearson Education, 2003, ISBN 0-13-790395-2.

  http://www.cs.berkeley.edu/~russell/aima

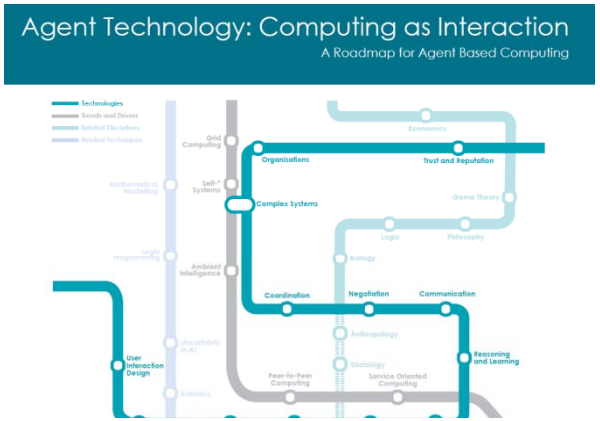
- AIMA repository, /afs/cu/class/cs451/aima

  Lisp, Java, Python

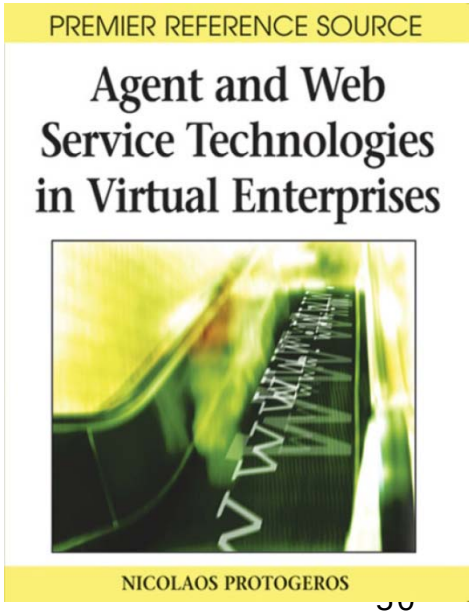


References - 1

- Agent Technology: Computing as Interaction
  A Roadmap for Agent-Based Computing
  Compiled, written and edited by
  M. Luck, P. McBurney, O. Shehory, S. Willmott
  and the AgentLink Community
  2005

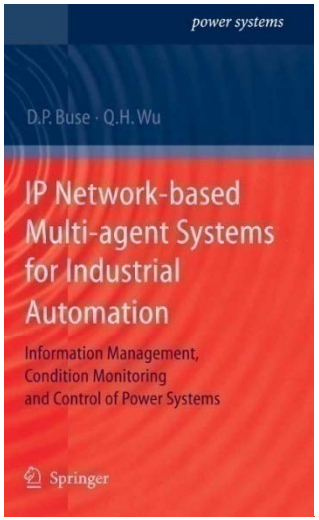


- Software Agent Technology: An Overview
  C. E. Georgakarakou and A. A. Economides
  in
  Agent and Web Service Technologies in Virtual Enterprises
  Nicolaos Protogeros, editor
  IGI Global, 2008

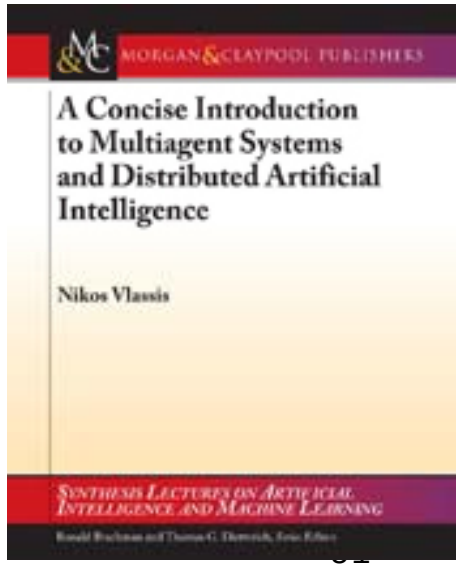


References - 2

- IP Network-based Multi-agent Systems for Industrial Automation D.P. Buse and Q.H.Wu Springer-Verlag 2007

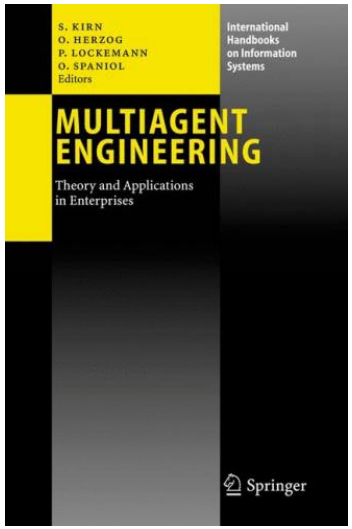


- A Concise Introduction to Multiagent Systems and Distributed Artificial Intelligence Nikos Vlassis Morgan & Claypool 2007



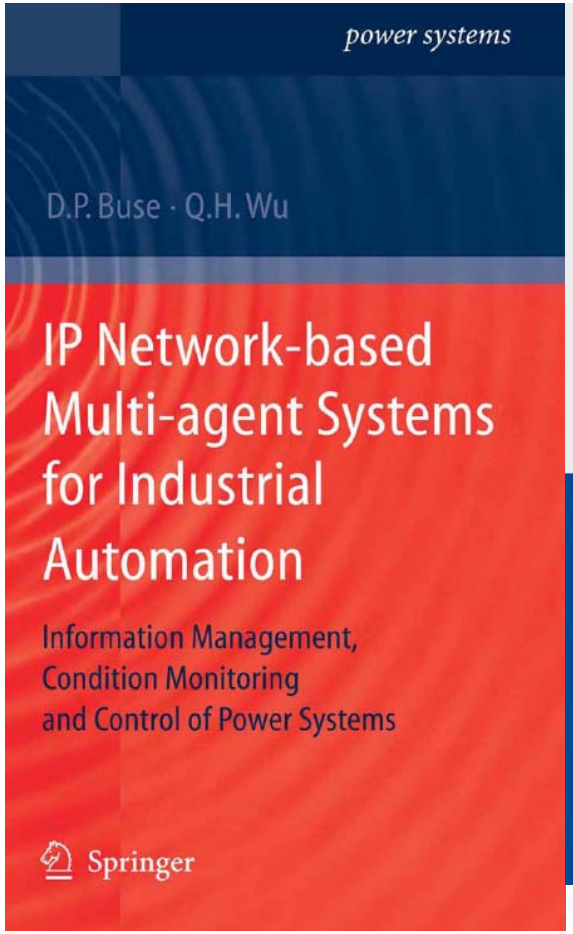
References - 3

- Multiagent Engineering:
  Theory and Applications in Enterprises
  By Stefan Kirn, Otthein Herzog,
  Peter Lockemann, Otto Spaniol,
  Springer 2006

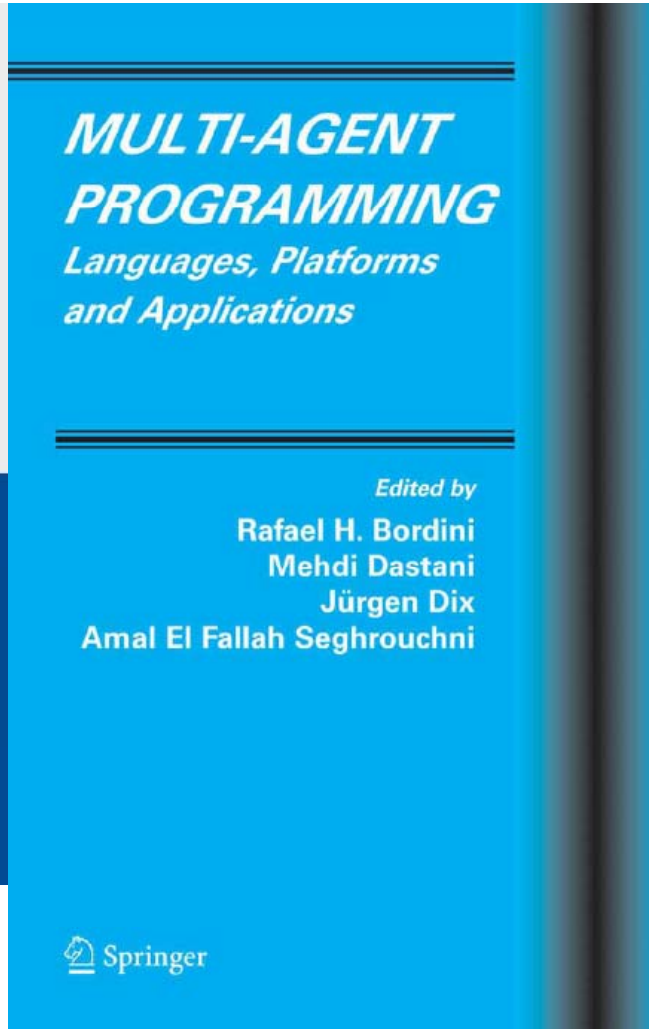


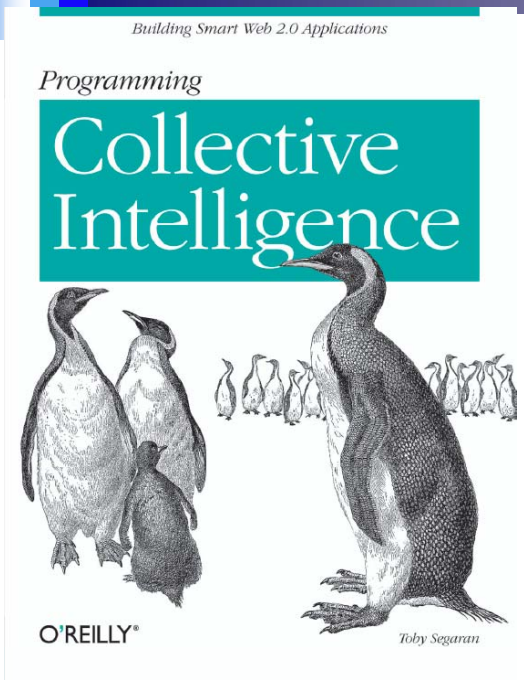
- **S**implifying **A**gent **C**oncepts (SAC): Survey of Agent Systems
  http://www.cs.rmit.edu.au/agents/SAC/
  http://www.cs.rmit.edu.au/agents/SAC2/index.html
- Foundation for Intelligent Physical Agents (FIPA)
  http://www.fipa.org/index.html

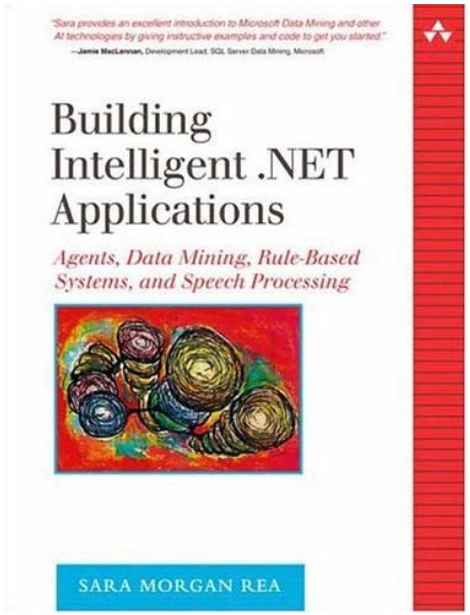
Textbooks

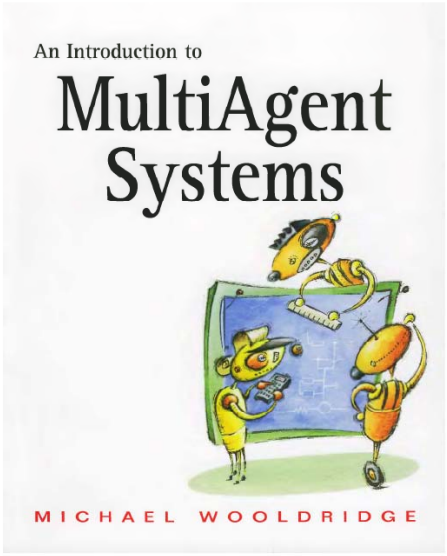

power systems
D.P. Buse · Q.H. Wu
IP Network-based Multi-agent Systems for Industrial Automation
Information Management, Condition Monitoring and Control of Power Systems
Springer

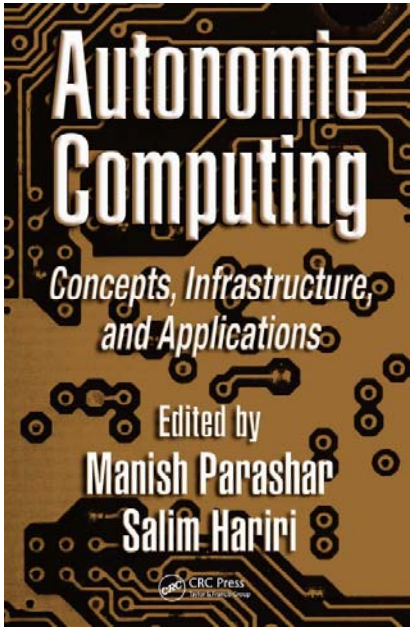

Whitestein Series in Software Agent Technologies
Roland Zimmermann
Agent-based Supply Network Event Management
Birkhäuser

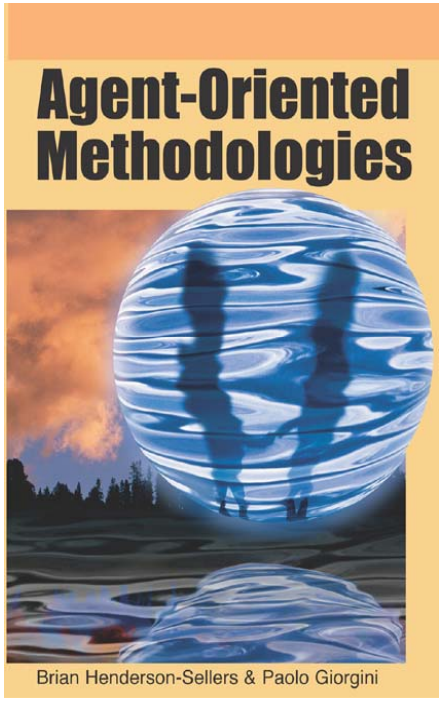

MULTI-AGENT PROGRAMMING
Languages, Platforms and Applications
Edited by
Rafael H. Bordini
Mehdi Dastani
Jürgen Dix
Amal El Fallah Seghrouchni
Springer

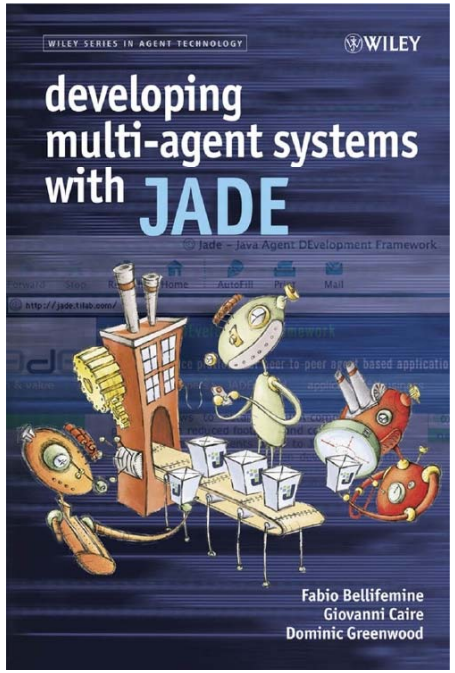

Building Smart Web 2.0 Applications
Programming
Collective Intelligence
O'REILLY
Toby Segaran


Building Intelligent .NET Applications
Agents, Data Mining, Rule-Based Systems, and Speech Processing
SARA MORGAN REA


An Introduction to
MultiAgent Systems
MICHAEL WOOLDRIDGE


Autonomic Computing
Concepts, Infrastructure, and Applications
Edited by
Manish Parashar
Salim Hariri
CRC Press


Agent-Oriented Methodologies
Brian Henderson-Sellers & Paolo Giorgini


WILEY SERIES IN AGENT TECHNOLOGY
WILEY
developing multi-agent systems with JADE
Fabio Bellifemine
Giovanni Caire
Dominic Greenwood

Thank You