Multi-Agent Systems

Thomas Moser

January 14, 2010

Institut für Softwaretechnik & Interaktive Systeme

Selbstorganisierende Systeme WS09/10

Folien übertragen[1] in Latex / Lyx. - modifiziert/gekürzt und möglicherweise mit Fehlern

Contents

[1] von Georg Regal

1 Introduction to Multi-Agent Systems

- MAS = distributed artificial intelligence system which embodies a number of autonomous agents within the same environment to achieve common goals
  - "environment" covers physical environments for robotic agents as well as runtime environments for software agents, ...
- MAS can be used to solve problems which are difficult or impossible for an individual agent or monolithic system to solve
  - E.g. online trading, disaster response, modelling social structures, ...

1.1 Characteristics of „Intelligent" Agents

- capable of acting in an environment
- communicate directly with other agents driven by a set of tendencies
- possess resources, skills and can offer services
- capable of perceiving their environment (only partial)
- may be able to reproduce themselves
- behaviour tends towards satisfying objectives
- Autonomy: the agents are at least partially autonomous
- Local views: no agent has a full global view of the system
- Decentralization: there is no one controlling agent

1.2 Acting Cycle of intelligent Agents

- Agent starts reasoning process by interpreting arriving information (perceive)
- by combining this information with the existing knowledge and specified goals, the agent infers and selects possible actions (infer, select)
- Agent executes inferred and selected action (act) and changes the state of the environment

1.3 Agent-Oriented Software Engineering (AOSE)

- "An Agent is a delimitable (software-/hardware-) unit which is in the position to trace the given duties in a flexible, interactive and autonomic way."
- using agents for solving special problems has roots in the (distributed) artificial intelligence

	OOP	AOP
Structural Elements		
	abstract class	generic role
	class	domain specific role
	class variables	knowledge, belief
	methods	capabillities
Relations		
	collaboration (uses)	negotiation
	composition (has)	holonic agents
	inheritance (is)	role multiplicity
	instantiation	domain-specific role + individual knowledge
	olymorphism	service matchmaking

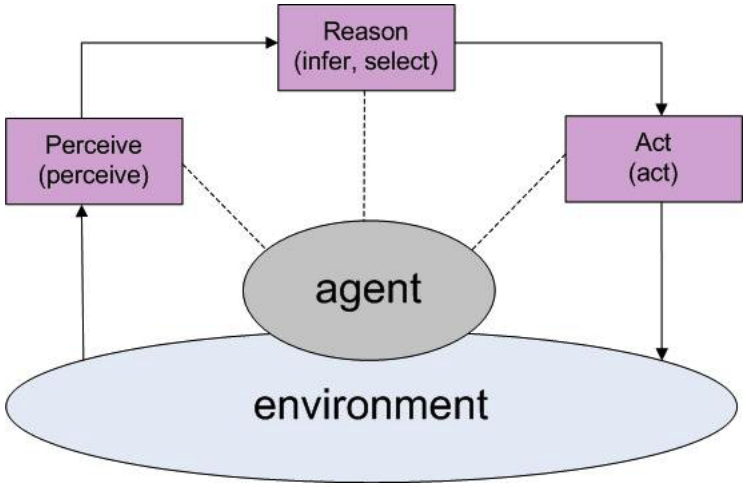

Figure 1: Acting Cycle

1.4 FIPA Standard

- "Foundation for Intelligent Physical Agents"
  - IEEE Computer Society standards organization (since 2005)
  - promotes agent-based technology and the interoperability of its standards with other technologies
  - www.fipa.org
- Originally Swiss organization (founded 1996)
  - software standards specifications for heterogeneous and interacting agents and agent based systems
- specification circle for development of agent technology

1.5 FIPA Agent Management Reference Model

- Agentmanagementsystem – is responsible for the administration of all agents
- MessageTransportSystem – is responsible for the transport of an agent communication language (ACL) message
- DirectoryFacilitator – is responsible for the administration of available services (comparable to a UDDI registry)
-

2 Agent Languages

- Agents in MAS need more then just communication to transmit information
- Agents achieve actions or make decisions which are affected by the attendance or the knowledge of other agents
- Agents need to interpret the semantics of the transmitted communicated information
- Act-of-Speech Theory
  - "While he makes this declaration, the speaker says not only something, but he does at the same time also something. The world changes with this statement, as for example <<I christen this ship with the name Queen Mary>>."

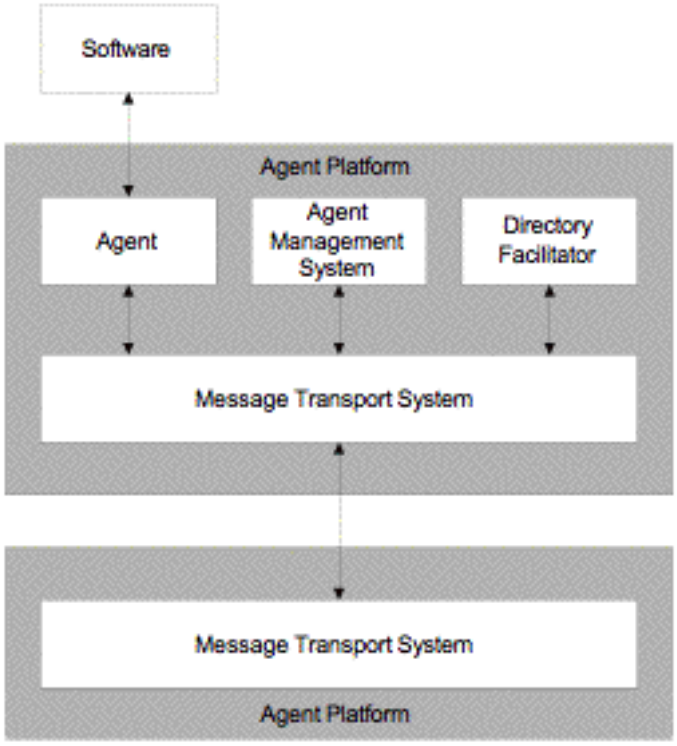

Figure 2: FIPA

2.1 Knowledge Query and Manipulation Language (KQML)

- originally developed as an interface for knowledge based systems in the early 1990's year as part of the DARPA (Defense Advanced Research Projects Agency)knowledge sharing effort – http://www.cs.umbc.edu/kqml/
- Language and protocol for communication among software agents and knowledge-based systems
- Specifies message format and message-handling protocol to support run-time knowledge sharing among agents
  - Defines the permissible operations that agents may attempt on each other's knowledge
  - Provides basic architecture for knowledge sharing

2.2 FIPA Agent Communication Language (ACL)

- FIPA standard language for agent communication relying on the act-of-speech theory
  - realized by a set of performatives and their meaning
  - content of the performatives is not standardized
- mechanism to execute an act of speech is like the sending of a message coding this act of speech
  - Receiver of the message can use the message together with his knowledge to interpret the transmission and act accordingly (agents need to have common specific knowledge of the required semantics)

- "The performative admits conclusions on the state of the internal model of the transmitter agent and the consequences expected by sender on the state of the internal model of the receiver's agent. Because both agents are autonomous, there is no guarantee that the expected consequences really arrive."

2.3 Application Areas for MAS

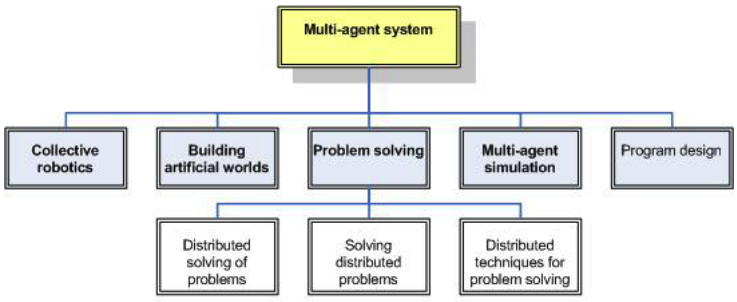

Figure 3: Application Area for MAS

2.4 Collective robotics

- assembly of robots acting in cooperation in order to reach a goal
- **"cellular" robotics**
  - construction of robots on a modular basis
    - each module of the robot is considered as a part of the multi-agent system and regarded as an agent
  - movement is result of the coordinative work of the agents building up the robot to fulfil the actual needed movement
- **mobile robotics**
  - uses at least two robots, which coordinate their movements to accomplish their tasks
  - MAS consists of robots which can be interpreted as agents

2.5 Building artificial worlds

- construction of synthetic worlds makes it possible to analyse certain interaction mechanisms in a more detailed way than a real application could do it
- constructed worlds can be manipulated with different parameters to create the demanded environment
  - allows to analyze interesting behaviours and situations which correspond to the real world without impacts from the outside
- "Social Simulation"
  - modeling or simulation of social phenomena (e.g., cooperation, competition, markets, social networks dynamics, etc.)
  - John Conway: Game of Life

2.6 MAS Problem Solving

- software agents accomplish tasks of use to human beings
- – contrast to robotics: agents are purely computing agents and have no real physical structure
- distributed solving of problems
  - all agents have to act in cooperation
  - like specialists for their own area
  - to solve the problem
- solving distributed problems
  - agents concerned do not have to but can have similar skills because the problem itself is distributed among them
- distributed techniques for problem solving
  - agents are used in interaction to solve problems in the classical way (find solution for problem which has been well formulated)
  - the MAS approach can impose a new mode of reasoning by breaking down the problem in a totally different way

2.7 MAS Simulation

- analysing the properties of theoretical models for real-life scenarios
  - try to explain or forecast problems by testing by running the constructed or designed models in a virtual world
  - compare and interpret the calculated results or phenomena
- E.g., simulation of production processes
  - MAS simulation allows a relatively simple reproduction of existing or planned to build assembly lines
  - Provides a practicable way to identify possible problems of the production plant design or improve the construction of the needed assembly line to optimize the throughput and production capacities

2.8 Program design

- Design computing systems which try to interact, adapt and reproduce by using relatively autonomous agents functioning in physically distributed environments (AOSE)
- New technologies for creating software can be based on the concepts of agents and their interaction principles among them
  - each program unit can take the form of an agent which has its own autonomy and its own objectives and which is embedded into the network and cooperates or negotiates with other units within it
  - users feed the network with various instructions to get out the requested information without further interaction

3 JADE

3.1 JADE History

- MAS software framework fully implemented in Java
  - Developed by Telecom Italia (started in 2000), the copyright holder, in open source software under the terms of the LGPL
  - Launch of JADE governing board in March 2003
    - Telecom Italia
    - Motorola
    - Whitestein Technologies
    - Profactor
    - France Telecom
  - Goal: JADE should become "standard middleware for mobile Peer-To-Peer intelligent agent applications"
  - http://jade.tilab.com/
    - latest version JADE 3.6.1

3.2 What is JADE?

- A middleware for the development and run-time execution of peer-to-peer intelligent-agent applications
- Runs seamlessly in mobile and in fixed environments
- Enables
  - Multi-party applications
  - Pro-activity
  - The Machine-to-Machine paradigm

3.3 JADE Key Characteristics

- simplifies the implementation of multi-agent systems through a middle-ware that complies with the FIPA specifications and through a set of tools that supports the debugging and deployment phases
- agent platform can be distributed across machines (which not even need to share the same OS) and the configuration can be controlled via a remote GUI
- configuration can be even changed at run-time by moving agents from one machine to another one

3.4 JADE Services

- JADE is an agent platform that implements the basic services and infrastructure of a distributed multi-agent application
  - agent life-cycle and agent mobility
  - white & yellow-page services
  - peer-to-peer message transport & parsing
  - agent security
  - scheduling of multiple agent tasks

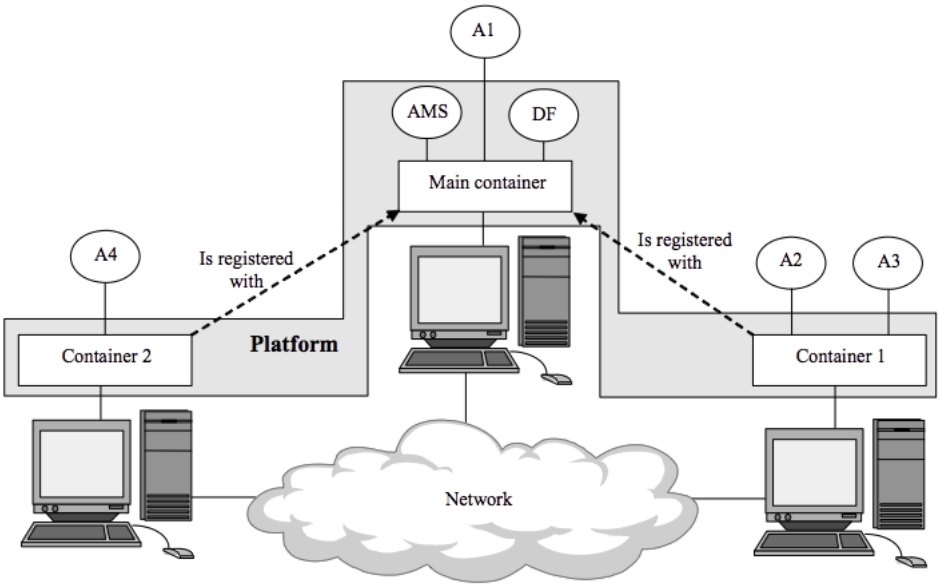

Figure 4: JADE-Archtitektur

  - set of graphical tools to support monitoring, logging, and debugging

- JADE Hides FIPA From Programmers!
- No need to implement the Agent Platform – Agent Management System (AMS) and Directory Facilitator (DF) executed at start-up
- No need to implement agent-management functionalities
  - An agent is registered with the AP within its constructor
  - The DF Service class provides a simplified interface to access the services of the DF (registration, lookup, . . .)
- No need to implement Message Transport and Parsing
  - Automatically (and possibly efficiently) done by the framework when sending/receiving messages
- Interaction Protocols must only be extended via handle methods
- AND it is standard FIPA

3.5 JADE Architectural Model (siehe 4)

- A JADE-based application is composed of a collection of active components called Agents
- Each agent has a unique name
- Each agent is a peer since he can communicate in a bidirectional way with all other agents
- Each agent lives in a container (that provides its run time) and can migrate within the platform
- One container plays the role of main (where AMS, DF live)
- The main-container can be replicated via replication service
-

3.6 The main graphical Tools of JADE

- RMA (Remote Monitoring Agent)
  - monitor and control the platform and all its remote containers
  - remote management of the life-cycle of agents
  - compose and send a custom message to an agent
  - launch the other graphical tools – monitor (just read operations) other FIPA-compliant platforms
- Dummy Agent
  - compose and send custom messages
  - load/save the queue of messages from/to a file
- Sniffer Agent
  - display the flow of interactions between selected agents
  - display the content of each exchanged message
  - save/load the flow to/from a file
- Introspector Agent
  - Monitoring agent internal state
    - received/sent/ pending msg
    - Scheduled behaviours (active, blocked) and subbehaviours
    - agent state
  - Debugging execution
    - step-by-step
    - Slowly
    - break points
- Log Manager Agent
  - GUI to modify at run-time the logging of the platform
  - based upon java.util.logging
- DF (Directory Facilitator) GUI
  - GUI of the yellowpage service
    - browse, register, deregister, modify, search agent descriptions
    - federate with other DFs
    - execute federated searchs

3.7 JADE Tutorials

- http://www.ifs.tuwien.ac.at/files/JADETutorial_Programming.pdf
- http://jade.tilab.com/doc/index.html
- Jade API doc: http://jade.tilab.com/doc/api/index.html

: JADE Tutorial und SAW wurden nicht aus den Folien übertragen

4 Summary

- MAS: distributed artificial intelligence system containing a number of autonomous agents within the same environment to achieve common goals
- Foundation for Intelligent Physical Agents (FIPA)
  - FIPA ACL: standard language for agent communication
- Java Agent Development Framework (JADE)
  - middleware for the development and run-time execution opeer-to-peer intelligent-agent applications
  - JADE Hides FIPA From Programmers
- Simulation of Assembly Workshops (SAW)
  - Multi-Agent Based Simulation of Production Processes
  - Research Project @ IFS