Multiagent Systems

Franco Zambonelli
March 2014

Outline

- Multiagent Systems
  - Motivations
  - Multiagent systems vs. Object and Service Systems
  - Applications
- Agent Communication and Coordination
  - Coordination and Communication models
  - Collaboration and Competition
  - Negotiation
- Agent Infrastructures
  - The JADE Infrastructure
  - The TUCSON and MARS Infrastructures
- Conclusions and Open Issues
Part 1

- Multiagent Systems

Multiagent Systems

- System or “organizations” of interacting autonomous agents
- Possibly distributed
  - over multiple computers and/or
  - in an environment
- Possibly belonging to different stakeholders or different organization
  - Composite multiagent systems
  - Open applications
- Collaborating and/or Competing
  - Striving at a shared global goal
  - Self-interested agents striving at maximising their own utility
  - According to various topologies of interaction networks
- Each Interacting with an environment
  - And possibly indirectly interacting with each other via the mediation of the environment
Characterization of a MAS

Why Multiagent Systems? (1)

- A single agent has finite rationality
  - Limit on the amount of knowledge he can rationally handle in a given time
  - Require splitting a decision between more agents
  - E.g., an agent in charge of analyzing a very large search space

- Distribution
  - Several problems are intrinsically distributed
  - Knowledge can be acquired only in loco
  - Require more agents to be allocated where the knowledge can be obtained
  - E.g., agents devoted to the control of physical processes must monitor in loco
Why Multiagent Systems? (2)

- Interactions between personal agents
  - Many problems require components of different stakeholders/organizations to interact
  - So, the problem is intrinsically composed of multiple agents
  - E.g., an E-commerce site requires both the buyer and the seller, which will be represented by two different agents interacting with each other
- Multiagent organizations for real-world organizations
  - Software systems devoted to support the work of some human organization (e.g., a work group or a network enterprise)
  - Should somehow “mimic” the structure of the real-world organization to minimize conceptual mismatch between the real-world and the software organization

Why Multiagent Systems? (3)

- More in general...
- It is easier to think in terms of “multiagent systems” to solve complex problems rather than in terms of “multi-object” or “service composition” systems...
  - After all, we live every day in a world which goes on by interactions of autonomous agents (humans), each with limited rationality and interacting with each other to achieve a common goal or a self-interest goal
  - This is how we live at work, with friends, when playing sports...
Why Multiagent Systems? (4)

- Multiagent systems are “paradigmatic” of modern distributed systems
  - Made up of decentralized autonomous components (sensors, peers, mobile devices, etc.)
  - Interacting with each other in complex way (P2P networks, MANETs, pervasive computing environments)
  - Situated in some environment, computational or physical
- We can this at these are sorts of multiagent systems

Object and Service Systems vs. Multiagent Systems (1)

- Objects and services are functional entities, agents are goal-oriented entities
  - In a multi-object and service system, objects and services participate by providing functions
  - In a multiagent systems, agent participate by providing the capability of achieving a goal in autonomy
- Objects and services are not autonomous, agents do are
  - The execution of an object or service is typically subject to a global flow of control (the orchestration scheme, the business plan), determined by the sequence of services invocations
  - The execution of an agent is autonomous, subject to its own internal decision
### Object and Service Systems vs. Multiagent Systems (2)

- Objects are not situated, agents are
  - In object and service systems, “everything is an object”
  - Agents, instead, live in a world of other agents and environmental resources
  - Maps more easily our normal perception of the world
- Objects interact via client-server, agents communicate and coordinate
  - In a object systems, the mean of interaction is service request
  - In service system, complex contractual interactions can be statically determined
  - Agents interact via
    - Exchanging knowledge via communication messages
    - Coordinating their actions (e.g., negotiate the execution of tasks, synchronize on their execution on the access to resources, etc.)
    - And possibly interact via a shared environment

### Object and Service Systems vs. Multiagent Systems (3)

- Objects are mechanical entities
  - The way of interacting in object-based systems resemble that of a mechanical system
  - Actions-reactions
  - Chain effects
  - They manage data and not knowledge
- Services can be a bit more social, but are not autonomous
- Agents are social
  - Autonomous entities that interact not because they are forced to
  - But because this helps either their own selfish goal or a global goal
  - Interactions resemble social interactions...exchange of knowledge, negotiation for actions, competitions for resources...
- Multiagent systems are social systems
  - “Agent organizations” “agent societies”
Applications of Multiagent Systems

- Distributed Control of Physical Processes
- Resource Management
- E-Commerce
- Workflow Management
- Multiagent Simulation
- Optimization

Control of Physical Processes

- Agents embedded in computer-based control systems
  - Devoted to control/assess the working of a specific tool in an autonomous way
  - Agents interact with each other to agree on common global working of a systems
- Example: manufacturing pipeline
  - Various agents devoted to control various stages of the pipeline
  - Interacting so as to ensure it overall works well → COOPERATION!
Control of Physical Processes: Where to Apply

- **Industry**
  - To control production of goods
    - Daimler Chrysler has already deployed multiagent systems to automate part of its manufacturing process
  - Agent control the allocation of pieces so as to overall optimize efficiency

- **Home and Offices**
  - To control heating systems or the air-conditioning systems
  - A multiagent system can save a lot of energy

- **Traffic Control**
  - Agents assigned to semaphores with the aim of maximizing local throughput
  - Depending on traffic conditions, agents can cooperate to each other to maximize global throughput

Resource Management

- Each agent devoted to maximize the utility/exploitation of a specific resource, in autonomy
  - Interacting with each other so as to ensure that the exploitation of the resource ensemble is overall balanced

- Example: allocation of channels in cellular networks
  - Some phone calls can be allocated to multiple cells
  - Cells are controlled by agents which interact with each other to agree on who gets a call
  - Competition and “economic approaches” can be a good solution to get the maximal utility
Resource Management: Where to Apply

- Cellular Networks (see previous slide)
- Power Grid
  - Agents allocated on each power plant
  - Monitor the production and the consume of electronic power
  - The agents interact with each other so as to ensure that all potential power is exploited
  - Agent limits cooperatively limit overall resource consumption to avoid blackouts
- Computers Grid
  - To distribute computationally intensive applications and task on a network of computers
  - So as to minimize execution time

E-commerce

- Commercial transactions can be delegated to agents
  - Each belonging to a specific commercial actor
  - In charge of acquiring goods on the network (physical or digital goods)
  - Agents interact with each other to negotiate and eventually to perform the transaction
- Example: e-auctions
  - There is not need for humans to attend auctions (and they do not want to be bored of)
  - The work can be delegated to agents
  - Look for a resource
  - Ask the auctioneer agents
  - Place bids
  - Agents compete with each other

Auctions at the fishmarket (courtesy of Pablo Noriega)
E-Commerce: Where to Apply

- Personal assistant agents
  - Sent to the Internet to buy things on our behalf
- B2B Agents
  - To perform commercial transactions
  - To take care of supply chains management
- A note:
  - Several experiments show that agents can do better in commercial transactions than humans
  - Can handle more knowledge
  - Can take more rationale decisions

Workflow Management

- Agents acting as avatars of humans in a distributed workflow process
  - Agents control that the overall process proceeds according to specified rules
  - Ensuring consistency of the process
- Example:
  - Control the production of a complex multi-author document
Agents for Workflow Management

○ Where to apply
  ● In humans organization
  ● To support complex or decentralized coordination activities

○ In temporary coalitions (e.g., dynamic network enterprises)
  ● To support effective coordination in the absence of previous experience of coordination

○ In scientific research
  ● To control and coordinate the execution of complex experiments by distributed research groups

Simulations

○ Analytical (e.g., equation-based) simulations
  ● Models the dynamics of a real-world system as a dynamical system
  ● Analyze the evolution of the systems as evolution in the phase space
  ● Problems: difficult to model the system properly, difficult to analyze its evolution, do not capture enough details,

○ Multiagent simulations
  ● Models the real-world system in terms of its composing agents
  ● Execute the system by having all its components execute in autonomy and interact with each other
  ● Advantages: easy to model (agents may have simply behaviors), Easy to analyze (“run and see”), more realistic

○ Example: traffic management
  ● Do not model the dynamic of flow in a street
  ● Models each and every car as an "goal-oriented” autonomous entity
Simulations: Where to Apply

- Traffic simulations
- Ecosystems simulations
  - Agents are the animals of the ecosystem
  - Looking for food
- Social simulations
  - Panic scenarios (how do people behave when escaping from a stadium?)
  - Economic scenarios (how do an economy behave in specific situations?)
  - Epidemic scenarios (how do an infection propagate based on the behaviour of infected people?)
- Entertainment-oriented simulations
  - Videogames (e.g., Quake Arena)
  - Special effects (e.g., "The Lord of The Ring" movie)

Optimization

- Instead of the more traditional “Operation Research” analytical approaches
  - Which globally parameterize and model the system
  - And analyze the possible parameter space
- Multiagent optimization
  - Rely on agents analyzing a local solution
  - And interacting with each other to build together a global solution
- Examples:
  - Ant-based agent systems for minimum path selection
Part 2

- Agent communication and coordination

Multiagent Interactions: Key Concepts

- Communication
  - Agents exchange information
  - "I have completed my task"
  - "I have the file Millennium.mp3"
  - "Do you have the file Millennium.mp3"

- Synchronization
  - Agent schedule their activities
  - "Wait!" "Go!" "Do not access file Millennium.mp3"

- Agent Coordination
  - Agents orchestrate their activities
  - May include (but not necessarily) exchange of information
  - May include synchronization
  - "I do task T1 and you do task T2 concurrently"

- Interaction Protocol
  - The sequence of event/messages that compose an interaction
Types of Interactions

- Collaborative
  - Agents cooperate together towards the achievement of some common application goals
  - Agents can trust each other (they tell the truth and they will do what they say)
- Competitive (non collaborative)
  - Agents interact together only to achieve their own interests
  - They do not necessarily tell the truth, and they cannot trust each other
- Collaborative competition
  - Agents compete each other but for the sake of achieving some global goal
  - They trust each other but they exploit sorts of “competitive negotiations” to interact

Means of Interactions

- What actual forms do agent interactions take
  - Direct: agents directly communicate with each other by exchanging messages
  - Indirect (or stigmergic): agents interact indirectly by accessing
    - Some common information space or
    - The environment in which they situate
Agent Communication Languages

- A direct way of interactions, with formalized "communication acts"
  - Grounded in sociology and adapted by the artificial intelligence community
- Messages between agents are considered "speech acts" or "performative"
  - What is the scope of the message ("I want to inform you that...") "I request you that..."
  - What is its content ("...that I know where the Milenium.mp3 file is")
- Some people say that an software component can be considered an agent only if it is capable of conversations with agent communication languages
  - I do not agree, too restricted definition!

Structure of an ACL Message

- Semantically defined standard vocabulary and syntax
  - Agent exchange
  - String-based messages
- An ACL Message
  - Structured message, targeted for flexible communication
  - Performative (INFORM, QUERY, REFUSE, ...) 
  - Addressing: TO and FROM
  - ConversationID – Used to link messages in same conversation
  - In reply to – Sender uses to help distinguish answers
  - Reply with – Another field to help distinguish answers
  - Reply by – Used to set a time limit on an answer
  - Language – Specifies which language is used in the content
  - Ontology – Specifies which ontology is used in the content
  - Protocol – Specifies the protocol
  - Content – This is the main content of the message
Agent Communication Example

INFORM
sender antagent
receiver bob martin
protocol status
conversation_id example6
reply_with 275
reply_by wed 3pm
language lisp
ontology ant
content (target (project "1hour") (platform "computer15") (author "sean") (time "8/07/01 4pm") (message "build failed") (target "compile") )

Primary Intent
Inform, Request, Failure, Refuse, ...

Addressing
broadcast, forwarding, ...

Dialogue Coordination
expected vs exceptional response

Detailed action or request
problem specific language, ontology, request

Interaction Protocols

- In general, an interaction protocol is
  - A sequence of messages between agents
  - With determined types of “performatives” exchange with each agents
  - And with determined content
  - Aimed at some form of coordination between agents
- An interaction protocol may also involve multiple agents
- Many types of interaction protocols have been standardised
  - E.g., client-server, contract-net, various types of auctions, etc.
- It is also possible to implement interaction protocols also via indirect interactions
  - In this case, the protocol specifies the sequence of access to the shared dataspace or the shared environment to be performed by agents
The Ontology Problem

- How can agents understand each other?
  - In a closed world, the programmer knows what knowledge terms (variables, data, string) means
  - And can build its agents accordingly
- In an open world, where agents developed by different stakeholders have to interact with each other
  - How can they understand with each other?
  - The meaning of the terms must be the same for all agents
- Agents have to rely on shared common “ontologies”, i.e., common perception of the world

Standards
- contract-net,
auctions,
planning,
negotiation,
management, ...

Tools
- State Machines
- Workflow
- UML

Models
- Meetings
- Organizations
- Context
Ontologies

- An ontology is an explicit formal specification of how to represent the objects, concepts, and other entities that are assumed to exist in some area of interest and the relationships that hold among them.

- In other words, an ontology for a software world is a set of "terms", each intended to represent a specific concept, put in specific types of relations with each other.
  - For software agents (as well as for the Web) an common ontology represents a common ground with which to share common knowledge in an understandable way.
  - And with which to reasons together about the “things” and the “fact” of their world.

- RDF and the various XML specifications are simply ways to define Web ontologies for specific domains.
  - Multiagent systems can similarly define shared ontologies for, e.g., electronic commerce.

Ontology: Terms & Relationships

**Standards**
- Shopping, Events, Travel, People, Organizations, Roles, Context, ...

**Tools & Models**
- UML, XML, AUML, DAML, OIL, RDF, Protégé, ...

**Event**
- type:
  - at:
  - Multiple:
    - oneOf:
      - allOf:

**Room**
- features:
  - address:

**Time**
- zone:
  - day:
  - hour:

**Person**
- email:
- role:
- project:

**Organizer**

**Participant**
- priority:

**where**

**when**

Negotiation Protocols

- Many types of interaction protocols are “common sense” also for non-agent applications
  - Client-server, simple exchange of information, synchronization commands
- However, agent autonomy and sociality enables agents to be able to “negotiate” with each other during execution
  - Negotiation protocols are of primary importance in multiagent systems
  - And find a variety of important applications in modern scenarios

Multiagent Negotiation

- “Negotiation is an economically-inspired form of distributed decision making where two or more partners jointly search a space of possible solutions to reach a common consensus” P. Maes
- Negotiation is a sort of dynamic agreement between two or more agents on how to act
- Competitive negotiation
  - Each agent tries to get the best for its own local utility
  - E.g., agents competing for a good on an Internet auction
- Non-Competitive negotiation
  - Each agent cooperates to maximize the global utility
  - E.g., agents negotiate for resource allocation so as to maximize the overall resource exploitation
Why Multiagent Negotiation?

- Agents typically have goals to achieve in a dynamic environment
  - Do not have a priori encoded solutions to deal with
  - The most appropriate action must be dynamically evaluated
  - And must be agreed with other agents
- Economics and Game Theory tell us that
  - Good equilibrium solutions can be reached based on negotiation
  - Both in the case of competitive and non-competitive interactions

Elements of Multiagent Negotiation

- A Negotiation between agents usually imply
  - A Protocol
    - The “rules of the game”
    - Determining how (with which sequence of messages) agents can participate in the negotiation
  - A Strategy
    - Each agent evaluates internally how to act, in accord to the rules, and based on its own goals
Cooperative Multiagent Negotiation: The Contract Net Protocol

- Useful to allocate in a satisfactory (sub-optimal) ways tasks in a complex multiagent system
- Cooperative:
  - It assumes that agents are willing to cooperate and are sincere
- Contract
  - The two agents reach a sort of agreement which correspond to a sort of contract

Hey guys, I have these task to be done: T1(attributes); T2(attributes)

I can do T1 very well
T2 so so
I can do T2 well
Can't do anything

You get T1
And your friend below get T2

Cooperative Multiagent Negotiation: The Contract Net

- The Contract Net
  - A “Manager” agent has a set of sub-task to allocate to other agents
  - It inform all other agents about these tasks (type of task, description, specific requirements, deadline, etc)
  - And ask all other agents to “bid” for that
- Prospective contractors
  - Place a bid specifying how they would be able to do that works (in what time, with what efficiency and accuracy)
  - For instance, an agent that would be able to do a task but it is already in charge of other task will make a weak bid
- For each task, the manager
  - Assign the task to the best bidder (“Sign the contract”)
  - And possibly repeat this until all tasks are assigned
- Clearly
  - All agents can possibly act as Manager or Bidders
  - There could be multiple chains of contracts
  - An agent could be a manager for a task and a bidder for other task
Applications of the Contract Net

- Task and Service Allocation in General
  - Cellular networks:
    - if a cell is saturated by phone calls,
    - it may ask neighbor cells to bid for tasks such as data transfers, normal phone calls, video calls, etc.
    - Similar considerations apply to IP networks
  - Manufacturing Process
    - A stage in the process which process items and then has to pass the items to the next state in the pipeline
    - when multiple machines can perform the next state
    - The contract net always select the best machines
    - E.g., consider the example of a set of car painting machines. The bid can be based on the currently loaded color and on the cost of changing the color cartridge
  - In Supply Chains
    - to determine the better contractor to supply a service or a good

Non Cooperative Multiagent Negotiation: Auctions

- Agents compete to access a “good”
  - A task, a digital resource, an information, or a physical artifact
- Valuable for their own interests
  - They need it to maximize their own utility
  - They can “pay” to get it
- They engage in an auction
The Actors of an Auction

- The seller (or the sellers)
  - Make a good available for sale
  - Wants to get the maximum possible price
- The Buyers (“Bidder”)
  - Place “bids” on the good to buy it (“Hey, I pay 10 dollars!”)
  - Try to get the good at the lowest price
- The Auctioneer
  - Control the execution of the auction
  - Enact the rules and check cheaters
  - Decide who wins
  - May not be necessary in collaborative systems

Types of Auctions

- English auction “ascending price”
  - The one we all know...
  - Start with the lowest acceptable price
  - Reach a price which is nearly (but never) the maximum anyone is willing to pay
- Dutch auction “descending price”
  - Starts with an excessive price
  - Lower the price monotonically
  - The first who bid wins
  - Can get higher prices than the English
- American auction (“first sealed bid”)
  - Secret bid
  - The highest wins
  - The value of the bid is usually the price of the highest bid
- Fishmarket auction (Double auction with multiple sellers)
  - The bidders increments the price
  - The sellers decrement the price
  - Eventually, a buyer and a sellers agree on a common price which satisfy both
Applications of Auctions

○ In Electronic commerce
  ● A very effective way to establish the right prices for things
  ● We humans do not want to participate in such a boring process
  ● But agents can be delegated of that!

○ In resource management systems
  ● By assigning virtual “prices” to resources
  ● Economic laws tell us that nice balancing in resource exploitations will be reached

Combinatorial Auctions

○ The agents wants to bid for a specific combination of goods/resources
  ● Flight + Hotel + Theatre Tickets
  ● A communication path across multiple carriers
  ● Air traffic control

○ In these cases:
  ● The Bidder should carefully evaluate its bid (economic theories)
  ● The Auctioneer should reserve a whole bunch of goods for individual bids, or it can propose alternative combinations
Combinatorial Auctions in Traffic Management (2)

- Do we really need redlights in intersections?
  - Little adaptivity
  - Low efficiency
- Even if there is no room to install a roundabout, some seems to be able to get rid of redlights

If we see an intersection like a resource
- We could have vehicles “bid” to reserve the intersection
- And an agent on the crossing to act as auctioneer

However
- A single vehicle do not need the whole intersection
- But only those sections needed to safely cross
- So let’s consider the intersection a composite resource
Combinatorial Auctions in Traffic Management (3)

- Eventually (P. Stone, Univ. Texas, 2008)
  - We can obtain dramatic efficiency
  - Reduce queues
  - Increase safety

- Related Applications
  - Air traffic control (it is being adopted!)
  - Schedule of composite hospital services

Other Negotiation Protocols

- Matrix Games and Prisoner Dilemma
  - Agents decide the best action to take on the basis of a personal revenue that may depend of other agents' behaviors
  - The system will reach the “Nash Equilibrium”

- Nature-inspired
  - Relying on the fact that the evolution of simple interactions in the system will eventually lead to stable satisfactory equilibria
  - E.g., “diffusion-inspired” models for resource management

- Tete-à-Tete
  - Whatever type of “discussion” two agents may engage in to negotiate the what to do...
  - E.g., “You have three task and I have one, let’s share them!”
Indirect Agent Interactions

- Two agents may indirectly interact
  - By accessing a common shared dataspace
    - E.g. a shared “knowledge blackboard” or a “tuple space”
  - By accessing a common shared environment
    - E.g., a physical environment or a computational environment

Tuple Spaces

- Shared containers “bags” of tuples
  - Data structures with multiple fields
  - In Java agent systems, tuples spaces can be containers of objects
- Agent typically get a reference to a tuple space (via a Discovery service of the middleware) and
  - Put tuples in the tuple space
  - Read or extract tuples from it
The Tuple Space Model

- Primitives to put, read, extract tuples
  
  ```
  write(int 7, char 'f', float 2.78);
  read(int a?, char c?, float 3.14);
  take(int 7, char 'f', float)
  ```

- Associative access to tuples: “pattern matching” of tuples with a provided template tuple (a tuple with some non-defined fields), e.g.:
  
  ```
  the template tuple (int i?, char 'f', float 2.78) matches with the tuple (int 7, char 'f', float 2.78)
  ```

- Associative access is a sort of attribute-based access
  - One specify the structure and some attributed of the data it is interested in
  - The tuple space get the tuple with the matching attributed and return it to the requesting agents

- It is also a coordination model
  - synchronization over tuple occurrence in the tuple repositories
  - an input operation blocks if no match occurs)

---

Pheromone and Field-based Interactions

- Agents interact indirectly with each other by
  - Spreading sort of “gravitational” fields or, “chemical pheromones” in the environment
  - Physical (e.g., a room) or computational (e.g., a network)
  - And being influenced in their behavior by the local “concentration” or by the “gradient” of fields or pheromones in the environment

- Will get back to this in detail later on in the course...
Part 3

- Agent Infrastructures
  - The JADE and the MARS examples

Agent Infrastructures

- The Overall ensemble of software components (e.g., the software framework) needed to develop, deploy, execute, agents and multiagent applications
- The "Agent Middleware"
  - Various types of services provided
  - Various types of interaction models and protocols supported
- In its basic, do not different to much from SOA middleware

- Various types of infrastructures for various types of applications
  - Simulation: e.g., Unreal Tournament Softbots for Videogaming, Jack Agents as used to implement ogres in "The Lord of the Ring" movie
  - Java Agent for Mobile Systems (e.g., the Aglet agent system) and Web-based Systems (e.g., the JADE system + J2EE)
  - Control of physical processes. Light weight agent systems for embedded computers.
  - Complex data management. Intelligence knowledge-oriented agents for complex data management (e.g., the RETSINA multiagent infrastructure)
FIPA Specifications

- The Foundation for Intelligent Physical Agents
- Specifies STANDARDS for multiagent infrastructures
  - to interoperate and be managed
- Formally specified ACL
  - Specifies encoding, semantics, and pragmatics of messages
- Includes: mobility, security, ontology, Human-Agent comm.
- FIPA reference architecture (see below)

![FIPA Architecture Diagram]

JADE (Java Agent DEvelopment Framework)

- JADE – A FIPA-compliant Agent Framework
  - [http://sharon.cselt.it/projects/jade/](http://sharon.cselt.it/projects/jade/)
- Is a software framework
  - simplifies the implementation of multi-agent systems
  - Attempts to be very efficient
  - Fully implemented in Java and fully distributed under LGPL
  - Mostly oriented to **AGENT COMMUNICATIONS** (via ACL)
- Definitely the most used systems
  - AND IT IS ITALIAN!!!
  - Developed by UNIPR and TELECOM-IT
JADE continued

- Is the middleware for MAS (Multi-Agent Systems)
  - Target users: agent programmers for MAS
  - Agent services
    - life-cycle (to handle creation and death of agents), yellow-pages (naming service), message transport (to have different platforms interoperate)
  - Agent Communication Languages
    - Support for Speech Act and Negotiation protocols
    - Support for Shared Ontologies
  - Tools to support debugging phase
    - remote monitoring agent, dummy agent, sniffer agent
  - Designed to support scalability
    - (from debugging to deployment)
    - (from small scale to large scale)

Distributed architecture of a JADE Agent Platform

- Host 1
  - DF Agent
  - AMS Agent
  - Application Agent
- Host 2
  - Application Agent
  - Application Agent
- Host 3
  - Application Agent
  - Application Agent
  - Application Agent
- Jade distributed Agent Platform
  - Jade Main-container
  - Jade Agent Container
  - Jade Agent Container
- Network protocol stack using RMI or BOP

JRE 1.2
JADE Agent Platform - GUI

- Remote Agent Management
  - Remote Monitoring Agent
  - Management Agent
  - White pages GUI – to find agents
  - Agent life cycle handling allowing start, stop, pause, migrate, etc.
  - Create and start agents on remote host
    - Assumes container already registered
  - Naturally uses ACL for communication

JADE Communication Sub-system

- Every agent has a private queue of ACL messages created and filled by the JADE communication sub-system
- Designed as a chameleon to achieve the lowest cost for message passing
  - The mechanism is selected according to the situation
  - The overheads depend on the receiver’s location and the cache status
- If you send a message to another agent and the sub-system can’t find target, then it sends it to the AMS to handle
- Graphics tools to analyse agent communications
JADE Interaction Protocols

- **Interaction protocols** are the FIPA way to manage interactions.
- JADE provides support for FIPA generic interaction protocols, e.g.:
  - FIPA Contract net;
  - FIPA English and Dutch auctions.
- JADE implements interaction protocols as FSM behaviors.
- Graphics Tools to Analyse Protocols

The MARS Infrastructure

- **Mobile Agent Reactive Spaces**
  - Developed at the University of Modena e Reggio Emilia
  - Ported on different agent systems (*Aglets, Java2Go, SOMA, JADE*)
- One shared data space on each node
- **“Tuple spaces”**
  - Attributed-based access to local resources
- Programmable tuple spaces
  - A “meta-level” can control and monitor all agent interactions
MARS Features

- Mobile agents roam the Internet
  - On each node, they connect to a local tuple space
- They can access it to retrieve/put data
  - Data can be accessed via attributes
  - Mediated interactions between agents via the local tuple space
  - Coordination and various interactions protocols as sequences of accesses to the tuple space
- Access to local resources
  - Appears to agents as access to data in the tuple space

Part 4

- Related and Open Issues
Related Issues (1)

- **Agent-oriented Software Engineering**
  - The discipline of building multiagent systems
- **How can we face the process of**
  - Starting from a problem
  - Identifying the agents that must be involved
  - Their interactions and negotiation
  - Their interactions with the environment
  - The overall architecture of the system
- **It is a matter of exploiting**
  - Methodologies
  - Guidelines
  - Tools

Related Issues (2)

- **Multiagent systems and complex systems**
  - Can multiagent systems lead to complex emergent behaviors?
  - YES! \( \rightarrow \) cellular automata, ant-systems, etc.
- **Complex vs. Simple Agents**
  - We can conceive systems with many simple reactive agents, leading to collective intelligence, and relying on system adaptivity
  - As well as systems with few complex – "very intelligent" – agents
- **Multiagent systems and social networks**
  - How systems of autonomous entities network with each other?
  - What is the structure of social networks?
- **Multiagent systems and natural systems**
  - Multiagent systems as a metaphor for naturally inspired systems
  - Useful to solve several classes of problems
- **Naturally inspired interaction models**
  - Can nature give us source of inspiration for other types of interactions models
  - Suitable to solve complex application problems?
- **We will analyze next in the course...**
Open Issues

- So many open issues...
  - Security and Trust
  - Monitoring of Distributed Agent Systems
  - CASE and Programming Tools for building Multiagent systems
  - Novel simple infrastructures for Multiagent systems
  - Evolving multiagent systems
- It’s a very active research area...