Multi-Agent Systems

Master 2: Systèmes Interactifs Intelligents & Autonomes

Pierre Chevaillier

Ecole Nationale d'INgénieurs de Brest

Fall'21

ABSTRACT: This document is related to the course on Multi-Agent Systems (MAS) delivered in the Module "Multi-Agent System and Interactive Simulations" of the parcours SIIA (Systèmes Interactifs Intelligents & Autonomes). It has been generated on October 4, 2021.

This document is incomplete in purpose. Students are supposed to annotate the document during courses, laboratories and home work. The digital version of the document contains clickable links to external resources.

This course is merely an introduction to multiagent systems. First, the course presents the basic principles related to the concepts of agent and multiagent systems. Next the course gives a broad overview of the different conceptual solutions and types of technology used for designing agent-oriented softwares.

The second part of the course is devoted to the coordination mechanisms in MAS.

MAS: Multi-Agent Systems

Master 2: Systèmes Interactifs Intelligents & Autonomes

- Distributed Artificial Intelligence – *swarm intelligence*
- Autonomous entities with *self-** properties
- Adaptation of the system to a dynamic environment
- *Emergence* of system-level behavioral properties
- Applications: problem solving & simulation
- Agent-Oriented Approach:
  General Artificial Intelligence + Object-Oriented Design

Self-* properties: self-adaptive, self-organisation, self-awareness, self-configuration, self-management, self-diagnosis, self correction, and self-repair (or self-healing). There are general concepts and there are many way to ensure —at least partially— these properties.

Part I

Concepts, Architectures & Technologies

Multi-Agent Systems

Basic concepts

To do: write down here your own summary of this section. Identify what are the most important points, what is perfectly clear for you and what is less.

Agents: autonomous entities

- Autonomous behavior:
  - Perception of the environment: *focus* and *awareness*
    The agent is able to act within partially unknown environments
  - Decision-making (deliberative behav.): goals and action selection:
    based on its percepts, the agent chooses the *right* action: rationality
  - Actions performed by the agent results from the decision-making process and are adapted to the context
- *Pro-Action*: goal-oriented behavior

Definition:

Autonomy: A is autonomous / Y for P of Y
iff A is able to decide by its own about P

MAS: system compounded of proactive autonomous entities

AGENTS: SOCIAL ENTITIES

Acting in a shared environment:

- *Influences* $\Longrightarrow$ *response*
  - between Agents (artificial or human)
  - from, and on, the Environment
- individual agents' activities contribute to the system (society)

Explicit communication (not always) implies:

- Agents exchange information, use *artefacts*, purposively or not.
- *Collaborative* behaviors
- *Organization*: constraints, rules / interactions

DEFINITION:

MAS: system compounded of collaborative loosely coupled entities

Agent: Rational behavior

- Rationality (goal-directed reasoning)
- Intentionality – Agency
- Planning
- Deliberation and negotiation

Agency (social sciences):

capacity of individuals to act independently and to make their own free choices.

Rationality (philosophy of practical reason):

agent selects and then performs the action that results in the optimal outcome for itself.

A *rational agent* has clear preferences, models uncertainty via expected values, and always chooses to perform the action that results in the optimal outcome for itself from among all feasible actions.
The decision a rational agent makes depends on (from Russel & Norvig):

- the preferences of the agent
- the agent's information of its environment, which may come from past experiences (the sequence of its percepts)
- the actions, duties and obligations available to the agent
- the estimated or actual benefits and the chances of success of the actions.

Agency (from K. E. Himma, *Artificial agency, consciousness, and the criteria for moral agency: what properties must an artificial agent have to be a moral agent?*, Ethics and Information Technology, 2008): "The concept of agency is associated with the idea of being capable of doing something that counts as an act or action". It depends on the agent's "mental state", its willing, or volition, to perform intentionaly the action.

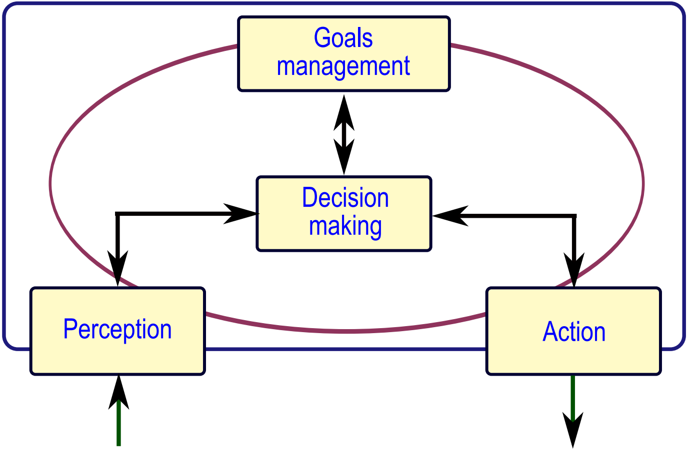
Agent: bounded capabilities

- Local perception and action
  no omnisicent entities
- Limited repertoire of competences
  agents with different (specific) capabilities
- Bounded rationality
  nobody's perfect ...
- Finite amount of time and computational resources to decide
  agents act in a changing world

Bounded rationality (source: wikipedia) is the notion that in decision making, rationality of individuals is limited by the information they have, the cognitive limitations of their minds, and the finite amount of time they have to make decisions. It was proposed by Herbert Simon as an alternative basis for the mathematical modeling of decision making. Thus the decision-maker is more a satisficer, one seeking a satisfactory solution rather than the optimal one.
The concept of bounded rationality accounts for the fact that perfectly rational decisions are often not feasible in practice due to the finite computational resources available for making them.

Agent abstract Architecture

Behavior-based Intelligence: building blocks for Autonomy



The above figure illustrates the general principle of an agent architecture, with its main functional building blocks. Actually no real agent is built according to this simplistic schema. Nevertheless, the many proposed architectures can be viewed as variations on this very general and abstract design.

MAS & Agents: definitions (summary)

Agent

- physical or logical, *autonomous* & *proactive* entity,
- *interacting* with its *environment* and other agents

Multi-Agent System

- self-organized set of agents immersed into an *environment* and acting together to reach a *common goal*
- Adaptive system, *self-organised*, with *emerging* properties

MAS as Systems

Set of autonomous collaborative *smart* entities

Agents System: set of autonomous entities

- Properties of the system are defined by its designer
- Entities adapt their behavior to satisfy these required properties
- "*Intelligent*" entities: *adaptive rational agents*

Multiagent System: autonomous system

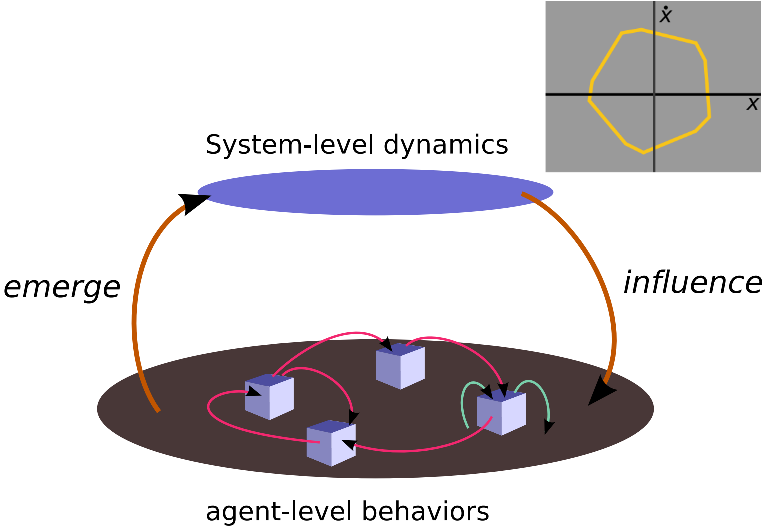
- Adaptation of the system
- self-* properties: self-organization ...
- System having emerging properties
- *"Distributed Artificial Intelligence"*

COMMENT: Not all the authors make the disctinction between *agents system* and *multiagent system*. Up to now, most of existing systems have fallen in the first category.
The *Distributed Artificial Intelligence* (DAI) is based on the principles underlying the theory that Marvin Lee MINSKY has started to develop with Seymour Papert in the early 1970s and that he published as a book entitled *The Society of Mind* in 1986 [Minsky, 1986]. According to this theory, mind is the emerging property resulting from the activity of individually simple processes named *agents* which are not themself *intelligent*.

Example of emergent behavior

Ants' trails

Emergence and behavioral dynamics



Emergence: definition

General definition

- A global (at the system level) behaviour, or *pattern*, which arises from the interactions between the agents (micro-level)
- the behavior is a dynamical construct which arises over time
- the agents have no representation of the global behavior
- interactions play a major role

Point of view of dynamical systems:

at a given moment in time, a bifurcation occurs, which leads the system to converge toward a new attractor

J. Holland (1998) *Emergence: from Chaos to Order*

Page 14

Adapted from [De Wolf and Holvoet, 2005]. The concept is not new and it is used in different scientific fields: physics, biology, social sciences ...

Moulay Aziz-Alaoui, Moulay, & Bertelle, Cyrille, 2006, Emergent Properties in Natural and Artificial Dynamical Systems, Springer-Verlag Berlin Heidelberg.

Self-* systems

Self-* property:

capability of the system to perform some action upon itself, by itself, without any external intervention (direction, control)

- self-organizing systems: *"a dynamical and adaptive process where systems acquire and maintain structure themselves, without external control "* [De Wolf and Holvoet, 2005].
- self-adaptive systems: *"capable of evaluating and changing in runtime its own behavior, whenever the evaluation shows that the system is not accomplishing what it was intended to do, or when better functionality or performance may be possible."* [Macías-Escrivá et al., 2013]

Associated terms:

self-awareness, self-regulation, self-healing, self-learning, autopoiesis, autonomic

Types of application

1. Simulation of complex systems
2. Software engineering: loosely coupled system, dynamic service composition
3. Problem solving: computational intelligence, DAI

MABS: MultiAgent Modeling & Simulation

- MABS and related approaches:
  - Cellular Automaton
    popularized in the 70s by J. H. Conway's Game of Life
    and latter by S. Wolfram
  - IBM: Individual-Based Models
    see [DeAngelis and Grimm, 2014] for their application in ecology
- Areas of application: complex systems
  - Ecology: population dynamics
  - Economy: market
  - Biology: medicine
  - Human & Social Science: crowd, traffic, decision making ...

Applications to engineering

- Concepts
  - Agents as *information retrievers*, *mediators*, *assistants* ...
  - Autonomy and adaptivity – Interactions
  - Autonomous service discovery and composition: loose coupling, dynamic planning Requires Service Description Language and ontologies
- Application areas:
  - *e-Business*, *e-Commerce* (Ashraf et al., 2019)
  - Transportation(Cruz et al. 2018), *Integrated Supply Chain Management* (Shukla et al., 2019) ...
  - *Ubiquitous computing* (Ambient Intelligence), distributed robotics
  - *Smart* -grid (Xie & Liu, 2017), -city (Postránecký *et al*, 2017) ...
  - Personal Assistants, ITS

Problem solving: computational intelligence

- Context
  - Distributed problem solving, distributed control
  - Dynamic and open environment: robustness, adaptivity
- Benefits of the MAS approach
  - Autonomy, adaptivity
  - Distributed-control, coordination
- Domain of Application
  - Manufacturing: Tasks allocation, job scheduling
  - Network: Adaptive routing, fault tolerant storage (Romito, 2012)
  - Collective mobile robotics (swarms)
  - Map generalization (Duchene *et al.*, 2012)
  - Surveillance: anomaly detection, alert triggering (Brax, 2013)
  - Computational creativity ...

Agents: issues & challenges

- Reactivity *vs* Pro-activity
- Autonomy *vs* Collaboration
- Context-specific behavior with general-purpose agent
- Planning, anticipation within a dynamic environment
- Action selection, continuous planning

MAS: issues & challenges

- Distributed control – Swarm intelligence – Heterogeneity
- Behavioral stability & plasticity
- Efficient collaboration
- Emergence towards the desired behavior
- Self-adaptation, (collective) learning
- Verification & Validation

AOSE: Agent-Oriented Software Engineering

Principle of MAS for the analysis, design and implementation of software systems

- Analysis: MAS as a metaphor
  - Complex system
  - Adaptation of the system to the users and context
  - Collaborative process supported by collaborative entities
- Design
  - Distributed loosely-coupled architecture
    set of interacting heterogeneous active components
  - Algorithms from AI
    soft computing, machine learning, behavior-based AI
- Implementation
  - *Smart* components
  - MAS frameworks: JADE, Janus
  - Specific AOP Language. E.g. SARL

AOP : *Agent-Oriented Programming*
SARL: `http://www.sarl.io`, [Rodriguez et al., 2014]

Comments: multiagent systems share many principles with the Object-Oriented paradigm. These principles are parts of UML: roles and services, asynchronous communication, collaboration.

See also: *holonic systems*.

Multi-Agent Systems

Design of MAS

To do: write down here your own summary of this section. Identify what are the most important points, what is perfectly clear for you and what is less.

AEIO: Agent

- Nature of the agents
  - Physical or conceptual entity?
  - Human or artificial (synthetic)?
- Behavioral Architecture
  - Reactive or cognitive?
  - Recursive, hybrid?
- Repertoire of capabilities & behavioral components
- Internal state, Knowledge, [explicit goals]
- Action selection, [Planning]
- Adaptation, [Learning]

Yves DEMAZEAU a introduit le concept de *voyelles* pour l'étude des systèmes multi-agents [?].

AEIO: Environment

- *Space* where agents are embedded in and act on
  - Information agents can access to: perception / action
  - Interaction medium
  - $E = \bar{A}$
- Properties (from Russel & Norvig, 2003)
  - Static or dynamic?
  - Deterministic or stochastic?
  - Continuous or discrete?
  - Physical or informational?
  - Topological, Euclidean?
  - Shared or distributed (fully accessible or not)?

L'environnement n'est pas toujours explicite dans les SMA. Sa nature est très variable d'un système à un autre et il est souvent traité de manière *ad hoc*. C'est certainement dans le domaine de la simulation multi-agents faisant appel à des agents situés que l'environnement fait l'objet d'une formalisation la plus avancée (topologie, dynamique).

AEIO: Interaction

- Influences and reactions
  - Agent – Environment
  - Agent – Agent (direct or not)
  - Intentional or contingent?

- Perception – Action:
  stimulus – reaction

- Communication
  - Agent – Agent
    - Protocole (conversations)
    - Agent Communication Language (ACL)
  - User – Agent
    - Natural Language Understanding & Generation
    - Non verbal signals: gesture, emotions, posture ...

AEIO: Organization

1. Organization as a *tool*
   a common frame of reference in which agents can act
   - Rules
     - Collective behavior: cooperation, action coordination
     - Social conventions: Norms (deontic logics)
   - Constraints on
     - **A**gents
     - **I**nteractions
     - **E**nvironnment
2. Organization as a *process*
   - resulting from the agents' activities and interactions
   - Self-organization
     can be intentional or contingent

La logique déontique (du grec *deontos* : devoir) tente de formaliser les rapports qui existent entre les quatre alternatives d'une loi :

- O : l'obligation,
- I : l'interdiction,
- P : la permission,
- F : le facultatif.

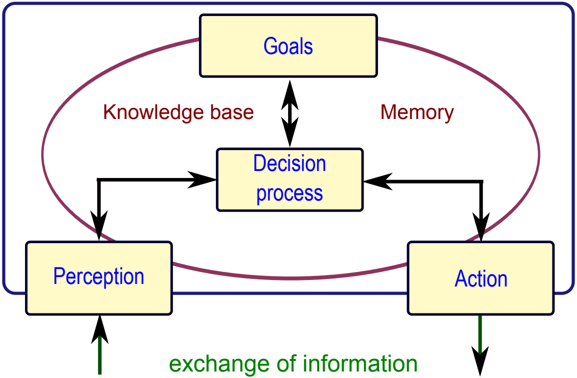
Dans [**?**], Hübner *et al.* ont utilisé la logique déontique pour la spécification d'organisations multi-agents.

AEI0... U: USER?

- Status / role?
  - Spectator
  - Actor
  - Creator
- In or Out the MAS?:
  is the User an Agent from the PoW of the system?
  - Out: Humans are not synthetic entities
    Person–System Interaction
  - In: users play roles that agents could also play, or share the authority with agents (eg. participatory simulation)
- Immersed User
  - How s/he is perceived by the Agents?
  - How can s/he interact with the Agents?
  - Is there a mediator agent b/w U and Agents?

Cognitive Agent

Deliberative behavior – planning / Goals

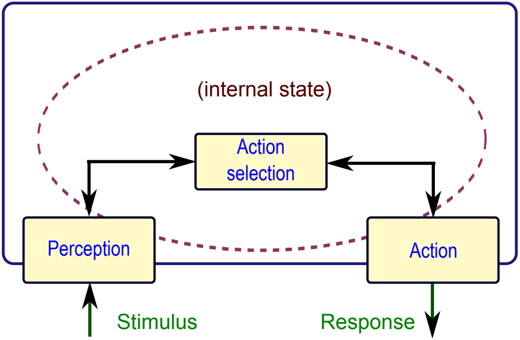


- Decision making: logical reasoning (inference), planning
- Goal-Oriented behavior: beliefs + intentions → actions

Cognitive architecture aims at providing human-like reasoning capabilities to agents. The approach is based on the so-called *unified theory of cognition* from cognitive psychology, as envisioned by authors like John Robert Anderson, Michael E. Bratman, Allen Newell. Nevertheless, due to computational constraints the proposed algorithms work in a very different way that the human brain actually does.

Reactive Architecture

Instantaneous reaction to stimulus



- Behavior = activation of (some) module(s)
- [pro-]activity = control of the activation / goal (ASM)
- Perception = behavior: [pro-]active perception

One may distinguish pure reactive agents from model-based reactive agents. Pure reactive agents have no internal representation of the world nor their self: they react to the stimulus they can perceive locally in their environment. Model-based reactive agents maintain a so-called internal representation of the environment, or of the current situation; they may also hold a representation of the feasibility conditions and/or the expected outcome of their actions. This information can be represented using finite state automata, or by 'internal' numerical variables. The reaction of such an agent depends on both the stimulus it perceives and its current internal state.

Reactive models are also called *behavior-based architecture*, that contrast with *plan-based architectures* (aka Knowledge-Based) or *motivational architectures*. Therefore, the agent's plans are not explicit: the agent is designed to perform right now the right action.

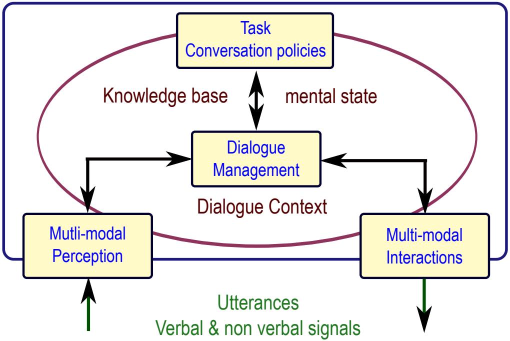
Reactive behaviors can be obtained by explicit and *ad hoc* algorithms that calculate output of the agent from its inputs and its internal state (if any). It is more suitable to use general purpose algorithms that separate the description of the behavior from its execution. Reactive behaviors can be described using state machines, production systems or behavioral architectures like those presented latter in this document.

Usually, reactive architectures are defined as a set of interconnected *modules*, often organized in layers. The activation of the modules depends on both data flow and flow of control. The algorithm responsible for the election of the "right" behavior is called the Action Selection Mechanism (ASM). These algorithms used an *utility function* to select the "best" action. Two kinds of policy may be used: *arbitration* (=competitive policy) or *fusion* (collaborative policy). To a certain extend, behaviors are seen as "micro-agents" that compete, or collaborate, for acting.

The design of multiagent systems compounded of reactive agents follows the principles of *society of Mind* proposed by Marvin Lee MINSKY, [Minsky, 1986], which is a model for human cognition. The principle of this paradigm is that cognition emerges from the activation of a large amount of simple agents and from their interactions.

ECA: Embodied Conversational Agent

Example of Domain-Specific architecture



- Natural Language Processing
  NLU/NLG: Syntactical and semantical analysis
  (based on ontology or not)
- Human-like interactions
  User model, task model, communication protocol

The concept of ECA (*Embodied Conversational Agent*) is closed to the one of *Intelligent Virtual Agents* (IVAs), which are autonomous embodied agents embedded into a virtual environment. These agents are able to interact intelligently with human users, other IVAs, and their environment.

Approaches in Artificial Intelligence

1. The Cognitivist paradigm
   - Cognition: manipulation of symbols (representation)
   - Central processing unit
   - Emblematic solution: Soar
2. The Embodied Cognition paradigm
   - Cognition: emergent state corresponding to the physical world
   - Distributed processing of what the agent perceives
   - Emblematic solution: Rodney A. Brooks's robots

Soar: Origin and Id

John E. Laird

- PhD in computer science (1983)
  Carnegie Mellon University
- Thesis advisor: Allen Newell (1927–1992)
  General Problem Solver (1959), Unified Theories of Cognition (1990)
- Research interest:
  cog. arch. for creating human-level artificial intelligent agents

Soar – U. Michigan

- Current stable version: 9.6.0
- Kernel, VisualSoar, SoarJavaDebugger, SML API
- [Laird, 2012], *The Soar Cognitive Architecture*,
  MIT Press, Cambridge, MA.

The “official” presentation from http://soar.eecs.umich.edu/, Univ. of Michigan (the former Soar’s website): Soar is a general cognitive architecture for developing systems that exhibit intelligent behavior. Historically, Soar stood for State, Operator And Result, because all problem solving in Soar is regarded as a search through a problem space in which you apply an operator to a state to get a result. Over time, the community no longer regarded Soar as an acronym.

Soar, John E. Laird *et al.*, Univ. of Michigan
[Laird et al., 1987]

Goal

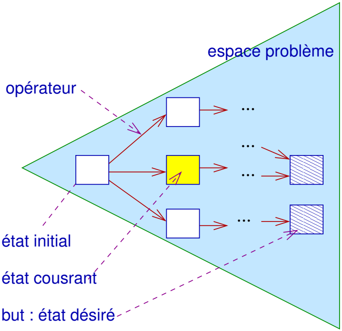
- Agent Architecture for the simulation human cognitive behavior
- General algorithm for problem solving + Knowledge representation
- Candidate / *Unified Theories of Cognition* (Newell, 90)

Principle: the cognitivist perspective

- Reasoning: moves into the *problem-space* (representations)
- Rationality: Agents with goals, process data / knowledge
- Knowledge representation and manipulation (projection & inference)
- Interaction with the environment: Perception – Action
- Learning: *chunking* and more

Soar: basic concepts

State, Operator And Result



- *problem space*: states, operators, goals
- *state*: $< \textit{augmentation}, \textit{value} >$
- Initial state – current state (subgoaling)
- Result: new state (final: goal)
- Operator: a "path" between 2 states

Soar: general principles

- Behavior: moves into the *problem-space*
  step by step application of *operators*

- Rationality: deliberative goal-oriented behavior
  if an operator contributes to the goal, the agent should select and then apply it

- Knowledge organized into *memories*:
  the current situation (state + ...) and long-term knowledge

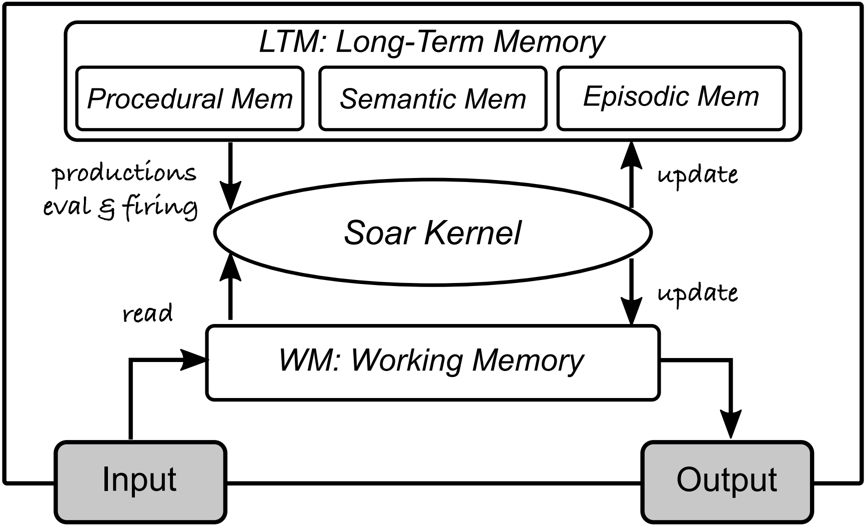
Architecture

- knowledge representation

- algorithm to compute the moves through the problem-space
  Simulates human problem-solving capabilities

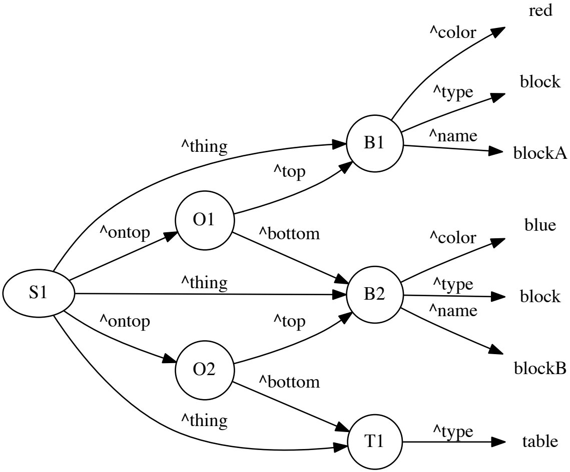
Soar: agent architecture

Conceptual view



Knowledge representation: example

WME's: Working Memory Elements



Organization of Memories in Soar – overview

WM: Working Memory = mémoire active

= representation of the current state by the agent

- Built by – and directly available to – the kernel
- Knowledge about:
  - data from sensors
  - active goals and operators
  - results of intermediate inferences

LTM: Long-Term Memory

= problem solving capabilities of the agent

- Not directly accessible, but via operators
- Contents $\mathcal{K}$ about:
  - how operators modify the state
  - monotonic inference: *state elaboration*

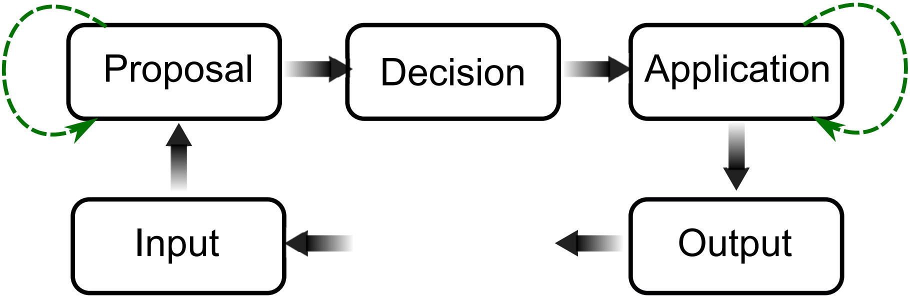
Soar's execution cycle

Conceptual view

State elaboration
Operator proposal
Operator comparison

Operator selection
or
substate creation

State elaboration
Operator application



add (delete) elements
from WM in response
to change in the environment

send motor commands
to the environment

Knowledge for decision-making in Soar

$\mathcal{K}$ to select and apply operators to a state

1. Proposing candidate operators for the current state
2. Comparing candidate operators
3. Selection of one single operator
4. Applying the selected operator
5. Making inferences / state = state elaboration

Programming in Soar:
defining the production rules that implement these functions

Rationality in Soar [1/2]

Means-Ends Reasoning

- Agent is able to decide what is the best action to reach its goal and actually performs that single action
- In Soar: the agent should have a goal, select one operator and apply it to the current state
- For the only one operator for which the conditions *meet* the current state, the production rule is *matched* and then *fired*
- When rationality is put in action?
  - no solution to progress towards the goal (or no goal)
  - cannot decide what is the best thing to do in the current situation
  - lack of $\mathcal{K}$ to perform the desired action

Rationality in Soar [2/2]

Impasse

- No operator is selected: none of them meet the current state
- Multiple operators are proposed and the 'best one' cannot be determined
- The current goal of the agent is to resolve this *impasse*

Response to an *impasse*

- Creation of a sub-state whose goal is to solve the impasse
- Agent must have some knowledge to resolve the impasse.
  The type of $\mathcal{K}$ depends on the type of impasse
- Impasses may be encountered when solving an impasse

The Subsumption Architecture [Brooks, 1986]

Embodied cognition - Priority-based Arbitration

Context: autonomous mobile robotics

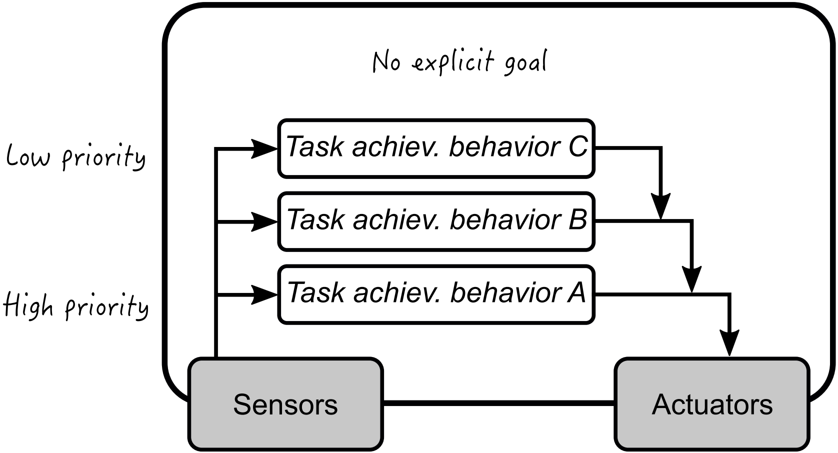
- No internal (symbolic) representation of the world
- Robustness of the control
- Modularity for an incremental building
  level of performance should be augmentable (Brooks & Flynn, 89)
- Parallel computation / loosely coupled processors
  No central control over the modules' activation
- *Task-achievement behaviors*: the control is decomposed in terms of behaviors rather than in terms of functional modules

Rodney A. BROOKS worked on the modular design of autonomous mobile robots. His goal was to be able to built robots by assembling existing modules that could be plugged into a general purpose hardware architecture.

Rodney A. BROOKS promoted the idea that robots can exhibit intelligent behavior without maintaining any internal symbolic representation of the environment [Brooks, 1991]. Brooks's famous statement is that *the world is its best own representation*. This non representational approach of intelligence was in conflict with traditional approaches in artificial intelligence that classically assume that intelligence is based on the symbolic representation of the current situation and results from logical inference processes.

The Subsumption Architecture

General principle, from Rodney A. Brooks, 1986



The Subsumption Architecture

Example, from [Brooks, 1986]

8	reason about behavior of objects
7	plan changes to the world
6	identify objects
5	monitor changes
4	build maps
3	explore
2	wander
1	avoid objects

The Subsumption Architecture

Rodney A. Brooks, 1986

Layered Architecture

Levels of competence: desired class of behavior

- The layer defines the priority of the corresponding behavior
  The lower the layer, the higher the priority
- A layer: a network of wired processors that exchange numerical messages and run asynchronously, with no shared memory.
  (Processor = AFSM: FSM + registers + alarm clocks)
- Agent's behaviors are coordinated thanks to their relative position and the existence of inhibitor and suppressor nodes.

AFSM: *Augmented Finite State Machine*.

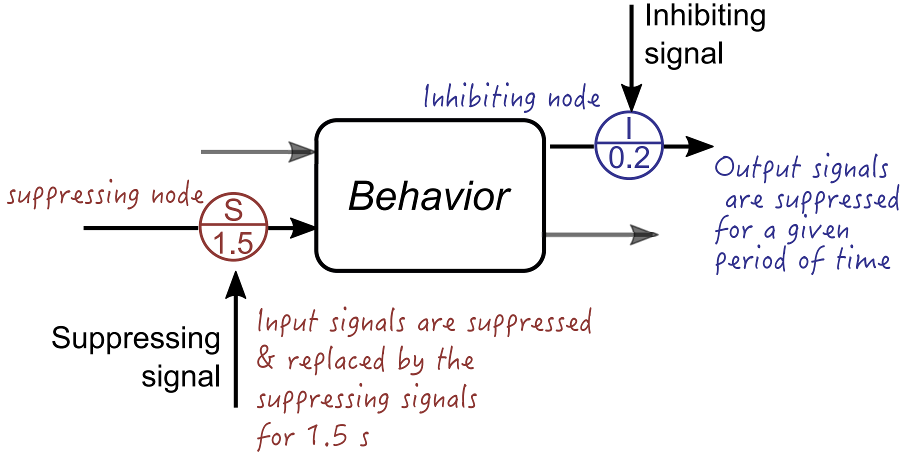
Subsumption Architecture

- Network of wired processors (activity, behavior)
- Output of a module can be wired to input registers of many other modules
- Augmentation of existing level of competence
  Resolution of conflits b/w behaviors: connection nodes
  - *output inhibition*
    When a message arrives on an inhibitory node, no messages can travel along the wire (for a short period of time)
  - *input suppression*
    When a message arrives on a suppressing node, no messages are allowed to flow from the original source (for a short period of time)

In complement to AFSM, Rodney A. BROOKS proposed the so-called *Behavior Language* (Lisp-like syntax) inspired from the work of Pattie MAES.
[**?**] gives a simple description of the subsumption architecture (in French).

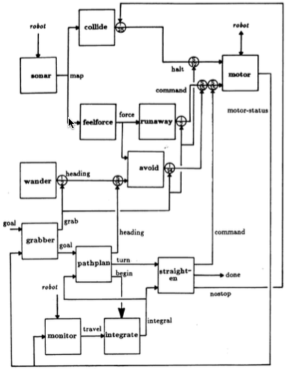
The Subsumption Architecture

General principle, from Rodney A. Brooks, 1986



Brooks's Architecture: strengths and weaknesses

- Strengths:
  - Theoretical framework
    perception – action
  - pragmatic results
    navigation of mobile robots
  - Simple *meta-control*
- Weaknesses:
  - All modules are hardwired to sensors, thus depending an a specific technology
  - Strong dependencies bw modules
  - The priority of the behaviors can not be changed
  - No explicit goal - no adaptation



Mutli-agent platforms

Definition:

- Offers support services and basic construct-blocks for the development of agent-oriented applications
- Developers focus on the agents' behavior and interactions

Two main classes:

1. Software framework
   - Middleware for the interoperability, distribution of apps.
   - Services: directory, communication,
   - Ex.: JADE – Telecom Italia Lab
     Open source – Latest version 4.5 June 2017
2. Simulation tools
   - Concurrent execution of agents in the same environment
   - Services: simulated time and space, agent scheduling, simulation-oriented tools
   - Ex.: Gama latest version 1.8 RC2 (summer 2018)

A recent survey: A Survey of Agent Platforms, Journal of Artificial Societies and Social Simulation, K. Kravari and N. Bassiliades, 2015.

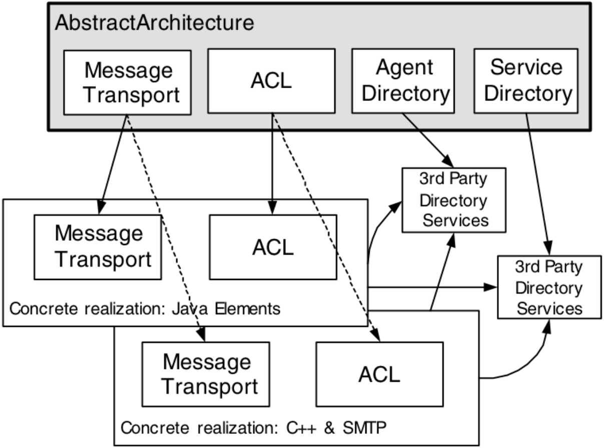
Fipa

Foundation for Intelligent Physical Agents (1996)

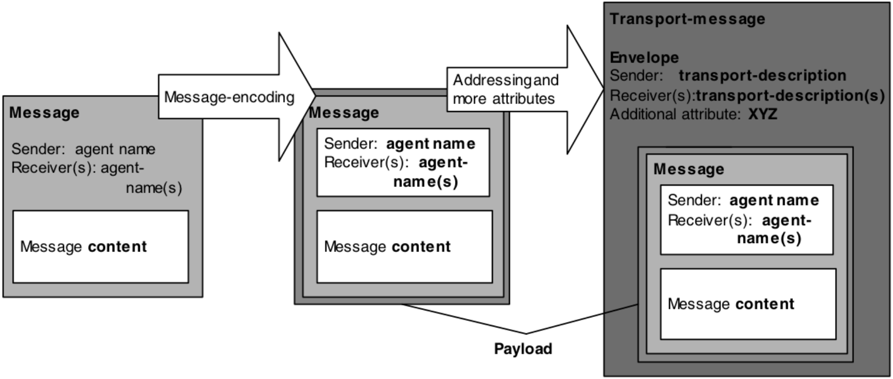
Objective: interoperability

- Abstract Architecture
- Agent Management Specification
- Agent Communication Language
- Content Language

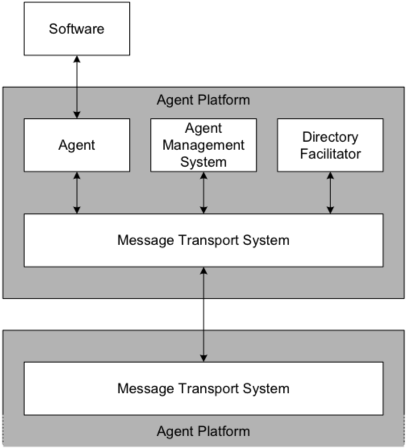
Fipa abstract and concrete architecture



Fipa Message transport



Fipa Agent management



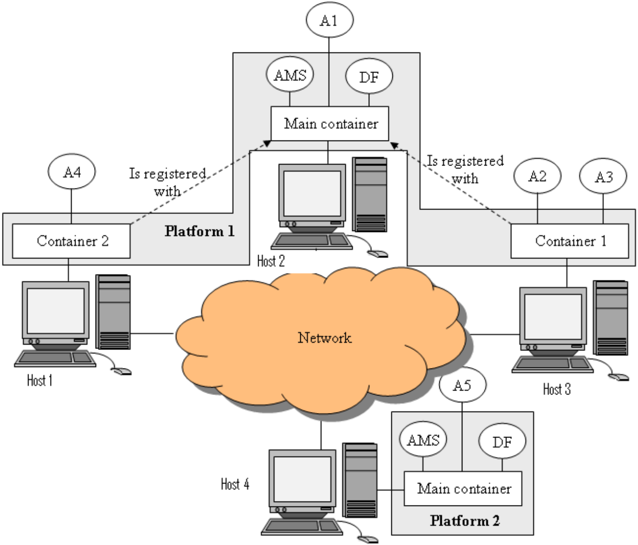
Functions of the AMS:

- register
- deregister
- modify
- search
- get-description

The AMS can ask the platform to execute the following actions: suivantes :

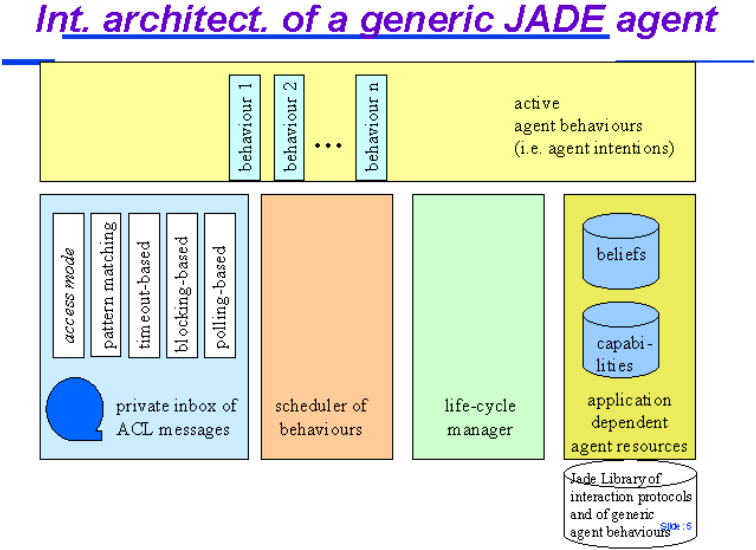
- Suspend agent,
- Terminate agent,
- Create agent,
- Resume agent execution,
- Invoke agent,
- Execute agent, and,
- Resource management.

JADE ARCHITECTURE



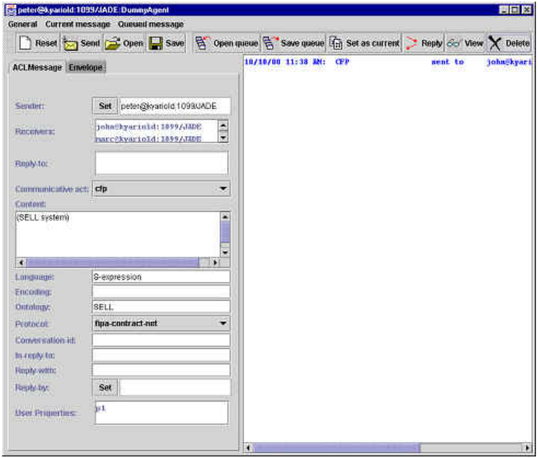
source: JADE web site http://jade.tilab.com/documentation/tutorials-guides/jade-administration-tutorial/architecture-overview/

JADE agent internal architecture



source: JADE web site http://jade.tilab.com/documentation/tutorials-guides/jade-administration-tutorial/architecture-overview/

Administration and monitoring tools

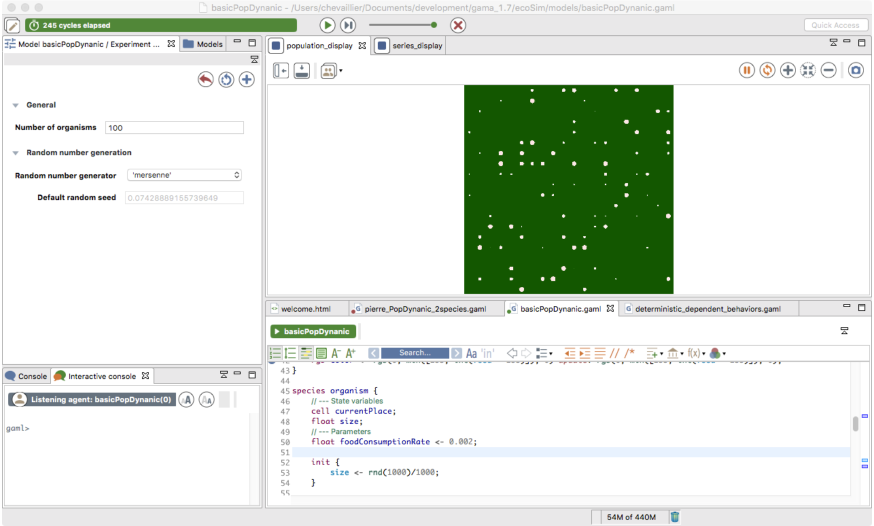


An agent responsible for inspecting message exchanges among agents

source: JADE web site http://jade.tilab.com/documentation/tutorials-guides/jade-administration-tutorial/architecture-overview/

The GAMA platform

[Taillandier et al., 2019]



https://gama-platform.github.io

Part II

Coordination mechanisms

Multi-Agent Systems

Communication

TO DO: write down here your own summary of this section. Identify what are the most important points, what is perfectly clear for you and what is less.

Communication in MAS

- One form of interaction
  the reception of a message influences the agent's behavior
  and it may respond to it
- A tool for collaboration
  collaboration needs Communication, Coordination and Cooperation
- Exchange of information among agents
  semantics attached to messages
- A collaborative behavior
  stakeholders should facilitate the communication

Characteristics of communication in MAS

- Basically asynchronous
- Do agents produce *messages* intentionally or not?
- Signals (stimulus) or structured, rich-content messages?
- What do agents share? Language, rules (protocoles), goals ...?
- Is communication free, or constrained by the Organization or by the 'law of physics'?

Differences bw O–O and A–A communication

Agents are autonomous entities, objects are not

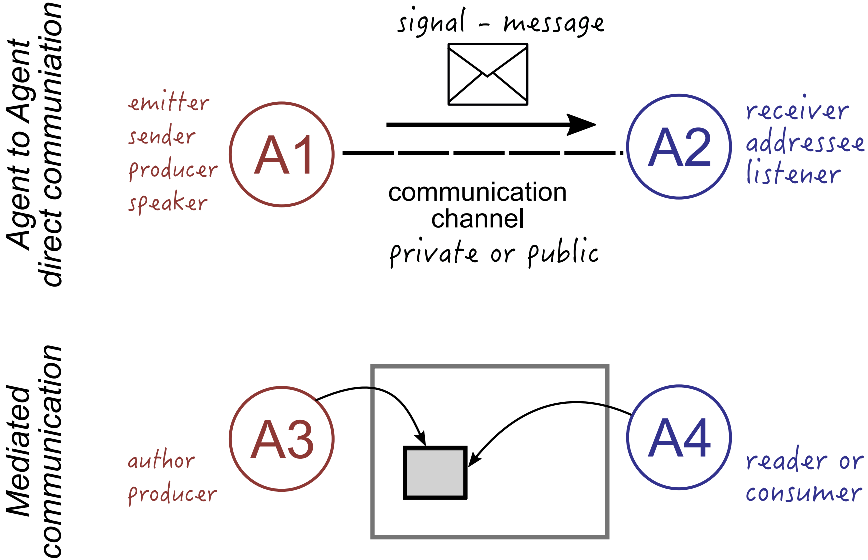
O–O: client-server $\neq$ A–A: peer to peer

Agents never access to the *features* of another agent

Characteristics of Agent–Agent communication

- The addressee may be (temporarily) unreachable
  O–O: null pointer exception...
- Agents are free to take into account received messages or to ignore them
  O–O: the server does what the client asks: the service
- Agents interpret the messages according to their own perspective
  O–O: the server executes the service, using its own the method

Communication channels



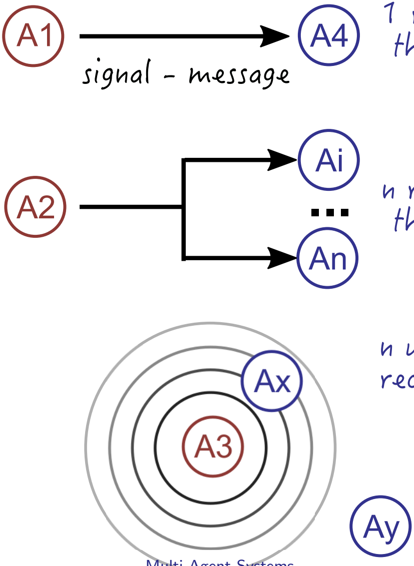
Whom the message is addressed to?

Point to Point

Multicast

Broadcast

situated (here)
or not



1 receiver
the sender knows

n receivers
the sender knows

n unknown
receiver(s)

This agent will not
receive the message

Communication protocols, or Conversation policies

- Heterogeneous MAS
- Agents' autonomy
- Protocole defines the level of interoperability of the agents
- Agents' communicative behavior must comply to a set of explicit *rules*: policies

Issues

- Robustness w.r.t. heterogeneity & autonomy
- Flexibility w.r.t. message ordering and handling
- Logical consistency: semantics & pragmatics

Speech Act Theory

Communication is a form of action

- *How to do Things with Words*, J. L. Austin, 1962
- Speech Acts:J. R. Searle, 1969
  distinction bw the *illocutionary force* and the *propositional content*
- Classification of illocutionary force of speech acts:
  - assertives: commit the speaker to the truth of the expressed proposition
  - directives: to cause the addressee to take a particular action (requests, commands and advice)
  - commisives: commit the speaker to some future action, e.g. promises
  - expressives express the speaker's attitudes and emotions towards the proposition, (e.g. agreement, thanks)
  - declaratives change the reality in accord with the proposition of the declaration

ACL: Agent Communication Language

- KQML Knowledge Query and Management Language
  Darpa project (1993), Labrou's PhD, 1996
- FIPA: an IEEE Computer Society standards organization
  interoperability between 'physical' agents
- DIT++: for Human-Agent spoken interactions
- MyACL: my application specific primitives
  since interoperability doesn't matter,
  and I have very specific needs (semantics, implementation),
  then why not defining my own ACL?

Modeling communication protocols

- Conversation: a sequence of interactions
  that take place within a (dynamic) context
- Protocols modeling (traditional approaches):
  - Uml: Statecharts or interaction Diagrams (e;g. Fipa)
  - FSM, Petri Nets, π-Calculus
  - Temporal Logic
- Behavioral-Intelligence:
  - Production rules
    Emission
    if <condition> then send Message_Type(<parameters>)
    Reception
    if <received message properties> then update_KB(<message content>)

Designing Communication protocols

- Identify participants requirements and their roles
- Reuse, or compose, existing protocols, if exist
- Define how messages affect commitments
  - message ordering should be encoded within antecedents and consequents of commitments
  - Nonoccurence of a message : does it violate agent's commitments?
- Make it independent to the underlying implementation

Kinds of Fipa performative

- Information management
  content: proposition
  - inform, inform_if, inform_ref, query_if, query_ref, subscribe, confirm, disconfirm, not_understood
- Task management
  content: action
  - request, request_whenever, agree, refuse, cancel, failure
- Negotiation
  content : action and proposition
  - cfp, propose, accept_proposal, reject_proposal

The complete set of FIPA performatives:

Accept proposal	Inform	Query if
Agree	Inform if	Query ref
Cancel	Inform ref	Refuse
Call for proposal	Not understood	Reject proposal
Confirm	Propagate	Requets
Disconfirm	Not understood	Request when/whenever
Failure	Proxy	Subscribe

Formal semantic of Fipa message (1/2)

Example: INFORM

The sender informs the receiver that a given proposition is true.

Mental state of the sending agent

Sending an inform message indicates that the sending agent:

- holds that some proposition is true,
- intends that the receiving agent also comes to believe that the proposition is true, and,
- does not already believe that the receiver has any knowledge of the truth of the proposition.

The first two properties defined above are straightforward: the sending agent is sincere, and has (somehow) generated the intention that the receiver should know the proposition (perhaps it has been asked).
The last property is concerned with the semantic soundness of the act. If an agent knows already that some state of the world holds (that the receiver knows proposition p), it cannot rationally adopt an intention to bring about that state of the world, that is, that the receiver comes to know p as a result of the inform act. Note that the property is not as strong as it perhaps appears. The sender is not required to establish whether the receiver knows p. It is only the case that, in the case that the sender already happens to know about the state of the receiverÕs beliefs; it should not adopt an intention to tell the receiver something it already knows.

Formal semantic of Fipa message (2/2)

Example: inform

From the receiver's point of view

receiving an inform message entitles it to believe that:

- the sender believes the proposition is true,
- and wishes the receiver to believe that proposition also.

Whether or not the receiver does, indeed, adopt belief in the proposition will be a function of the receiver's trust in the sincerity and reliability of the sender

Formal semantic of Fipa message

Example: inform

Content: A proposition ϕ.

Formal Model:

$$< i, inform(j, \phi) >$$

$$\begin{array}{rl} \text{feasibility preconditions :} & B_i\phi \wedge \neg B_i(Bif_j\phi \vee Uif_j\phi) \\ \text{rational effect:} & B_j\phi \end{array}$$

$Bif_i\phi \equiv B_i\phi \vee B_i\neg\phi$: either agent i believes ϕ or that it believes $\neg\phi$.

$Uif_i\phi \equiv U_i\phi \vee U_i\neg\phi$: either agent i is uncertain (in the sense defined above) about ϕ or it is uncertain about $\neg\phi$.

$U_i\phi$: agent i is uncertain about ϕ but thinks that ϕ is more likely than $\neg\phi$.

Ref.: http://www.fipa.org/specs/fipa00037/SC00037J.pdf

Properties of a Fipa SL Content Language

profiles (source: http://www.fipa.org/specs/fipa00008/)

- FIPA–SL0: *Minimal subset*
  allows the representation of actions, the determination of the result of a computation, the completion of an action and simple binary propositions
- FIPA–SL1: *Propositional Form*
  adds boolean connectives to represent propositional expressions.
- FIPA–SL2: *Decidability Restrictions*
  allows first order predicate and modal logic, but is restricted to ensure that it must be decidable.

Properties of a Fipa message

Message
≪type of communicative act≫ performative: PerformativeKind
≪participant≫ sender: AgentID receiver: AgentID reply-to: AgentId
≪content≫ content: string
≪description of the content≫ language: string encoding: string ontology: string
≪control of the conversation≫ protocol: string conversationId: Id reply-with: MessageId in-reply-to: MessageId reply-by: Time

A very simple conversation based on the FIPA-CNP protocol
Code sample in GAML

```
species initiator skills: [ fipa ] {
  reflex send_cfp_to_participants when: (time = 1) {
    do start_conversation (to: list(participant),
      protocol: 'fipa-contract-net', performative: 'cfp',
        contents: [ 'Go swimming' ] );
  }
  reflex receive_refuse_messages when: !empty(refuses) {
    loop r over: refuses {
     write name + ' received refuse from ' + r.sender + ' : ' + r.contents ;
    }
  }
}
```

A very simple conversation using the FIPA-CNP protocol

Code sample in GAML

```
species participant skills: [ fipa ] {
  reflex receive_cfp_from_initiator when: !empty(cfps) {
    message cfp <- cfps[ 0];
    write 'receives a cfp message from ' + agent(cfp.sender).name;
    do refuse (message: cfp, contents: [ 'I am busy today' ] );
  }
}
```

Comments: In Gama, message sendings are actions (do <action_name>). The name of the action is the name of the performative. P. ex. do refuse.

Multi-Agent Systems

Distributed-decision making

To do: write down here your own summary of this section. Identify what are the most important points, what is perfectly clear for you and what is less.

Coordination in MAS

What coordination problems are?

- Agents have to make decision about their actions that depend on what other agents actually do or decide
- coordination: the process by which agents reason about their own course of actions and the others' actions, so as the group acts coherently
- Agents need to coordinate their *expectations* about their *intentions* and actions
- What agents have to decide on:
  - How to elaborate plans for an interleaved execution of their actions
  - How to manage conflicts about resources usage
  - How to combine private goals with shared goals

Multi-Agent coordination

- Coordination is necessary because [Jennings, 1996]:
  - there are dependencies in the agents' rationality:
    local decisions made by one agent impact the decision of others
  - some global constraints must be satisfied
  - no one agent has sufficient competence, resources or information to solve the problem
- Coordination problems are solved through
  social rules (conventions, commitments...),
  communication, *observation* and *inference*

Remenber: in a MAS there is no global controller ...

Necessary conditions for the coordination to occur

From Durfee *et al.* 1989

1. Existence of *structures* which enable agents to interact in predictable ways
2. Flexibility to cope with the partial and imprecise perception of the other agents (and the environment)
3. Agents having sufficient knowledge and reasoning capabilities to exploit the available structures and the flexibility to do it

Approaches in Multi-Agent coordination

1. Bio-inspired approach: models from ethology
   - No explicit shared goal, nor explicit intention to cooperate
   - Reactive behavior
   - Self-organisation
2. Human-inspired model: models from social sciences
   - Goal-oriented behavior
   - Agents that share knowledge
   - Team coordination
3. Computational-based approach: Distributed computing
   - Specific coordination mechanisms
   - Protocols for interoperability
   - Distributed-problem solving

Reactive coordination

Coordination as an emergent behavior

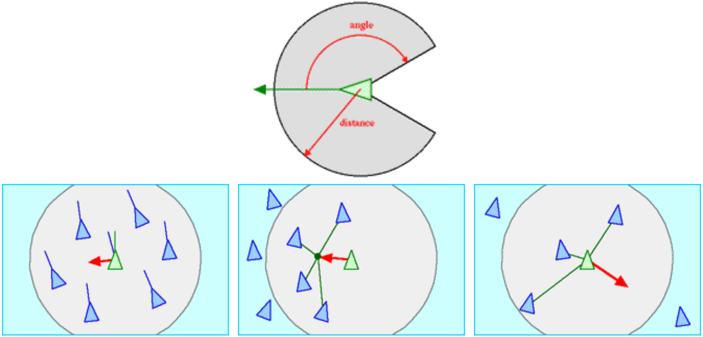
- *Situated reactive Agents*
- strong locality, no goal-oriented behavior
- Information written in the environment or direct interactions amongst Agents
- Examples:
  - Coordinated displacemments: *Boids* [Reynolds, 1987]
  - Eco-resolution [Ferber, 1995]
  - Conflict management for altruist agents

Page 85

boid : néologisme pour *bird-oid* (voir www.red3d.com/cwr/boids/).

boids: EMERGENT BEHAVIOR FROM 'SIMPLE' INTERACTIONS

BEHAVIOR-BASED: NO COMMUNICATION – DISTRIBUTED CONTROL



Applications: computer animation [Reynolds, 1987], crowd simulation, swarm intelligence, IoT [Roca et al., 2016]

Mediated interactions: *pheromones trails*

The *stigmergy* principle

- Social insects: ants, bees, spiders ...
  ant foraging, nests construction, reproduction...
- Principles
  - Agents lay down pheromone trails that serve as a shared situated memory stored in the environment
  - Perception of the trail drives the agent's behavior
  - Agents reinforce the trail while they use the path to reach their goal
- Applications
  - Agents with few memory and low communication capabilities
  - Situated informations and Agents
  - Non persistent information, possibly mobile

Deliberative coordination

Team Coordination

- Managing dependencies between agents' activities
- the complementary temporal sequencing (or synchronicity) of behaviors among team members in the accomplishment of their goal
- the attempt by multiple agents to act in concert in order to achieve a common goal by carrying out a script they all understand
- the intent to work together to align goals and to invest effort to sustain common interest

Context: joint activity

from Garland & Alterman, 04

- A set of heterogeneous agents
- Agents' goals are private knowledge, unless the agent communicates it to others
- Agents make their own decision / the ordering of their goals
- Agents make their own decision / whether or not to cooperate
- Actions are not guaranteed to succeed
- Agents implicitly share the desire to solve coordination problems
- Agents do not hold all the information about other agents' reasoning strategies

Inter-dependencies between agents
(Sichman, Demazeau, 95)

- strong:
  A_i cannot achieve G_x without A_j
- weak:
  A_i could achieve G_x without A_j, but maybe less efficiently
- unilateral:
  A_i depends on A_j to achieve its goals, but A_j does not
- reciprocal:
  A_i depends on A_j to achieve G_x, and A_j to achieve G_y
- mutual:
  A_i depends on A_j to achieve G_x and A_j too (same goal)

Communication, observation, inference

- Communication
  exchange of information, request, joint commitment, negotiation, argumentation ...

- Observation to achieve *mutual belief*.
  Requires the *copresence* of the agents: agents in a common situation where they can perceive the same objects and also each others (at least partially)

- Inference
  copresence is also necessary for agents to infer their perception, their beliefs about others' perception ...

The *Contract Net Protocol*

From [Smith and Davis, 1981] – FIPA 02

- Cooperative distributed problem solving
  eg.: decentralized allocation of tasks among agents
- No one entity has the control on the whole
  none of them holds enough information to solve the problem
- the most appropriate agent for performing the task cannot be a priori defined
  no pre-ordered plan of actions – unpredictable environment
- agents spend more time in computation rather than communication
- Commmunication: broadcast + point-to-point

Contract Net Protocols

- Many variations:
  - FIPA Contract Net Interaction Protocol
  - [Aknine et al., 2001]
  - ...
- Applied to various types of problem, for instance:
  - job scheduling in flexible workshops
  - sensors network configuration (Chen et al. 2012)

CNP: TWO AGENT ROLES

1. Role *manager*:
   agents that need a task to be executed
   and that cannot perform it by their own

2. Role *bidder* = *contractor*:
   agents that can eventualy perform some task(s)
   and able to assess their own capability to perform the task

CNP: scenario for task sharing

1. A *manager* broadcasts a message (= *call for proposal*)
   to advertise the task it want to be done.

2. Potential *bidders* send a *bid*
   to the requesting *manager*

3. *Managers* evaluate the bids they have received
   and award the most appropriate *bidder* ("*best*" bid)

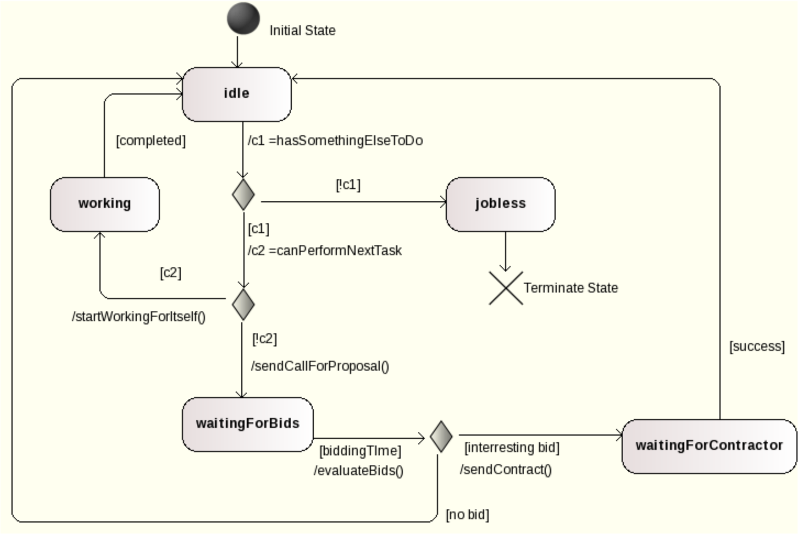
4. At the end, the bidder (=*contractor*) and the manager
   are linked by a *contract*.
   The *contractor* is then accountable to the manager
   for the actual execution of the task

CNP: TASK ANNOUNCEMENT

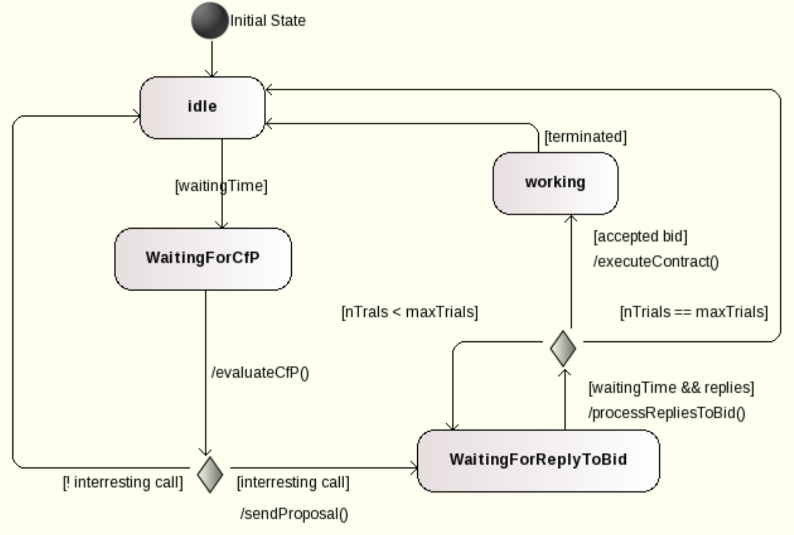
INFORMATION SHARED BW AGENTS – FIPA PROTOCOL

- *task abstraction*:
  informations about the task and relative to the *manager*
  eg.: type of task, due date, manager's location ...
- *eligibility abstraction*:
  information about the potential *contractor*
  eg.: required skills, location ...
- *bid specification*:
  information the potential contractor (=*bidder*) must provide in its bid
  eg.: delay and duration for the execution of the task, capabilities

CNP: BEHAVIOR FOR THE ROLE *manager*



CNP: BEHAVIOR FOR THE ROLE *contractor*



CNP: EXAMPLE

1. One agent (*manager*) broadcasts a task announcement (Call for Proposal CfP, a.k.a. *bid invite*)
2. Agents having the capabilities to perform the task send a bid (PROPOSE)
3. When the due date is reached, the manager evaluates the bids is has received so far
   - If it has received no bid, then the manager broadcast an new call for proposal (with a new due date)
   - else it evaluates the bid and select the best one (criteria: execution date...)
4. The manager sends a contract to the bidder corresponding to the selected bid (ACCEPT_PROPOSAL) and informs other bidders that it has rejected their bid (REJECT_PROPOSAL)

CNP: EXAMPLE

- The selected bidder (= *contractor*) performs the task and, at the end, informs the manager (not in the primitive protocol):
  - success or INFORM
  - FAILURE

PRINCIPLES (EXTENSIBLE)

- Manager: one single bidding process at a time
- Bidder: commits to only one call for proposal at a time
- comment: the same agent could play alternatively the two roles

Multi-Agent Systems

Multi-agent organizations

TO DO: write down here your own summary of this section. Identify what are the most important points, what is perfectly clear for you and what is less.

Definition of organizations

From Edgar Morin, 1977

An arrangement of relationships between components or individuals which produces a unit, or system, endowed with qualities not apprehended at the level of the components or individuals.

Organization: A social system existing in physical and social environments over time

- Predefined: explicit rules known by the agents
- Emerging property resulting from the agents' activity

Origins of organizational models:

- Bio-inspired: animal societies, esp. insects
- Social models: human societies or institutions

Benefits of Organization

- Introduction of constraints reduces the complexity of the problem
- Information sharing shapes the interactional space
- Agents can reason about the organization
- Defines stable, mandatory (or expected), features of the system

Relevance of organization

from Virginia Dignum and Julian Padget

From the agents perspective

- insures a better integration of the agents in the system
  in order to better adapt themselves to change
- A tool for the delegation of tasks/beliefs between the agents:
  structures that need to be represented such as teams, coalitions...

From MAS perspective

- insures some global behavior at the MAS level (eg. collaboration) To be sure that the global goals of the system are achieved
- represents observed patterns of interaction

Regulation *versus* Autonomy

- Autonomous behavior
  - Ability to make decisions about it's own activity
  - Individual rationality is insufficient to deal with social behavior (helpfulness, greater good, etc)
  - (Informal) structures are necessary for coordinating processes and stability
- Regulated, or directed, behavior
  - Pre-determined behavior, external to agent: leads to lack of agility
  - Do not consider differences in individual capabilities
  - Strict obedience to rules often does not get work done

Regulation *with* Autonomy

- Can we have the best of both?
- Combination of individual rationality with laws of social interaction

- Internal autonomy requirement: Specify organization independently from the internal design of the agent
  - Enables open systems
  - Heterogeneous participants
- Collaboration autonomy requirement: Specify organizations without fixing a priori all structures, interactions and protocols
  - Enables evolving societies
  - Balances organizational needs and agent autonomy

Relationships within organizations

- The organization should create order among entities and a coherence of the whole
- Relationships enable results that cannot be accomplished by any of the individuals alone
- The organization is responsible for the achievement of the overall goal of the system
- Possible relations between entities are:
  - Access to resources and services
  - Transfer of information, Delegation of tasks
  - Synchronization of actions
  - Responsibility between agents, obligation and norms

Requirements of Organizational Models

- Reflect and Support Organizational Design
  - Structure: roles, norms, interactions
  - Global goals and requirements
- Specify interaction independently from the internal design of the agent (internal autonomy requirement)
  - Interaction structures are not completely fixed in advance
  - Enables open systems
  - Heterogeneous participation
- Balance organizational design and agent autonomy (collaboration autonomy requirement)
  - Explicit agreements concerning individual performance
  - Explicit agreements concerning interaction
  - Enables evolving societies
  - Balances organizational needs and agent autonomy

(some) Organizational models

- Aaladin, a.k.a. AGR (=Agent-Group-Role), [Ferber et al., 2000]
- Moise+ [Hübner et al., 2007]
- OperA, [Dignum et al., 2004]
- ...

Multi-Agent Systems – Conclusion

- An ambitious approach
  but the interconnection of AIs isn't the next step?
- The synthesis of all the AI
  the Graal of the General Artificial Intelligence
- A new way to design and to use (collaborate with) artificial systems
- New acceptability issues: trust, privacy, authority sharing...

Aknine, S., Pinson, S., and Shakun, M. F. (2001).
Négocation multi-agent : analyse, modèles et expérimentations.
Revue d'Intelligence Artificielle, 15(2):173–217.

Brooks, R. A. (1986).
A robust layered control system for a mobile robot.
IEEE Journal of Robotics and Automation, 2(1):14–23.

Brooks, R. A. (1991).
Intelligence without representation.
Artificial Intelligence, 47:139–159.

De Wolf, T. and Holvoet, T. (2005).
Emergence versus self-organisation: Different concepts but promising when combined.
In *Engineering Self-Organising Systems: Methodologies and Applications*, volume 3464, pages 1–15. Springer Berlin Heidelberg.

DeAngelis, D. L. and Grimm, V. (2014).
Individual-based models in ecology after four decades.
F1000Prime Reports, 6(39):(on line).

Dignum, V., Dignum, F., and Meyer, J.-J. (2004).
An agent-mediated approach to the support of knowledge sharing in organizations.
Knowledge Engineering Review, 19:147–174.

Ferber, J. (1995).
Les systèmes multi-agents. Vers une intelligence collective.
InterEdition.

Ferber, J., Gutknecht, O., Jonker, C. M., Treur, J., and Müller, J. (2000).
Organization models and behavioral requirements specification for multi-agent systems.
In *4th International Conference on Multi-Agent Systems, ICMAS 2000,*, pages 387–388, Boston, MA, USA. IEEE Computer Society.

Hübner, J. F., Sichman, J. S., and Boissier, O. (2007).
Developing organised multiagent systems using the moise+ model: programming issues at the system and agent levels.
International Journal of Agent-Oriented Software Engineering, 1(3/4):370–395.

Jennings, N. R. (1996).
Coordination techniques for distributed artificial intelligence.
In O'Hare, G. M. P. and Jennings, N. R., editors, *Foundations of Distributed Artificial Intelligence*, pages 187–210. John Wiley & Sons.

Laird, J. E. (2012).
The Soar Cognitive Architecture.
MIT, Cambridge, MA, USA.

Laird, J. E., Newell, A., and Rosenbloom, P. S. (1987).
Soar: An architecture for general intelligence.
Artificial Intelligence, 33:1–64.

Macías-Escrivá, F. D., Haber, R., del Toro, R., and Hernandez, V. (2013).
Self-adaptive systems: A survey of current approaches, research challenges and applications.
Expert Systems with Applications, 40(18):7267–7279.

Minsky, M. (1986).
The society of mind.
Touchstone Book, Simon & Schuster.

Reynolds, C. W. (1987).
Flocks, herds, and schools: A distributed behavioral models.
ACM SIGGRAPH Computer Graphics, 21(4):25–34.

Roca, D., Nemirovsky, D., Nemirovsky, M., Milito, R., and Valero, M. (2016).
Emergent behaviors in the internet of things: The ultimate ultra-large-scale system.
IEEE Micro, 36(6):36–44.

Rodriguez, S., Gaud, N., and Galland, S. (2014).

SARL: a general-purpose agent-oriented programming language.
In *the 2014 IEEE/WIC/ACM International Conference on Intelligent Agent Technology*.

Smith, R. G. and Davis, R. (1981).
Frameworks for cooperation in distributed problem solving.
IEEE Transactions on Systems, Man, and Cybernetics, 11(1):61–70.

Taillandier, P., Gaudou, B., Grignard, A., Huynh, Q.-N., Marilleau, N., Caillou, P., Philippon, D., and Drogoul, A. (2019).
Building, composing and experimenting complex spatial models with the GAMA platform.
Geoinformatica, 23(2):299–322.