USD

Odense
06 August 2010

MAS Course 3

Yves Demazeau
Yves.Demazeau@imag.fr

SCHEDULE OF THE COURSE + EXAMINATION

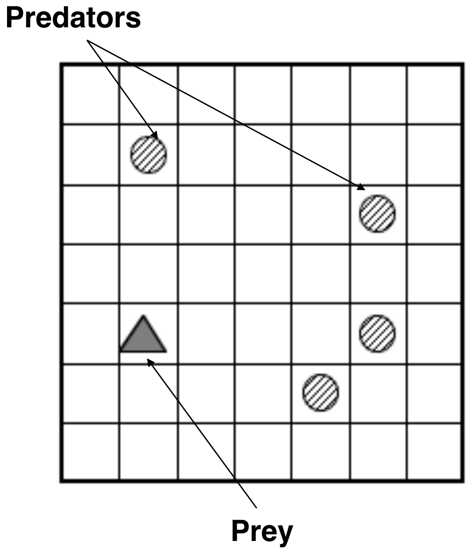
MAS 01	04 Aug.	Introduction, Methodology, Agents
MAS 02	05 Aug.	Agents, Environments, Interactions
MAS 03	06 Aug.	Dynamics, Organisations, Example
MAS 04	09 Aug.	
MAS 05	10 Aug.	
MAS 06	11 Aug.	

attendance ; handouts ; individual work
[Ferber 95] [HERMES 01] [OFTA 04]

MAS Ex.	13 Aug.	Written Control and Oral Delivery

Dynamics

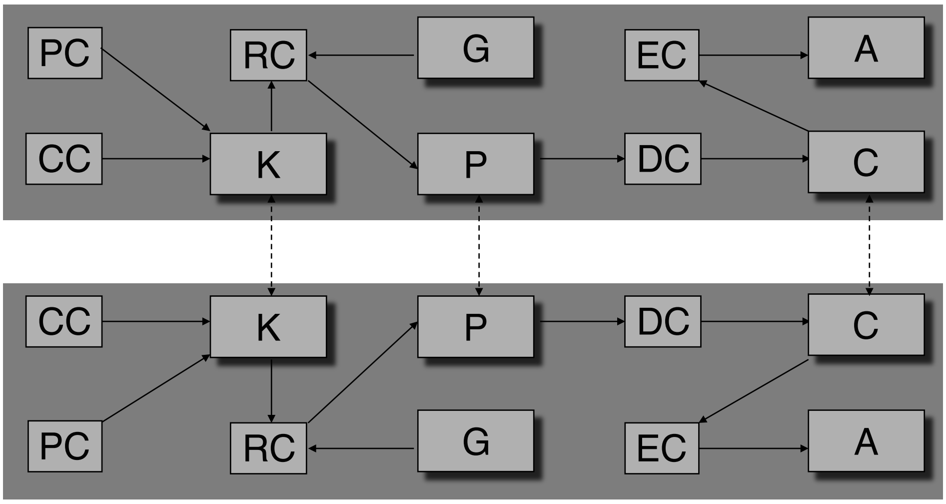
A classical example : The Pursuit Problem



Goal
To capture the prey by surrounding it with the predators

Problem
how to coordinate the predator actions

Example of Coordination Control



Examples and use of coordination

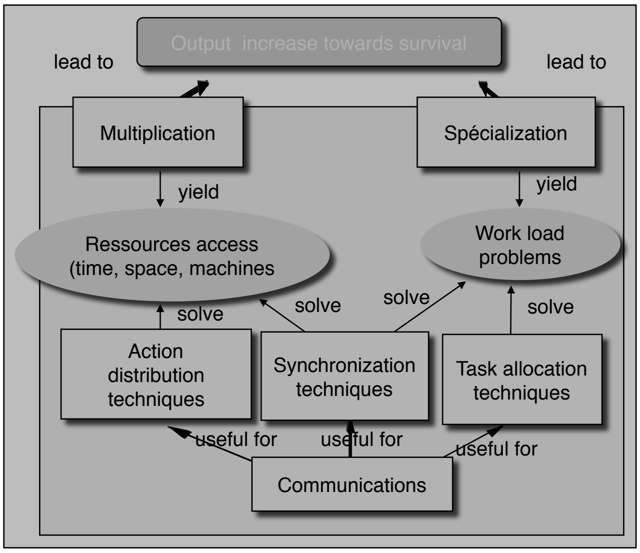
Allocation of tasks and resources
- Electronic commerce, Robotics, Energy distribution

Synchronisation
- Resource sharing

Action distribution (in time and space)
- Motion coordination, Distributed planning, Scheduling

Some usages
- Perform tasks that an agent cannot perform alone
- Increases productivity of agents by specialisation and scale economy
- Increases number of tasks performed in given time
- Optimize ressource usage by "smoothing" actions

Systems of coordination



Task Allocation

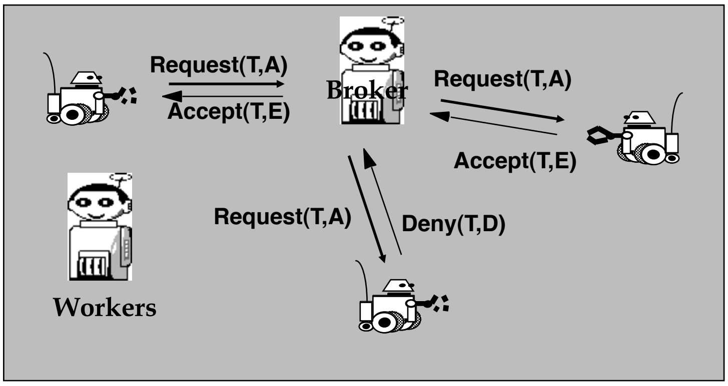
A mean to solve the question: who does what?

Different types of task allocation

1. Centralized allocation
2. Distributed allocation via accointance network
3. Distributed allocation via call for offer

Centralized allocation

Use of a broker to centralize requests and offers



Allocation via Accointance Network

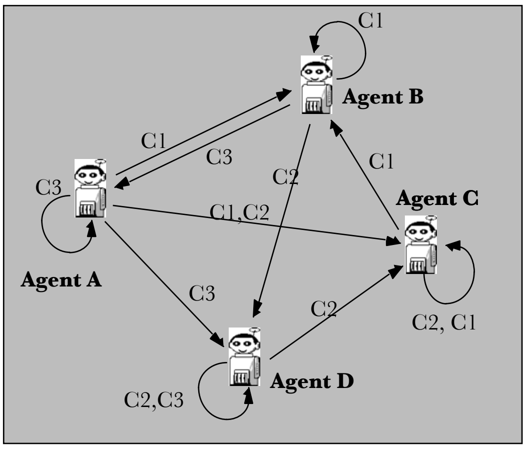
Each agent "knows" the skills of a set of agents

Direct allocation

- An agent request a service to agents it directly knows

Indirect allocation

- Speech acts are propagated through the network
- Uses algorithms adapted from distributed operating systems classics



Distributed Allocation via Call for Offer

The Contract Net : the most widely known algorithm in DAI [Smith-87]

Uses the protocol of state markets and two types of agents (two roles)

- Managers and Bidders

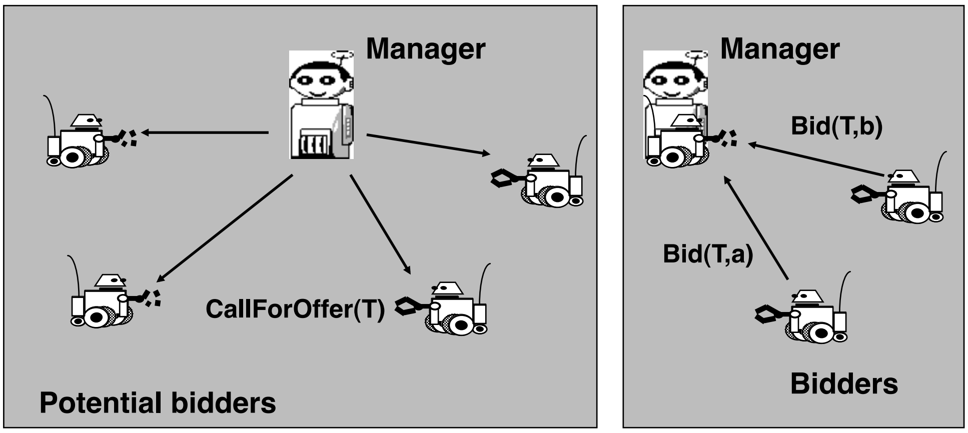
It is realized in 4 steps : Call for offer, Offers (bids), Awarding contract, Contracting and work

Problems with the Contract Net

- Subcontractors and commitment
- Deadlines
- Multiple managers and optimality

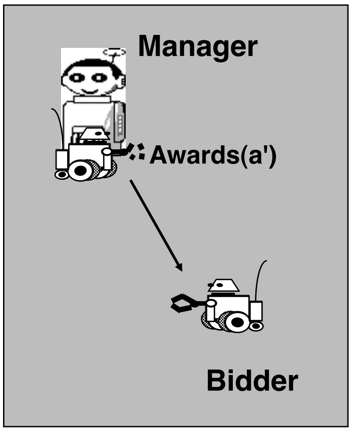
Contract net protocol: steps 1 and 2

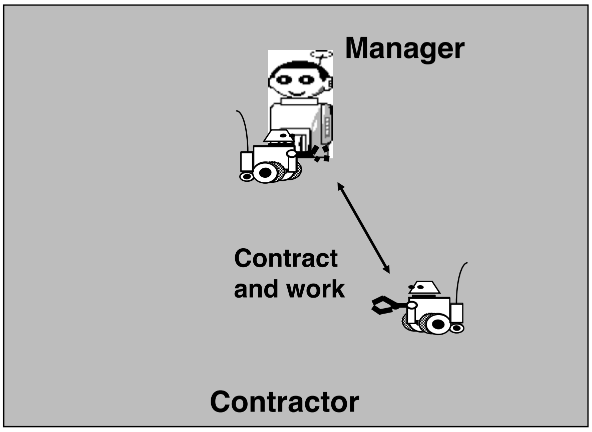
Call for offer **Offers (bids)**



Contract net protocol: steps 3 and 4

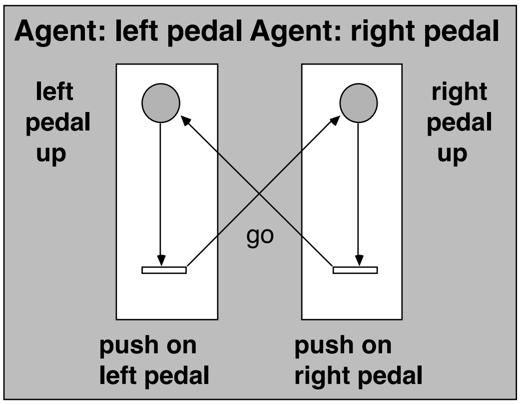
Awarding contract | **Contracting and work**





Synchronization (1) : Synchronisation of Motion

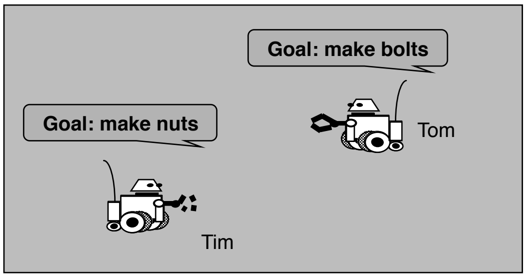
The cyclist example



Synchronization (2) : Access to a Resource

behavior Tim
- go to the machine
- places metal in machine
- makes a nut
- carry nut to stock

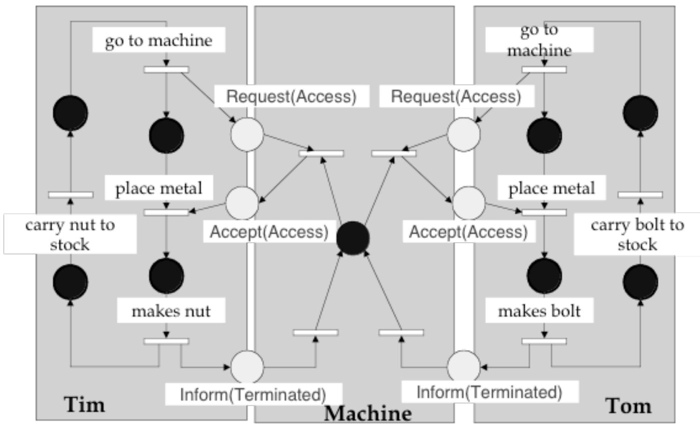
behavior Tom
- go to the machine
- places metal in machine
- makes a bolt
- carry bolt to stock



Pb: conflict between Tim and Tom to access the machine

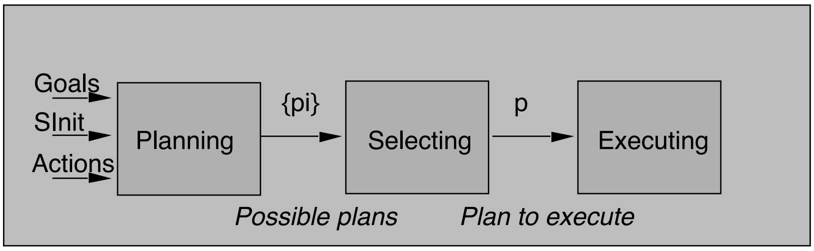
Synchronization (2) : Access to a Resource

Consider the machine as an agent (at least as a process) which gives access to one agent at one time



Multi-Agent Planning

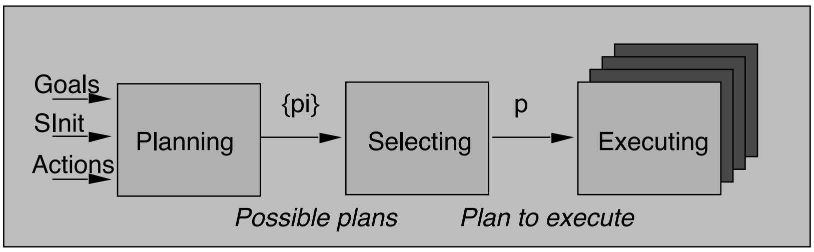
Find a sequence of operators Oi such that Sfin = On (... O2(O1(Sinit)) ...). Each operator is seen as a transition in a state space. The solution is obtained by finding a path from initial state to final state.



Centralized planning for multiple agents
Centralized coordination for partial plans
Distributed coordination for partial plans

Centralized planning (1)

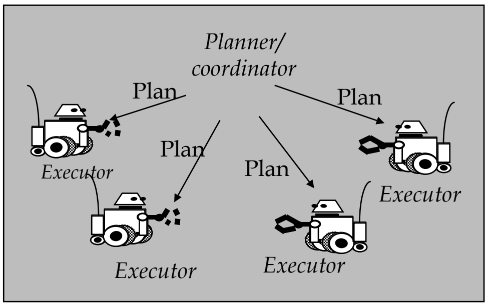
Central system which both plan and distribute plans among agents. Agents are just executors.

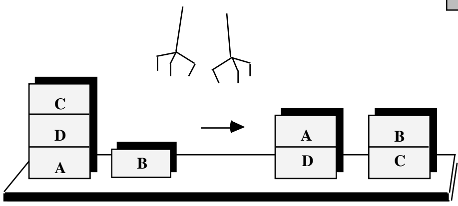


Coordination

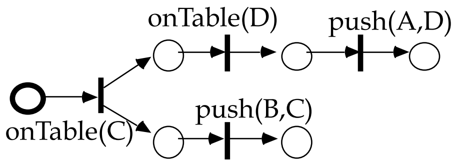
Centralized planning (2)

Planning for multiple agents
=
planning
+
coordination





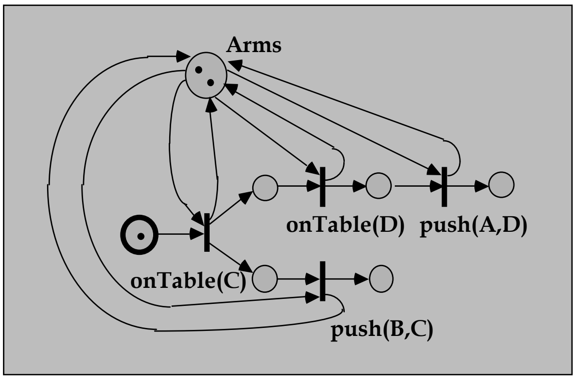
The problem



The plan

Centralized planning (3)

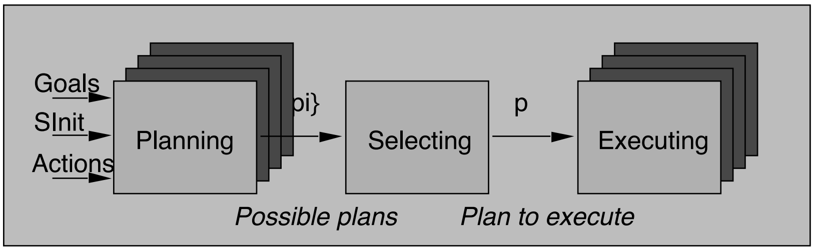
Coordination
=
resource allocation
+
synchronization



Each arm is considered as a resource and is represented as a token in the Petri net

Centralized coordination (1)

The planning process is distributed but coordination of partial plan is centralized

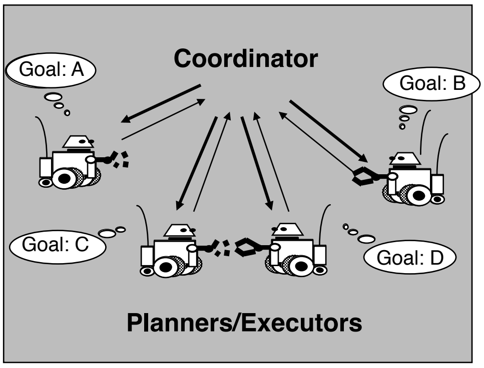


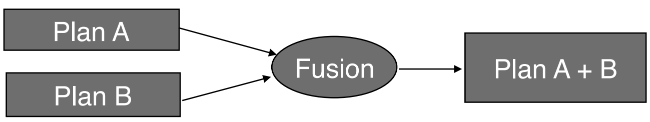
Centralized coordination (2)

Fusion of partial plans

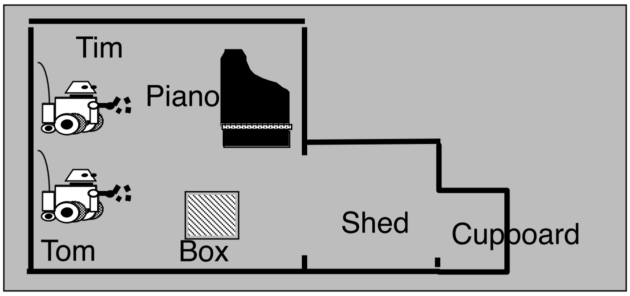
Partial plan

Coordinated plan





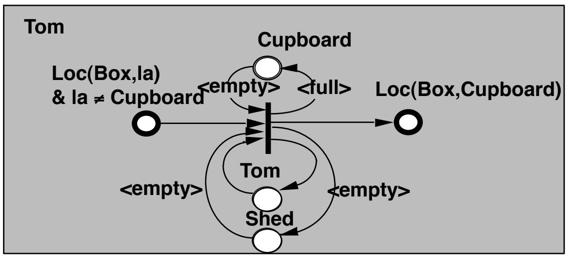
Fusionning partial plans (1)

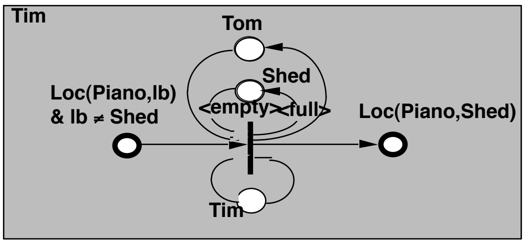


Operator: movePiano
nbAgents: 2
pre: Loc(Piano,la), Empty(Shed)
del: Loc(Piano, la), Empty(Shed)
add: Loc(Piano, Shed), Full(Shed)

Operator: moveBox
nbAgents: 1
pre: Loc(Box,lb), Empty(Shed), Empty(Cupboard)
del: Loc(Box, lb), Empty(Cupboard)
add: Loc(Box, Cupboard), Full(Cupboard)

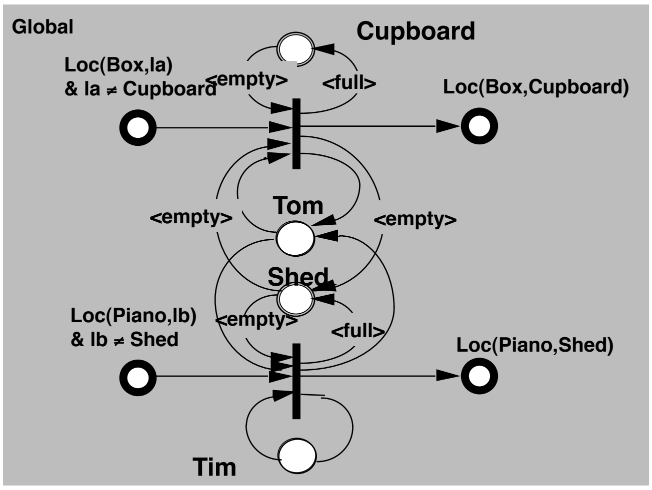
Fusionning partial plans (2)





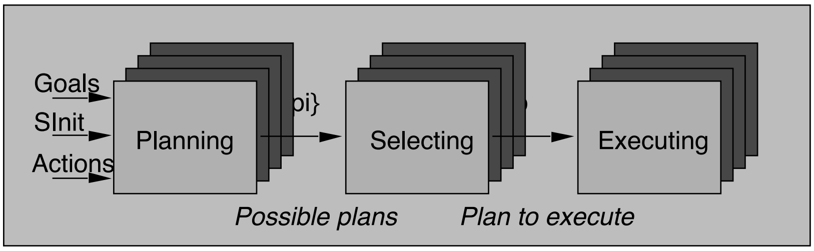
Fusionning partial plans (3)

Principle: stick plans together by identifying places



Distributed coordination (1)

Distributed planning and distributed coordination



Distributed coordination (2)

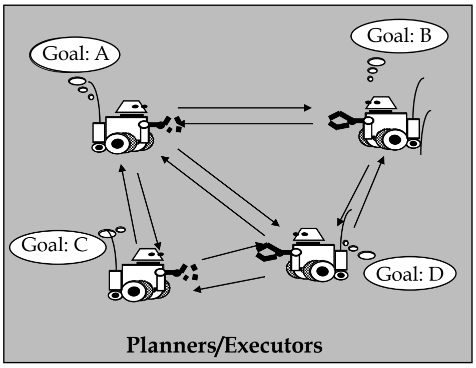
Two questions:

- Find the other agents with whom one should coordinate its plan
- When does it have to be coordinated

Problems

- Detect conflict and synergies
- Convergence of plans

  See for instance: PGP (Lesser, Durfee, Decker,..)



Request infos and partial plans

Distributed coordination (3)

Each agent produces a partial plan. Existence of possible conflicts between the different plans (more difficult situation, non exhaustively solved by now)

Possible solutions

- **One agent receives all the partial plans** (centralising, fusionning, synchronizing partial plans)
- **Every agent sends its partial plans to everybody** (each agent analyses the potential conflicts and identifies the conflicts with its own plans)
- **The partial plans are executed. As soon as some conflict occures during the execution, it is identified and handled** (which means that dynamic replanning and execution is possible)

Conflict resolution

Agents pursuing similar or different goals will have to face conflicts : • resource accessibily • alternative solutions • conflicting interests or goals

Example of conflict resolution techniques :

- **A priori solution** using strength, authority, ...
- **Mediation** by a third agent which knows about the different points of view, and tries to solve the conflict
- **Negociation** agents in conflict enter a transactional phase (exchanges, compromises, persuasive arguments, disagreement with the compromise or argument, requests for additional information, reasons for disagreement, utilities / preferences for the disagreed-upon issues) in order to reach an agreement, i.e. an equilibrium state
- **Flip a coin**

Negociation

Negociation structure

- **step 1 :** A propose a solution
- **step 2 :** B evaluates the solution, determines its satisfaction
- **step 3 :** If B is satisfied, ok, otherwise B propose another solution with regards to its own goals and constraints
- **step 4 :** goto step 1 exchanging A and B roles

Negociation control

- **by compromise :** each party relaxes its weekest cosntraints. The solution is found as soon as every constraint is satisfied
- **by integration :** each part tries to induce the deep goals of the others and then tries to find a solution which will satisfy these deep goals, even not the fully surface solutions

Reactive coordination and Situated Actions

Agents are situated
- they live in a "here and now" world

No planning
- reaction of agents depends only of their perception

Goals and informations are in the environment

Behavior is related to the state of the environment
- no mental states!!
- not even internal states in pure situated actions

Rule of action
- If <internal state> and <perceived state> then <action>

Other Agents may be perceived differently from the environment but in the same terms

Coordination through Marks Control

Behavior handling

Mode determination
Crumb handling
- if I carry a sample, I drop 2 crumbs
- if I carry no sample and crumbs are detected, I pick up one crumb

---

Obstacle avoidance
Path attraction
- if I am not carrying a sample and I sense crumbs, I move towards the highest concentration of crumbs

...

Reactive coordination and Force Fields

Force fields

- Forces are defined as the gradient of a potential field
  F(p) = -grad(U(p))
- Goals are represented as attractive fields
- Obstacles are represented as repulsive fields

Motion is obtained as a combination of attractive and repulsive forces

The resulting field U is defined as the sum of an attractive and repulsive field

$$U(p) = U_{attr(p)} + U_{repul(p)}$$

Importance of the environment

Coordination through Cohesion Fields

Force fields

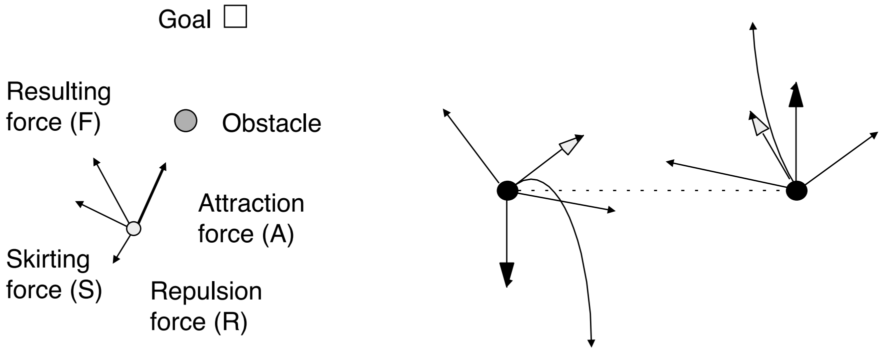
- Forces are defined as the gradient of a potential field
  F(p) = -grad(U(p))
- Goals are represented as attractive fields (external field)
- Obstacles are represented as repulsive fields (external field)
- Coordination is represented by cohesion fields (internal fields)

Motion is obtained as a combination of internal and external forces

$$U(p) = U_{inter(p)} + U_{exter(p)}$$
$$U(p) = U_{inter(p)} + (U_{attr(p)} + U_{repul(p)})$$

Collision avoidance in Sailing

Avoidance means remaining at a "good" distance of obstacles while going towards the goal



S(p) is such that : $dir(R(p)).dir(S(p)) = 0$

$F(p) = \alpha A(p) + \beta R(p) + \gamma S(p)$

ORGANISATIONS

Motivations

A trick to filter potential interactions
A way to control the increasing number of agents
A heuristics to improve systems performance
A guarantee for the system to converge
A way to control MAS along a normative vision

A way to support and improve cooperation
A model of real societies organisations
A way to integrate MAS with human organisations
A complementary vision than the agent-centered
A grounding brick to MAS methodology

History (1)

-> 1992 : Introducing the concept

- 77 : Distributed Interpretation - Hearsay-II [Lesser 80]
- 80 : Contract Net [Smith 80]
- 83 : Organizational Self-Design - DVMT [Corkill 83]
- 87 : Organizational Structures [Pattison 87]
- 89 : Organization Knowledge - MACE [Gasser 89]
- 90 : Roles and Social Structure [Werner 89]

1990 -> 2000 : Settling the concept

- 92 : Organisations and Coordination [Bouron 92]
- 92 : Social Laws [Shoham 92]
- 93 : ASIC [Boissier 93][Ricordel 99]
- 93 : Conventions [Jennings 93]
- 94 : Dependence Networks [Sichman 94]
- 95 : AEIO [Demazeau 95]

History (2)

1990 -> 2000 : Settling the concept

- 96 : PopOrgs [Demazeau 96][Costa 96]
- 96 : Norms [Dignum 96] [Conte 99]
- 96 : Learning [Prasad 96] [Camps 98]
- 98 : Agents, Groups, Roles [Gutknecht 98]
- 98 : Rôles [Kendall 98] [Stone 98]
- 99 : Dynamics [Baeijs 98] [Van Aeken 99] [Kozlak 00]

1998 -> : Exploiting the concept

- 98 : Organisation Oriented Programming [Lemaitre 98]
- 98 : MADKIT platform (AGR) [Gutknecht 98]
- 99 : Institutions [Sergot 99] [Esteva 01]
- 01 : VOLCANO platform (AEIO) [Ricordel 01]
- 01 : MESSAGE methodology [Garijo 01]
- 01 : VOWELS [Demazeau 01]

Definitions

An arrangement of relationships between components, which results into an entity, a system, that has unknown skills at the level of the individuals [Morin 77] : emergence

An organisation is characterized by : a division of tasks, a distribution of roles, authority systems, communication systems, contribution-retribution systems [Bernoux 85] : norms

A decision and communication schema which is applied to a set of actors that together fullfil a set of tasks in order to satisfy goals while guarantying a global coherent state [Malone 87] : design

Types and Classes [Baeijs 96]

Types

- **Teams** : shared environment in which agents interact
- **Communities of practice** : formation of groups independently of predefined schemas
- **SIGs** : gathering of agents sharing some interest
- **Markets** : common value sharing
- **Groups** : goal sharing, heterarchical decision
- **Hierarchies** : system sharing, hierarchical decision

Classes

- **Centralized** : simple hierarchies, multi-level hierarchies, recursive structures, …
- **Decentralized** : multiple hierarchies, Markets, Markets, …
- **Unstructured** : groups, teams, SIGs, communities of practice, …

From Mathematics [Boissier 93] [Stefanini 93]

Cognitive MAS in computer vision, NLP, HCI, to improve design issues in classical problems

O as predefined authorities or shared social laws, that translate the controller-controlled relationships between possible pairs of agents

O are implemented as explicit external rules

TALISMAN (academic project)

Traitement Automatique des Langues par un Système Multi-Agents Crisstal (F), LIFIA-CNRS (F)

Multi-level morpho-syntactic analysis of the written french minimizing ambiguities by interactions

Approach

- simplified Demazeau's COHIA agent model
- agents : preprocessing, morphological analysis, syntactical anaysis, segmentation into propositions, transformations, ellipses, coordinations, negations
- Sian's interaction protocols to initialize, to regulate communication exchanges, to realize agent control
- full implementation on Bim/Prolog - Sun WS

TALISMAN : The Society of Agents

Restriction of multiple solutions (usage of local grammars, interaction between levels)

- ---> cooperation between modules

Language variations, complex linguistic phenomena :

- ---> evolutive system : modification of strategy,

Introduction of new modules :

MORPH : morphology	TRANSF : transformation
PRET : preprocessing	COORD : coordination
SEGM : segmentation	ELLIP : ellipsis
SYNT : syntax	NEGA : negation
STAT : statistics	+ interaction protocols

TALISMAN : The coordination between agents

Interactions are based on both safe (not ambiguous) information (e.g. syntactical structures, morphological categories, ...) and uncertain information or linguistics heuristics

When new knowledge is validated by an agent, it modifies its local memory and restart its reolution process.

The agent may also update the interpretation of the sentence that is accessible to every component of the system

The system stops on its own when there is no further updating to be done on the interpretation

From Social Psychology [Sichman 94]

MAS in human sciences, to computationalize the Social Power Theory developped by Castelfranchi

O express social dependences wrt actions and resources, enable to build dependence networks forming coalitions

O is represented as a network of accointances and links

Dependency Networks (1) [Sichman 94]

Based on Castelfranchi 's theory on Social Power

Social Reasoning is performed on the basis of

- An explicit representation of the others
- An explicit reasoning about the others
- A belief revision mechanism about the others

Hypotheses

- Non-willingfullness
- Sincerity
- Self-knowledge
- Coherence

Dependency Networks (2) [Sichman 94]

Each agent evaluates its social (goals or resource) dependencies from the external description of other agents, in terms of goals, actions, and resources

social-autonomy (g) = action-autonomy (a,g)
and resource-autonomy (r,g)

- For a given goal, there is a plan where every action (resource) can be performed (controlled) by the agent

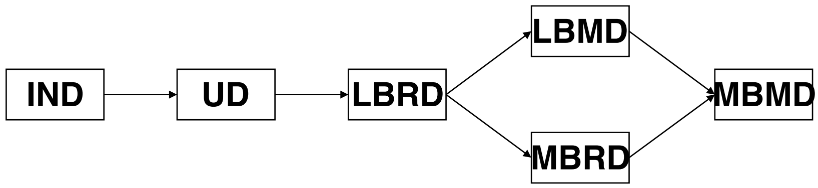
social-dependency (g) = action-dependency (a,g)
or resource-dependency (r,g)

- For a given goal, every plan involves an action (resource) of the agent is not able to perform (is not controlling)

Dependency Networks (3) [Sichman 94]

Dependency Relationships between two agents

- Independency (IND)
- Unilateral dependency (UD)
- Mutual dependency (**MD**)
  - ✓ For the same goal
- Reciprocal dependency (**RD**)
  - ✓ For different goals
- Locally believed dependency (**LB**)
  - ✓ A1 cannot infer the dependency from A2's description
- Mutually believed dependency (**MB**)
  - ✓ A1 can infer the dependency from A2's description



From Sociology [Costa 96] [Demazeau 96]

MAS in human sciences, to maintain the functional integrity of the system

O as states of organisations, to be matched with the set of interacting agents (population)

O are implemented as autonomous structures composed of organizational roles and organizational links

PopOrgs (1) : Populations [Demazeau 96]

The Population structure is the set of agents, the set of possible behaviors of the agents, and the set of all interaction processes between agents

Pop = (Ag, Bh, Ip; bc, ic)

Ag : set of agents
Bh : set of behaviors agents are able to perform
Ip : set of interaction processes
bc : Ag ---> P(Bh), behavioral capability,
bc(a), set of behavior a is able to perform
ic : Ag x Ag ---> P(Ip), interaction capability,
ic(a1,a2), set of interaction processes agents a1 and a2 may perform together

PopOrgs (2) : Organisations [Demazeau 96]

The Organization structure is composed of organizational roles and organizational links

Org = (Ro; Li)

Ro is defined in a relational way

- e.g. Ro $\subseteq$ Lp x Gp : global processes (Gp) and local processes (Lp), the role is the part of agent's behavior that is integrated in the global process.
- e.g. Ro $\subseteq$ Fo x Lv : foci of interest (Fo), representation levels (Lv), the role is the agent's behavior for a given focus at a given level.

Li $\subseteq$ Ro x Ro

PopOrgs (3) : Pop $\Re$ Org [Demazeau 96]

The suitable relation between the Pop and the Org is the system's organization implementation

It is any relation imp = Pop $\Re$ Org, on (Ro x Ag) $\cup$ (Li x Ip), Pop = (Ag, Bh, Ip; bc, ic), Org = (Ro; Li).

- if (r,a) $\in$ imp, r is said to implemented by a
- if (l,p) $\in$ imp, l is said to implemented by p

imp is said "proper" iff $\Re$ is an homomorphism.

- $\forall$ r $\in$ Ro, $\exists$ a $\in$ Ag / (r,a) $\in$ imp, and r is properly implemented by some behavior b $\in$ bc(a)
- $\forall$ l = (l1,l2) $\in$ Li, $\exists$ ip $\in$ Ip / { (l,ip) $\in$ imp $\wedge$ $\exists$ (a1,a2) $\in$ Ag x Ag / ip $\in$ ic(a1,a2), (r1,a1) $\in$ imp, (r2,a2) $\in$ imp, and r1, r2 are properly implemented by the behaviors of a1 and a2, respectively }

From Thermodynamics [Van Aeken 98 99]

MAS in WWW, to dynamically structure it, to optimize its social organisation

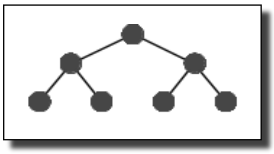
O as recursive pairs of agents looking like the same, to be permanently restructured to optimize balance and entropy

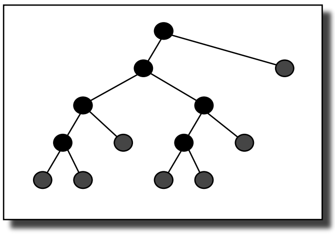
O are represented by non-ordered binary trees

Minimal Multi-Agent Systems (1) [Van Aeken 98]

Atomic agents and more complex agents

$\Lambda = \{ S \mid S = \Delta \text{ or } S = (G\ D) = (D\ G) \text{ with } G, D \in \Lambda \}$





Agents behaviour

- *« qui se ressemble, s'assemble »*

MAS behaviour

- *Size of a closed SMAM is constant over time*
- *Equilibrium of a closed SMAM is maximizing over time*
- *Entropy of a closed SMAM is maximizing over time*

Minimal Multi-Agent Systems (2) [Van Aeken 98]

Measuring SMAMs

- size
- equilibrium
- entropy

$$EQ(S) = \frac{E(S)}{\log_2(TR(S))}$$

$$EQ(\Delta) = 1$$

$$E(S) = \sum \frac{N(A)}{2^{N(A)}}$$

FRIENDS (industrial project)

A joint project between INPG and France Telecom

Agents behaviour

- *« qui se ressemble, s'assemble »*

MAS behaviour

- *Size of a closed SMAM is constant over time*
- *Equilibrium of a closed SMAM is maximizing over time*
- *Entropy of a closed SMAM is maximizing over time*

Applications

- Augmented SMAMs vs. pure SMAMs
  - introduction of symbols
  - Adding attributes
- Friends (Off-Line, On-Line, Final, Numbercruncher)

From Mechanics [Demazeau 93] [Baeijs 98]

Reactive MAS in image analysis, robotics, GIS, to extract solutions of unsolved problems

O as global structures to emerge from local interactions between agents of different classes, locally organized as groups sharing common features of interest

O are implemented as neighbourhood graphs linking agents

SIGMA (academic project)

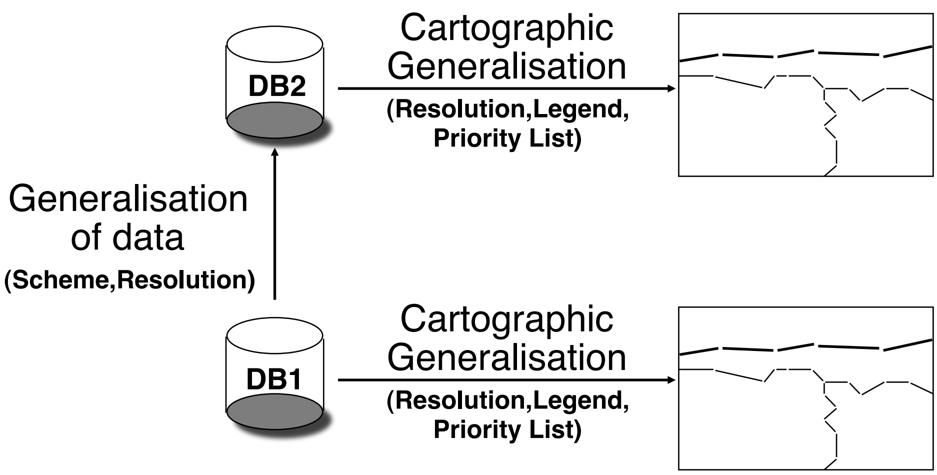
A reactive multi-agent approach to cartographic generalization LIFIA-INPG (F), IGN (F)

Interaction and organisation modelling to study their reciprocal interdependencies

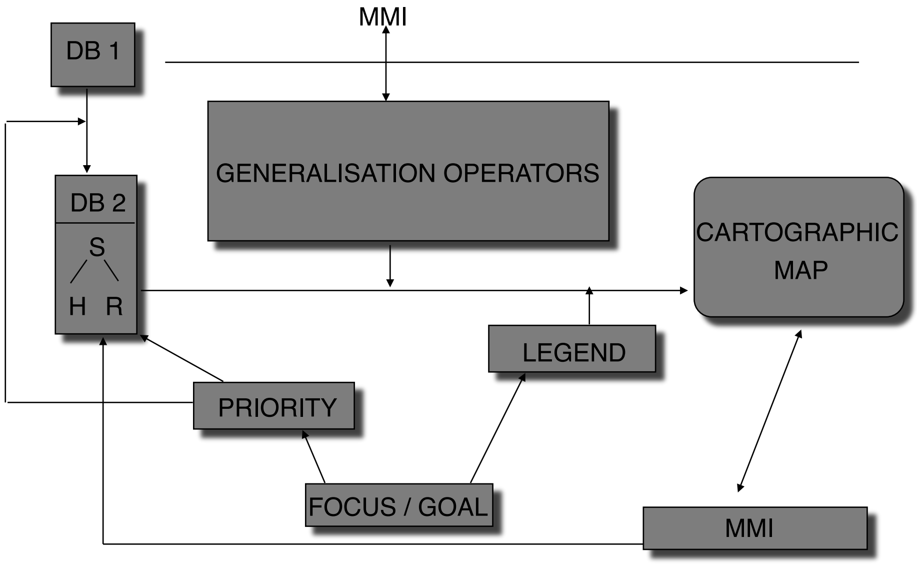
Approach

- following the PACO approach (multiple types + organizational knowledge)
- reaching the relative importance of data types according to a desired global goal
- operators to transforms the representations of the data and the possible changes of scale
- interactive validation
- Implementation on C/C++ on Sun WS - LAN/XENOOPS

SIGMA : Types of Generalisation



SIGMA : The Architecture of the System



SIGMA : Principles

Partial automatizing of cartographic generalization

- Creation of a readable and useful cartographic map from a geographical database given the aim of the map (pre-order) and using a non-holostic approach
- Modelling agents, interactions and organizational structures, and studying the convergence effects

Extension of the PACO paradigm

- Geographical objects are represented by a collection of "geographical entities" which "may" become agents
- Introduction of organizational knowledge to study their impact on a local level (behaviour of the agents) as well as on a global level (convergence of the system)

SIGMA : Model : E and A

Environment

- Geographical entities placed on a 2D grid, initially corresponding to the raw data (World of Reference)
- Active work on a copy (Active World) of the initial world to offer the opportunity to later geographical verification mechanisms

Agents

- A geographical entity becomes an agent as soon as its position in the organization (its mass) is important enough with respect to the aim of the map
- Each agent possesses several self-controled scopes:
  - ✓ Perception (local environment)
  - ✓ Communication (class, object, proximity, groups)
  - ✓ Action (class, object, proximity, groups)

SIGMA : Model : I and O

Interactions

- Between artificial agents (or objective groups)
  - ✓ Repulsion Force
  - ✓ Proportional Following (against local deformation of objects)
  - ✓ Unconditional Following (agents "sticking" together)
  - ✓ Change of symbolization
- Between the user and the agents (or subjective groups)
  - ✓ Change of symbolization
  - ✓ Formation or breaking of topological structures
  - ✓ Displacement of agents

Organizations

- Pre-orders, figuring "power"- relationship between geographical classes
- Groups, consisting of agents sharing the same local environment to realize a common task

SIGMA : Implementation and Results

Implementation

- Full implementation in C++ under Unix with acceptable results

Results

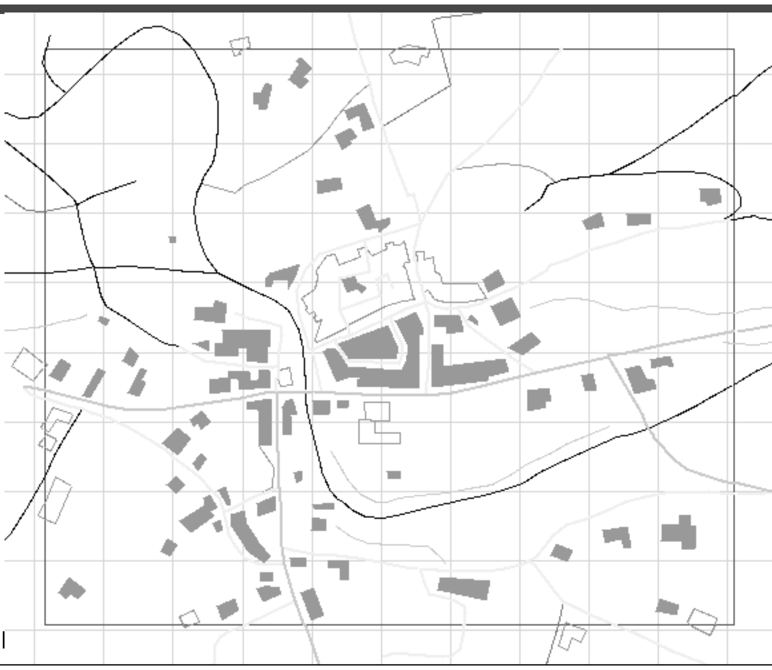
- "Les Matelles": • 300 objects -> 1800 geo. entities acceptable results in quality and computation
- "Neighbourhood of Les Matelles": • 2000 objects -> 15000 geo. entities • acceptable results in quality

Perspectives

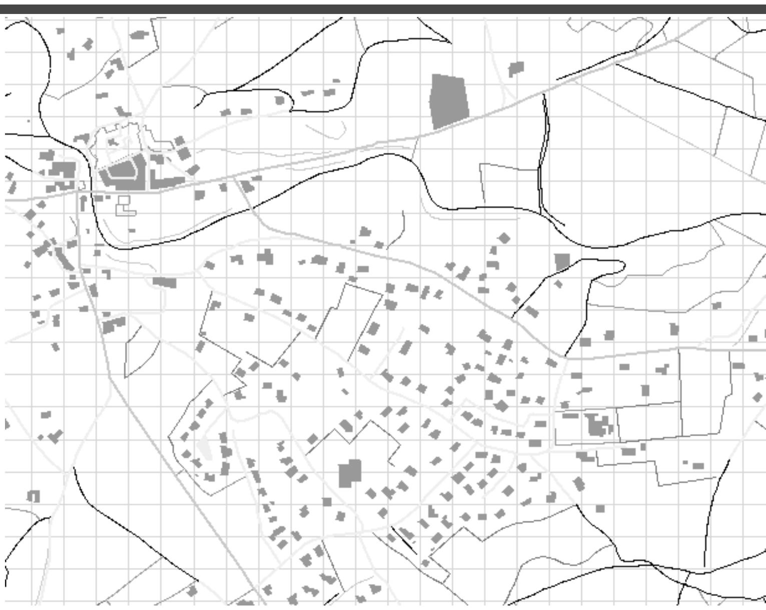
- Full explicitation of the organizational issues in the system
- Distribution/Parallellization of the system using XENOOPS
- Followup within the CEC-IT-LTR AGENT project

SIGMA : Les Matelles : initial map

SIGMA : Les Matelles : final map



SIGMA : Les Matelles Surroundings : initial



SIGMA : Les Matelles Surroundings : final