Tandil, 6 September 2000

Multi-Agent Systems Methodology

Yves Demazeau
Yves.Demazeau@imag.fr

Contents

Introduction : Multi-Agent Systems
MAS Analysis : A possible way of doing
MAS Design : An historical way of doing
MAS Models : The MAGMA example
MAS Development tools : MAOP
MAS Deployment tools : A critical Analysis
Comparizon with other Methodologies
Conclusion : The VOWELS Method

Introduction : Multi-Agent Systems

MAS Analysis : A possible way of doing
MAS Design : An historical way of doing
MAS Models : The MAGMA example
MAS Development tools : MAOP
MAS Deployment tools : A critical Analysis
Comparizon with other Methodologies
Conclusion : The VOWELS Method

What is an Agent ?

External Definition : a **real** or **virtual** entity that evolves in an **environment**, that is able to **perceive** this environment, that is able to **act** in this environment, that is able to **communicate** with other agents, and that **exhibits** an **autonomous** behaviour

---> autonomous agents

Internal Definition : a **real** or **virtual** entity that **encompasses** some **local control** in some of its **perception** , **communication** , **knowledge acquisition** , **reasoning** , **decision** , **execution**, **action** processes.

---> personal assistants, mobile objects, AI systems

But there is no agent without multi-agent systems !

Agents Environments Interactions Organisations

Agents
- internal architectures of the processing entities

Environment
- domain-dependent elements for structuring external interactions between entities

Interactions
- elements for structuring internal interactions between entities

Organisations
- elements for structuring sets of entities according to their roles in the MAS

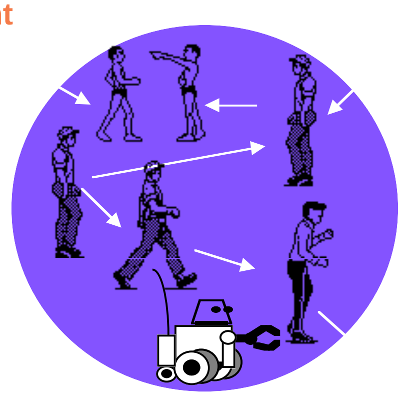
What is a Multi-Agent System ?

A set of possibly organized agents which interact in a common environment

MAS main interests :

To revise classical mono-agent AI models and tools (Agent-centered)

To study specific multi-agent models and tools (MAS-centered)



Multi-Agent System, Emergence, Recursion

The Declarative Principle
MAS = A + E + I + O

The Functional Principle
Function(MAS) = ∑ Function(entities)
+ Emergence Function

The Recursive Principle
entity = basic entity | MAS

MAS Micro and Macro Issues

Micro issues (Agent oriented)
- how do we design and build an agent that is capable of acting autonomously
- are oriented towards mental and environmental issues
- are typical of agent theories (Cohen & Levesque, Rao & Georgeff, Shoham, Singh, Wooldridge & Jennings, ...)

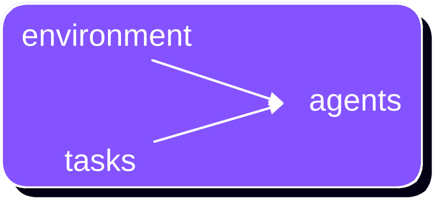
Macro issues (MAS oriented)
- how do we get a society of agents to cooperate effectively?
- are oriented towards interactions and organisations issues
- are typical of multi-agent theories (Durfee, Ferber, Gasser, Hewitt, Lesser...)

How to bridge between Micro and Macro Issues

Distributed Problem Solving

global conceptual model
global problem
global success criteria
division of :

knowledge
resources
control
authority



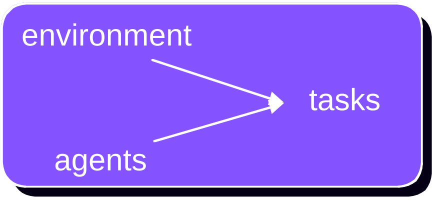
focus on the collaborative resolution of global problems by a set of distributive entities

society goals directed
input : tasks, environment
output : model of the distributed entities
schema to solve the tasks

Decentralized System Simulation

local conceptual models
local problems
local success criteria
division of :

knowledge
resources
control
authority



focus on the coordinated activities of a set of agents evolving in a multi-agent world

agent goals directed
input : agents, environment
output : tasks which can be solved
schema to solve the tasks

Domain Problem Characteristics

Natural decomposition of action, perception, or control, sharing of resource, environment, ...

No constraint about the heterogeneity of agents

Agents are perceived as being autonomous entities behaving rationally

No constraint about the grain of the agent model

Need for 3 or more coordinating agents or environments : interactions, organization, ...

Which applications are better handled by MAS ?

MAS methods cater for distributed intelligence applications : Network based, Human involved, Physically distributed, Decentralized controlled, ...

It suits when only local computational models are available whilst global ones are unknown

- Telecommunications, Internet Applications, Vision, NLP, ...

It is adequate for application domains and kinds of problem as soon as non-predictabiliy is acceptable

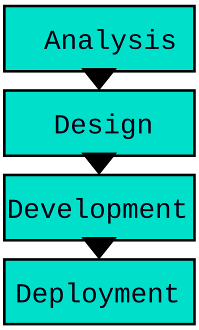
- Vision, Robotics, NLP, GIS, Societies Simulation, ...

It suits when the human is involved in the life cycle of a distributed system

- Internet Applications, Groupware, CSCW, GIS, ...

MAS Methodology

Methodology
= Approach + Model + Tools + Problem + Domain
= Analysis + Design + Development + Deployment



Identify the problem and the domain

Get rid of the domain / Define the solution

Implement the solution / Plug the domain

Apply the solution to the problem/domain

MAS domains and problems

...	**...**
Ecosystem	**Maintenance**
Electronic	**Business**
Entreprise	**Models**
Image	**Analysis**
Manufacturing	**Systems**
Natural Language	**Processing**
Network	**Monitoring**
Robotics	**Control**
Societies	**Simulation**
Spatial Data	**Handling**
Traffic	**Management**
...	**...**

How MAS Methodology is specific ? (1)

= Approach + Model + Tools + Problem + Domain
= Analysis + Design + Development + Deployment

It caters for distributed intelligence applications

...

MAS ANALYSIS : A possible way of doing

Introduction : Multi-Agent Systems

MAS Analysis : A possible way of doing

MAS Design : An historical way of doing
MAS Models : The MAGMA example
MAS Development tools : MAOP
MAS Deployment tools : A critical Analysis
Comparizon with other Methodologies
Conclusion : The VOWELS Method

Extrinsic Decomposition [Alvares 96]

Characteristics

- each agent is able to solve the whole problem
- the use of many agents in parallel speeds up the problem solving
- it is a purely physical (spatial or temporal) decomposition of the work between the agents

Examples

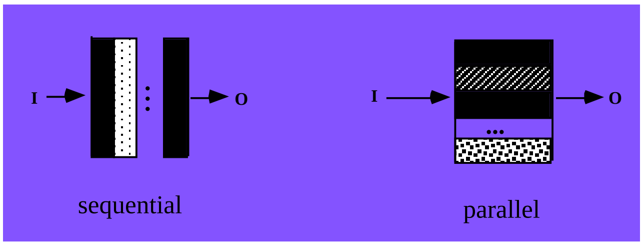
- there is an examination to be prepared by several professors. Each one wil be responsible to prepare a given number of questions (spatial)
- each professor will work for a given time (temporal)

Intrinsic Decomposition [Alvares 96]

The decomposition is based on a specialization

Two possible ways

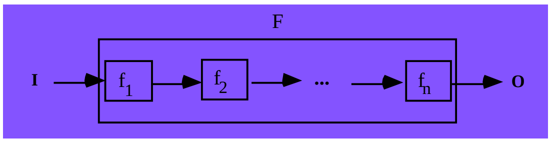
- to solve the problem partially for any case
- to solve the problem entirely for some cases



Sequential or Task-based [Alvares 96]

Exemple: to prepare an examination subject, we can divide the work in three subproblems

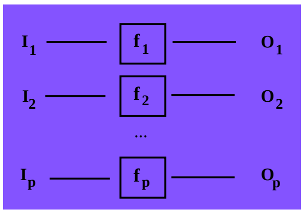
- to determine the number of questions by topic
- to really conceive each question
- to revise the questions



$F(I) \dashrightarrow O : f_n R ... R f_2 R f_1(I) \dashrightarrow O$,
where R is a temporal relation between the functions, and can be "precedes" or "succeeds"

Parallel or Domain-based [Alvares 96]

Example: to prepare an examination subject, we can imagine some domain division like by type of question (to fill in, discursive, multiple choice, ...) or by subject (topic)



$I = I_1 \cup I_2 \cup ... \cup I_m$, $O = O_1 \cup O_2 \cup ... \cup O_n$, $f_i(I_i) \dashrightarrow O_i$

Using many criteria (1) [Alvares 96]

The criteria are not mutually exclusive, we can combine them

At every level, the decomposition criteria are exclusive

Example: to prepare an examination subject

- Determine the number of questions and the respective value by topic (sequential)
- There will be people to prepare questions about topic t1 and people to prepare questions about topic t2 (parallel)
- In topic t1, there will be discursive and simple choice questions (parallel).
- There will be people to revise all questions (sequential)
- Each question will be revised for technical aspects and for linguistic aspects (parallel)

Using many criteria (2) [Alvares 96]

The problem is decomposed into :

- 1 determine topics 2 prepare questions 3 revise questions

The subproblem 2 is decomposed into

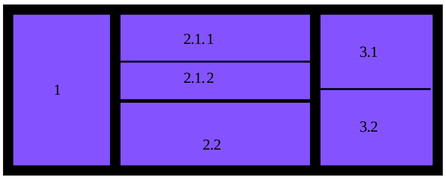
- 2.1 topic t1 2.2 topic t2.

The subproblem 2.1 is decomposed into

- 2.1.1 discursive questions 2.1.2 simple choice questions.

The subproblem 3 is decomposed into

- 3.1 technical review; 3.2 linguistic review.



Comparative Properties [Alvares 96]

	extrinsic	sequential task-bsd	parallel domain-bsd
ag's competence and behaviour	same	different	different
allowance of parallelism	yes	no	yes
allowance of ag's simplification	no	yes	yes
type of decomposition	quantitative	qualitative	qualitative
communication between agents	minimal	maximal	minimal

MAS DESIGN : An historical way of doing

Introduction : Multi-Agent Systems
MAS Analysis : A possible way of doing

MAS Design : An historical way of doing

MAS Models : The MAGMA example
MAS Development tools : MAOP
MAS Deployment tools : A critical Analysis
Comparizon with other Methodologies
Conclusion : The VOWELS Method

The COHIA Approach

Structuring the knowledge representation

- criteria : abstraction and decentration
- horizontal decoupling levels of representation
- vertical first-hand **interactions** : perception

Structuring the knowledge processing

- criteria : foci on space, time, features, models, tasks
- vertical decoupling into foci of attention
- horizontal second-hand **interactions** : communication

Identifying the basic entities of the system

- definition : intersection of level-agents & focus-agents
- choices : **agents**, **organisation**, **environment** models

Identifying the behaviour of the system

- System simulation : driven by the nature of the agents
- Problem solving : guided by the goals of the society

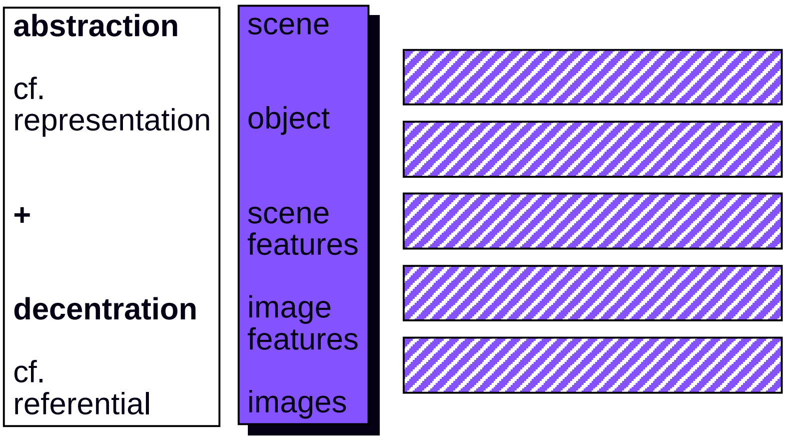
SATURNE : Origin of the studies

Building, maintaining, using a world description from data issued by several sensors
Building an open, domain-independent system

- Decomposing the knowledge representation problem into level-agents (cf. abstraction, decentration)
- Decomposing the knowledge processing problem into focus-agents (cf. focalisation / characteristics)
- Intersecting the level-agents and the focus-agents into basic agents
- Two behaviours to be exhibited by the society :

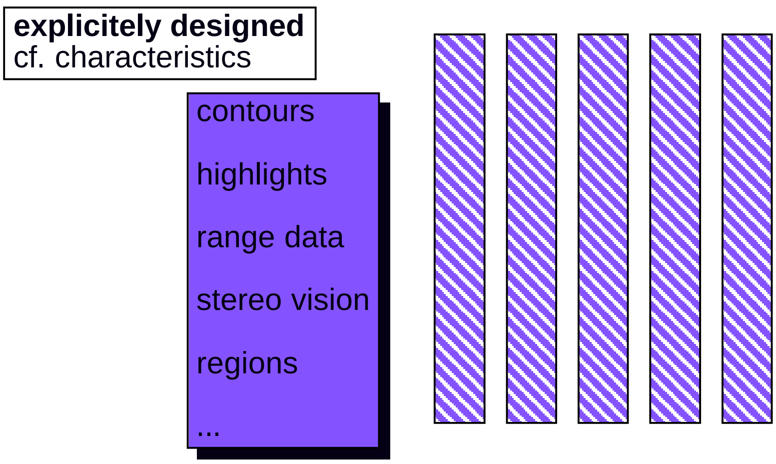
---> modelling : scene understanding
---> interpreting : recognition and localisation

SATURNE : Horizontal levels of representation



CNRS / Leibniz IMAG Y. Demazeau

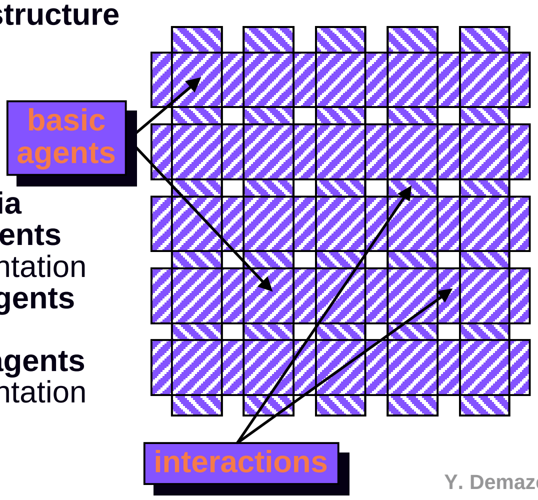
SATURNE : Vertical foci of attention



CNRS / Leibniz IMAG Y. Demazeau

SATURNE : Agents and Society of Agents

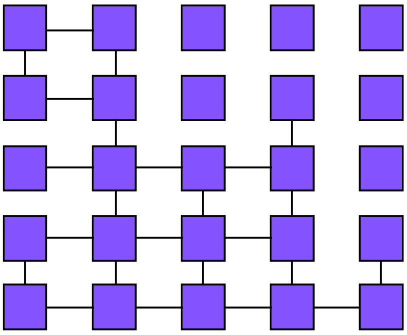
organisational structure
horizontal links
vertical links



interaction media
between foci agents
levels of representation
between level agents
foci of attention
between basic agents
levels of representation
x foci of attention

SATURNE Behaviour : Scene Understanding

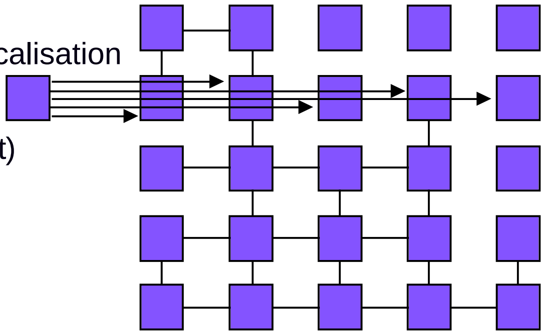
input
image
(environment)
basic agents
output
scene understanding
(global goals)



data driven
no explicit goal
no centralised representation
information exchange towards local coherence
decentralized system simulation

SATURNE Behaviour : Recognition Localisation

input
recognition, localisation
(global goals)
image
(environment)
output
basic agents



goal directed
explicit goal
purposive, centralised representation
information exchange towards global coherence
distributed problem solving

MAS Approach : Decomposing into Entities

A new approach to analyze and design SS

1. **MAS are situated, and the real environment differs from the perceived environment**
2. **The methods are mainly process-centered, but non-only task-based**
3. **The methods involve both declarative and computational specifications**
4. **The control is mainly decentralized, highly modular, it is distributed among entities and partly in an emergence engine**
5. **The entry point of the design is not unique nor imposed, even usually focused on Agents first**
6. **VOWELS decomposes the MAS into A, E, I, O**
7. **...**

How MAS Methodology is specific ? (2)

= Approach + Model + Tools + Problem + Domain
= Analysis + Design + Development + Deployment

It caters for distributed intelligence applications

It provides a new analysis and design approach

...

CASSIOPEE : General Issues

From the Analysis of natural organisations to the Design of artificial organisations

Based on several applications and experiments

Three Abstraction Levels

- individual agents, interactions, organizations

Agents is defined as a set of Roles

- individual roles, interactional roles, organizational roles

Lacks of Models and Tools

CASSIOPEE : Abstraction Levels

Agents

- Which architecture to choose to implement the agents ?
- Which scope of knowledge and how to best use it ?
- Which competences and how are they distributed ?

Interactions

- How do agents communicate ?
- Which content ?
- Can agents influence / alterate other's behaviour ?

Organisations

- How do the agents cooperate ?
- Is there a global goal, how to build a plan to reach it ?
- Which structure to organize, which evolution of the structure ?

CASSIOPEE : Composing Roles

Domain & Problem Dependent Typology of Roles

Individual Roles

Getting abstracted from the Domain
by Resource / Functional Dependence
(conflicts, permits, facilitates, needs, ...)

Problem based Typology of Relational Roles

Interactional roles (influencing, influenced)

Getting abstracted from the Problem
by Identification of Potential Groups
(SIGs, ...)

Typology of Organizational Roles

Organizational roles (initiator, participant)

MAS MODELS : The MAGMA example

Introduction : Multi-Agent Systems
MAS Analysis : A possible way of doing
MAS Design : An historical way of doing

MAS Models : The MAGMA example

MAS Development tools : MAOP
MAS Deployment tools : A critical Analysis
Comparizon with other Methodologies
Conclusion : The VOWELS Method

The MAGMA models

Mathematics
- Maths : Logics, Graphs and Trees
- Maths : Geometry, Topology
- Maths : Analysis
- Maths : Algebra

Physics
- Physics : Mechanics, Statistical Mechanics
- Physics : Automata, Control

Other Sciences
- H&S Sciences : Social Psychology, Sociology
- H&S Sciences : Philosophy
- H&S Sciences : Economy
- N&L Sciences : Ecosystems

Models : Agents

Agents

- Maths : Logics : COHIA, ASIC
  - Boissier 96 - ASIC Architecture
  - Boissier 97 - ASIC Applied to Vision
- Maths : Graphs and Trees : SMAM
  - Van Aeken 98 - SMAM (cf. thesis)
- H&S Sciences : Social Psychology
- Physics : Mechanics : PACO, PACO+
  - Baeijs 96 - PACO Extension to multiple types
  - Ferrand 98 - Reactive Spatialized Agents
- Physics : Automata : SMARRPS
- Physics : Control : ASTRO
  - Occello 97 - Real-time agents
  - Occello 98 - Real-time organized agents
- H&S Sciences : Social Psychology
  - Chicoisne 99 - Rational Agents

Models : Environments

Environnements

- Physics : Mechanics + Maths : Geometry : PACO
- Maths : Geometry : SMARRPS, SIGMA, AGENT
  - Ferrand 97 - T&C Development environment (cf. thesis)
  - Baeijs 98 - Geographical Information (cf. thesis)
- Maths : Topology : SMAM
  - Van Aeken 99 - WWW structures (cf. thesis)
- H&S Sciences : Social Psychology
  - Pesty 97 - Cognitive Agents and Environments
- Natural Sciences : Ecosystems
  - Fianyo 98 - Temporal Issues for Simulation

Models : Interactions

Interactions

- Physics : Mechanics : PACO, SMARRPS
- Maths : Logics + H&S Sciences : Philosophy : IL, IL2
- Maths : Graphs and Trees : IL Interaction Protocols
  - Ferrand 96 - Negociation Protocols (cf. thesis)
  - Koning 98 - Protocol Design
  - Koning 98 - Protocols Prevalidation
  - Koning 99 - Formal Specification
- Maths : Graphs and Trees : Dynamic Interaction Models
  - Ribeiro 98 - Dynamic Interaction Mechanics
  - Ribeiro 99 - Passive and Active Mechanisms
- H&S Sciences : Social Psych. + Philosophy : Dialogism
  - Pesty 96 - From coaction to cooperation
  - Chicoisne 98 - Dialogism
  - Ricordel 99 - Multiple Agents Interactions
  - Pesty 99 - Simulating conversations

Models : Organisations

Organisations

- Maths : Logics : PACORG
  - Baeijs 96 - Kinds of and Representations
  - Baeijs 98 - Organised reactive MAS (cf. thesis)
- Physics : Mechanics : SIGMA
  - Baeijs 97 - Organised reactive MAS
- H&S Sciences : Social Psychology : Social Power
  - Sichman 96 - Dependence Networks
- Maths : Analysis + H&S Sciences : Economy : Markets
  - Kozlak 99 - Dynamic Organisations
- Maths : Graphs and Trees : SMAM
  - Van Aeken 98 - Organisational Dynamics

Models : Recursion

Recursion

- Maths : Graphs and Trees
  - Occello 97 - Agent centered
  - Van Aeken 98 - Organisation centered (cf. thesis)
  - Mezura 99 - A E I O centered

Models : Emergence

Emergence

- Physics : Mechanics : PACO, SMARRPS
  - Ferrand 98 - reactive dynamics
- Physics : Statistical Mechanics : PHAMUS, SMAM
  - Perram 97 - PHAMUS
  - Van Aeken 98 - Functional Integrity Maintenance
- H&S Sciences : Social Psychology : Social Power
  - Sichman 96 - social reasoning
- Maths : Algebra + H&S Sc. : Sociology : ((A + I) +O) + E)
  - Costa 96 - Functional Integrity Maintenance
- N&L Sciences + H&S Sciences
  - MARCIA 96 - Self-organisation
  - M.R.Jean 96 - Emergence and MAS

MAS Models : Modelling these Entities

New models supported by existing formalisms

1. At higher abstraction level than other existing methods, closer to natural human way of thinking and reasoning about systems, not only devoted to computer scientists
2. It does not supply any new formalism currently, but entities are formalized using existing formalisms like traditional logics, Petri nets, algebraic languages, design patterns,...
3. VOWELS As range from reactive to cognitive
4. VOWELS Es range from spatial to topological
5. VOWELS Is range from forces to speech acts
6. VOWELS Os range from groups to markets
7. ...

How MAS Methodology is specific ? (3)

= Approach + Model + Tools + Problem + Domain
= Analysis + Design + Development + Deployment

It caters for distributed intelligence applications

It provides a new analysis and design approach

It is supported by existing formalisms,

...

MAS DEVELOPMENT TOOLS : MAOP

Introduction : Multi-Agent Systems
MAS Analysis : A possible way of doing
MAS Design : An historical way of doing
MAS Models : The MAGMA example

MAS Development tools : MAOP

MAS Deployment tools : A critical Analysis
Comparizon with other Methodologies
Conclusion : The VOWELS Method

CNRS / Leibniz IMAG   Y. Demazeau

Programming Paradigms

1950's
- Machine and assembly language

1960's
- Procedural programming

1970's
- Structured programming

1980's
- Object-Based programming
- Declarative programming

1990's
- Frameworks, design patterns, scenarios, and protocols

CNRS / Leibniz IMAG   Y. Demazeau

Features of Languages and Paradigms

Concept	Proc. L.	Object L.	Agent L.
abstraction	type	class	society
building block	data	object	agent
computational model	procedure call	method message	perceive reason / act
design paradigm	tree of procedures	interaction patterns	cooperative interaction
architecture	functional decompos.	inheritance polymorph.	managers assistants,peers
modes of behavior	coding	designing and using	enabling and enacting
terminology	implement	engineer	activate

Overview of AOP framework [Shoham 93]

A complete AOP system will inculde three primary components

- a restricted formal language with clear syntax and semantics for describing mental state: the mental state will be defined uniquely by several modalities, such as belief and commitment
- an interpreted programming language in which to define and program agents, with primitive commands such as REQUEST and INFORM: the semantics of the language will be required to be faithful to the semantics of the mental state
- an "agentifier", converting neutral devices into programmable agents.

Interaction Oriented Programming [Huhns 96]

Motivations

- errors will always be in complex systems;
- Error-free code can be a disadvantage;
- Where systems interact with the real world, there is a power that can be exploited

Example : children forming a circle

- conventional approach: create a C++ class for each type of object, write a control program that uses trigonometry to compute the location of each object
- interaction-oriented approach: children approach is robust due to local intelligence and autonomy, write the program based on objects having attitudes, goals, agent models

IOP : Active modules, declarative specification, modules that volunteer, modules holdbelief about the world, especially about themselves and others

Organisation Oriented Programming [Lemaitre 98]

Designing, Maintaining, Using MAS utilize different integrative frameworks that include features to deal with agents, interactions, environments, ... MAS programming itself follows history of programming.

The most well-known effort towards MAOP is AOP [Shoham 93] ... IOP [Huhns 97] is an alternative...

OOP is another one [Lemaitre 98] ... EOP does not actually exist as a trend but looks like Artificial Life.

These approach respectively focus on Agents, on Interactions, on Organisations, on Environments, as being the respective basic bricks at the disposal of the designer / MAS / user...

Multi-Agent Oriented Programming

Not Object-Oriented Programming
- S = Objects + Message passing

Not Logic nor Expert Systems Programming
- S = Knowledge + Inference Mechanism

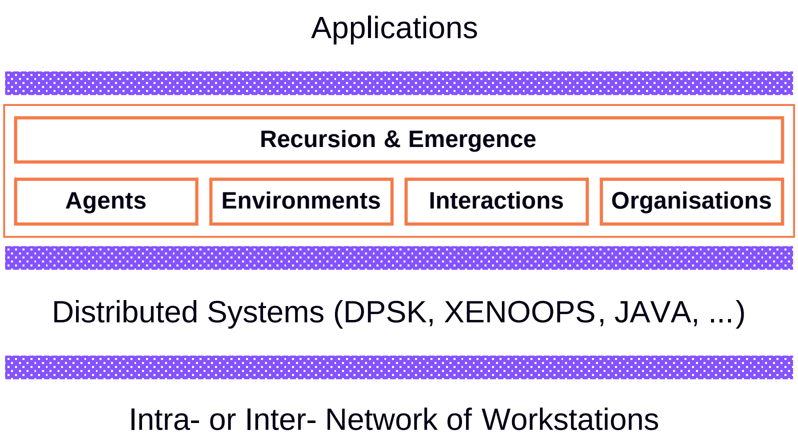
Not Ontology-Oriented Programming
- S = Knowledge + Problem Solving Methods

But Agent-Oriented Programming
- S = BDI Agents + KQML (Interactions)

But (((A + I) + O) + E)-Oriented Programming
- S = ((A + I) + O) + E)

But VOWELS Programming
- S = [A*; E*; I*; O*] + (Recursion & Emergence) Mechanism

The historical MASK tool



VOWELS Perspectives

Computational Equivalence (extending contingency ?)

- (((A + I) + O) + E) ?=? (((A + E) + I) + O)
- which semantics for the "(", the "+" as an operator
- which computational equivalences ?
- which possible pairs ?
- which possible recursions ?
- which contraints imposed on A, E, I, O ?

Domain Dependence (extending STS perspective ?)

- (((A + E) + I) + O) Computer Science
- (((E + A) + I) + O) Life Science
- (((A + I) + O) + E) Social Science
- (((A + I) + E) + O) Cognitive Science
- (((O + I) + A) + E) Military Science
- (((O + I) + E) + A) Economic Science

MAS Tools : Developing these Entities

New tools integrating existing paradigms

1. MAS is not (yet?) an implementation model and MAS oriented tools are usually not specific
2. Agents themselves just begin to have their own languages
3. MAS Development relies on existing languages and programming paradigms
4. The trend of the work is towards Multi-Agent Oriented Programming, meaning programming MAS with MAS tools
5. The closest related tools for VOWELS seems be frameworks but are still under investigation from the computational point of view
6. ...

How MAS Methodology is specific ? (4)

= Approach + Model + Tools + Problem + Domain
= Analysis + Design + Development + Deployment

It caters for distributed intelligence applications

It provides a new analysis and design approach

It is supported by existing formalisms,

It integrates existing programming paradigms,

...

DESIRE : General issues

Design and Specification

- Complex reasoning systems in general
- Proposes a powerfull design tool
- A design approach more than an analysis approach

A Formal Framework

- Formal specifications to automatically generate a prototype

Interacting Components based

- Input/output components

Reflective

- reasoning
- architecture

DESIRE : A Specification Framework

Components Decomposition

- Components Hierarchy
- Primitive and composed components

Information Exchange between Components

- Information links for information flows (channels)
- different levels of dynamic interaction models

Sequencing of tasks

- a local control process in each component
  - rules set (facts)
  - required data (required interactions)

Hierarchical knowledge structures

- adapted to components granularity

DESIRE : Modeling Agents

Models

- Agents as composed components
- Modeling of specific types of Information Exchange
  - more communication than interaction
  - MAS interaction = components interaction
  - interaction is embedded in components

Approach

- A task based approach (functional)
  - no explicit AEIO models

Design

- An agent centered approach
  - no external expression of interaction
  - no external expression of organisation

MAS DEPLOYMENT TOOLS : A critical analysis

Introduction : Multi-Agent Systems
MAS Analysis : A possible way of doing
MAS Design : An historical way of doing
MAS Models : The MAGMA example
MAS Development tools : MAOP

MAS Deployment tools : A critical Analysis

Comparizon with other Methodologies
Conclusion : The VOWELS Method

MAS Advanced tools

Academics

- Firefly (MIT before Microsoft) (no more accesible)
- MadKit (LIRMM Montpellier - Ferber's group)
- Simula (II Porto Alegre - Alvares's group)
- dMARS (-> Jack, by Agent Oriented Software)
- ...

Industrials

- Voyager (ObjectSpace) - freeware (linked with OMG)
- JINI (Sun) - freeware
- Aglets (IBM) - freeware
- Javabeans (Sun) - freeware (based on components)
- Agentbuilder (Reticular) - freeware + product (AOP based)
- ZEUS (BT) - freeware product (FIPA compliant)
- ...

Qualification criteria

Four *qualities* for each stages:

- Completeness: quantity & quality
- Applicability: scope, restrictions
- Complexity: competence required, workload
- Reusability: reuse of previous work

16 criteria + availability & support

	Analysis	Design	Development	Deployment
Completeness				
Applicability				
Complexity				
Reusability				

Selected platforms

Platforms requirements :

- based on a strong academic model
- high quality software, well maintained
- cover as many aspects as possible of MAS
- cover the four methodological stages

AgentBuilder, Jack, Madkit, Zeus

- As of first semester 2000

AgentBuilder®

Developed by Reticular Systems Inc.

Grounded on Agent0/Placa BDI architecture
Almost all stages covered
Complete graphical tools
Limited to a single agent model

	Analysis	Design	Development	Deployment
Completeness	ontology	agent definition	behavoural rules	RT Agent engine
Applicability	universal	cognitive agents	AgentBuilder's BDI	Small societies
Complexity	OO, GUI	MAS design, GUI	logic prog., GUI	GUI
Reusability	ontology	protocols	agents	none

Jack™

Developed by Agent Oriented Software Pty.

Including the dMARS BDI model
Great versatility
Focus on the development stage

	Analysis	Design	Development	Deployment
Completeness	none	ident. of classes	Extended Java	manual
Applicability	n/a	Jack's BDI model	Any MAS	n/a
Complexity	n/a	Jack's BDI model	Java & logic prog.	n/a
Reusability	n/a	difficult	classes	n/a

MadKit

Developed by O. Gutknecht & J. Ferber, LIRMM

Based on the AALAADIN organisational model
Graphical multi-agent runtime engine
Good versatility
Light methodology, no BDI

	Analysis	Design	Development	Deployment
Completeness	none	Aalaadin, no sw tools	Pure Java	G-Box
Applicability	n/a	broad range	simple agents	small to large MAS
Complexity	n/a	intuitive	few code base	GUI
Reusability	n/a	design patterns	classes	dynamic reconfig.

Zeus

Developed by British Telecom

All stages covered, from analysis to deployment
Methodological and Software tools
Limited to a single agent model

	Analysis	Design	Development	Deployment
Completeness	role modelling	finding solutions	5 activities	tools, docs
Applicability	role oriented MAS	task oriented agents	Zeus agent model	debug, visualisation
Complexity	UML	design skills	GUI tools	GUI
Reusability	role models provided	reusable formalism	partial	agent reconfigur.

Pitfalls of current MAS offers

Completeness
- Much on development… nothing about analysis/design
- Much focus on approach… but poor technical aspects
- Nothing about deployment
- Every stage must be developed in the platform !

Applicability
- An agent platform…but not a multi-agent platform
- A generalisation of a specific multi-agent system …multi-domain, but single-problem platform
- Fixed models, and no way to escape
- The platform must be as versatile as possible !

Complexity
- The documentation is sparse
- You have to code a lot
- The user interface is unfriendly
- Understanding, (re)using the platform must be facilitated !

How MAS Methodology is specific ? (5)

= Approach + Model + Tools + Problem + Domain
= Analysis + Design + Development + Deployment

It caters for distributed intelligence applications

It provides a new analysis and design approach

It is supported by existing formalisms,

It integrates existing programming paradigms,

It is striving towards industrial quality,

…

Volcano

Developed by PM. Ricordel & Y. Demazeau, LEIBNIZ

A multi-agent platform to fulfil all these criteria

- Based on the AEIO MAS decomposition [Demazeau]
- Full analysis-to-deployment chain
  - Problem/domain decomposition
  - AEIO modelling
  - Open library of models (simplicity, versatility, reusability)
  - Intelligent deployment tools

But

- Still under development…
- To be fully evaluated...

COMPARIZON WITH OTHER METHODOLOGIES

Introduction : Multi-Agent Systems

MAS Analysis : A possible way of doing

MAS Design : An historical way of doing

MAS Models : The MAGMA example

MAS Development tools : MAOP

MAS Deployment tools : A critical Analysis

Comparizon with other Methodologies

Conclusion : The VOWELS Method

State of the art of MAS Methods

Univ. of Amsterdam, NL (DESIRE)
- Treur, ...

Univ. of Paris 6, F (CASSIOPEE)
- Drogoul, ...

Univ. of Grenoble, F (VOWELS)
- Demazeau, ...

AAII, AUS
- Kinny, ...

RMIT, AUS
- Kendall, ...

Univ. of Stanford, USA (AOP)
- Shoham, ...

Univ. of Michigan, USA

Univ. Of Liverpool, UK

...

MAS vs. Systemic methods

Systemic Methods meaning...
- Information Systems

Characteristics of the Systemic Methodology
- data-centered
- centralized
- almost not modular

Characteristics of the MAS Methodology
- mainly process-centered
- decentralized
- highly modular

MAS vs. Formal methods

Formal (Specification) Methods meaning...
- Logics, Algebraic languages like Z, Automatas, Petri Nets, ...

Characteristics of the FS Methodology
- mainly used for validation
- include automatic generation

Characteristics of the MAS Methodology
- very low supported by a dedicated formal framework, but...
- ... possible use of existing formalisms to specify MAS components
  - logics-based approach [Fischer 94], [Huntbach 95], ...
  - Z, algebraic language approach [Luck 95], ...
  - Petri Nets approach [Elfallah 96], ...

MAS vs. Knowledge methods

Knowledge (Representation) Methods meaning...
- KADS, CML, KSM [Molina 95]...

Characteristics of the KR Methodology
- mainly declarative specifications
- control lays in the system inference engine

Characteristics of the MAS Methodology
- both declarative and computational specifications [Glaser 96], ...
- control lays in processing units and an emergence engine
  - (agent) control lays in the processing units [Occello 97], ...
  - (MAS) control lays in the system emergence engine, this engine involves the processing units with a recursion principle, whichever they are agents, environments, interactions, organisations [Demazeau 95], ...

MAS vs. Functional methods

Functional Methods meaning...

- SART, ...

Characteristics of the Functional Methodology

- task-based
- hierarchical
- decision as automata
- global context

Characteristics of the MAS Methodology

- non-only task-based [Alvares 97], ...
- hierarchical and possibly recursive [Occello 97], ...
- reactive and cognitive decision [Brazier 95], [Jonker 98], ...
- global and local contexts [Drogoul 98], ...

MAS vs. Object methods (start)

Object Methods meaning...

- OO analysis and design, modelling, implementation

Characteristics of the Object Methodology

- continuity Approach / Modelling / Implementation
- ...

Characteristics of the MAS Methodology

- no full continuity Approach / Modelling / Implementation
  - MAS is not (yet?) an implementation model
  - Agents just begin to have their own languages [Shoham 93], [Thomas 95], ... but the programming is not always based on Agents [Demazeau 97]
  - MAS design is based on existing languages and programming paradigms [Poggi 94], ...
  - towards multi-agent oriented programming [Demazeau 97], ...
- ...

MAS vs. Object methods (cont'd)

Characteristics of the Object Methodology

- object classes
- inheritance mechanism
- no organisation nor group primitives
- objects are built first, and then their dynamics
- ...

Characteristics of the MAS Methodology

- Agents, Environments, Interactions, Organizations [Demazeau 95], ...
- component groups, recursive mechanism [Fisher 94], [Kinny 96], [Occello 97], ...
- organisation and group primitives [Occello 97], ...
- entry point of the design is not unique nor imposed [Demazeau 97], ... even it often corresponds to agents
- ...

MAS vs. Object methods (end)

Characteristics of the Object Methodology

- environnement of an object does not exist, even if the environment of an object system does
- fixed Data Interaction Model
- global control, RPC mechanism,

Characteristics of the MAS Methodology

- MAS are situated, the real environment differs from the perceived environment [Moulin 95], [Kendall 95], ...
- free Data interaction Model [Demazeau 95], ...
- global (protocols) [Demazeau 95], [Koning 98], ... and local control (agent's decision) [Shoham 93], [Kendall 95], ...

MAS vs. Components methods (start)

Components Methods meaning...

- Components meaning JavaBeans, MS-COM, ...

Characteristics of the Components Methodology

- continuity Approach / Modelling / Implementation
- fixed Data Interaction Model between components
- no organisation nor group primitives
- components are built first, and then their dynamics

Characteristics of the MAS Methodology

- no full continuity Approach / Modelling / Implementation
- free Data interaction Model [Demazeau 95], ...
- organisation and group primitives [Occello 97], ...
- entry point of the design is not unique nor imposed [Demazeau 97], ... even it often corresponds to agents

MAS vs. Components methods (end)

Some common features between the methods

- introspection, persistence, mobility of basic entities
- event-driven communication between entities
- entities design and integration into applications

Characteristics of the Components Methodology

- customisation of entities at design time only
- existing de facto standards towards interoperability
- application independent reusable interoperable entities

Characteristics of the MAS Methodology

- possible dynamic allocation of roles during run time
- efforts to standardisation through the FIPA foundation
- still frequently application dependent entities

How MAS Methodology is different ? (start)

An enriched process-centered, decentralized, highly modular information system methodology

A currently poorly formalized formal specification methodology, reusing existing formalisms

An enriched knowledge representation methodology with computational specifications, a decentralized control and an emergence engine

An enriched functional methodology, not-only task-based, with possible recursion, cognitive decision, and local contexts

...

How MAS Methodology is different ? (end)

An enriched but incomplete object methodology

- **with extended classes (A, E, I, O), groups, organizations, recursive mechanism, and where the design is not always based on agents,**
- **with situated agents, free interactions, local control,**
- **where the programming is not always based on agents, but where no full continuity Analysis / Design / Implementation is not yet acheived**

An close component methodology, more flexible but still to be standardized

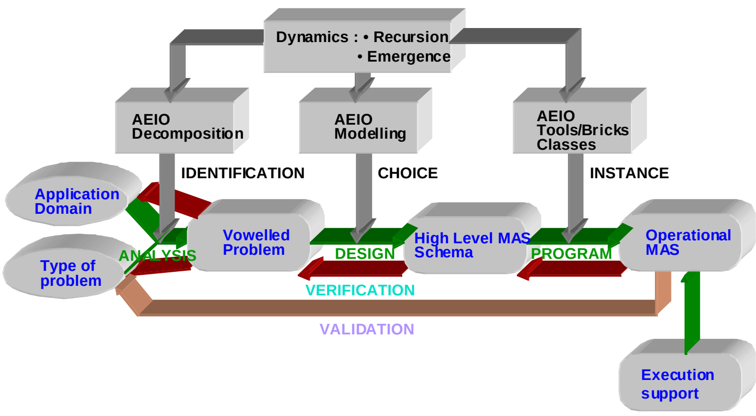
An enriched UML methodology which is not restricted to the design of systems

CONCLUSION : The VOWELS Method

Introduction : Multi-Agent Systems
MAS Analysis : A possible way of doing
MAS Design : An historical way of doing
MAS Models : The MAGMA example
MAS Development tools : MAOP
MAS Deployment tools : A critical Analysis
Comparizon with other Methodologies

Conclusion : The VOWELS Method

VOWELS General Approach



SIGMA (academic project)

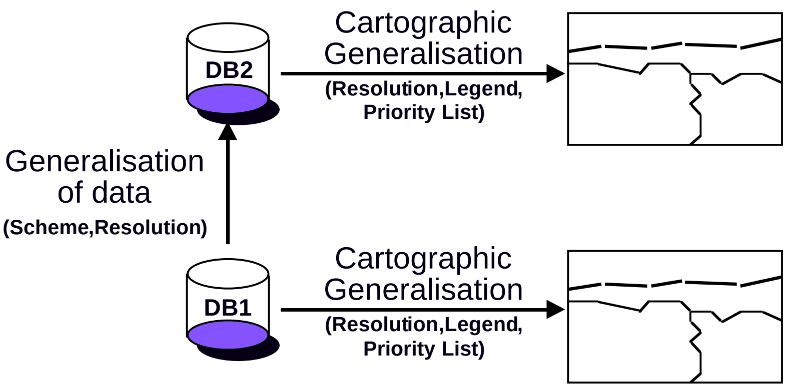
A reactive multi-agent approach to cartographic generalization LIFIA-INPG (F), IGN (F)

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 : 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

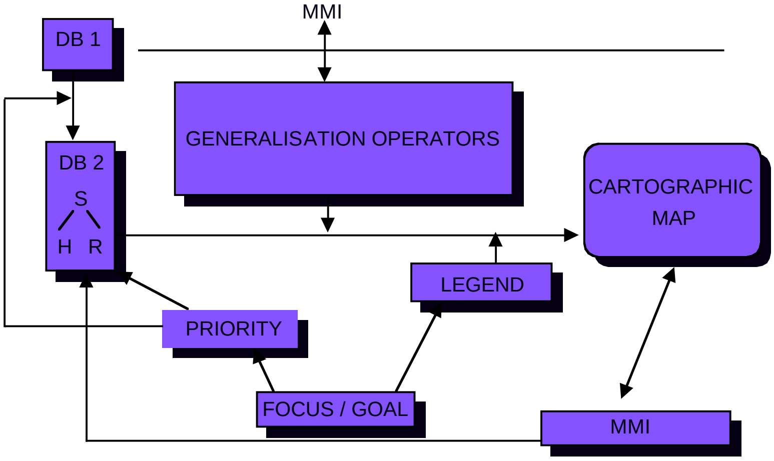
Interactions

- 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

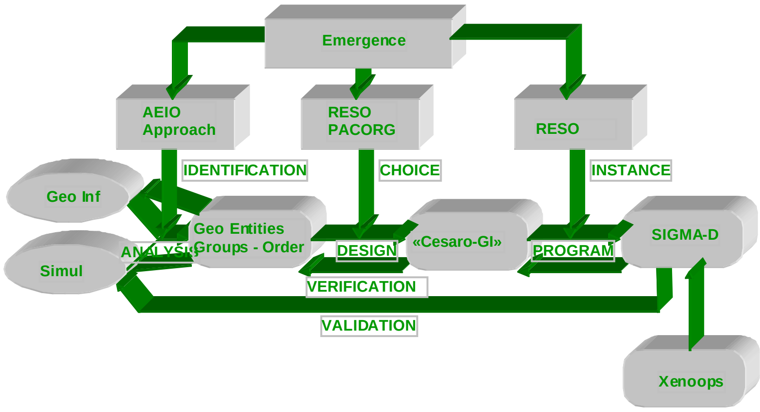
Organizations

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

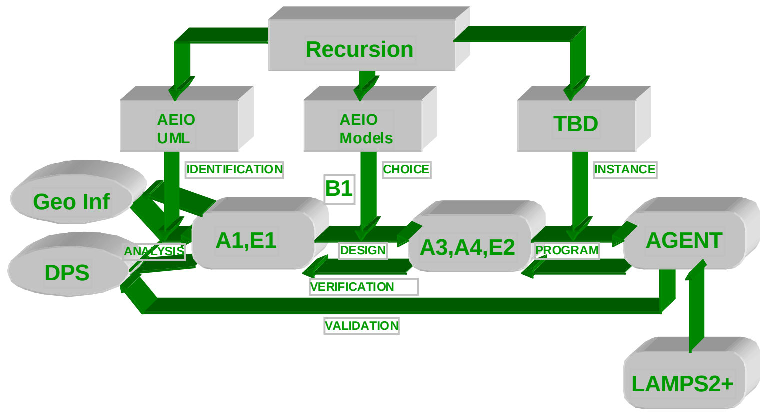
SIGMA : The Architecture of the System



VOWELS : SIGMA-D



VOWELS : AGENT



How MAS Methodology is specific ? (6)

= Approach + Model + Tools + Problem + Domain
= Analysis + Design + Development + Deployment

It caters for distributed intelligence applications

It provides a new analysis and design approach

It is supported by existing formalisms,

It integrates existing programming paradigms,

It is striving towards industrial quality,

It will always implies a possible non-provability.

The industrial impact of MAS

LES THEMES DES APPLICATIONS INDUSTRIELLES

L'IA a passé le flambeau à la modélisation multi-agent, IA distribuée, vie artificielle. L'approche multi-agent est au coeur de la conception de services et applications distribuées

Extrait du Rapport de Synthèse "Recherche Publique et Coopérations Industrielles dans le Secteur Informatique " établi par SPECIF, pour la Direction de la Technologie du MENRT - Juin 1999