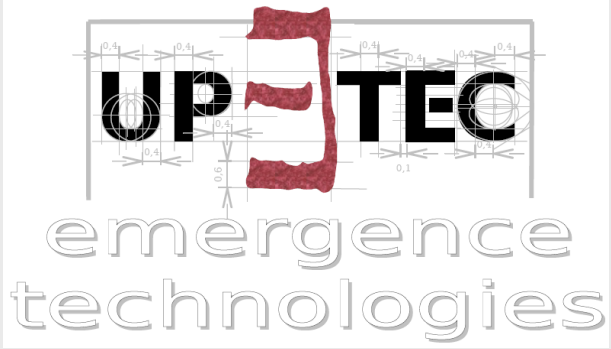


Reducing Complexity and Increasing Adaptivity by Engineering Self-Organizing Large-Scale Systems

Pierre Glize

pierre.glize@upetec.fr

http://www.upetec.fr

- IRIT laboratory spin-off
  - SMAC team www.irit.fr/SMAC
    - Multi-agent systems
    - Emergence
    - Self-organisation

- UPETEC Objectives
  - Diffuse into industry the emergent problem solving technology
  - Create engineers R&D jobs on adaptive complex systems

Outline

- Self* technology in Upetec
- Projects examples
- Challenge to academics
- Conclusion

Why emergence technologies?

- Technical motivations
  - More powerful computers (Moore's law)
  - Systems become more open (Internet, Ambient…)
  - Interdependencies, non linearities between them
- Theoretical consequences
  - Incompleteness theorems of Gödel
  - Law of requisite variety of Ashby
  - "No free lunch theorems" of Wolpert and Mac Ready

Emergence technologies for computer science

- We cannot formally prove that sufficiently « complex » software are safe
- Systems openess goes against their a priori checking

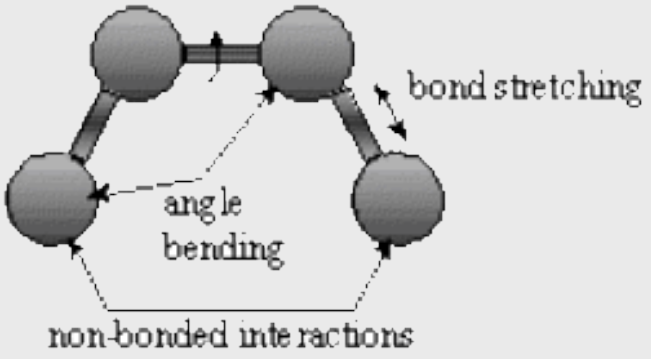
☞ Standard design is today inappropiate

☞ New systems must be self-* (self-control, self-repair, self-healing, self-organisation….)

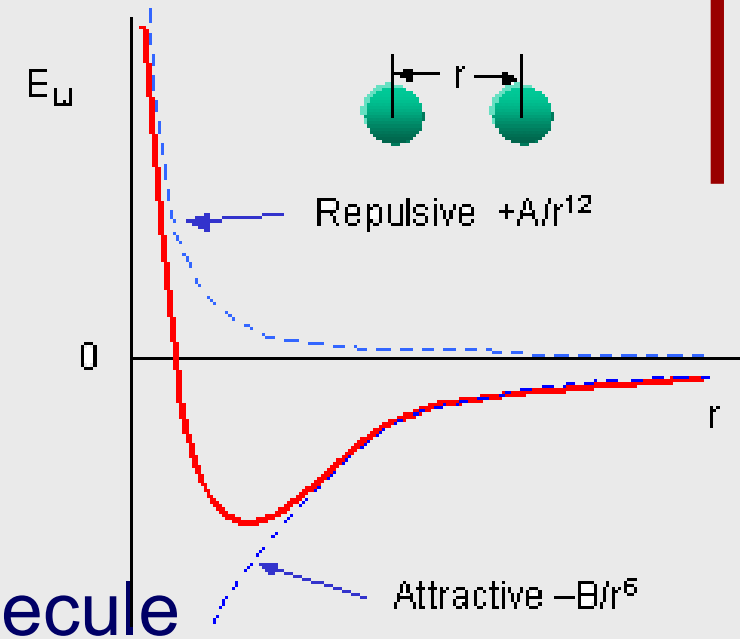
➔ New theories and methods
for designing emergent functionality systems

Protein folding

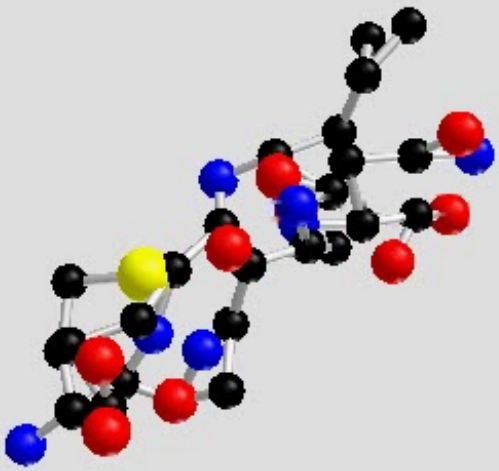
- Finding particular three dimensional shape
- An inadequate structure produces inactive proteins with different properties
- Set of atoms linked by two types of liaisons
- Strong interactions
  - Common electrons ➔ quasi-stable interatomic distances
- Weak interactions
  - Van der Waals
  - Electrostatics
  - Di-hydrogen
  - ….



Energy of weak interactions



- Potential energy function
  - Van der Waals interactions
    - Equilibrium (radius of van der Waals)
  - Lennard-Jones potential ➔
- Global potential energy of a molecule
  = Σ weak local energies
- Emergent protein folding
  - Respect to strong interactions
  - Only local computation of L-J potential
  - Minimisation of global potential energy ?



- Self-organising process avoids local minima
- Dynamic stability ➔ micro-fluctuations
- Computation time in $\mathcal{O}$(atoms number)

Basic Agents Characteristics

- An agent realises a function (not a global goal to achieve)
- All agents try to realise their own function according to their neighbors criticity

- Self* agents behaviors
  1. Self-tune the agents parameters
  2. Self-organise the agents
  3. Self-evolution of the system (agents adding/removing)

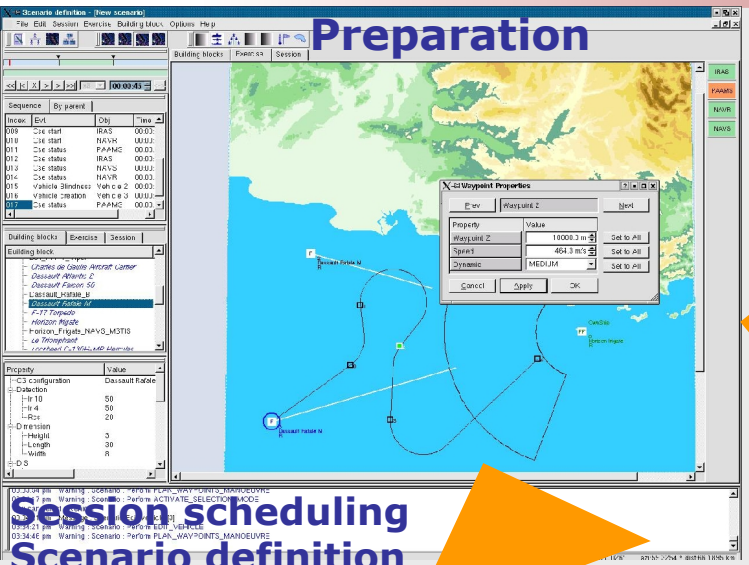
The collective function is not required

Emergence of the global behavior

Outline

- Self* technology in Upetec
- Projects examples
- Challenge to academics
- Conclusion

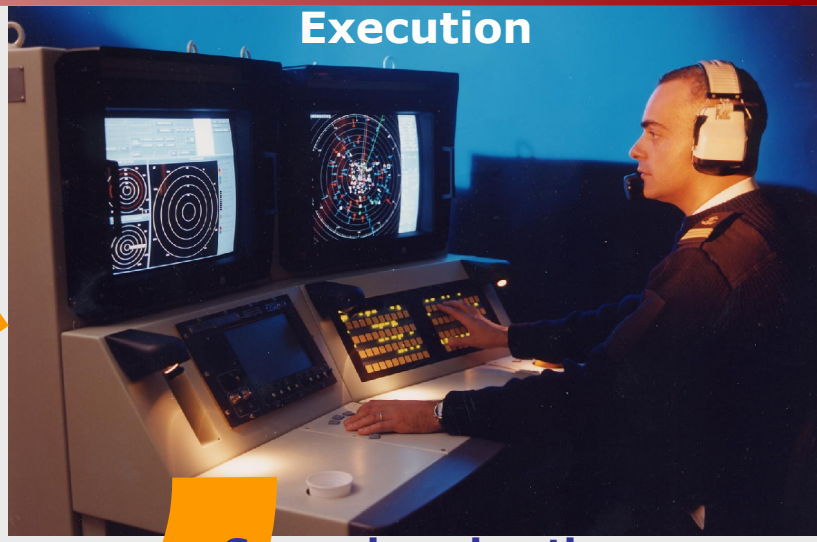
Adaptive Real-Time Strategy



Preparation

Session scheduling
Scenario definition
Objectives
- **Mission**
- **Assessment criteria**

Pre-requisite definition
- **CMS configuration**
- **Trainees selection**



Execution

Scenario animation
Training staff monitoring
Trainees under staff control

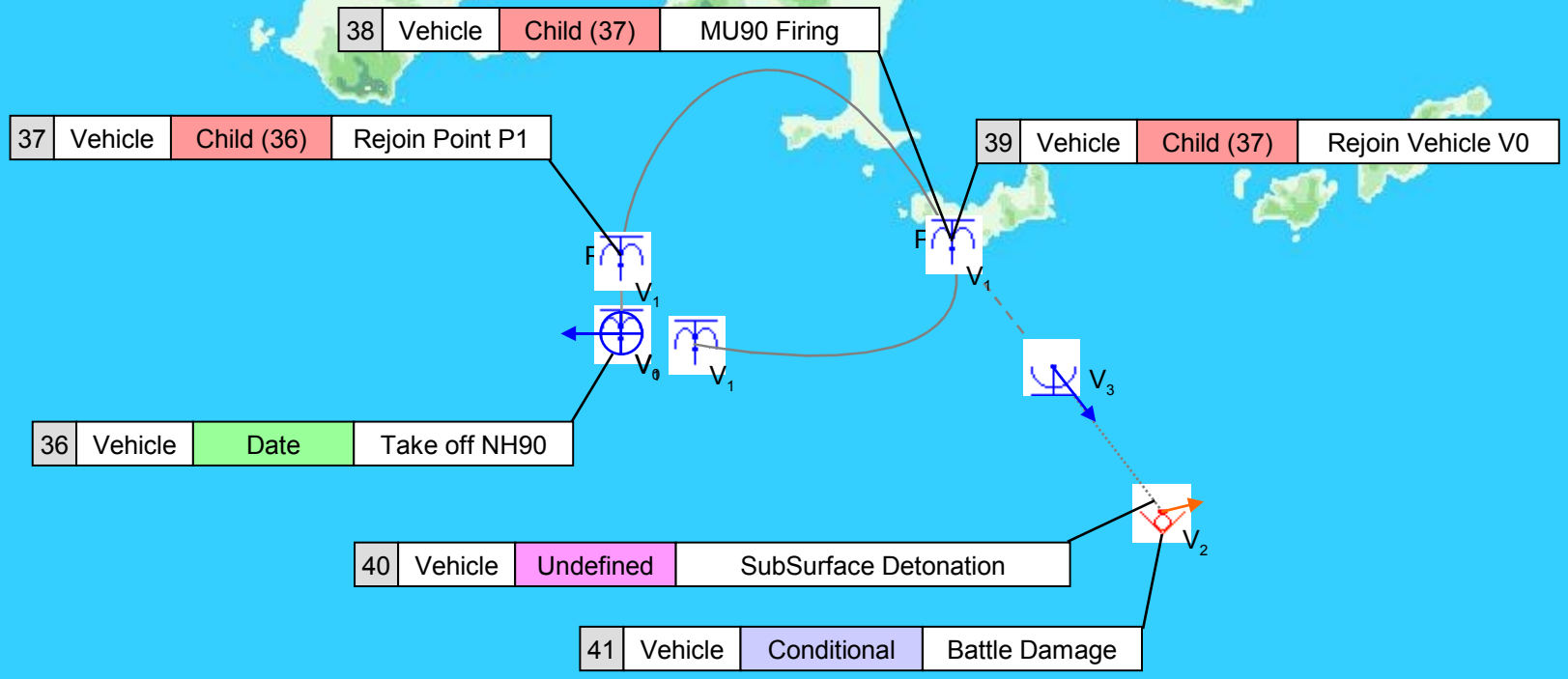
Re-entrance process

Enhancement trainees behaviour

Debriefing

On-line Exercise Replay
Recording off-line analysis
Assessment automaton facilities

Interception Mission



legend

Id	Class	Trigger	Name

-	Weather	Undefined	Sea Conditions Modifications

From Scenarii to Agents

- Hundred of agents
  - Two naval fleets are involved with several vehicles
  - A vehicle manages many plans (instantiated)
- Agents (plans and vehicles) have a quite coarse granularity

- Constraints
  - Real time execution
  - Collective dynamic behavior of a fleet
  - Relevant dynamic management of plans inside a vehicle

Vehicles Agentification

- Sending requests (with priorities) to neighbors
- Evaluating answers
- Sending agreements (according to priorities)
- Mixing individual and neighbors plans

Collective fleet optimal behavior

Without a global function cost

Scenario-independent

Plans Agentification

- A scenario is a sequencing of predefined plans
  - Unexpected events (trainee and trainer)
  - Self-organisation of plans
    - Cooperation leads to the activation of more « useful » plans
    - Not dependent of the number of plans
    - Environment dynamics lead to reconsider organisation

A more relevant planning than a predefined sequencing

Collective plan optimal behavior without a global function cost

Scenario-independent

Ontology Evolution

- French National Research Project (3 labs, 3 companies)

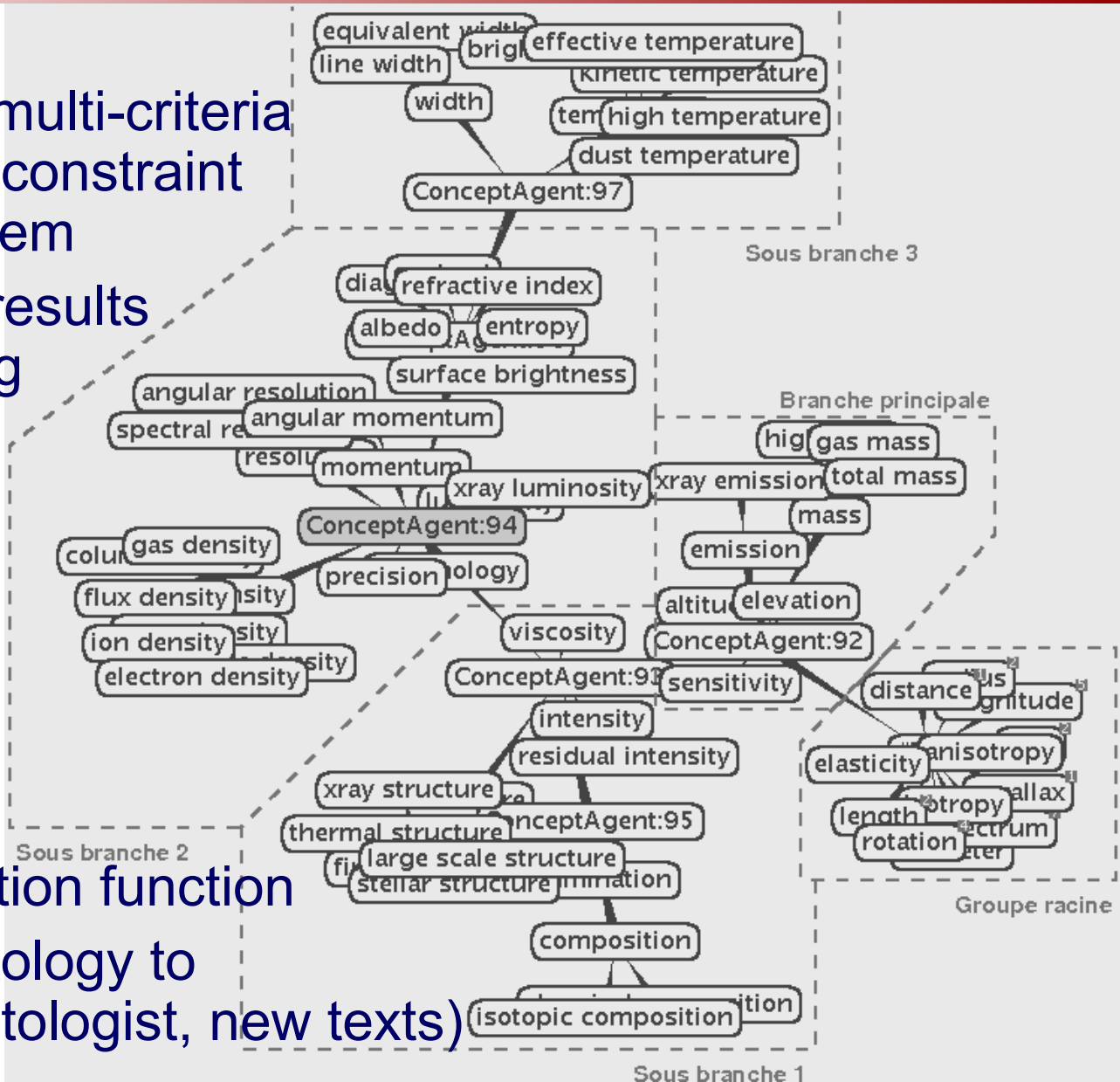
- Changes in a domain
  - Texts as knowledge sources
  - Users' needs & Ontologist focus
  - Applications

- Consecutive changes in the ontology
  - Conceptual structure (concepts, concept hierarchy, relations)
  - Instances of concepts, relations, terms

➔ Adapts ontology with reduced efforts

The MAS is the Ontology

- Multi-objective &multi-criteria optimisation and constraint satisfaction problem
- Process solving results from local self-org



- No global evaluation function
- Self-adapting ontology to perturbations (ontologist, new texts)

Outline

- Self* technology in Upetec
- Projects examples
- Challenge to academics
- Conclusion

Automatic Software Debugging

- Debugging, a lengthy and tiresome task (Billions $)
- Depends heavily of the programmer's skill
- Debuggers tools are passive

- Plug an adaptive agent to each component
- Detect and propagate local abnormal functioning

➔ Emergence of collective breakdown detection

An Academic Nightmare?

- A debugger is a complex system (Ashby's law)
- The organisation of the software system is unknown
- The roles of software components are unknown
- Real-time learning of the "Jacobians" components
- Real-time functioning of the MAS debugger

➔ Under these constraints

➔Learning "from scratch" without any a priori cost function

➔A true emergent functionality system using self-adaptive and self-organizing capabilities

Outline

- Self* technology in Upetec
- Projects examples
- Challenge to academics
- Conclusion

Concluding Remarks

- Upetec Business
  - Design of adaptive complex systems software
  - Works exclusively on applications requiring self-adaptation
  - Self-adaptation has three axis: tuning, organisation, evolution

- Upetec research
  - Time spent : 50%Analysis+Conception, 50% Specif-Dev
  - Business allows to maintain research works (such as Irit cooperation)

Concluding Remarks

- R&D domain on self*-systems is highly dynamic
- Defining today standards would be useful but quite constraining

Working on Self*-systems
to be more than a fashion effect
(such as AI in 80's) ?