# Improving dynamic skills searching in virtual social communities using agent' network self-organization

**Self-organized communities in virtual social network**
annabelle.mercier@iut-valence.fr
michel.occello@iut-valence.fr
jean-paul.jamont@iut-valence.fr

*LCIS Laboratory, Valence*
*Grenoble University, France*



## Outline

- Introduction
  - Goal & contribution
  - MultiAgents systems (MAS)
- Proposition
  - Agentification of the social network
  - MWAC Model
  - Simulation results
- Future works & conclusion

## Goals

- Skill searching in a large network
  - Access to the information with low effort, low cost
  - Incapacity to know the whole graph
  - Dynamic search
- Organizational structure
  - Social network representation
  - Minimizing the number of transmitted messages
  - Structure group detection (ease information access)



## Existing network structure representations

- Local interplays vs. Global behaviour
- Big networks
  - Modeling by mathematical objects
    - Small worlds
    - Random graph
  - Properties
- Adapted to network analysis and simulation
  - Theoretical methods
  - Sociological approach
  - Agent approach
- Static analysis vs. Dynamic use of the networks

➔ **structure representation is involved in the efficiency of experiments in social network**

## Our proposition

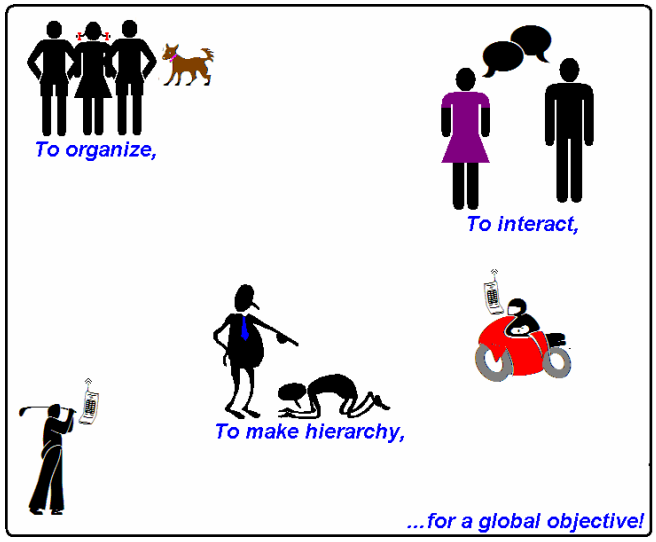
- From global to member point of view
  - Member
    - is a node
    - provides services & skills
    - can access services & skills from its neighbourhood
- Adaptative network structure representation
  - Existing static representation
  - Member evolution
    - Takes into account member changes
    - Adaptative network topology
  - Routing queries
- Our contribution
  - A tool to access the services or skills provided by the social network



## What is a MultiAgent System ?

- About an agent
  - Automous behaviour
  - Own goal
  - Environment
- MultiAgent : more than 2 agents in interaction
  - Distributed problem
  - Emergent collective behaviour more than the sum of the agents' behaviour

## Multiagent systems & complex systems



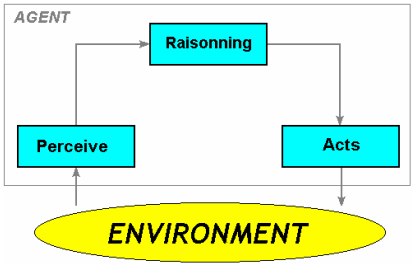



## What's an agent ?

In this context, an agent is node of the social network in an environment which it can *perceive* and in which it *acts*. It is endowed with *autonomous* behaviours and has *objectives*.

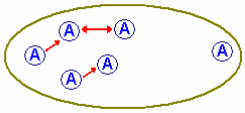
Autonomy is the main concept in the agent issue: it is the ability of agents to control their actions and their internal states.

The autonomy of agents implies no centralized control



# What is MultiAgent System ?

A MAS $(\Gamma)$ is a set of agents situated in a *common environment*, which *interact* and *attempt* to reach a *set of goals*.



$$\Gamma = \{A_1, A_2, ..., A_i, ..., A_n\}$$

Through these interactions, a *global behaviour*, more intelligent than the sum of the local intelligence of multiagent system components, can *emerge*. The *emergence* process is a way to obtain, through *cooperation*, dynamic results that cannot be predicted in a deterministic way.

In the social network, each member node has been modelled as an agent. MAS must make an organization of agents emerge. These agents :

• Supply a dynamic and reliable path from the asking node to the reply node.

• Save as much autonomous node workload as possible.



# Agentification of the problem
## What is an agent in this context ?

- An agent is a list of services or keywords
  - *service list* = < [*keyword*]*, *service name* >
- With several features
  - Privacy
    - *p* = (*nb public services* ÷ *total nb services*)
  - Sociability
    - *S* = *K* × *number of friends*
  - Workload
    - Artificial agent
    - Human assistant agent

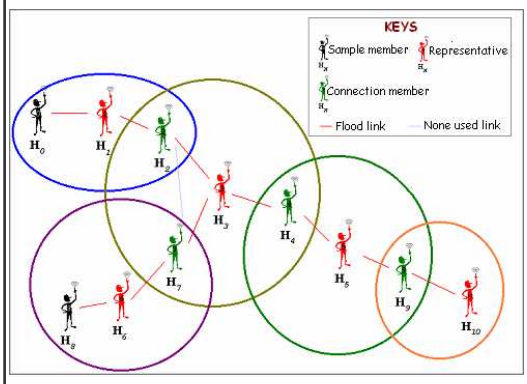
# The MWAC model

## Multi Wireless Agent Communication

*LCIS Laboratory, Valence*
*Grenoble University, France*



# Making the organisation structure (1)




• A *representative agent* manages the messages of the members of its group,

• A *connection agent* enables the representative agent message transmission,

• A *simple member agent* is active only for its own messages.

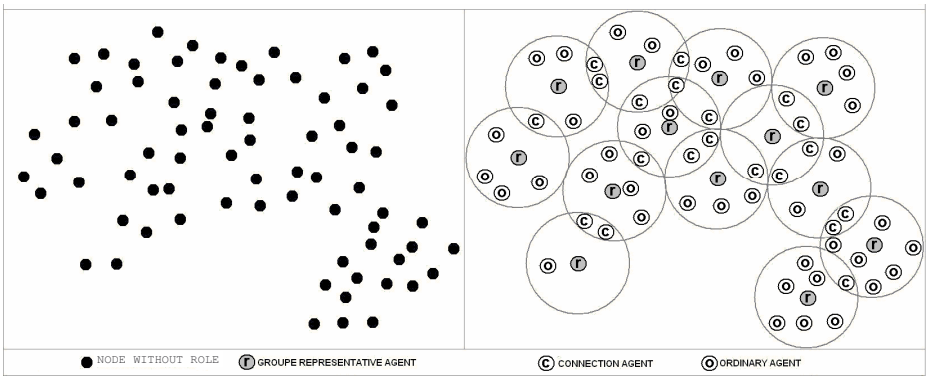
- A self-organization process is based on role allocation techniques,
- The representative agent election will integrate several parameters (use of a *score* function)
- The organization is modified only when a communication problem occures (use of *eavesdropping* to reduce the number of periodical update messages).

# Role allocation algorithme

IF neighbourNumber>0 THEN
** One has neighbours*
IF neighnorRepresentativeNumber=0 THEN
** None of our neighbours is representative: one decides to become one. This * case intervenes when one has just created the agent or when he is isolated. One does not proceed to a vote because one makes the system unstable (the member surely goes to carry on its path)*
myRole = *REPRESENTATIV E*;
ELSE IF myRole = *REPRESENTATIV E* THEN
** I am a representative agent too: I enter in conflict with the other applicants to this role an election will take place and the agent with the best score will remain in place.*
RepresentativeElectionProcedure()
ELSE IF neighnorRepresentativeNumber=1 THEN
** One of our neighbours is representative: one subjects oneself to its authority and this even if the organization is less effective than otherwise. One privileges, for the moment, stability to performance in the organization. One will await the member leaves the network or wishes to leave its mandate.*
myRole = *SIMPLEMEMBER*;
ELSE
** There are, in our vicinity, several representatives: one becomes connection agent for these representatives*
myRole = *CONNECTION*
ENDIF
ELSE
** One does not have a neighbour: one has no role any more*
myRole = *NOTHING*
ENDIF

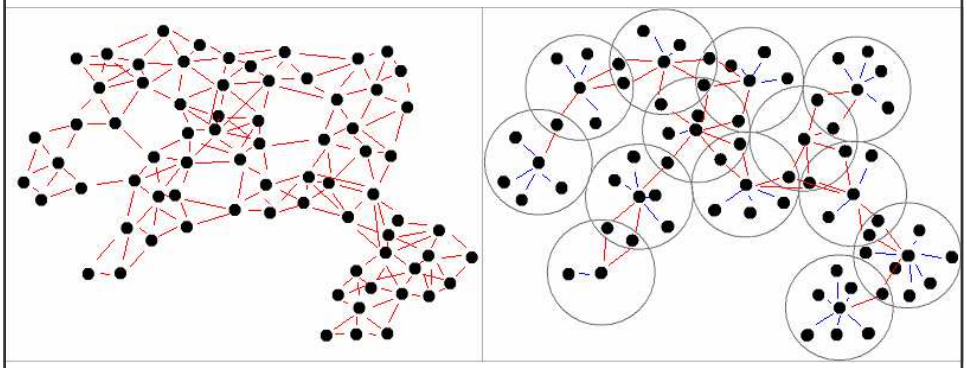


# Making the organisation structure (2)



# Simulation steps

Flood link  Hierarchical link



Minimising exchange numbers



# Information access

- For each agent (member)
  - Local view of the network
  - Launch or provide a skill
- For the network
  - No border, infinite
  - Service location is not stable
- ➔ Asynchronous messages

  ***msg***::=< *id – msg, date, id – sender, id – receiver, status,< content >>*

  with : ***status*** ::= *request*|*reply* and

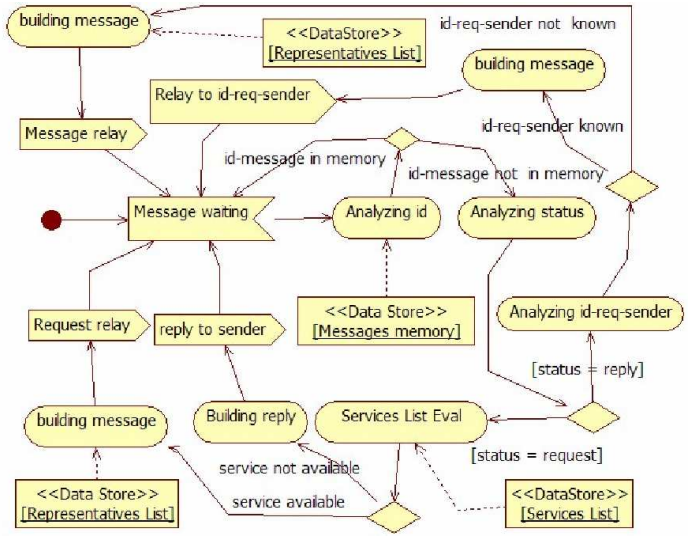
  ***content*** ::= [*keyword*]*| *id – req, id – req – sender, id – service – owner, service name*

# Message treatment

- Message id
  - To avoid loops, the same message is forgotten in a node
- At reception, an agent makes a
  - Reply
  - Transmission another agent who is well-known to have the reply
  - Relay (routing request)
- Message path between the source :
  - From a simple member (a)
  - To a receiver a simple member (b)
    - $(a, r), *[(r, c), (c, r)], (r, b))$.



# Message treatment

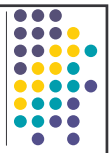


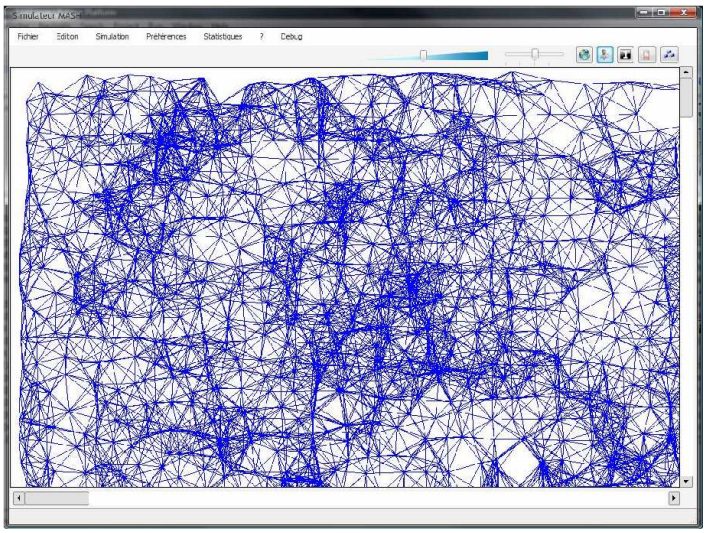
## The simulation results with MASH

*LCIS Laboratory, Valence*
*Grenoble University, France*

19

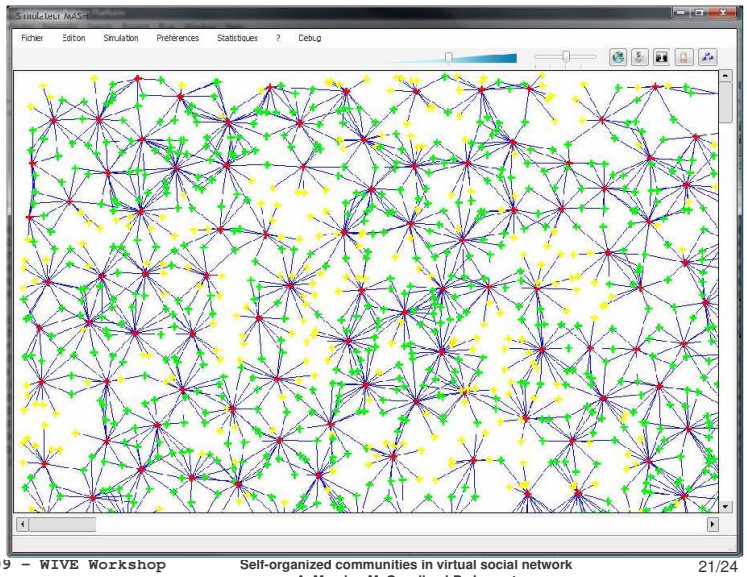
## Initial network each node as a simple member







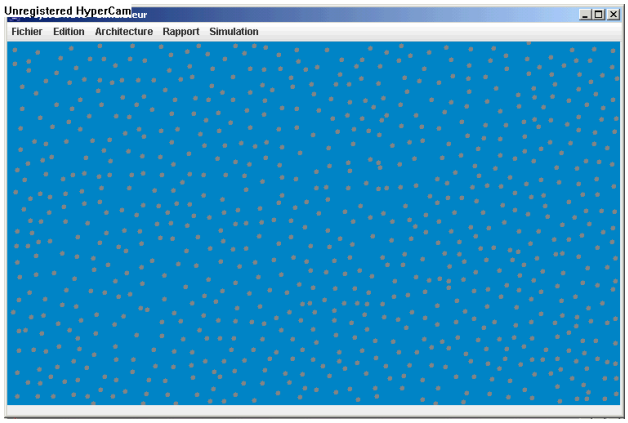

# The organisation structure with roles






# The organisation structure with roles

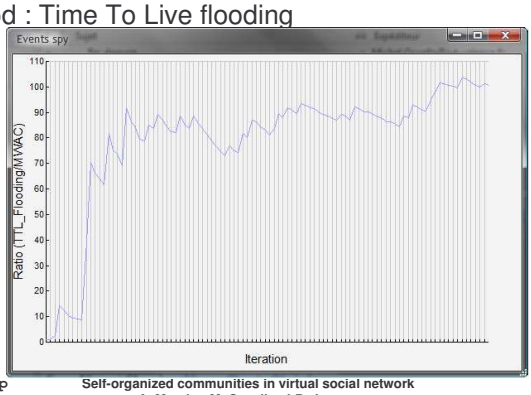
The simulation step allows us to:
- evaluate performance,
- quantify the emergence inferred by the MAS approach.




- Representative agent
- Connection agent
- Simple member agent

# Simulation statistics

- Simulation
  - A network of 1200 nodes
  - 40 services by agent
  - Searching 3 skills launched by 25 to 100 agent randomly chosen
  - 100 iterations
- Baseline method : Time To Live flooding






# Conclusion & future works

- The advantages of this approach using an adaptation of the MWAC model are :
  - the local vision of the network,
  - the independence of nodes,
  - the absence of path (definitively defined) to take into account user preferences and available members skills which can evolve rapidly.
- Extensions :
  - Semantic information in the choice of receiver
  - Completing the friend definition
  - Recursion in the MAS model to improve efficiency and representation