Agent-Based Participatory Simulation Activities for the Emergence of Complex Social Behaviors

Stefano Cacciaguerra, **Matteo Roffilli**

{scacciag,roffilli}@cs.unibo.it

University of Bologna - Department of Computer Science

Background

Social organizations shows different levels of abstraction:

1. macro-level : complex self-organizing systems
2. micro-level : behaviour of individuals

A Multi Agent System can be employed in order to describe self-organizing systems:

1. To mimic real societies by implementing artificial societies
2. To create a quasi-experimental observation-generation environment

Taxonomy

Two types of agents (Stone and Veloso 2000):

Homogeneous

Heterogeneous

Two kinds of configurations:

- Not communicating
- Communicating

Two ways of acting:

- Reactive
- Deliberative

Traditional approach

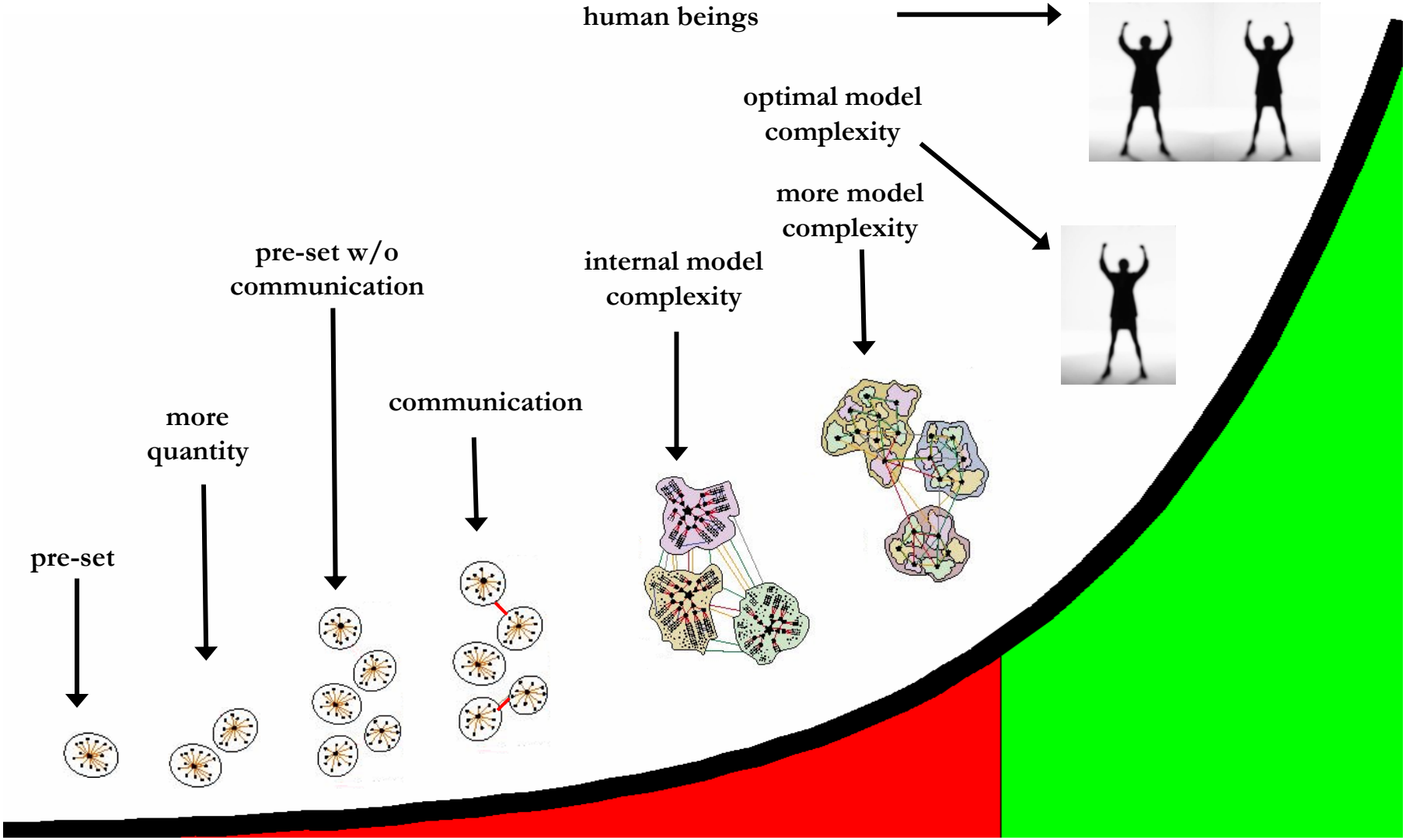
Accurate simulations require:

Heterogeneous, Communicative and Deliberative agents.

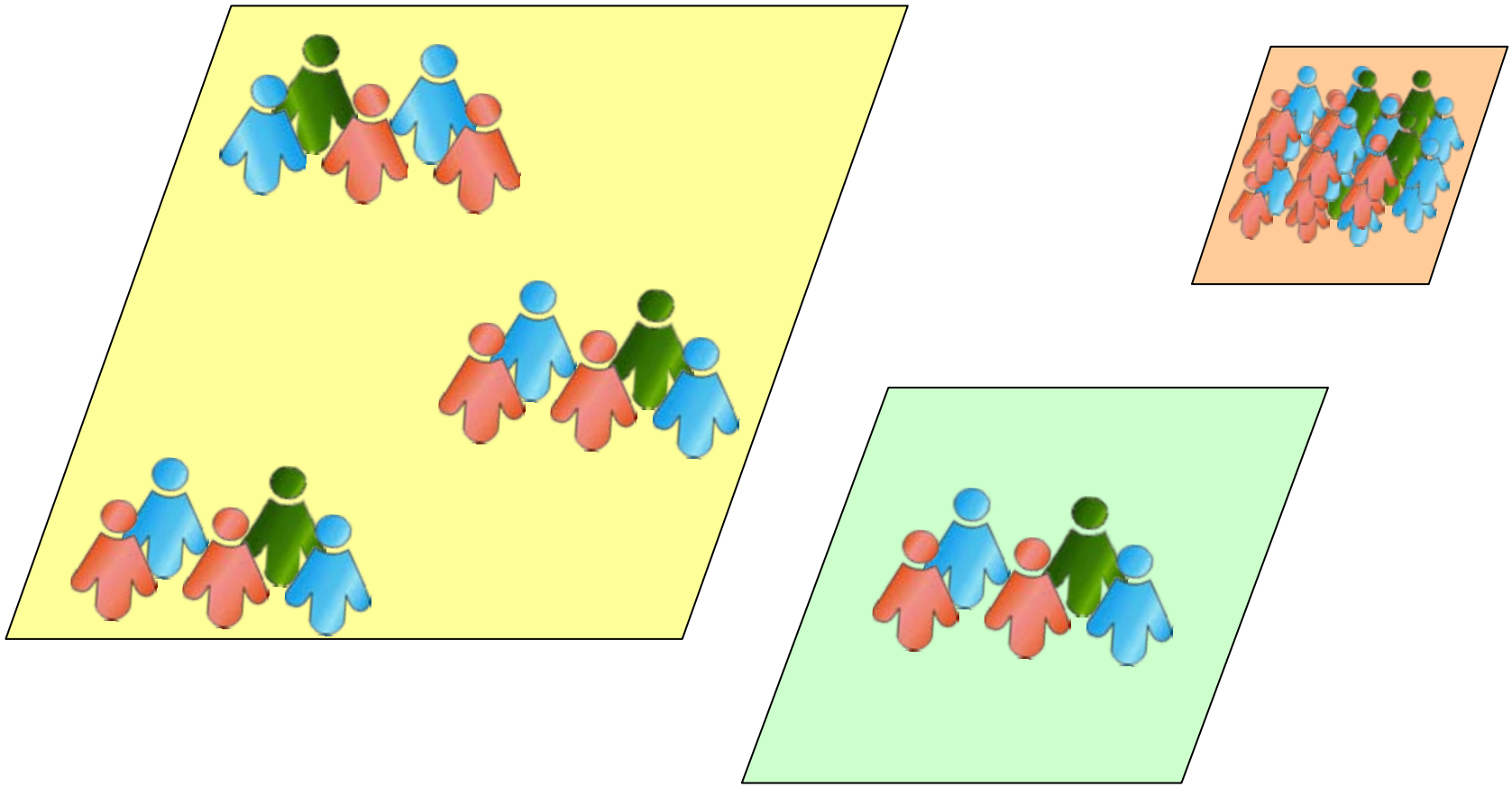
The historical approach suggests:

- **to increase the model complexity**
- **to scale up the number of agents**
- **to improve the behavior of agent**

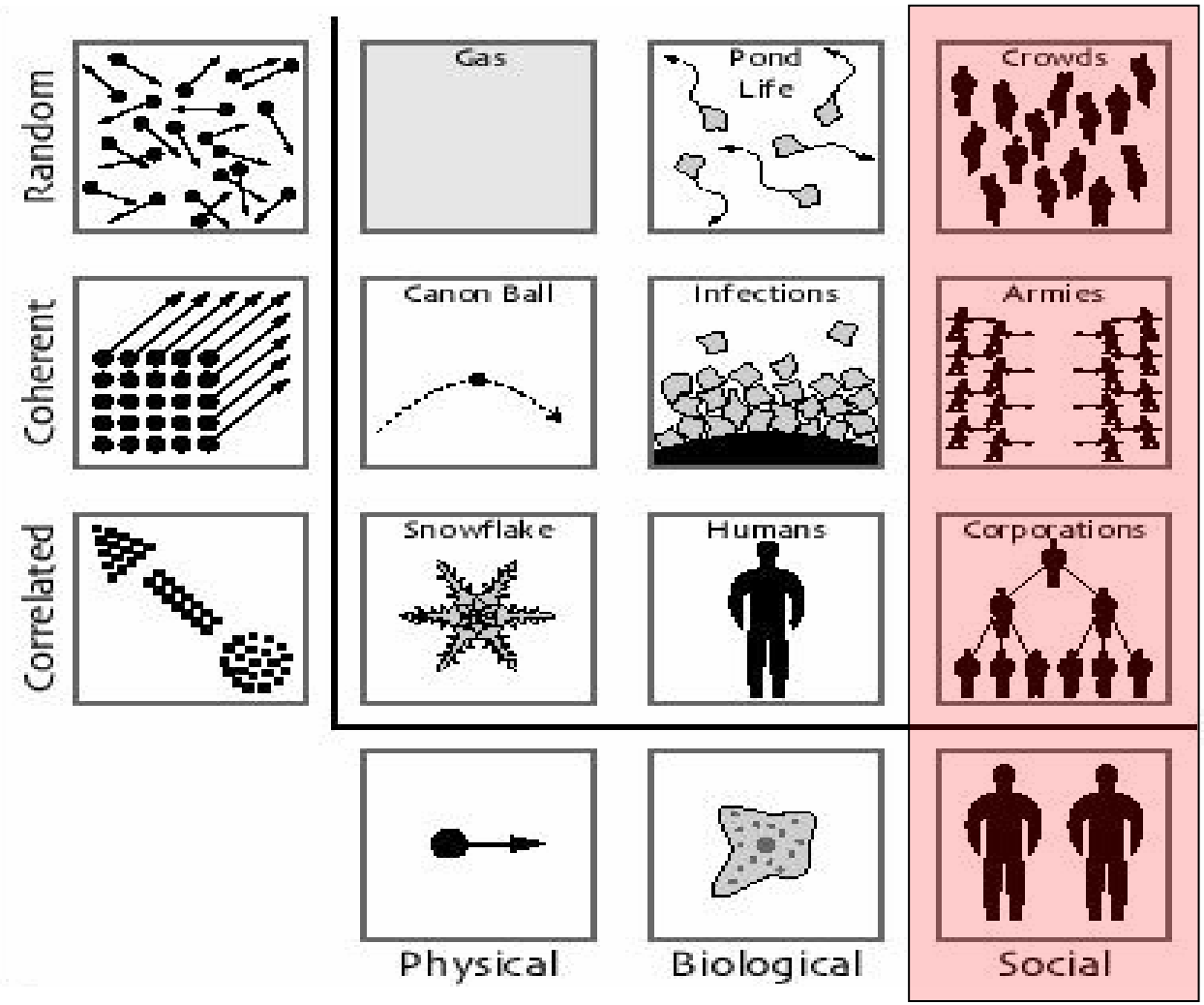
Complexity


human beings
optimal model complexity
more model complexity
internal model complexity
communication
pre-set w/o communication
more quantity
pre-set

Number



Behavior


Random
Coherent
Correlated
Gas
Pond Life
Crowds
Canon Ball
Infections
Armies
Snowflake
Humans
Corporations
Physical
Biological
Social

Main question

Are we sure that is the

best approach to make

accurate simulations

?

Participatory Simulation

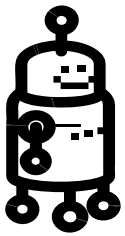
Participatory **S**imulation represents an alternative approach that expands the capability of interactions at **run time**

- Each user can play the role of individual system entities and can see how the behaviour of the system as a whole can emerge from it participation

- PS promotes the interaction among agents controlled by pre-fixed behavioural models and driven by humans

Complex social behaviour

To promote the **emergence of complex social behaviour** we propose to exploit the PS to play games in which:



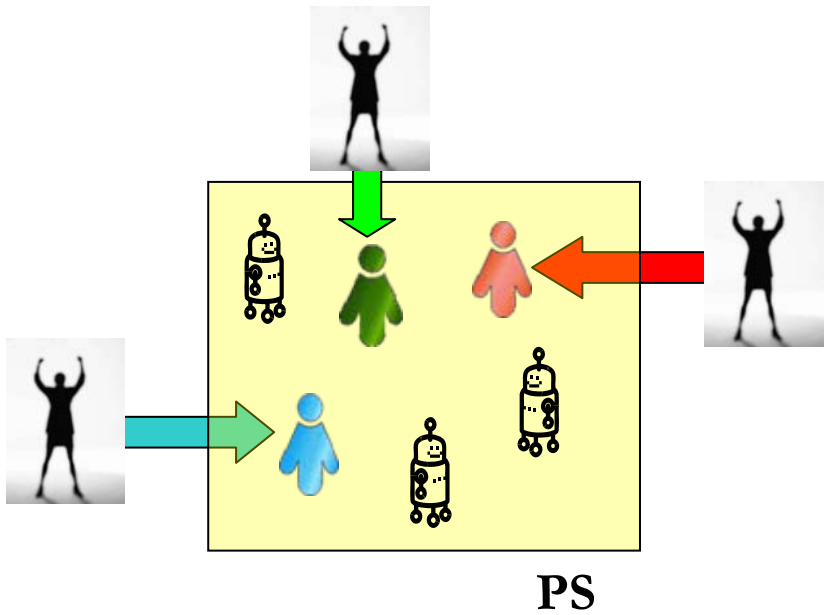
Each agent can be controlled by a **software** that implements hand-made behavioural model

Each **human being** is represented in the game by his digital avatar that can be fully controlled

Framework for PS

Participatory **F**ramework supports the management of the interaction between humans and their agents into any PS

- A user can participate in the evolution of the (remote) simulated complex system by means of PF
- PF handles a connection between a user and a **remote agent** by implementing a session level over a TCP stack
- The user drives a specific agent by means of a client at application level that communicates over a network connection to the synthetic environment



Speed of PS evolution

The participation of multiple users can slow down the evolution of the simulated complex system to unacceptable speed depending on:

- a **momentary interruption** due to congestions or outages of the network communications,
- a **permanent interruption** due to the client or server disconnection and
- a **low level of reactivity** due to the lack of attention from user.

Today frameworks for PS

Open issues:

1. the **responsiveness** is not guaranteed
2. the lost connections **can not be resumed**
3. the agent's behaviour is **prefixed**

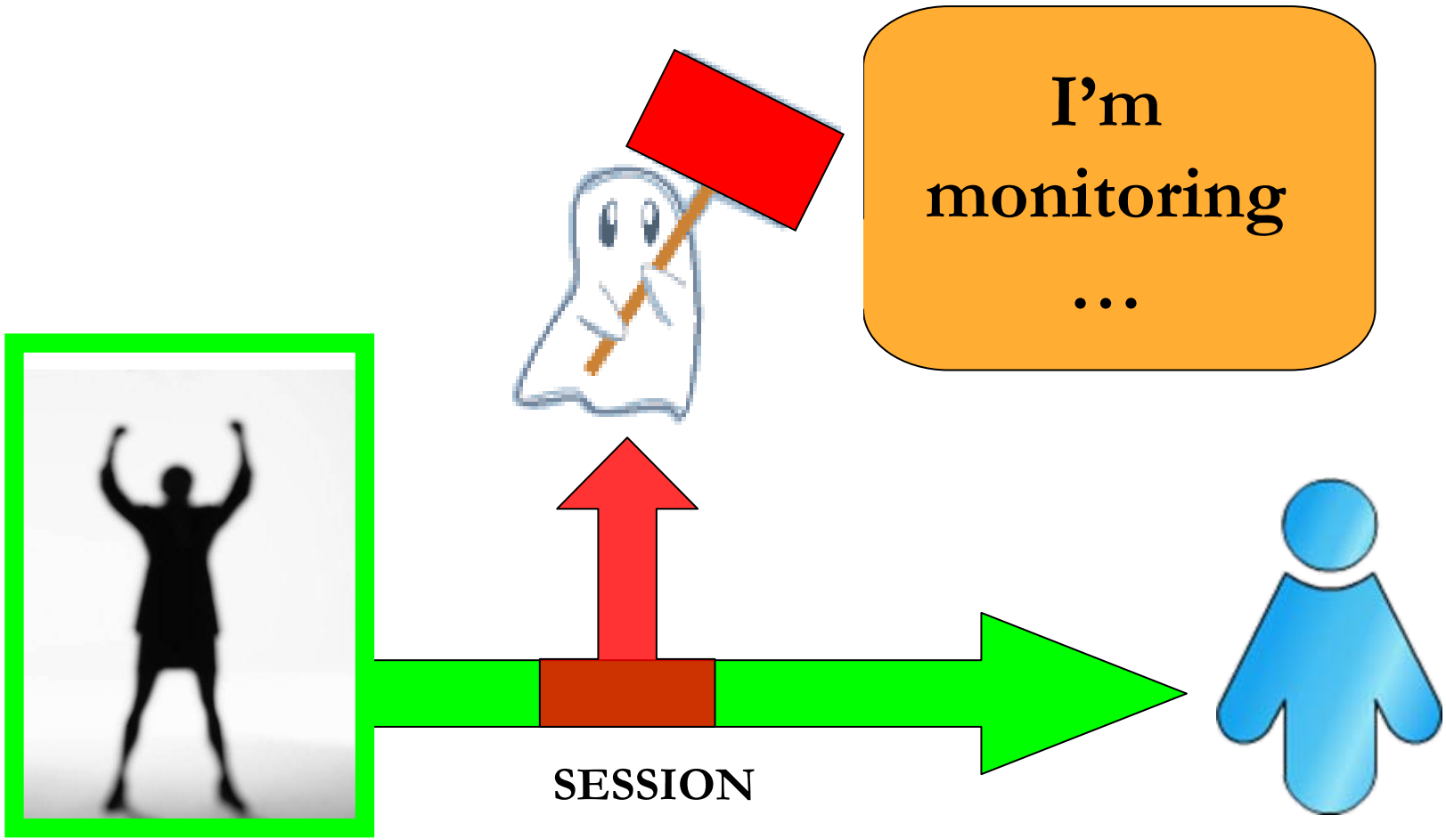
Goal of PF

To maintain the speed of the system evolution over a certain threshold by supporting the human playability

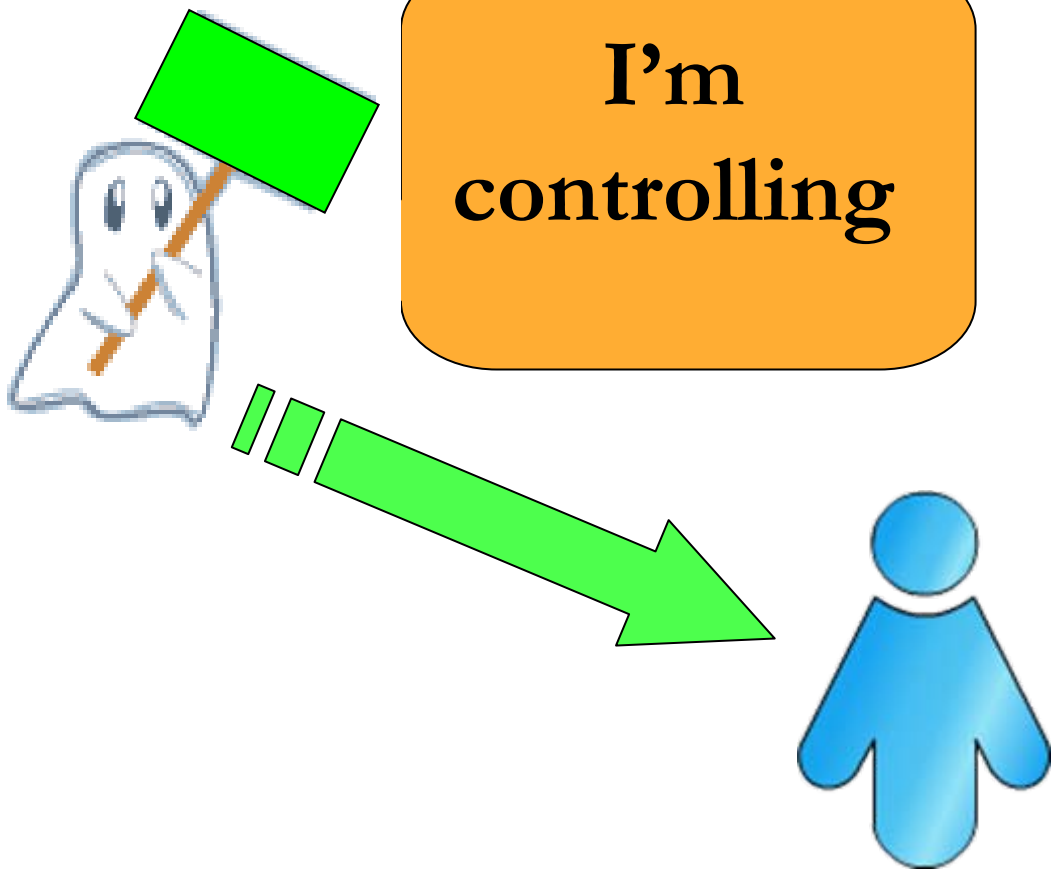
- If a human player is not able to participate in all the turns on time, PF guarantees the correctness of the sequence serialization by imposing to the slow agents to be played by their ghost mimic players

- PF implements a **session recovery** mechanism that allows users to control their agents once again, after the interruption of the network communications

Ghost player



Ghost player in action



Network implementation of PF

Application	**Client**	**Agent**
Session	Participatory framework	Participatory framework
TCP	•	•
IP	•	•
Datalink	•	•
Physical	•	•

Communication Management

The CM mechanism of the ghost player consists of:

- The **Action Timeout Handler (ATH)** avoids that a user low reactivity slows down the evolution of the entire systems
- The **TCP Timeout Handler** decides if the connection between an agent and a client is closed, based on statistical calculations related to the previous performance according to the agent responsiveness on client side and user responsiveness on agent side

Action Timeout Handler

ATH controls the responsiveness of the client (at agent side). ATH monitors the time to receive a new action

If action timeout does not expires before the response from the client, the agent will execute user actions

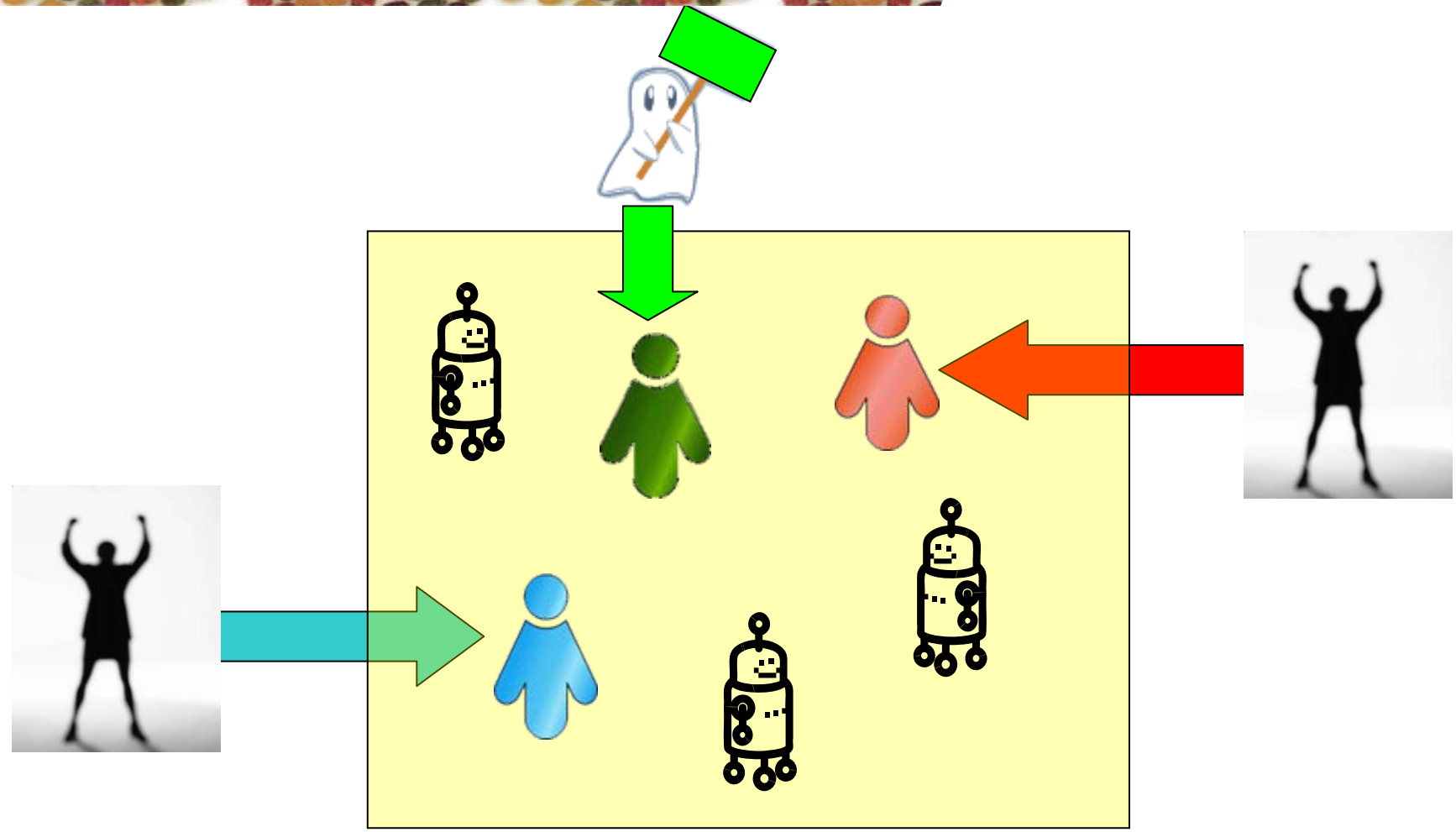
Else, the ghost mimic player drives the agent in place of the human being making 1 move

TCP Timeout Handler

- **From the agent side**, TCP TH sets the state of the connection as **"broken",** when a maximum number of consecutive action timeout occurs

- **From the client side**, TCP TH sets the state of the connection as **"broken",** only after an amount of time (called TCP timeout) has passed without receiving any session ack from the agent

PF with Ghost player



Mimicking capabilities

Which move should the Ghost player choose?

- Random model
- Prefixed behavioural model
- Adaptive behavioural model
- **Mimic** model

The Ghost player tries to mimic human player's strategy

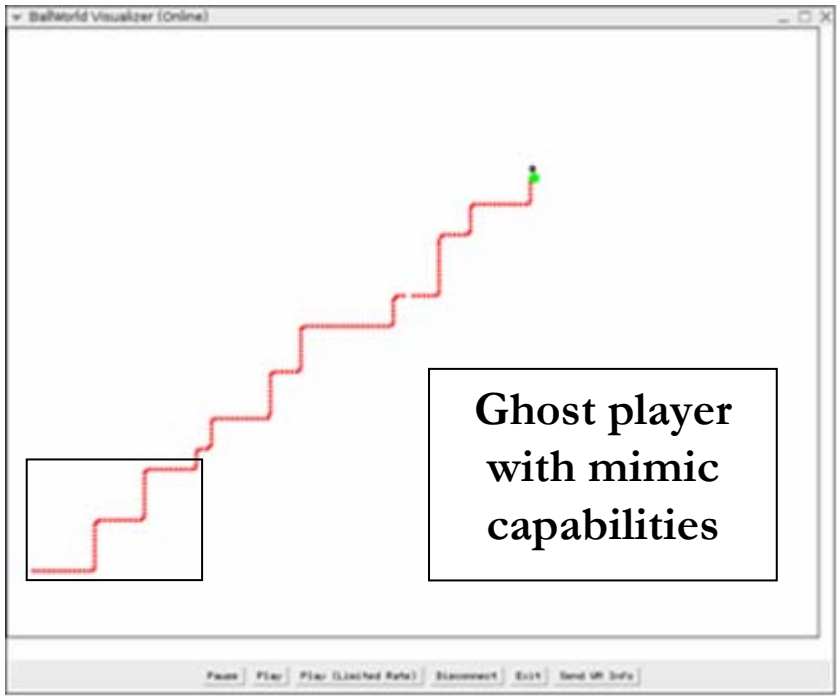
Preliminary results

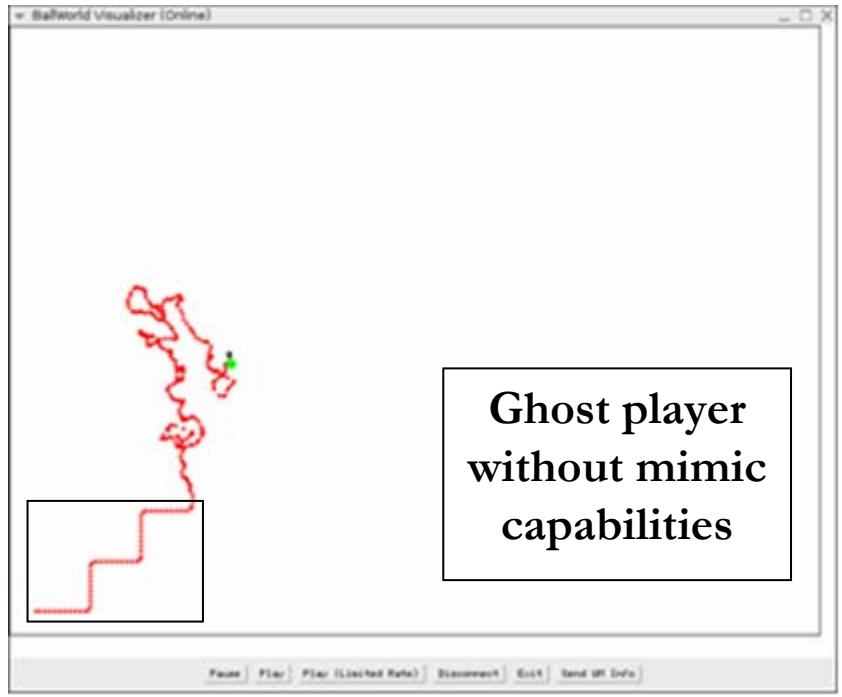
We develop a predator-prey artificial ecosystem (**pursuit domain**)

- The prey goal is to run away, while the predator one is to pursue the prey ●
- Once a predator reaches a prey, it kills it. Otherwise, if a long period passes, predator dies for starvation

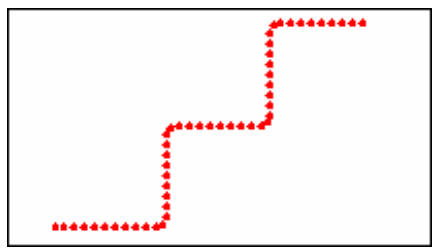
In these preliminary tests, we focus on the **escape trajectory** of the prey-agent ●●●●●●●●

Visualization of escape trajectory

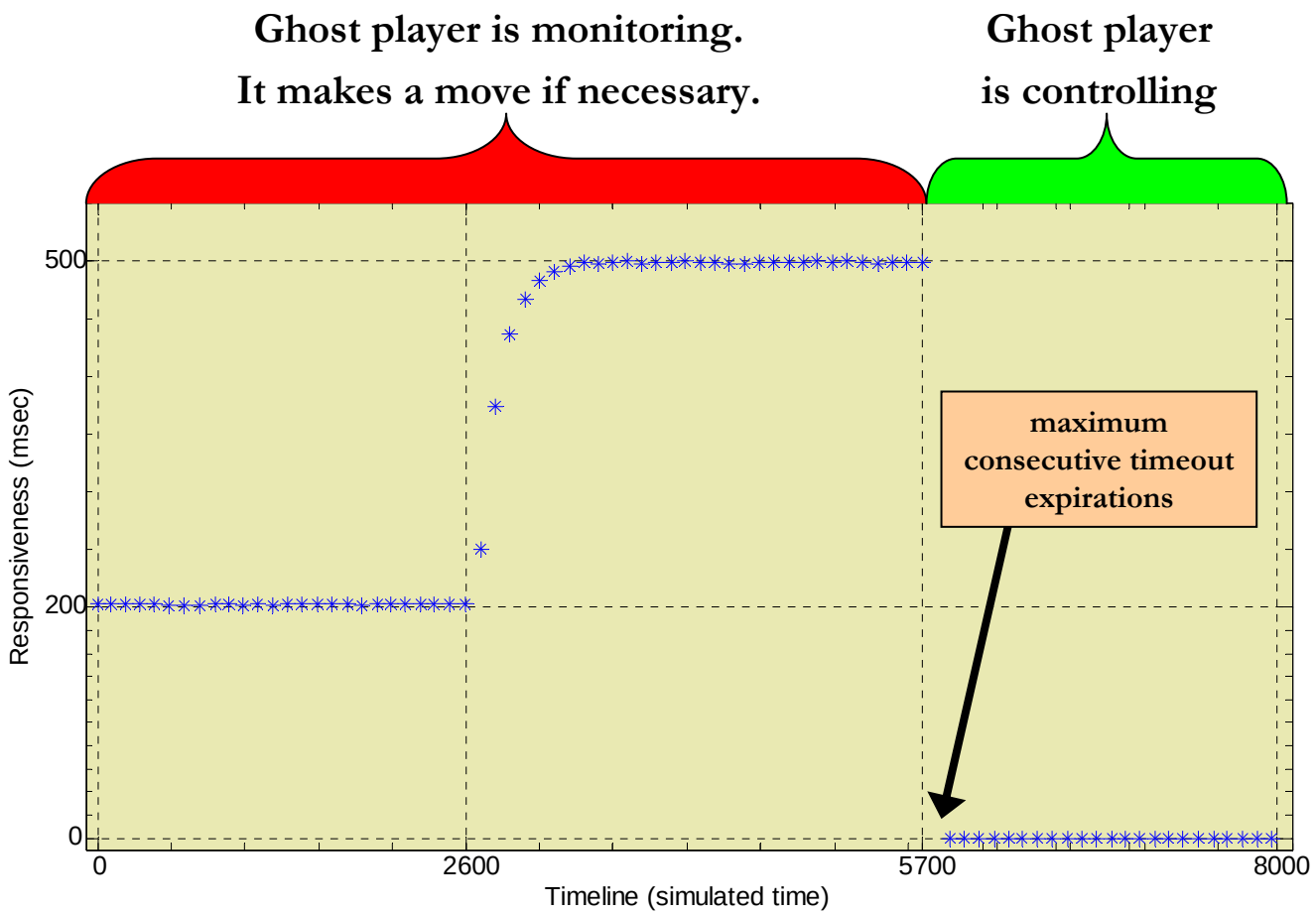




The **pattern of moves** related to the human being is similar to a stairway



Plot of responsiveness



Agent is driven by remote human player

Action timeout expirations

Agent is driven by ghost player

Towards a new Turing Test

Can we construct an agent so that no human being can recognize it as a software while playing with it for a long time

?

If this mimic game is successful, we could safely assert that this software has passed a new version of the Turing test (Turing, 1950)

Conclusions

- This prototype supports the participants with an **endless session level** that allows the human player to disconnect from the synthetic environment while a ghost player takes the control of his agent

- A **mimicking strategy** has been proposed to drive the ghost player

- Preliminary results confirmed the **efficacy** of our approach

Future works

- We are designing our software prototype to pass to a **new version** of the **Turing test** using some methodologies gathered from the field of Machine Learning

- We are planning a **massive experimental campaign** to study the performance of our PF

- These trained behavioral models may be very effective in digital cinema, edutainment, and multiplayer games

Agent-Based Participatory Simulation Activities for the Emergence of Complex Social Behaviors

Stefano Cacciaguerra, **Matteo Roffilli**

{scacciag,roffilli}@cs.unibo.it

University of Bologna - Department of Computer Science