
Pietro Terna - DipEco

0. From Swarm to SLAPP with AESOP

Why a new tool and why **SLAPP** (Swarm-Like Agent Based Protocol in Python) as a preferred tool?

- For didactical reasons, applying a such rigorous and simple object oriented language as Python
- To build models upon transparent code: Python does not have hidden parts or feature coming from magic, it has no obscure libraries
- To create the new AESOP (Agents and Emergencies for Simulating Organizations in Python) way of interaction with agents
- **To use the openness of Python**
- **To apply easily the SWARM protocol**

The openness of Python (www.python.org)

- ... going from Python to R
(R is at <http://cran.r-project.org/> ; rpy library is at <http://rpy.sourceforge.net/>)
- ... going from OpenOffice (Calc, Writer, ...) to Python and viceversa (via the Python-UNO bridge, incorporated in OOO)
- ... doing symbolic calculations in Python (via <http://code.google.com/p/sympy/>)
- ... doing declarative programming with PyLog, a Prolog implementation in Python (<http://christophe.delord.free.fr/pylog/index.html>)
- ... using Social Network Analysis from Python; examples:
 - lgraph library <http://cneurocv.s.rmki.kfki.hu/igraph/>
 - libsna <http://www.libsna.org/>
 - pySNA <http://www.menslibera.com.tr/pysna/>

The SWARM protocol

What's SLAPP: basically **a demonstration that we can easily implement the Swarm protocol** [Minar, N., R. Burkhart, C. Langton, and M. Askenazi (1996), *The Swarm simulation system: A toolkit for building multi-agent simulations*. Working Paper 96-06-042, Santa Fe Institute, Santa Fe (*)] **in Python**

(*) <http://www.swarm.org/images/b/bb/MinarEtAl96.pdf>

Key points (quoting from that paper):

- *Swarm defines a structure for simulations, a framework within which models are built.*
- *The core commitment is to a discrete-event simulation of multiple agents using an object-oriented representation.*
- *To these basic choices Swarm adds the concept of the "swarm," a collection of agents with a schedule of activity.*

The SWARM protocol

An absolutely clear and rigorous application of the SWARM protocol is contained in the original SimpleBug tutorial (1996?) with ObjectiveC code and text by Chris Langton & Swarm development team (Santa Fe Institute), on line at

<http://ftp.swarm.org/pub/swarm/apps/objc/sdg/swarmapps-objc-2.2-3.tar.gz>
(into the folder “tutorial”, with the texts reported into the README files in the tutorial folder and in the internal subfolders)

The same has also been adapted to Java by Charles J. Staelin (*jSIMPLEBUG, a Swarm tutorial for Java*, 2000), at

<http://www.cse.nd.edu/courses/cse498j/www/Resources/jsimplebug11.pdf>

(text) or

<http://eco83.econ.unito.it/swarm/materiale/jtutorial/JavaTutorial.zip> (text and code)

The SWARM protocol as *lingua franca* in agent based simulation models e

Have a look to Swarm basics

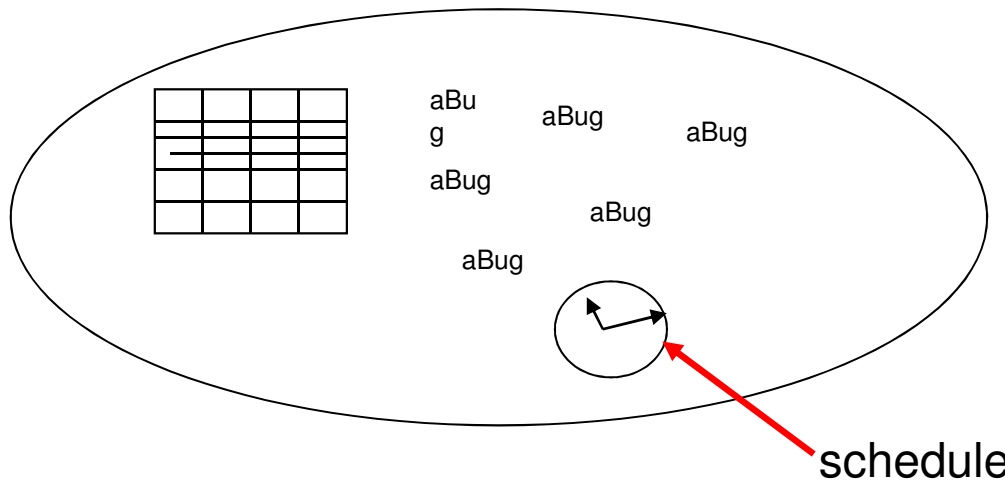
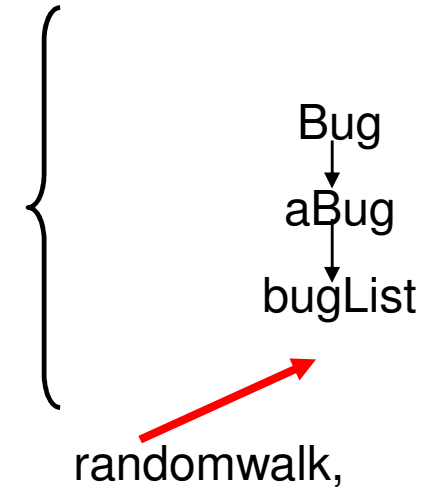
Swarm = a library of functions and a **protocol**

modelSwarm

create objects

create actions

run modelSwarm



Swarm = a library of functions and a **protocol**

modelSwarm

create objects
create actions

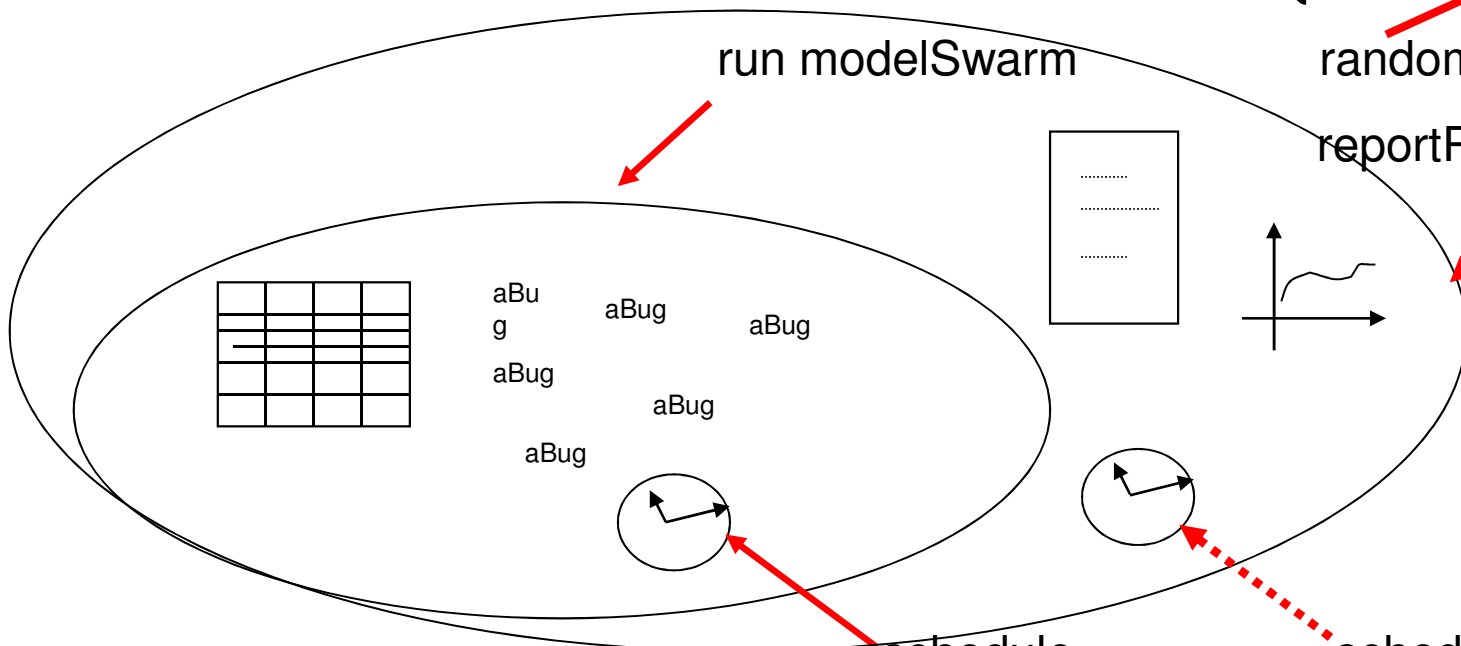
```

Bug
  ↓
aBug
  ↓
bugList
    
```

run modelSwarm

randomwalk,
reportPosition

run
observerSwarm



Swarm = a library of functions and a **protocol**

