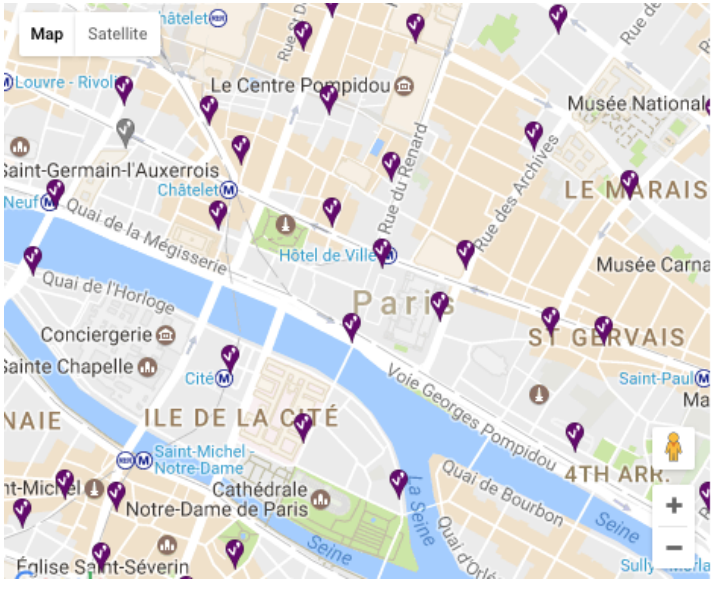
The QUANTICOL project: A Quantitative Approach to Management and Design of Collective and Adaptive Behaviours

EaPEC'17 conference

Nicolas Gast (Inria)

28 September 2017

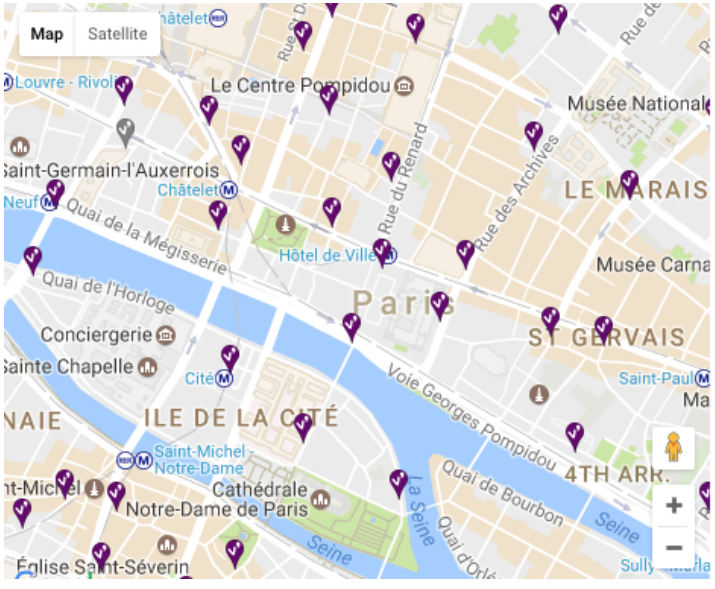
Collective Adaptive Behaviours



Users:

- Will there be a bike/slot when I need one?

Collective Adaptive Behaviours



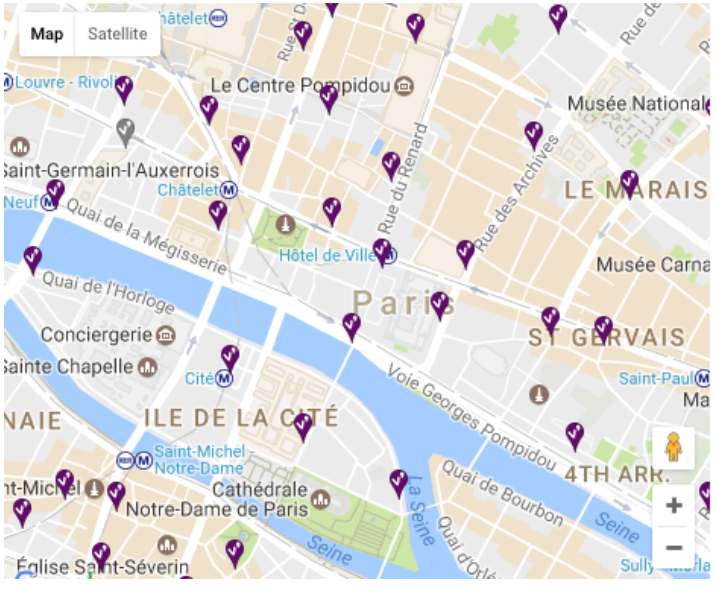
Users:

- Will there be a bike/slot when I need one?

System operators:

- How to maximise revenue within the system?

Collective Adaptive Behaviours



Users:

- Will there be a bike/slot when I need one?

System operators:

- How to maximise revenue within the system?

Answering either question involves quantitative reasoning.

QUANTICOL Research Vision

Our objective was to develop an innovative formal design framework for quantitative reasoning consisting of:

1. an unambiguous way of describing the behaviour of the systems we are interested in;

2. a logic or requirements language which allows us to express the behaviours we wish our designed system to have;

3. automatic ways to check the description against the requirements, captured in software tools;

QUANTICOL Research Vision

Our objective was to develop an innovative formal design framework for quantitative reasoning consisting of:

1. an unambiguous way of describing the behaviour of the systems we are interested in; **a process algebra-based language**
2. a logic or requirements language which allows us to express the behaviours we wish our designed system to have;
3. automatic ways to check the description against the requirements, captured in software tools;

QUANTICOL Research Vision

Our objective was to develop an innovative formal design framework for quantitative reasoning consisting of:

1. an unambiguous way of describing the behaviour of the systems we are interested in; **a process algebra-based language**
2. a logic or requirements language which allows us to express the behaviours we wish our designed system to have; **a logic**
3. automatic ways to check the description against the requirements, captured in software tools;

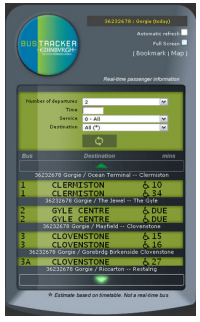
QUANTICOL Research Vision

Our objective was to develop an innovative formal design framework for quantitative reasoning consisting of:

1. an unambiguous way of describing the behaviour of the systems we are interested in; **a process algebra-based language**
2. a logic or requirements language which allows us to express the behaviours we wish our designed system to have; **a logic**
3. automatic ways to check the description against the requirements, captured in software tools; **model checking, approximations, implementation**

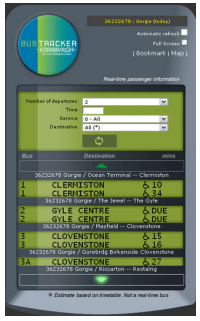
Research Vision – Case Studies

The development of our methodology has focused on the challenges of provisioning smart urban transport and smart grid.



Research Vision – Case Studies

The development of our methodology has focused on the challenges of provisioning smart urban transport and smart grid.



These case studies have provided a rich set of problems against which to design and test our methodology.

The applicability of our tools and techniques is not limited to these scenarios (see `http://blog.inf.ed.ac.uk/quanticol/tools/`).

Quanticol's Overall Strategy

1 Theoretical foundations.

- We developed new stochastic models that are more amenable to scalable analysis (mean-field analysis)
- We focused on multi-scale, spatial representation and formal verification.

2 Specification language and tool support

- Models are described by process algebra and are built by specific languages (e.g., CARMA)
- Tools allow transparent use of scalable analysis techniques.

Quanticol's Overall Strategy

1 Theoretical foundations.

- We developed new stochastic models that are more amenable to scalable analysis (mean-field analysis)
- We focused on multi-scale, spatial representation and formal verification.

2 Specification language and tool support

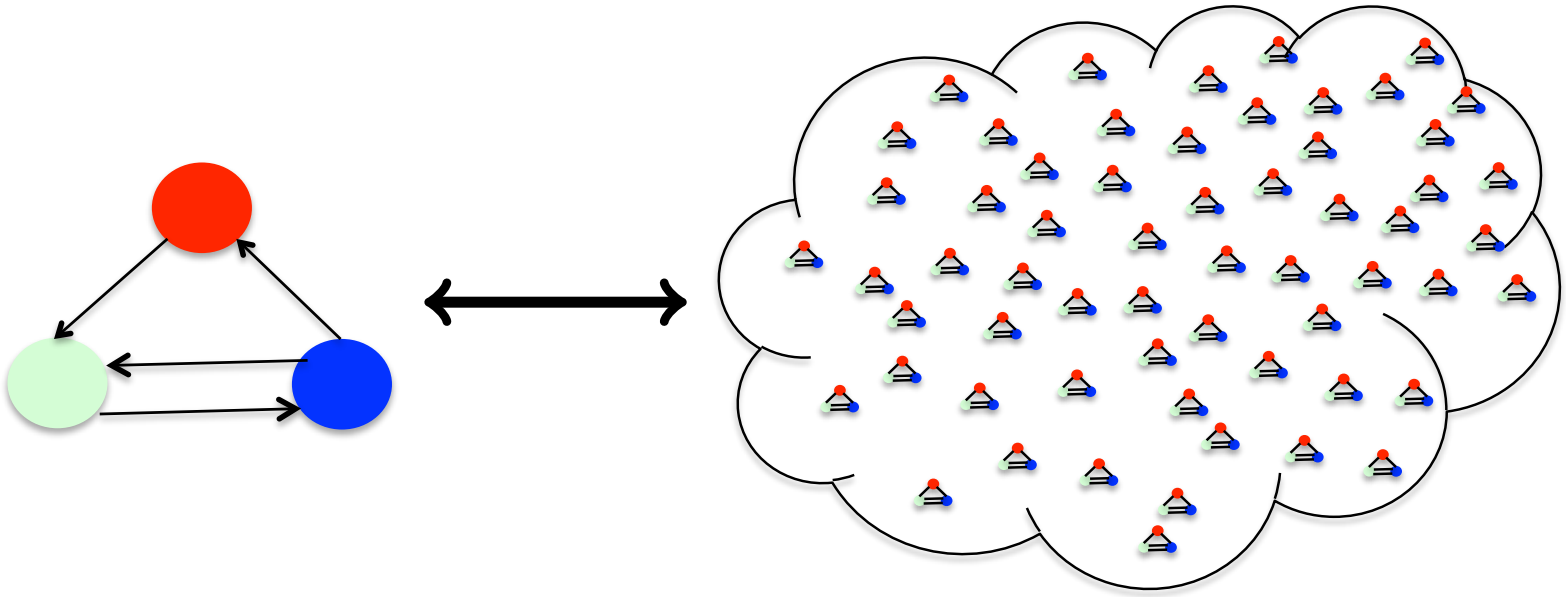
- Models are described by process algebra and are built by specific languages (e.g., CARMA)
- Tools allow transparent use of scalable analysis techniques.

Overview of the presentation

CAS as a system of interacting objects

Problem: the curse of dimensionality

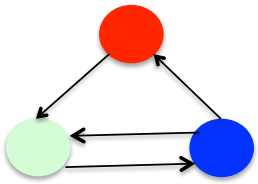
Building a Markov model is (relatively) easy



CAS as a system of interacting objects

Problem: the curse of dimensionality

Building a Markov model is (relatively) easy but the state space grows exponentially with the number of objects.

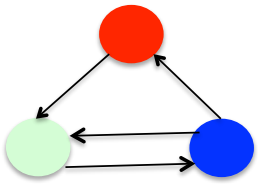


1 object
3 states

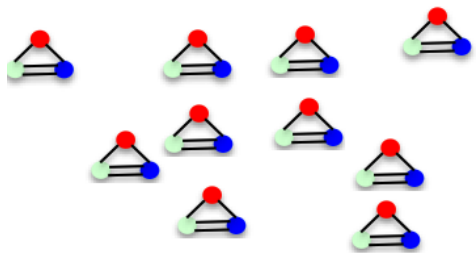
CAS as a system of interacting objects

Problem: the curse of dimensionality

Building a Markov model is (relatively) easy but the state space grows exponentially with the number of objects.



1 object
3 states

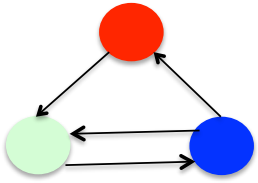


10 objects
$3^{10} \approx 10^5$ states

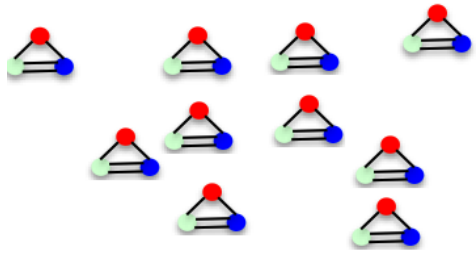
CAS as a system of interacting objects

Problem: the curse of dimensionality

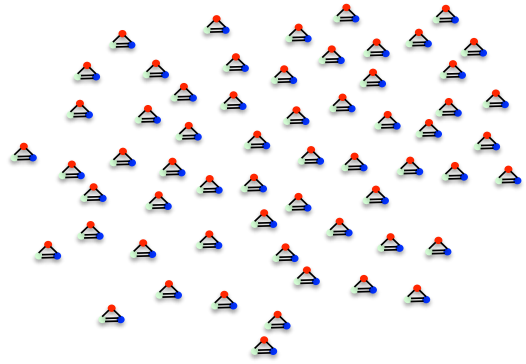
Building a Markov model is (relatively) easy but the state space grows exponentially with the number of objects.



1 object
3 states



10 objects
$3^{10} \approx 10^5$ states



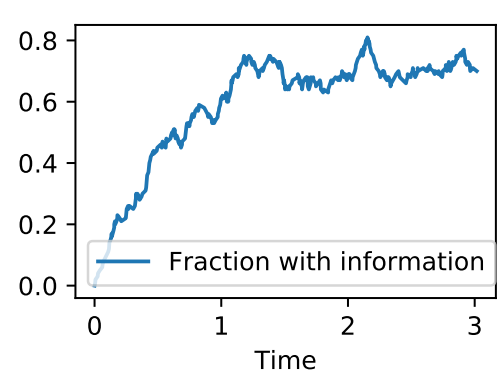
60 objects
$3^{60} \approx 10^{28}$ states

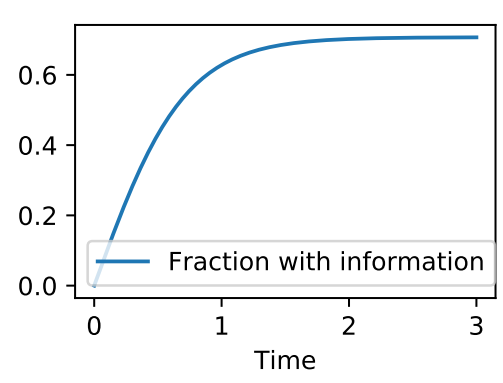
The mean-field approximation

Theorem

For a regular system, the behavior of the stochastic model converges to an ODE as N goes to infinity :

$$\lim_{N\to\infty} \left(\right) =$$





Example – Consider that x/N individual has an information.

- One individual gets the information at rate $1 + 2x$.
- Each individual "looses" the information at rate 1.

$$\text{Approx: } \frac{d}{dt}x(t) = \underbrace{1 - x(t)}_{\text{Get info from external source}} + \underbrace{2x(t)(1 - x(t))}_{\text{Get info from peer}} - \underbrace{x(t)}_{\text{looses the information}}$$

The above approximation requires $N = +\infty$

What about $N = 100$? $N = 50$? $N = 10$?

The above approximation requires $N = +\infty$

What about $N = 100$? $N = 50$? $N = 10$?

Theorem

For a large class of model, there exists a constant C s.t.:

$$\mathbb{E}[X] = x + \frac{C}{N} + O(\frac{1}{N^2}).$$

C can be computed in polynomial time.

Our approximation depends on the system's size. **Example**, for the previous model:

		$N = 10$	$N = 20$
Mean-field approx.	$1/\sqrt{2}$	0.7071	0.7071
Value from simulation		$0.687 \pm .001$	0.698 ± 0.01
Refined model	$\frac{1}{\sqrt{2}}(1 + \frac{1}{4N})$	0.6894 . . .	0.6982 . . .

Modelling space

Capturing the spatial location of agents and how it affects their behaviour is essential but poses significant challenges to the modeller.

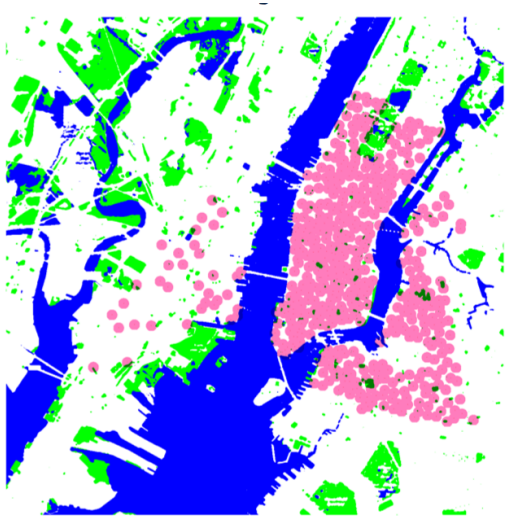
We have investigated the use of space within models from various different perspectives:

- Which language constructs can be used to represent space and movement
- How to use spatial data to build a model of a existing or non-existing system.
- How space can be abstracted to improve the efficiency of analysis.

From Spatial Data to Spatial Models

The case of Bike Sharing Systems (BSS)

System's manager problem : improve the existing BSS



Existing BSS system (NY city)

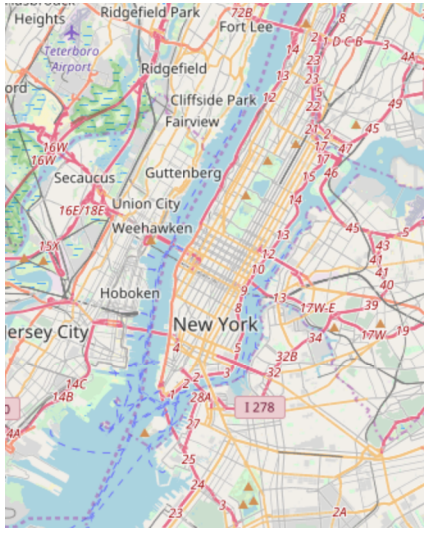
You can study and reduce congestion by :

- Record traces
- Build a simulator to compare different rebalancing policies

From Spatial Data to Spatial Models

The case of Bike Sharing Systems (BSS)

System's designer problem : you want to install a new BSS



openstreetmap
(NY city)

$\mapsto$

Challenge : the system does not exist.

- Where will station will be placed?
- Which traffic flow?

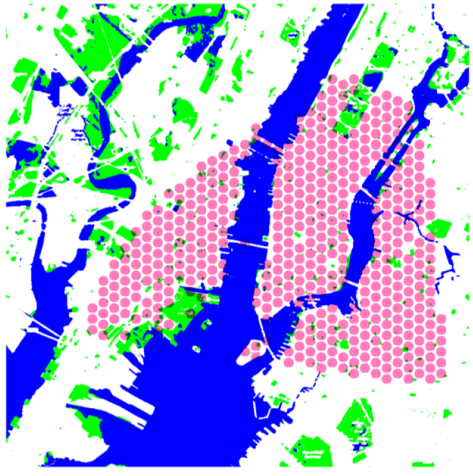
D. Reijsbergen. Probabilistic Modelling of Station Locations in Bicycle-Sharing Systems, in Proceedings of DataMod 2016 From Data to Models and Back, 2016

From Spatial Data to Spatial Models

Example of New-York's BSS

Real system

Regular



From Spatial Data to Spatial Models

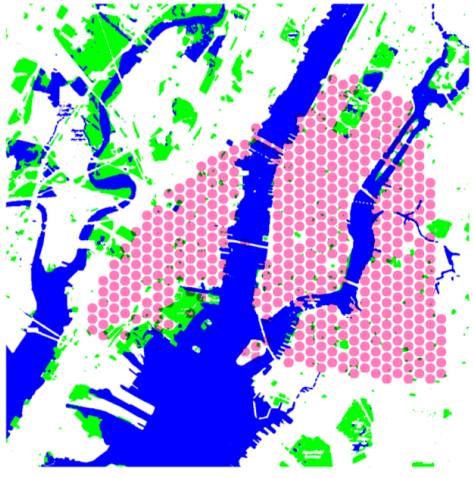
Example of New-York's BSS

Real system

Regular (Noisy)

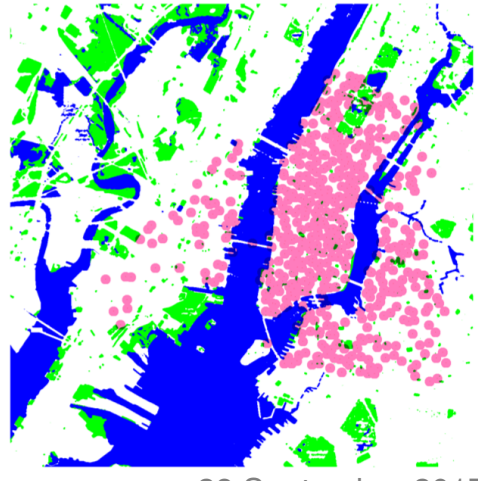
Poisson

Regular



Ginibre

Rating-Weighted

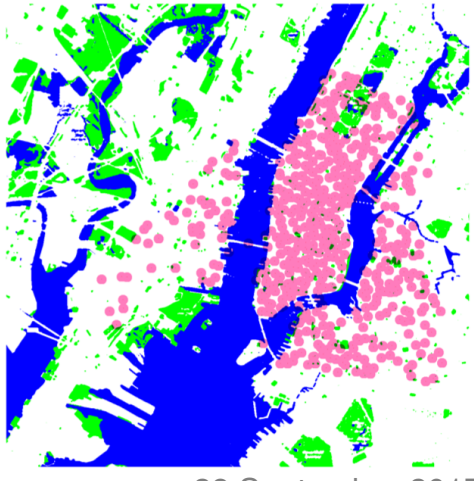


From Spatial Data to Spatial Models
Example of New-York's BSS

Real system

Ginibre + densities of population, of shops, ...

Rating-Weighted



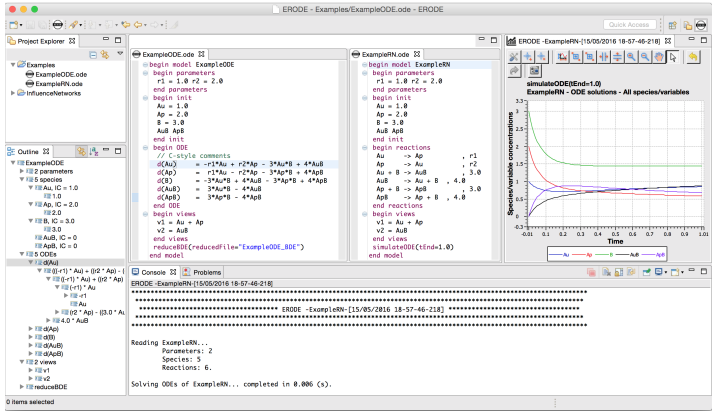
D. Reijsbergen. Probabilistic Modelling of Station Locations in Bicycle-Sharing Systems, in Proceedings of DataMod 2016 From Data to Models and Back, 2016

QUANTICOL software tools

http://blog.inf.ed.ac.uk/quanticol/tools/

Extensive software tool suite. For instance :

Mean-field model checking with *FlyFast*

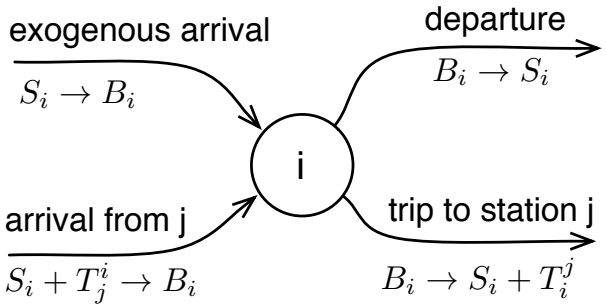


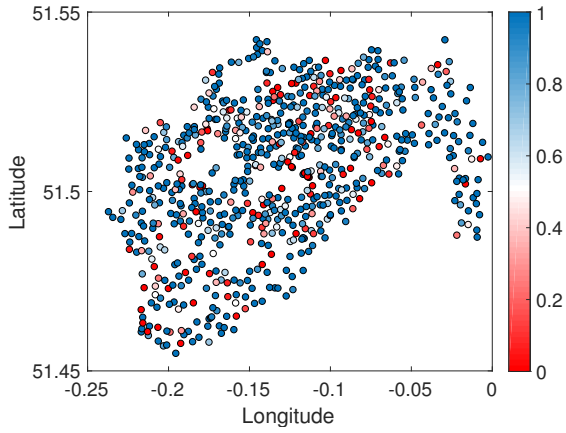
Reduction of differential equations with *ERODE*

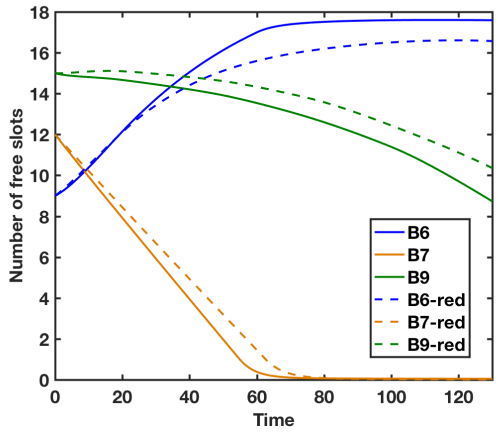
- These are publicly available under an open-source license.
- Many techniques have been implemented as a proof of concept (*e.g.* to experiment with the techniques on realistic case studies).

Validation of tools via Case Studies

The case of BSS





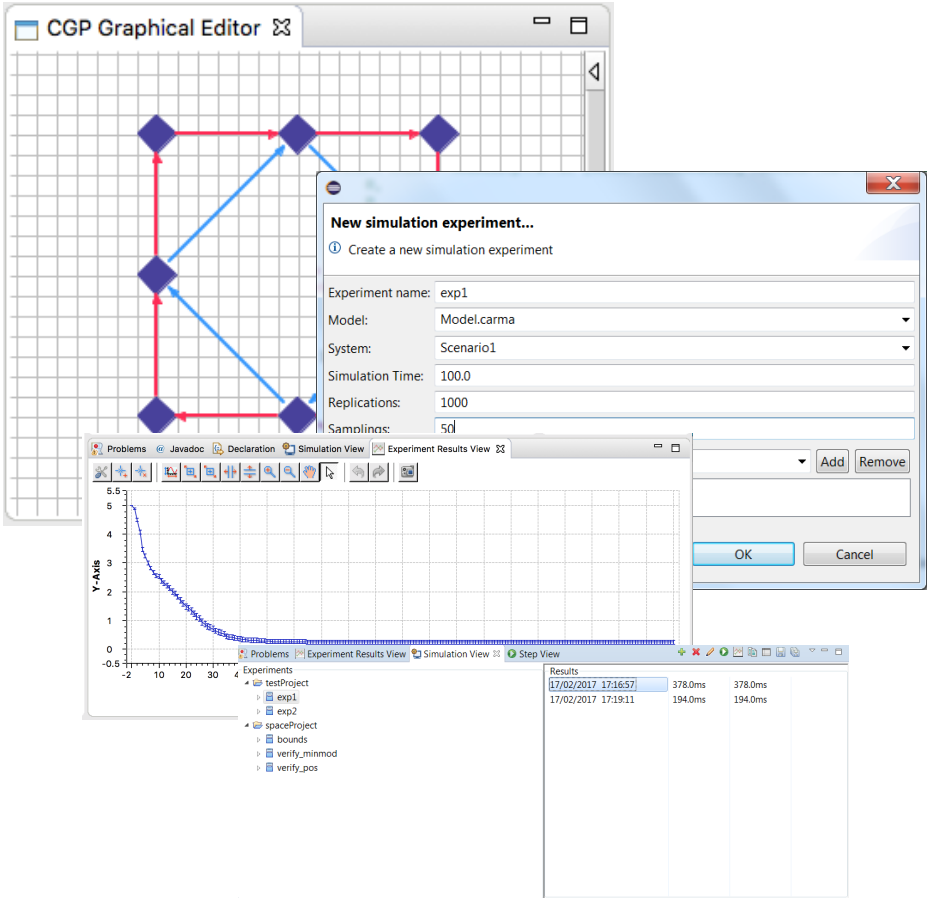


- Tools at work with a model of London's bike sharing network
- Signal spatio-temporal logic to check user satisfaction
- Model reduction to identify clusters of stations with similar dynamics

CARMA : A language to govern them all

CARMA tool-suite:

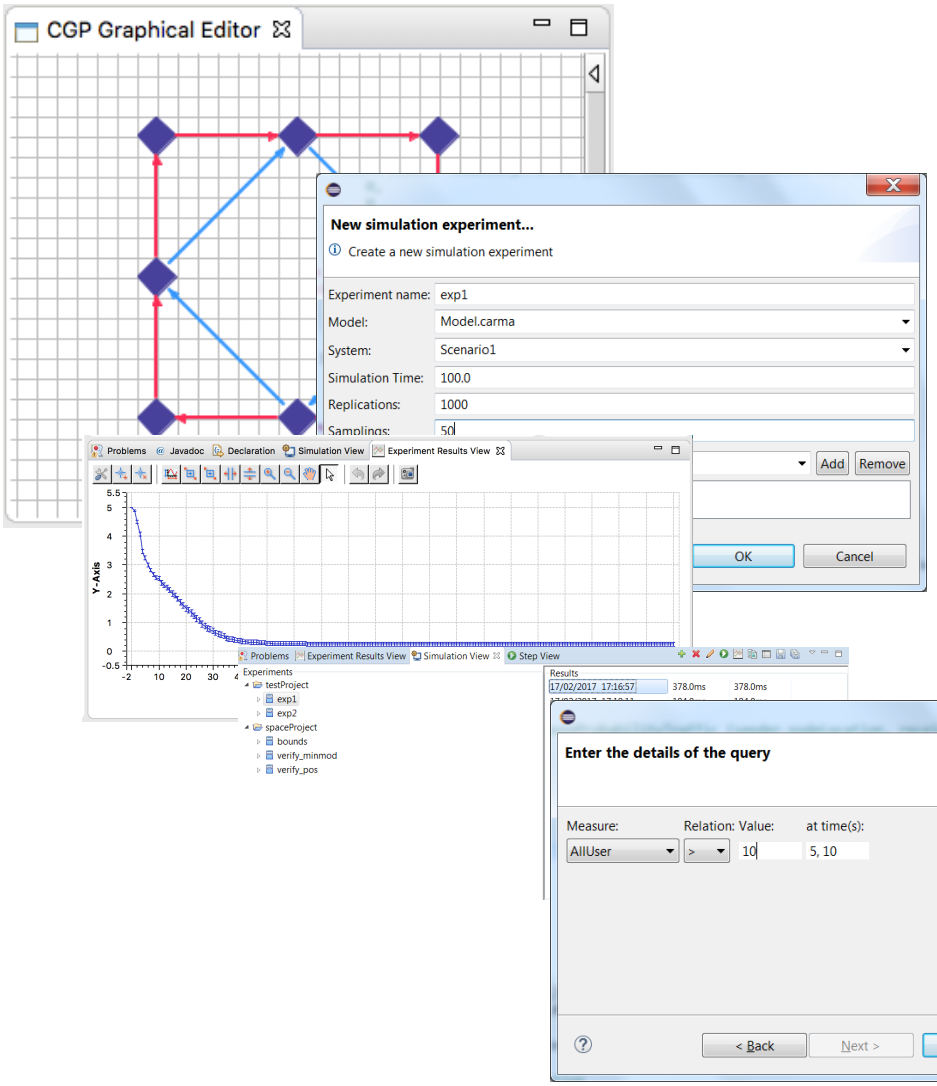
CARMA : A language to govern them all



CARMA tool-suite:

- define model (graphical editor)

CARMA : A language to govern them all



CARMA tool-suite:

- define model (graphical editor)
- Integration with MultiVeStA, jSSTL, FLYFAST,...

Conclusion / Recap

We developed an innovative formal design framework for quantitative reasoning, via :

- Math modeling and new techniques of approximation
- Tools that can build models from data and analyse them.
- Unified approach and many software tools.

What I did not talk about (not exhaustive list)

- Application to smart grids (electricity markets and distribution networks)
- Forecasting and model checking

Thank you!

http://mescal.imag.fr/membres/nicolas.gast

nicolas.gast@inria.fr

Links :

- http://blog.inf.ed.ac.uk/quanticol/tools/
- https://github.com/Quanticol/
- https://github.com/ngast/rmf_tool/