Deceptions and Delectations, Fears and Hopes induced by a few decades in Performance Evaluation

Raymond Marie
IRISA / Université de Rennes 1
UEB

Introduction

- During these last four decades :

  tremendous transformations of the field of modeling in general and of performance evaluation of discrete event systems (DES) in particular.

- Special event motivating this meeting :

  an occasion to have a look back at all this evolution, trying to retain some singularities from the past.

Summary

- Delectations
- Deceptions
- Fears
- Hopes

Delectations

- In the beginning :
  - Our scientific ambition was limited by computing power,
  - we were excited by the novelty of the disciplinary field,
  - Existence of success stories…

→ M/M/1/PS

→ 3 stations PFQN

Delectations

- – All young : sharing national and international responsibilities,
- – Setting up of few but high quality international conferences (no Internet, just postal mail)

- Emergence of new concepts :

  Petri nets, stochastic Petri nets, neural networks,…

Deceptions

- Huge amounts of available computing resources increase the trend to solve models through simulation (do not encourage researchers to look for tractable analytical solutions).
- Researchers are exited by new subjects but forget the past.
- Performance evaluation tasks are not often included in the engineering process.
- Part of the performance evaluation in university regular curses is decreasing…
- Number of international conferences per year is increasing : Why ? When does a lot becomes too much ?

Fears

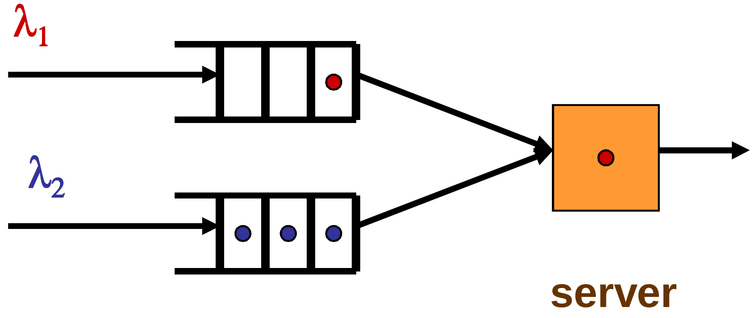
List of dangers threatening the quality of the work of a modeler of DES

Risk 1 : bad comprehension of the system

Remark : both simulation and analytical approaches are concerned !

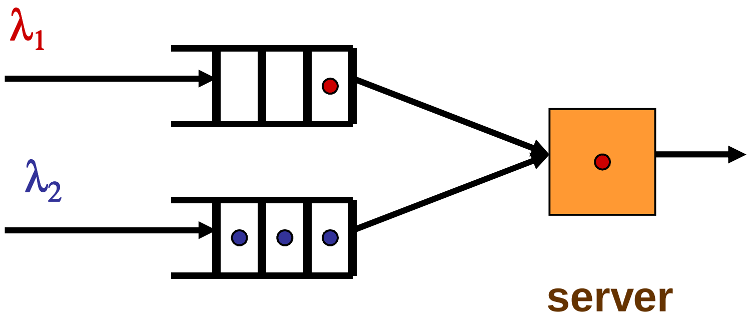
Example

A two class preemptive repeat priority queue



Fears / bad comprehension of the system

A two class preemptive repeat priority queue



For this queue, λ_1 = 1.0, m_1=0.01 (the server stays 1% of its time with the first class customers)
and λ_2 = 0.01 et m_2=1,

The service time distribution of class two is **exponentially** distributed

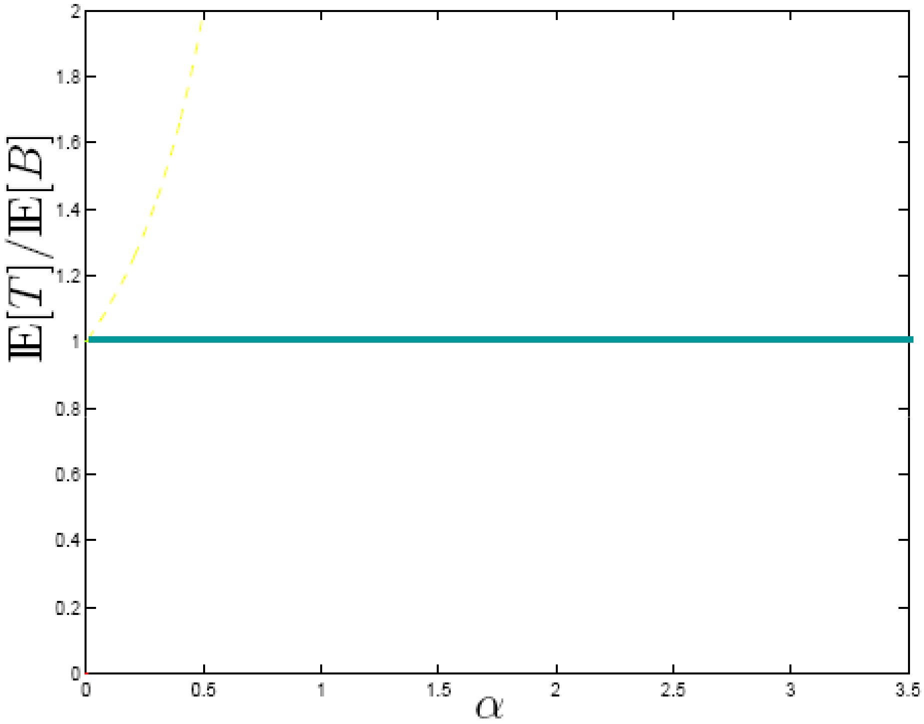
Memoryless property → "the residual service time at the preemption instant is still **exponentially** distributed".

Fears / bad comprehension of the system

e.g., The randomness of the service time comes from the randomness of the Web response time…

→**preemptive resume discipline =**

preemptive repeat discipline



T_i = service duration between the $(i-1)^{th}$ and i^{th} preemptions

B = service duration

E[T] = ΣT_i = expected busy time of the server

E[T] = $1/\mu_2$ **= E[B]**

$\rho = 0.01 + 0.01 = 0.02$

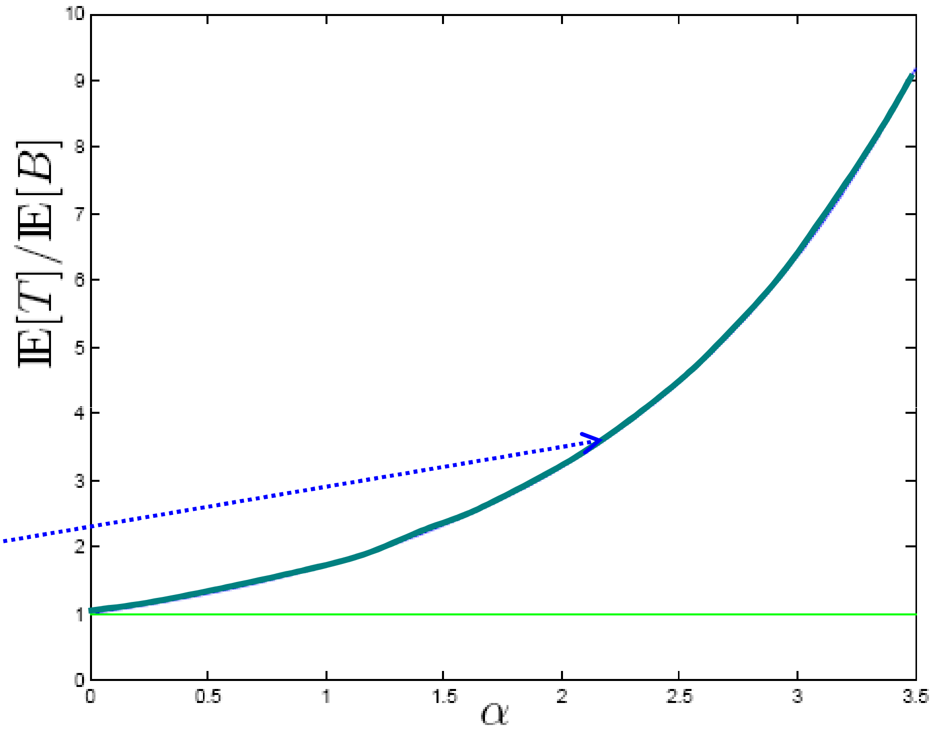
α = expected number of preemptions during the service

→ $\alpha = \lambda_1$ E[B] (Poisson arrival process)

Fears / bad comprehension of the system

If B was a constant service time ?

$$\frac{\mathbb{E}[T|B=x]}{x} = \frac{(e^{\alpha}-1)}{\alpha}$$



Remark : a random service time looks better than a constant service time !

Fears / bad comprehension of the system

Assume that the class 2 customers represent the many different routines of an application such that, globally, the service time can be seen as exponentially distributed !

→ The preempted job will always take the same time to be completed

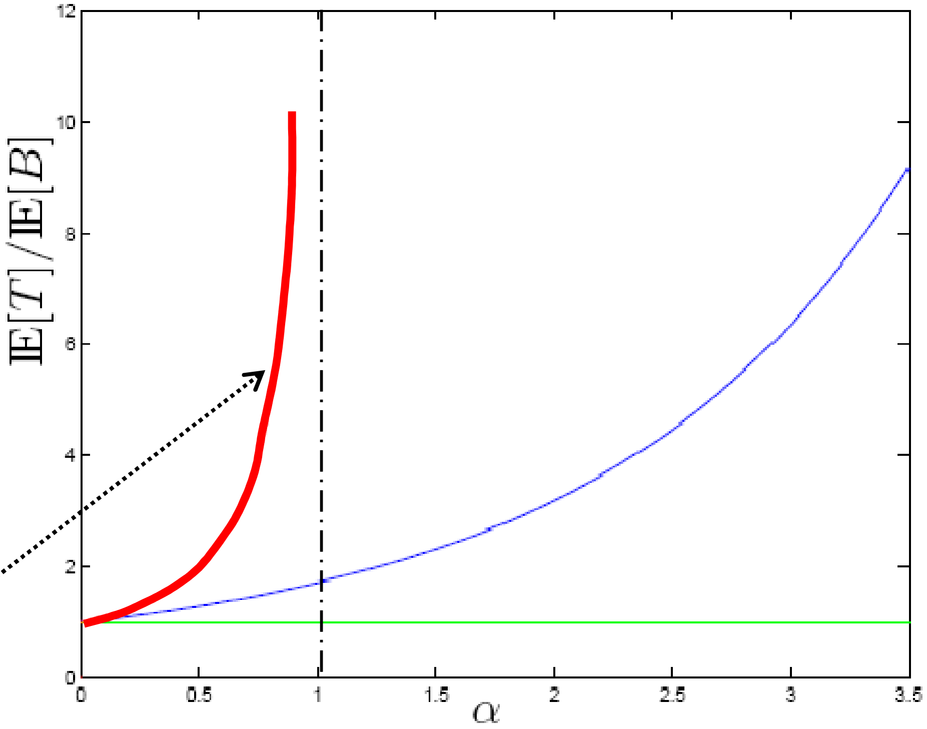
→ preemptive repeat identical discipline !!!

Fears / bad comprehension of the system

$$\boxed{\mathbb{E}[T] = \frac{1}{\lambda_1}\left(G^*(-\lambda_1) - 1\right)}$$

$$\begin{aligned}\mathbb{E}[T] &= \frac{1}{\lambda_1}\left(\frac{\mu_2}{\mu_2 - \lambda_1} - 1\right)\\ &= \frac{1}{\mu_2 - \lambda_1}\end{aligned}$$

$$\frac{\mathbb{E}[T]}{\mathbb{E}[B]} = \mathbb{E}[T]\mu_2 = \frac{1}{1-\alpha}$$



In the example, α = 1 and the queue blows up !

Fears

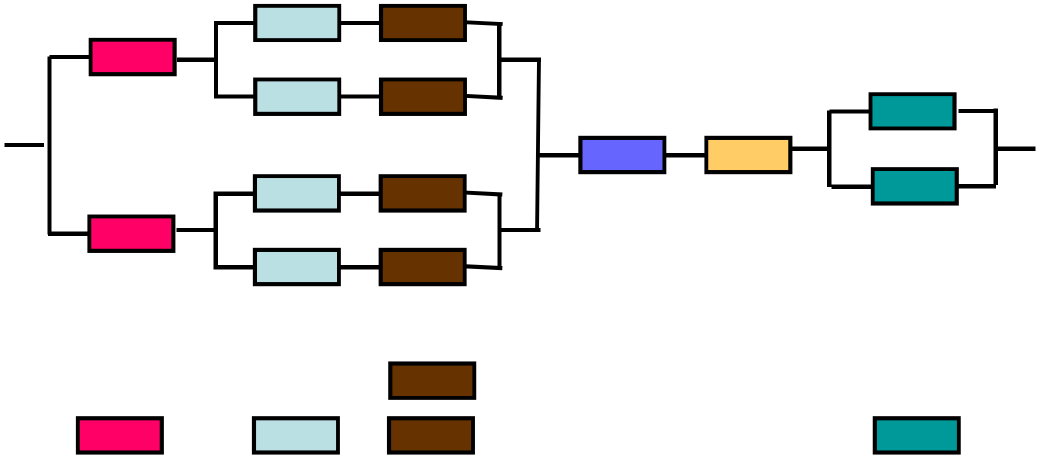
Risk 2 : Bad mastery of approximations

- Approximations done consciously (we assume that the service time is exponentially distributed),

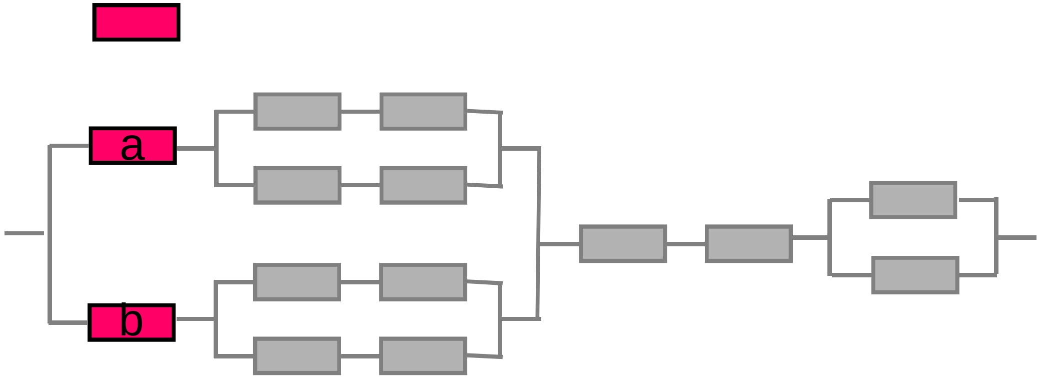
- Approximations done unconsciously (considering that two events are independent while they are not),

Example

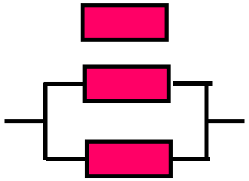
Unavailability at time T



Fears / bad mastery of approximations



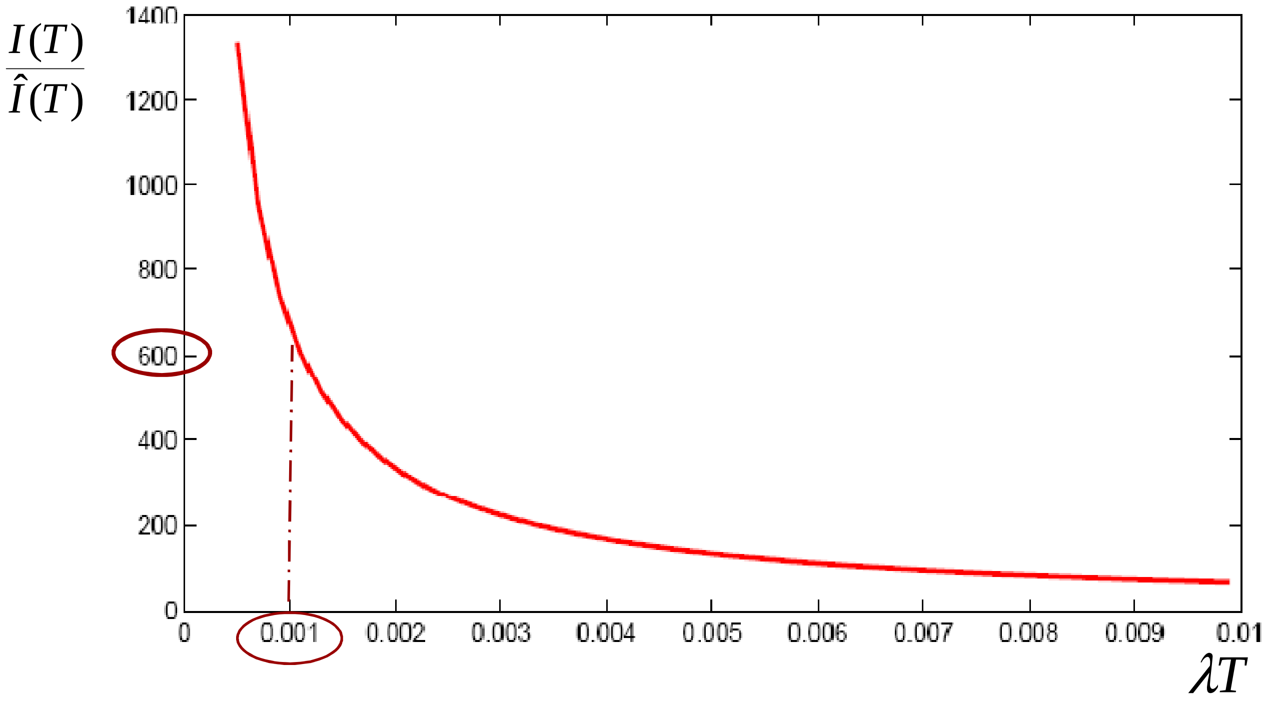
$p_A(T)$



$$\hat{I}(T) = (p_A(T))^2$$

$$I(T) = 1 - e^{-\lambda T}[4 - e^{-\lambda T}(3 + 2\lambda T)]$$

Fears / bad mastery of approximations



Fears

Risk 3 : use of numerically unstable algorithms

Remark : **Difference of two (smaller and smaller) positive numbers*!***

Need to look for stable algorithms,

Note that simulating is generally also approximating ! (confidence intervals)

Fears

Risk 4 : inadequate use of Markovian models

When is it dangerous to use Markovian models ?

Remark :

Almost everything can be modeled by a MM ! The difficulty arises from the potentially extremely large state space (*e.g.*, Coxian rv) .

This is why we try to decrease the dimension of the state space by approaching task durations by exponential distributions.

In some situations, it is without any consequences,
In some others (concurrent activities), it can be very dangerous !!

Fears

Risk 5 : lack of technical background ?

Remark 1 : highly correlated with risk 1 (bad comprehension of the system)

Remark 2 : Unfortunately, being technically good is not a sufficient condition for building good models !

Fears

Risk 6 : lack of scientific background

Remark 1: **in particular in applied mathematics !**
(not award of the different families of models)

Remark 2: **both simulation and analytical fields are concerned !**

Hopes

Thanks to research (and D), computational power and data storage have increased the possibilities of performance evaluation studies during the last decades.

Development of libraries and graphical interfaces has increased our productivity !

But, if we compare with other industries (space, transport, nuclear), or disciplines (physics, chemistry,…)…

We should be more ambitious wrt our evaluation tools

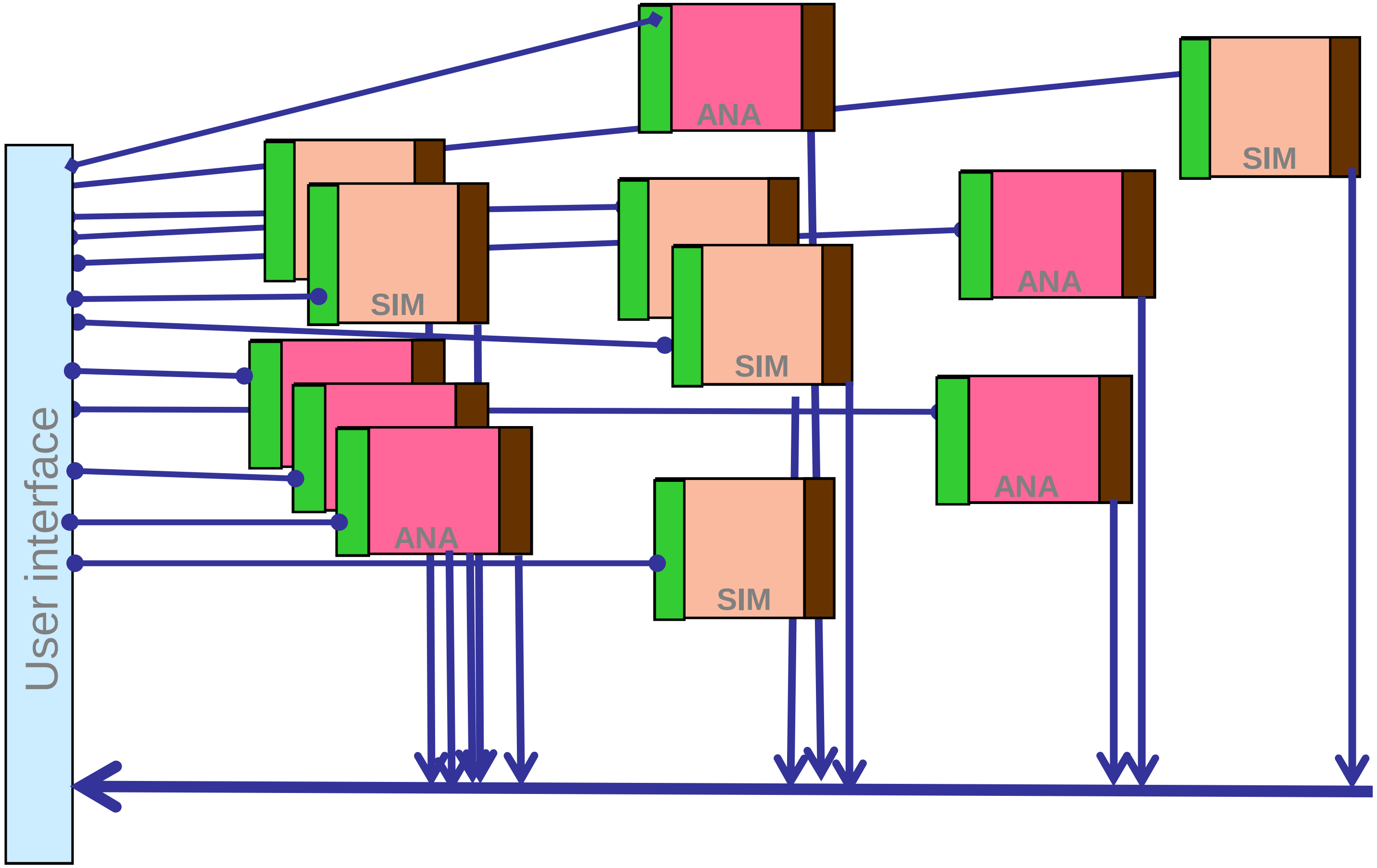
Hopes

There is place for a large evaluation tool built as a set of *cooperative agents* including simulation agents and analytical/numerical agents *(virtual machine built on a computer cloud ?)*

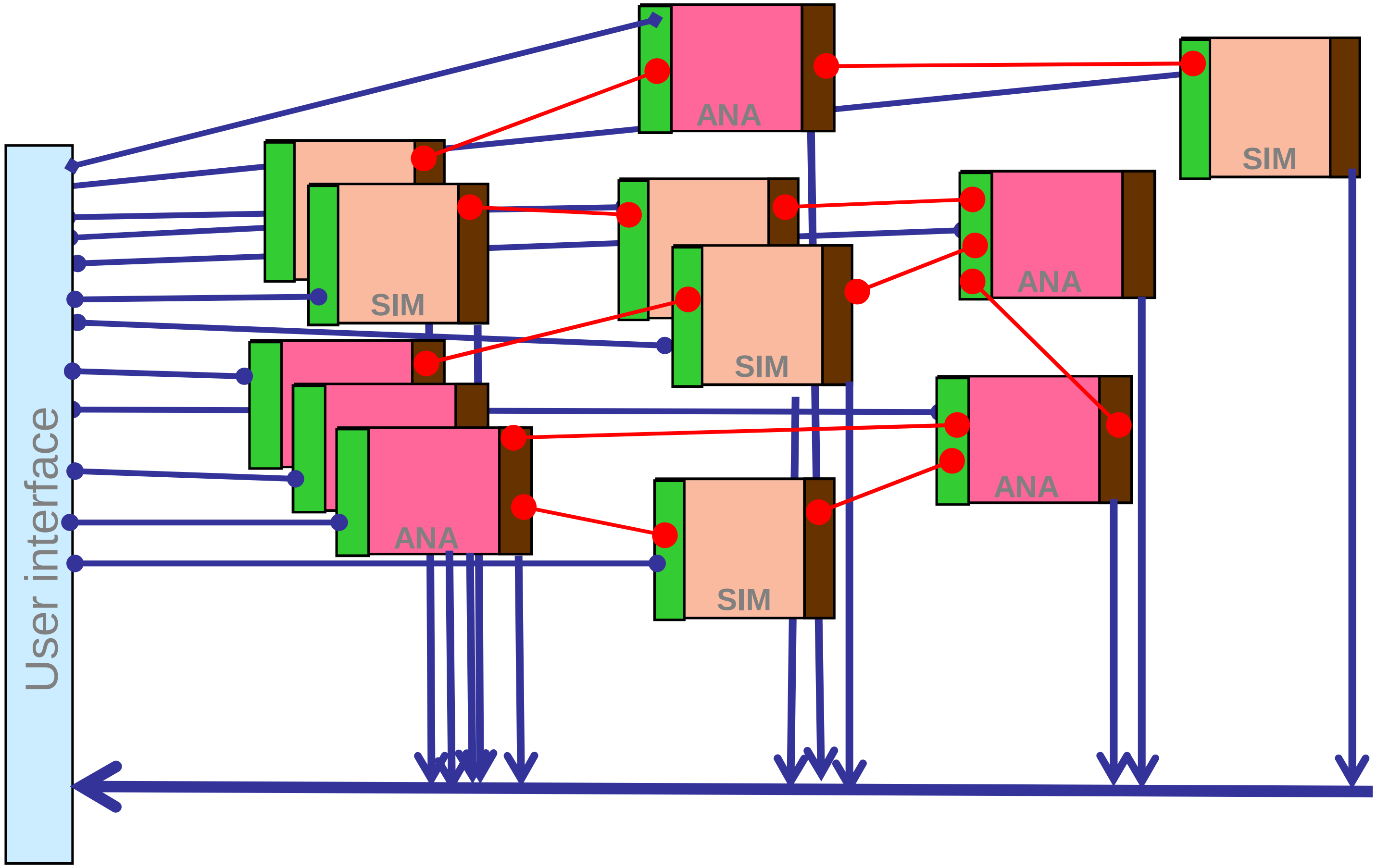
- with help of standard commissions to define tasks of agents, interface protocols,
- with groups testing each agent, the possibilities of each method, ranking them wrt the specifications of the application in case of multiple choice (eg, importance sampling wrt importance splitting)…

Remark : more Model Checking… more progress in manipulation of UML models…

Hopes



Hopes



As a Conclusion

Development of libraries and graphical interfaces during the last decades has increased our productivity !

But do the students still understand what the tools are doing below the graphical interfaces ?

→ **Risk 6 : lack of scientific background**

Shared feelings :

- Attraction of scientific studies is decreasing ; why ?
- The student who comes in CS wants more and more to practice !

New responsibility : to find the good levers to keep our young discipline as a branch where research remains a lovely way of life. For the new generation ? *(Ετσι ειναι η Ζωη !)*

Vielen Dank !

Ευχαριστω πολυ !

Thank You !

Merci !