Multiformalism modeling

An introduction to multiformalism modeling for performance evaluation

Prof. Ing. Mauro Iacono, PhD.
President of the European Council for Modelling and Simulation
Director of the ECMS International School for Young Researchers

1st ECMS International School
for Young Researchers
Politechnika Krakowska, 3/6/2024

Università
degli Studi
della Campania
Luigi Vanvitelli

ECMS
EUROPEAN COUNCIL FOR MODELLING AND SIMULATION

What will we do today?

- What is performance modeling? Why do we use it?
- Examples: Queuing Networks, Stochastic Petri Nets
- Multiformalism modeling

Performance modeling

- *Performance Evaluation* is the quantitative and qualitative study of systems, to evaluate, measure, predict and ensure target behaviors and performances
- It is usually carried on using *models of a system*
- A model is an abstraction of a system:

  "an attempt to distill, from the details of the system, exactly those aspects that are essentials to the system behavior"....

  (E. Lazoswka)

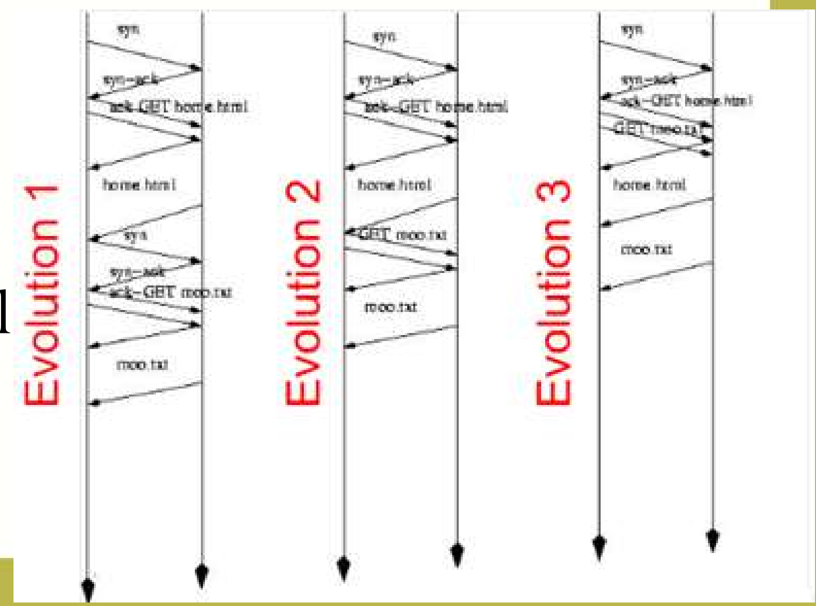
- After a model has been defined, it is usually exploited in three steps:
  - Validation
  - Projection
  - Verification

Performance modeling

- During the *validation* phase, results predicted by the model are compared against measurement of the real system to check if they match
- The *projection* changes some of the parameters of the model (i.e. the speed or the quantity of a component) and computes the corresponding indices to see if a goal is met in the new configuration
- *Verification* actually modifies the real system according to the new parameters tested in the model, to check whether the objective has been really achieved

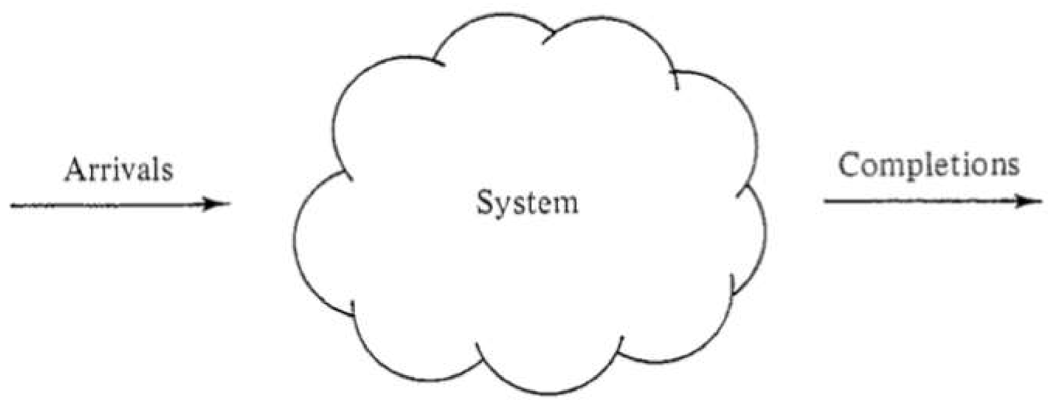
Solution techniques

- Once a model has been defined, its performance indices are computed using suitable solution techniques. Different procedures exists, and the most common are:
  - *Analytical* and *numerical* techniques are based on the application of mathematical techniques, which usually exploit results coming from the theory of probability and stochastic process
    - If the technique uses expressions in closed form (i.e. a formula or an exact algorithm to compute them can be derived), it is said to be *analytical*
    - If the solutions can only be obtained using numerical procedures, the tech-niques are said to be *numerical*
  - Simulation is based instead on the reproduction of *traces* of the model
    - A trace is a possible sequence of events that can characterize one possible evolution of the model



A problem with arrivals and departures

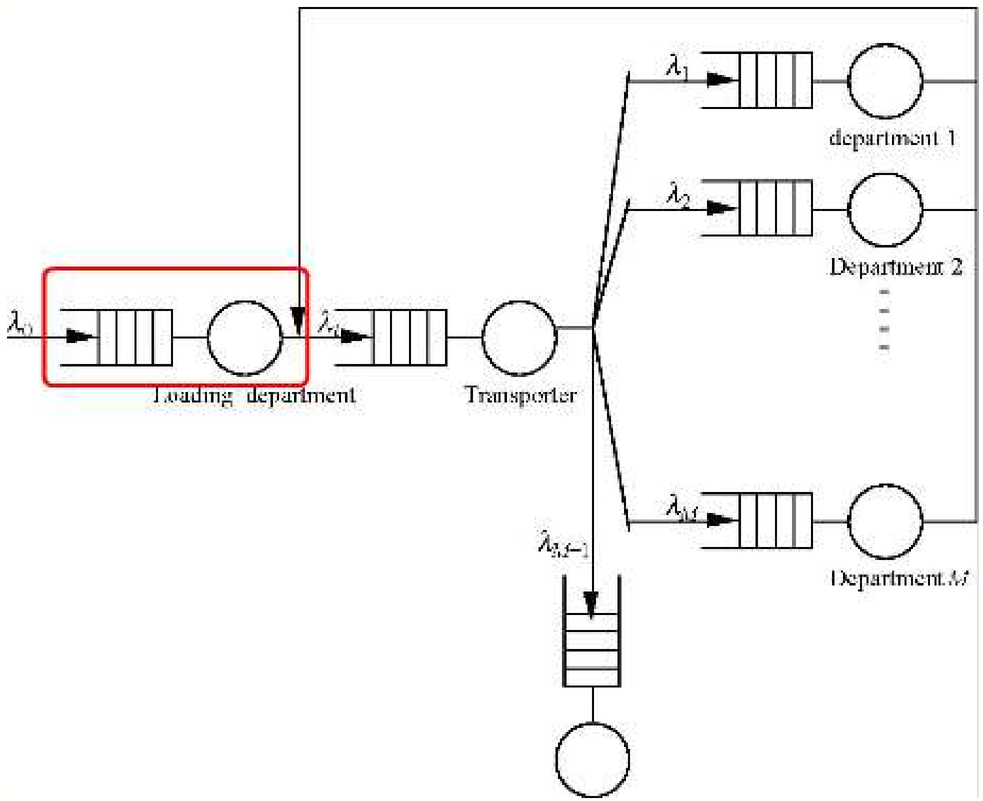
- Let us observe an abstract system, where we can only focus on its entry and exit point, for a given amount of time T



- By simply counting the jobs that enters and leaves the system in the considered time frame, we can determine its main parameters (*arrival rate* and *average service time*), and performance indices (*utilization*, *throughput*, *average service time*, *average number of jobs in the system*)

Example: Queuing Networks

- *Queuing Networks* represent systems using a set of interconnected entities called queues
- Born in the TLC domain, they are widely adopted whenever queuing effects impact on system performances

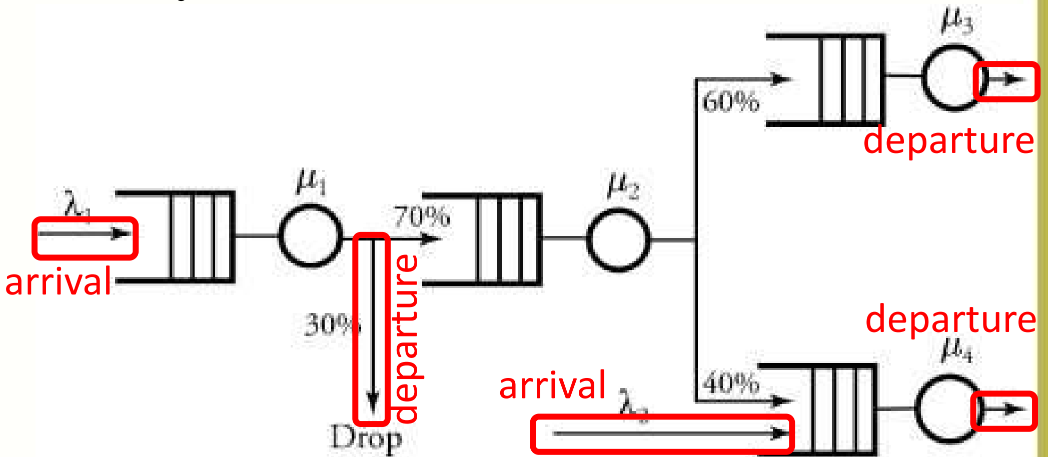


Queuing stations

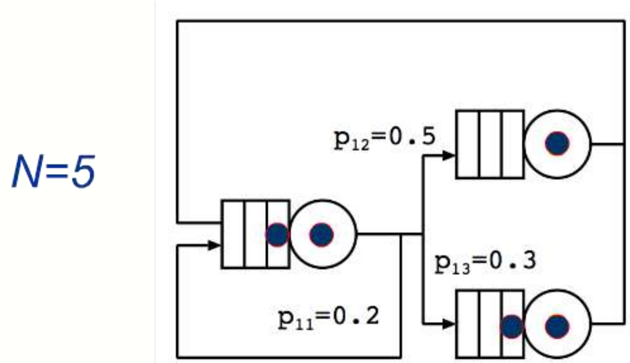
- *Queuing stations* can be used to model several elements of a system:
  - CPUs in multi-tasking S.O., disks, web services, communication channels, routers with buffers
  - Manufacturing machines
  - ...
- Queues are populated by entities that require services
- Depending on the context, such entities are called customers, clients, jobs, tasks, packets, tokens, ...
- Stochastic characterization of *service times* and *arrival rates*
- Evaluation of the stability conditions
- Strong mathematical foundations allow analytical evaluation, event-based simulation as alternative

Open and closed networks

- Queuing networks can be either *open* or *closed*
  - In *open models* jobs arrive from outside at a specified arrival rate
  - In *closed models* there is a fixed population of jobs that moves between the queues inside the system
- *Open Models* are characterized by *arrivals* and *departures* from the system



- In closed models we have a fixed population of *N* jobs that continuously circulate inside the system



Performance indices

- From a service station, several performance indices can be computed, depending on the problem
- The most important are:
  - the *utilization*: is the fraction of time a server is busy (not waiting for new jobs to arrive)
  - the *response time*: is the average time spent by a job at a service center
  - the *average queue length*: accounts for the mean number of jobs in a service station (both the ones being served and the ones in the queue)
  - the *throughput*: describes the rate at which jobs are served and depart from the station

Network performance indices

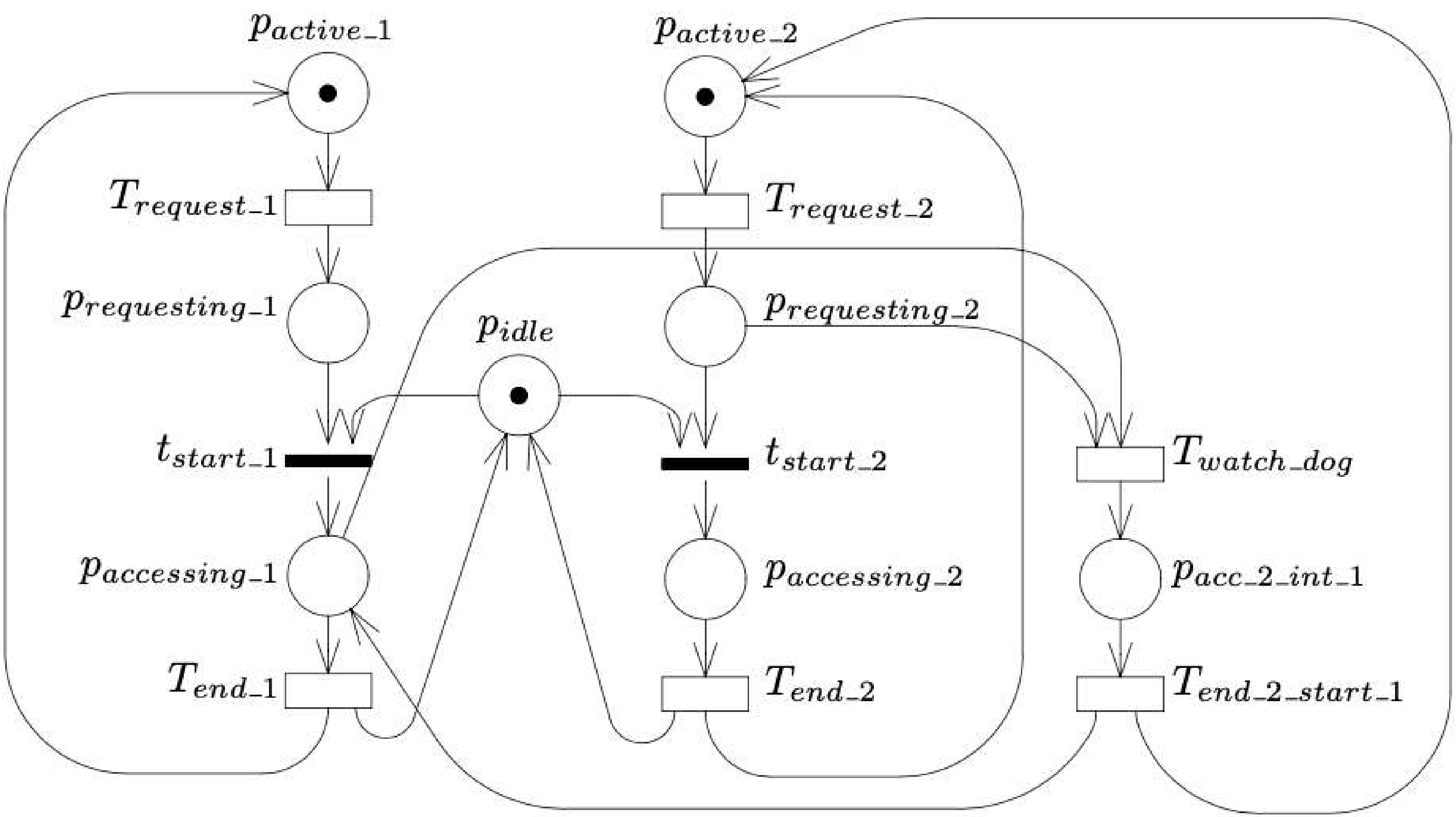
- On network models, extra performance indices can be defined:
  - System throughput
  - Total system population
  - System response time
- For what concerns utilization, there is no unique definition of a system-wise measure:
  - The fraction of time in which there is at least one job in the system
  - The average utilization of all the stations
  - The utilization of the bottleneck station
- All definitions have strength and weaknesses

Example: Stochastic Petri Nets

- Queuing networks are perfect to model systems where jobs are executed through a set of stations
- They are characterized by convenient high-level performance indices such as throughput, response times and utilization
- However, they cannot easily model *resource contentions* and *concurrency*
- Other formalisms, such as *Stochastic Petri Nets*, are used to model systems characterized by such features

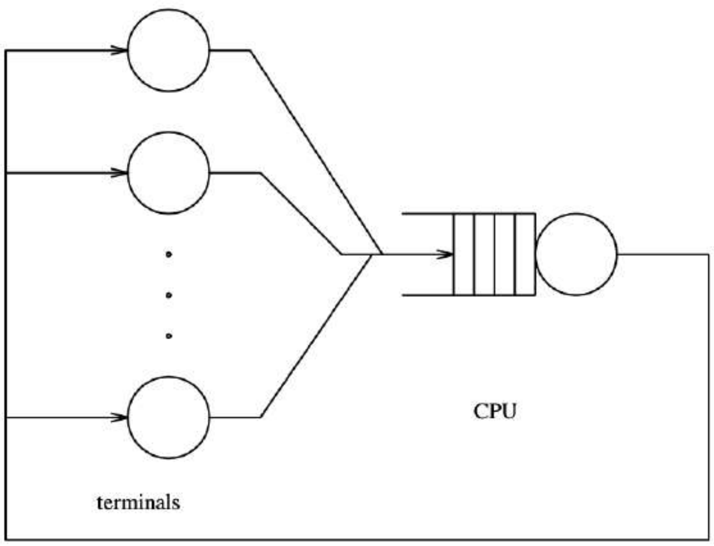
Petri Nets

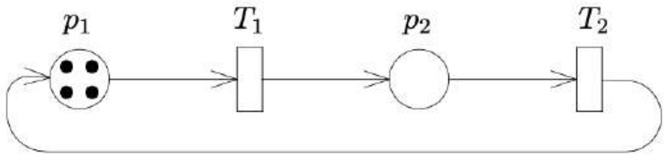
- Petri Nets are *bi-partite graphs*, characterized by two set of elements: *places* and *transitions*



Petri Nets performance models

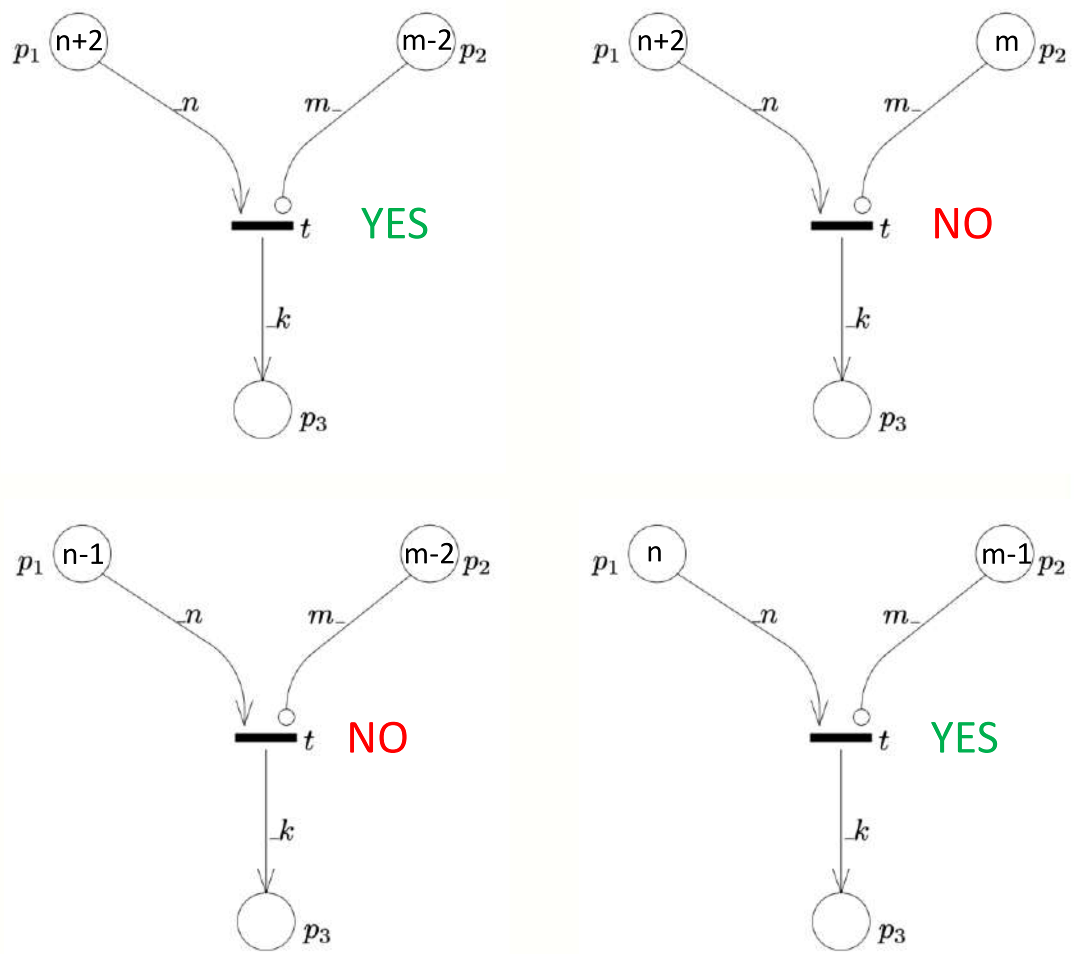
- The meaning of the different type of arcs is the following:
  - *input arcs* model *preconditions*, that must be satisfied for an event to take place
  - *output arcs* are used to specify the *effect* of the actions
  - *inhibitor arcs prevent* an event from taking place
- In performance models created using Petri nets, tokens are used to represent jobs, places to define queues or resource occupancy, and transitions correspond to services





Enabled transitions

- A transition is said to be enabled if:
  - Each input place has at least as many token as the weight of the corresponding input arc
  - Each place connected with an inhibitor arc has less tokens than the weight of the connection



Firing time distribution

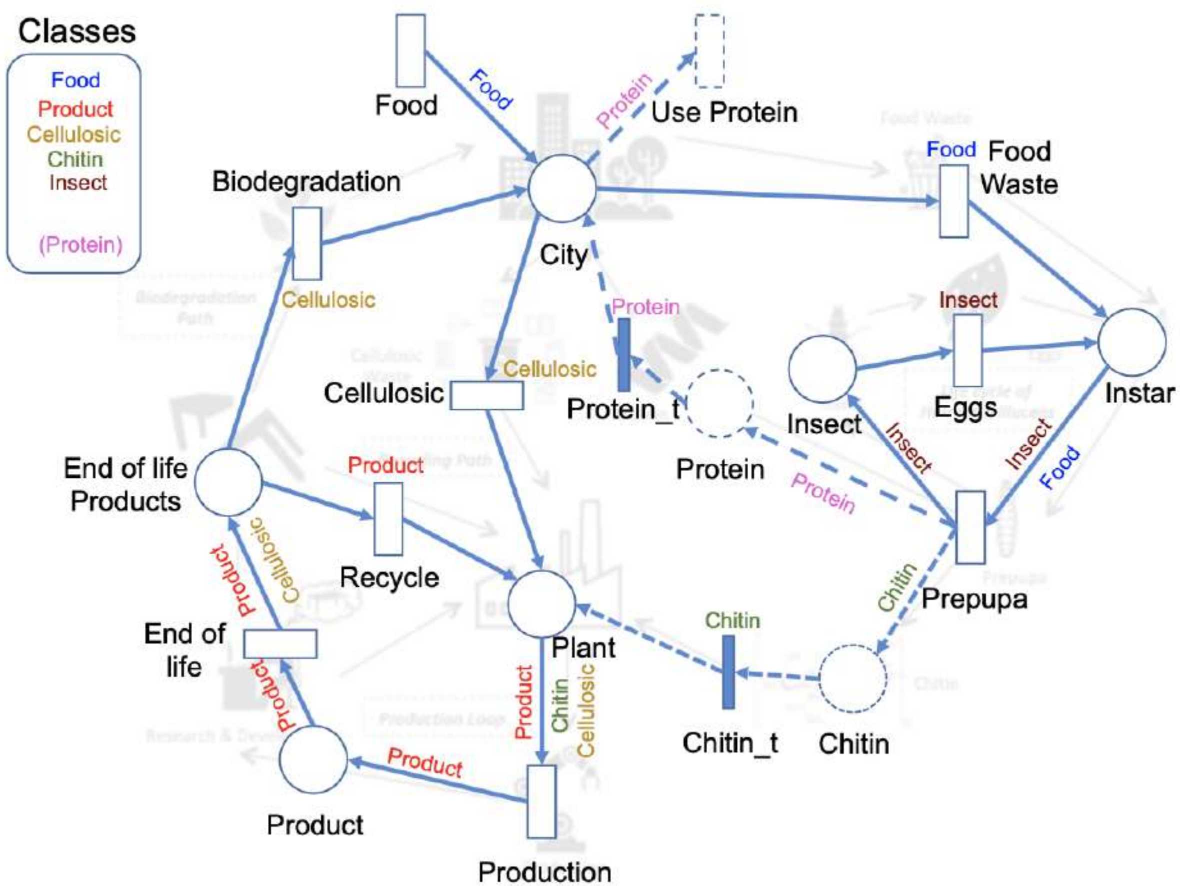
- The original definition of *Stochastic Petri Nets*, associate to each transition an *exponentially distributed random* firing time
- Extended models, called *Non-Markovian Stochastic Petri Nets*, allow to use general firing time distributions
- Conflict are solved with a *race policy*: the system will evolve according to the transition that will fire first

Performance indices: transition throughput

- Only three types of performance indices are defined on PNs
  - the transition throughput counts the average number of firings per time unit done by a transition
  - for places, it is relevant to compute either the probability distribution of having a given number of tokens, or the average number of tokens inside it

Colored Petri Nets

- In *Colored Petri Nets*, tokens are divided into classes called *colors*: each place might contain a different number of tokens for each color



Modeling complex systems

- Problem: modern complex systems have *different aspects* related to *different domains* and *different expertise* but must be represented and evaluated as a whole
- Example: ERTMS/ ETCS

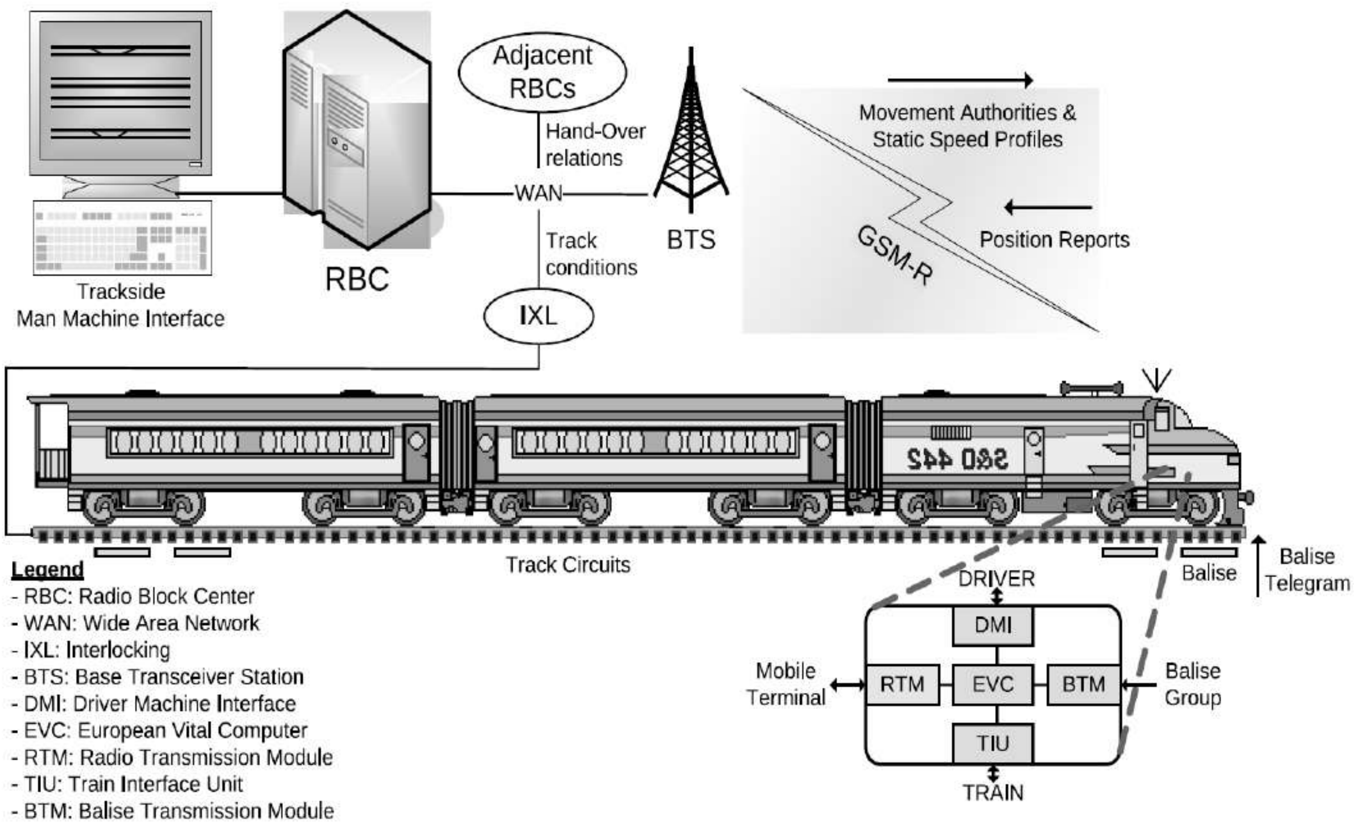


Fig. 4. ERTMS/ETCS system

Multiformalism models

- Multiformalism models allows to exploit different formalisms to describe different components of a system
- They can use the *most appropriate modelling primitive* for each part, and can combine different modeling languages

- One of the main advantages of multi-formalism modelling is its ability to represent the system at *multiple levels of abstraction*
- This allows a *more comprehensive and accurate representation* of the system, which can *facilitate the analysis* and optimization of the behavior of the system

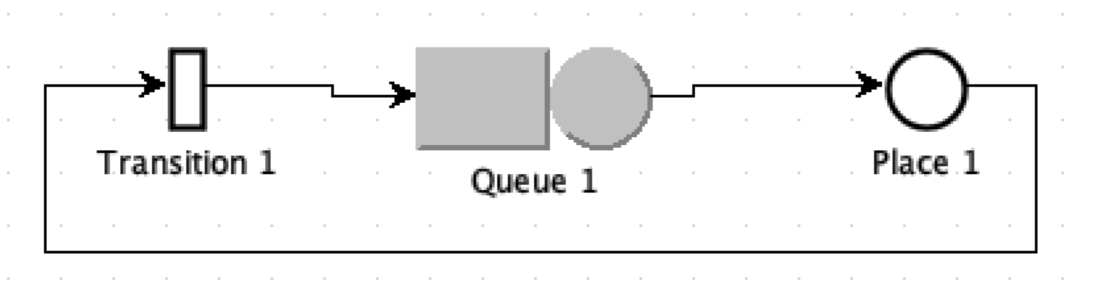
Definition and challenges

- "Multi-formalism modeling is the combination of different formalisms, such as Petri nets, process algebras, and queuing networks, to capture different system behaviours and characteristics"

- Main problems: *semantics, analysis*, *representation*, *modularity/compositionality*, *coherence*

Example of a simple multiformalism model

- One of the most popular example of multiformalism modeling technique is the combination of Queuing Networks and Petri Nets
- If a transition is connected to a queue, whenever the transition fires, it inserts as many jobs as the weight of its arc in the destination queue
- The class of the jobs corresponds to the color of the tokens



Example: architecture

- The lineside subsystem: it is mainly responsible for providing geographical position information to the on-board subsystem;
- The on-board subsystem: it is the core of the control activities located on the train;
- The trackside subsystem: it is in charge of monitoring the movement of the trains.

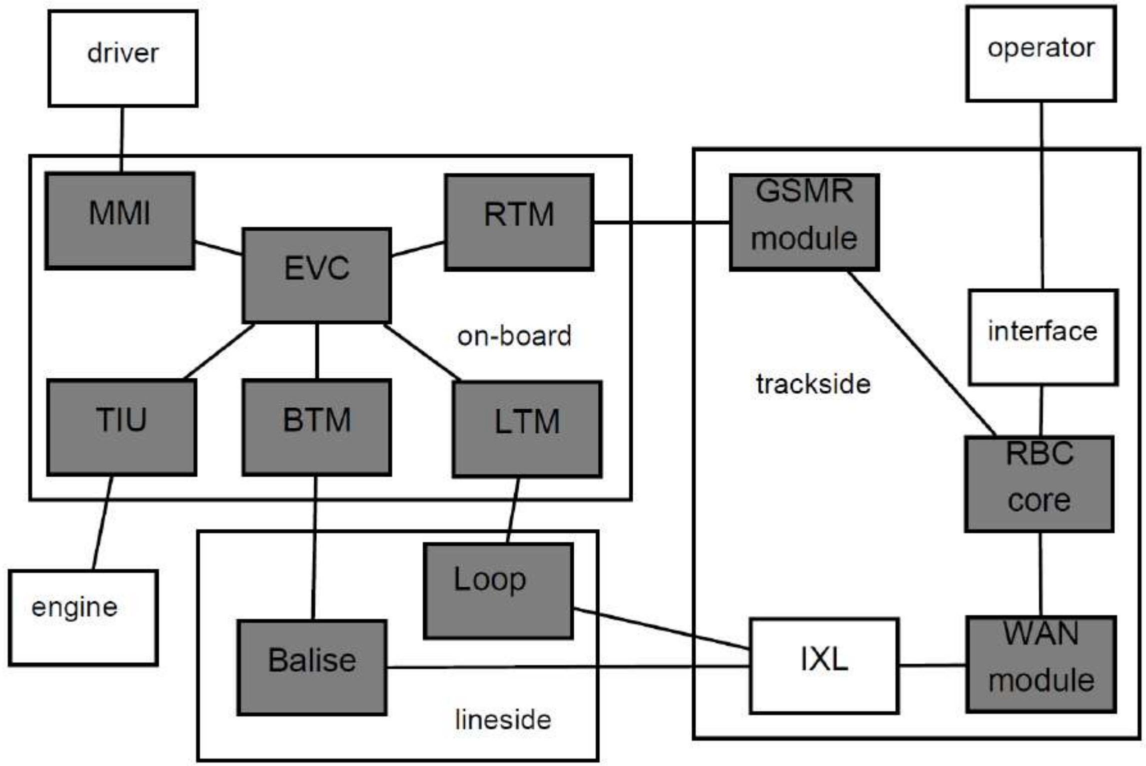


Fig. 1. ERTMS/ETCS architecture

Example: UML component diagrams

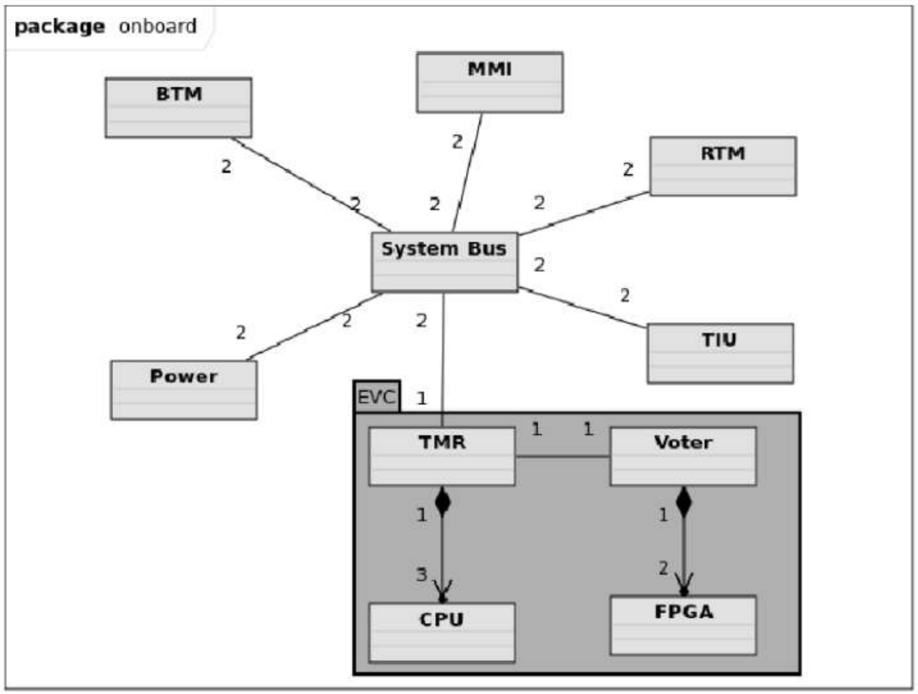


Fig. 2. Class diagram of the ONB subsystem

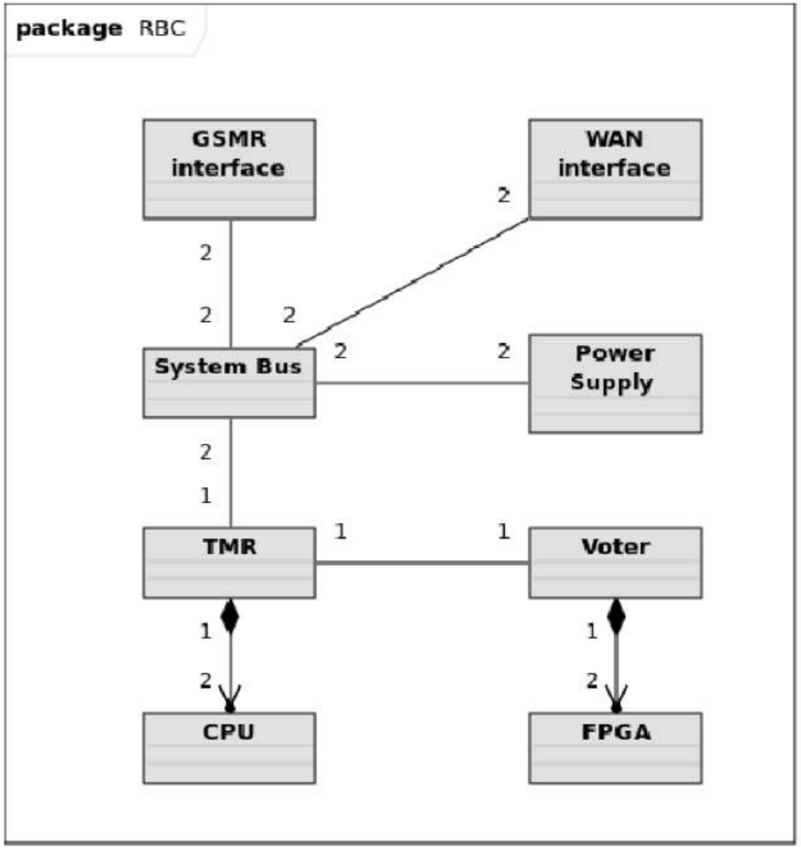


Fig. 3. Class diagram of the RBC component

Example: scenario components

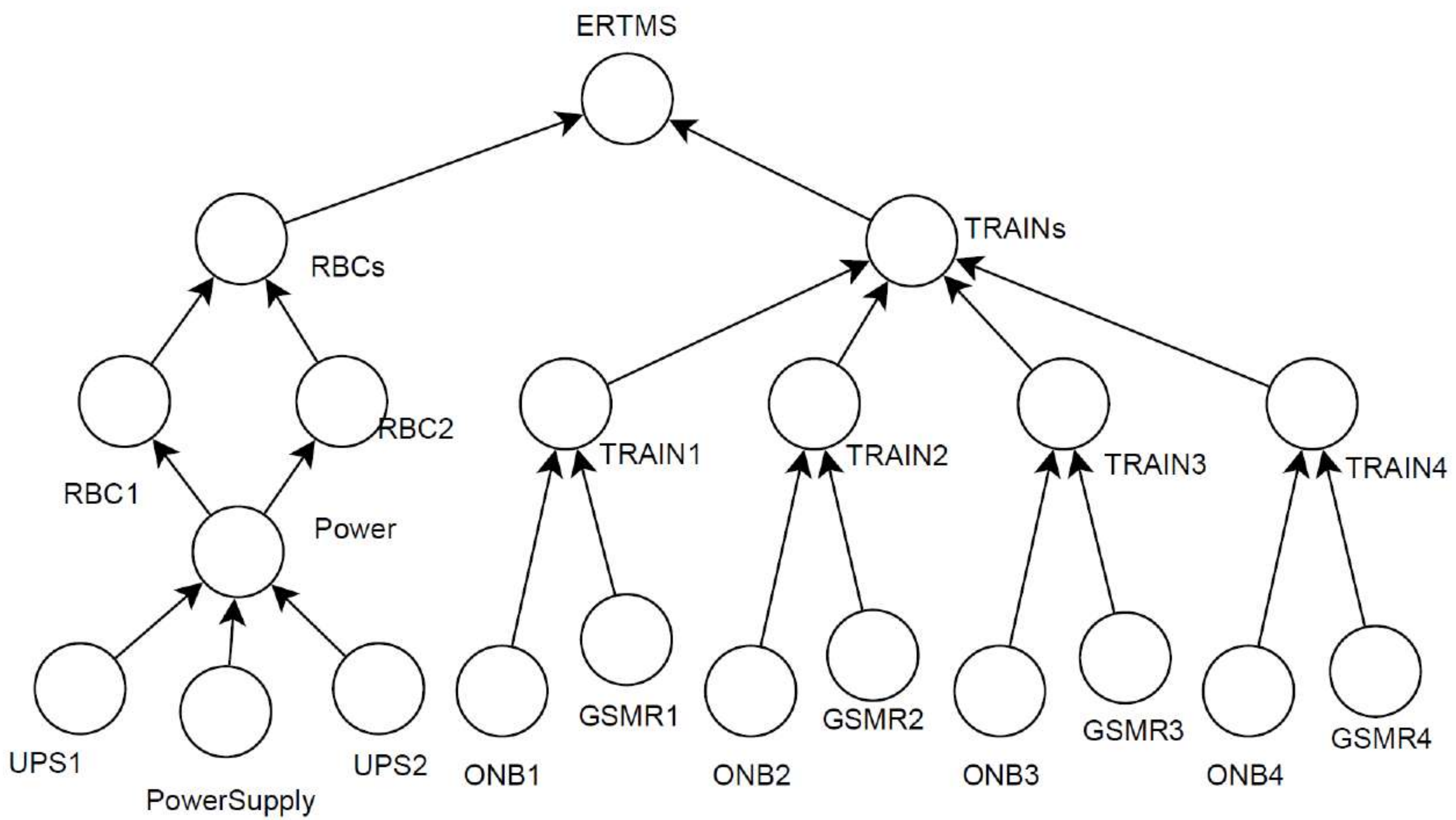


Fig. 5. Contributions of the different sub-models to the overall analysis

Example: reliability

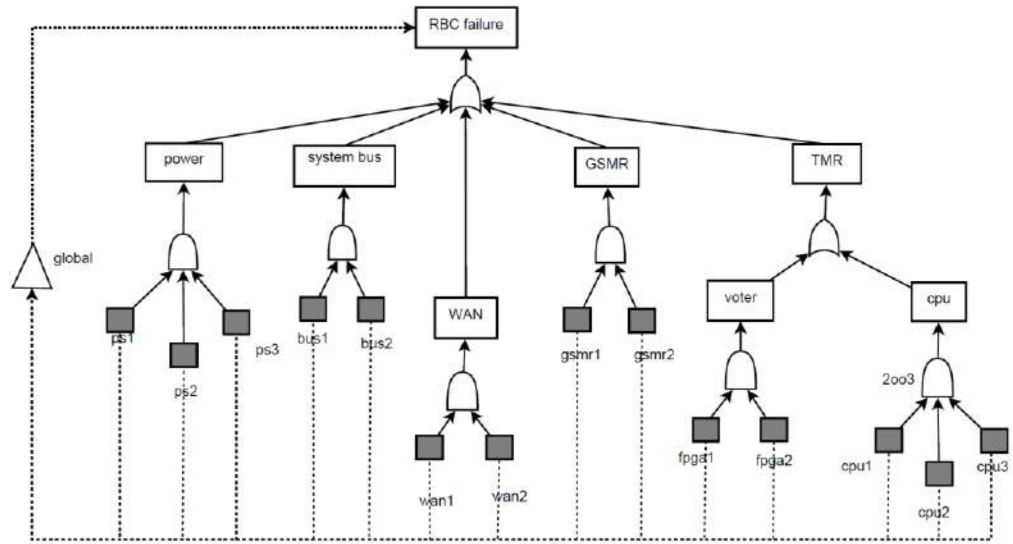


Fig. 7. RFT model of the RBC

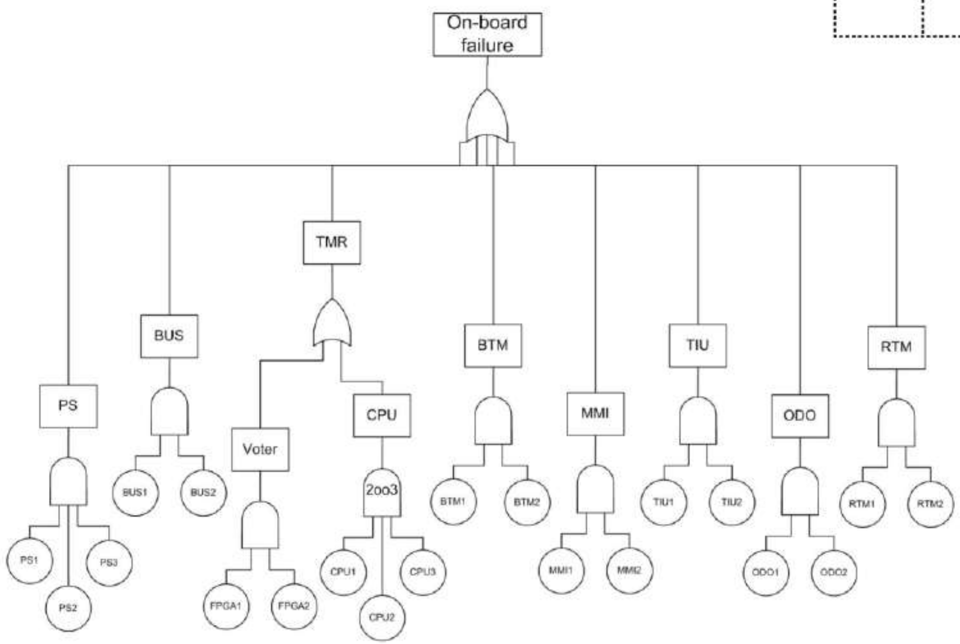


Fig. 6. The FT model for the On-board subsystem

Example: performability

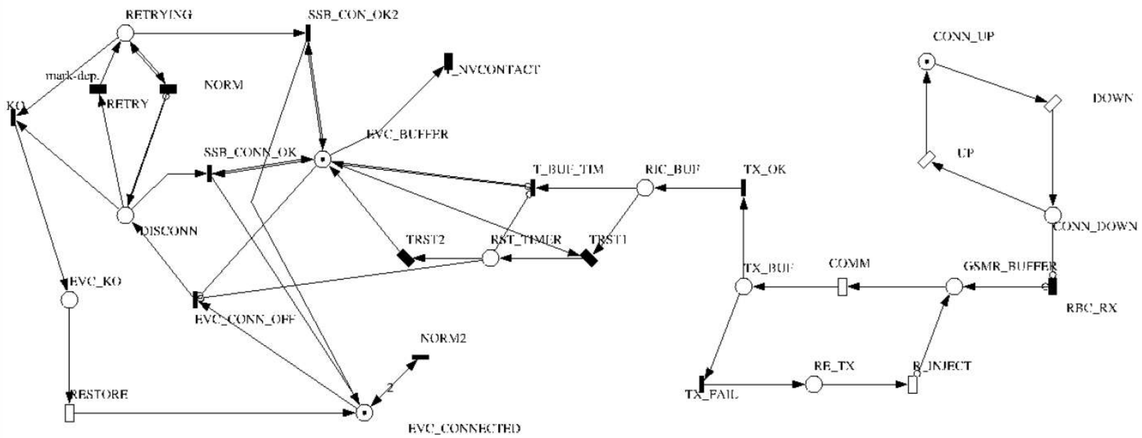


Fig. 9. The structure of the GSPN performability model

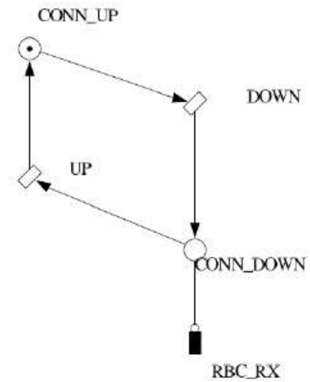


Fig. 10. RBC communication module in vital communication model

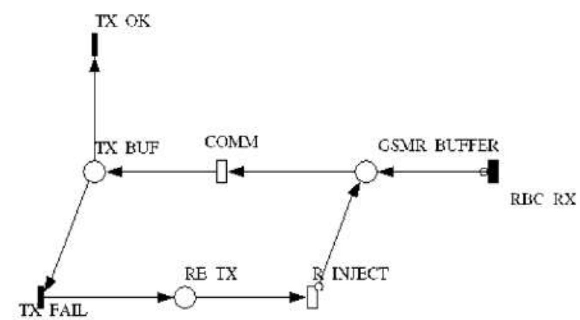


Fig. 11. GSM-R module in vital communication model

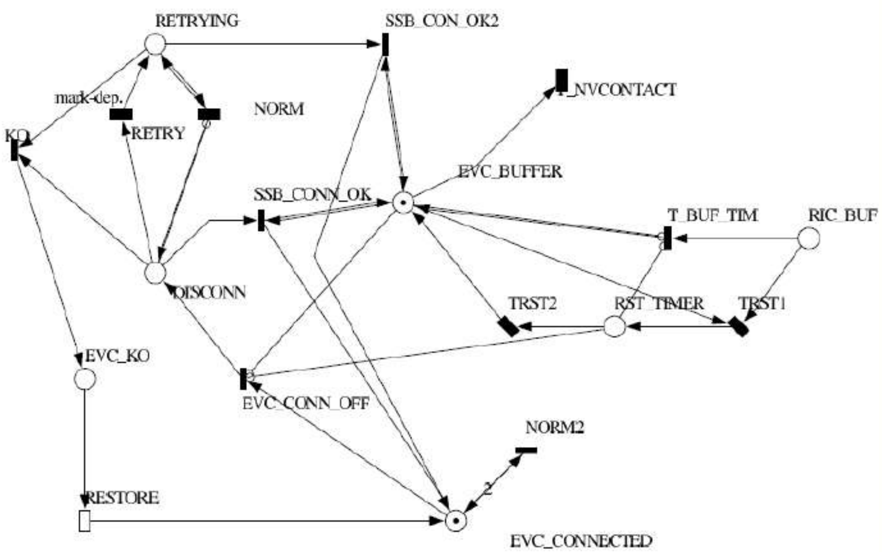


Fig. 12. On-board timing module in vital communication model

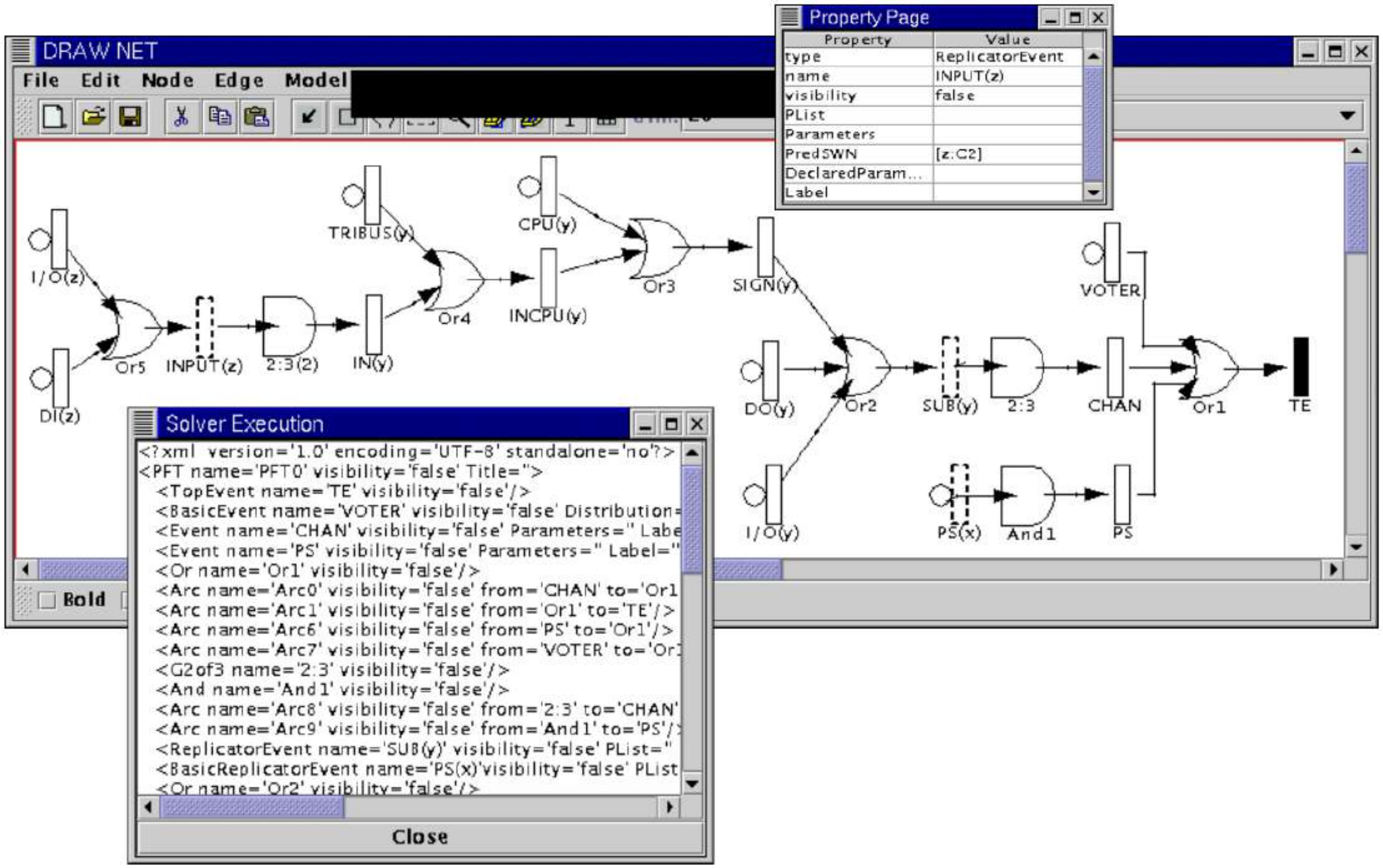
So what?

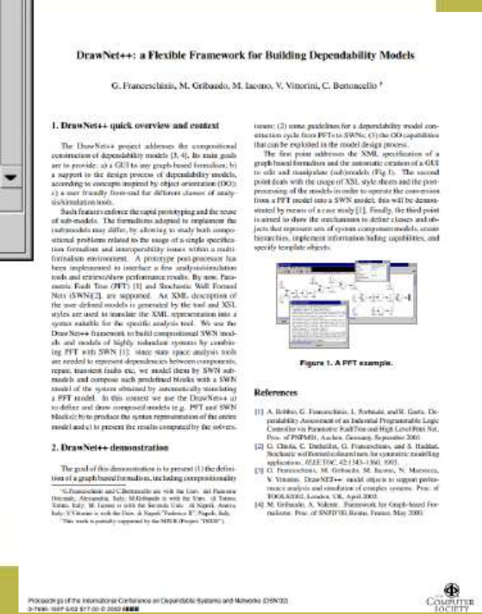
- We analyzed the different submodels for the different aspects of the system
- All submodels showed compliance with specifications and expected system behavior
- **The overall multiformalism system analysis showed that there is a possibility of system failure with no component failure because of combined effects of a legitimate combination of delays of computing and transmissions and a misinterpretation by the control logic**
- Such a problem is a *higher level system effect* not trackable to any of the logs of the components, which show absolutely no error

Doing multiformalism

- What do we need to use multiformalism modeling?
  - Proper model composition tools
  - Proper model analysis strategies
  - Proper model analysis tools
  - A novel theoretical approach

Tools to define multi-formalism models





DrawNet++: a Flexible Framework for Building Dependability Models

G. Franceschinis, M. Gribaudo, M. Iacono, V. Vittorini, C. Bertoncello

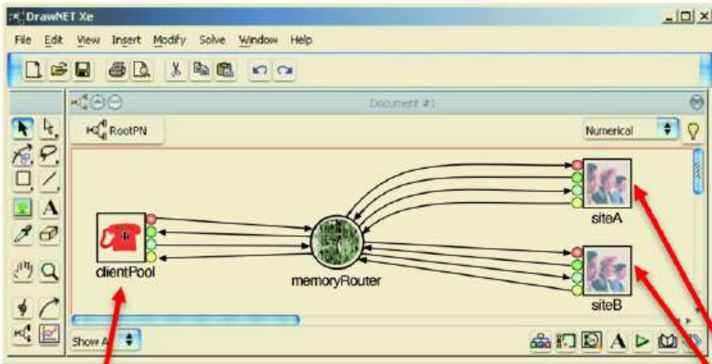
1. DrawNet++ quick overview and context

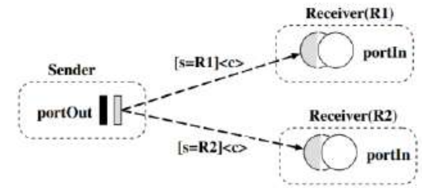
2. DrawNet++ demonstration

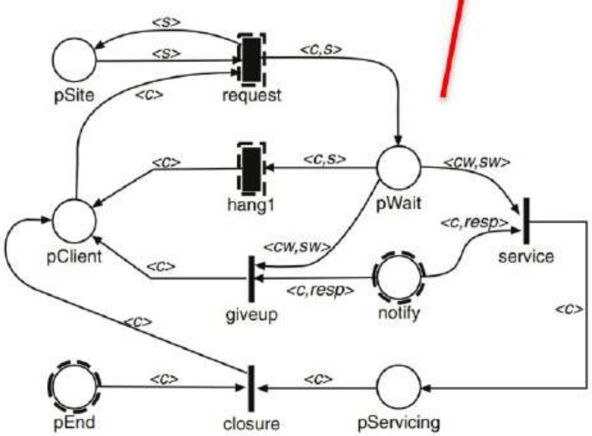
Figure 1. A PFT example.

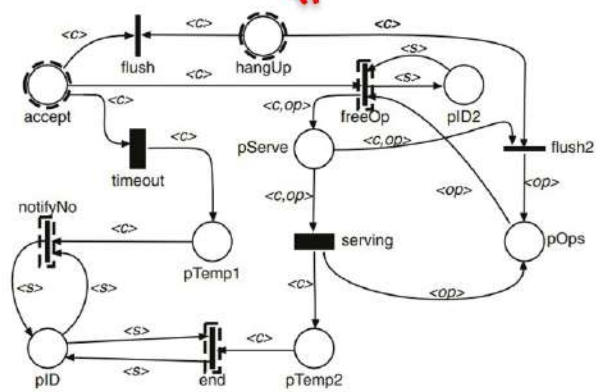
References

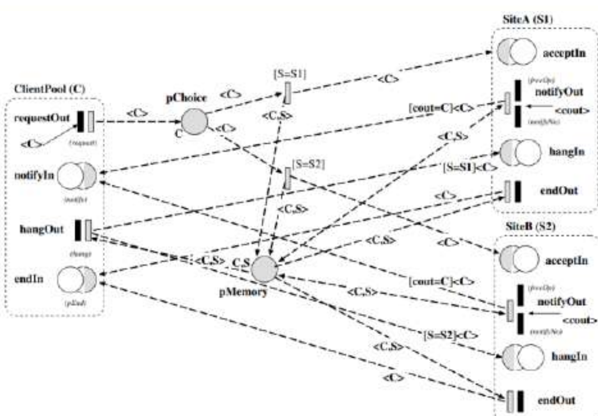
Multi-solution by orchestration

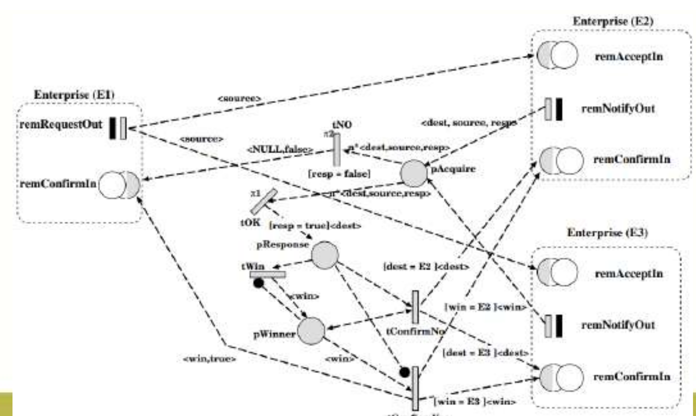












Compositional Modeling of Complex Systems: Contact Center Scenarios in OsMoSys

Giuliana Franceschinis[1], Marco Gribaudo[2], Mauro Iacono[3], Stefano Marrone[3], Nicola Mazzocca[3], and Valeria Vittorini[4]

[1] Dipartimento di Informatica, Università del Piemonte Orientale, Piazza Ambrosoli 5, 15100, Alessandria, Italy
giuliana@mfn.unipmn.it
[2] Dipartimento di Informatica, Università di Torino, Corso Svizzera 185/B, 10149 Torino, Italy
marcog@di.unito.it
[3] Dipartimento di Ingegneria dell'Informazione, Seconda Università di Napoli, Via Roma 29, 81031 Aversa, Italy
{mauro.iacono,stefano.marrone,nicola.mazzocca}@unina2.it
[4] Dipartimento di Informatica e Sistemistica, Università degli Studi di Napoli "Federico II", Via Claudio 21, 80125 Napoli, Italy
vittorin@unina.it

Abstract. In this paper we present the application of a compositional modeling methodology to the re-engineering of Stochastic Well Formed net (SWN) models of a contact center. The modeling methodology is based on the definition of proper operators to connect submodels and it is supported by the OsMoSys modeling framework. The paper describes the implementation of a library of reusable SWN submodels of the contact center components and the definition of proper SWN connectors to easily develop models of different configurations of the system. We also describe the solving process of the composed models and its integration in the OsMoSys framework. Moreover, we discuss the advantages that this approach, based on the definition of classes and instances of submodels, can provide to the application of SWN to complex case studies.

1 Introduction

Compositionality is an effective mean to cope with the complexity of real system modeling and analysis. It can also be exploited to set up a framework where designers, who may not be expert in formal modeling languages, compose models of complex scenarios by just choosing building blocks from a library and connecting them. The building blocks could hide complex models built by experts.

In the context of Petri nets (PN) a number of proposals have appeared in the literature to add model (de)composition features to a formalism that is not compositional in its basic definition (e.g. [1,3,10,11]). In particular, in [13, 14] PNs are used to describe the behavior of object classes and communication mechanisms are defined to allow the object models to inter-operate. The approach proposed in [7] adapts this model composition paradigm to Stochastic

J. Cortadella and W. Reisig (Eds.): ICATPN 2004, LNCS 3099, pp. 177–196, 2004.

"Real" multi-formalism models

- Multiformalism potential is not limited to a joint analysis of heterogeneous submodels
- In the Mobius approach a superformalism towards which all formalisms are translated allow the generation of a single analysis model
- In the SIMTHE*Sys* approach *elements from different formalisms may actually natively interact* by specifying their elementary interactions and complex state change conditions, to account for actual behaviors of a system

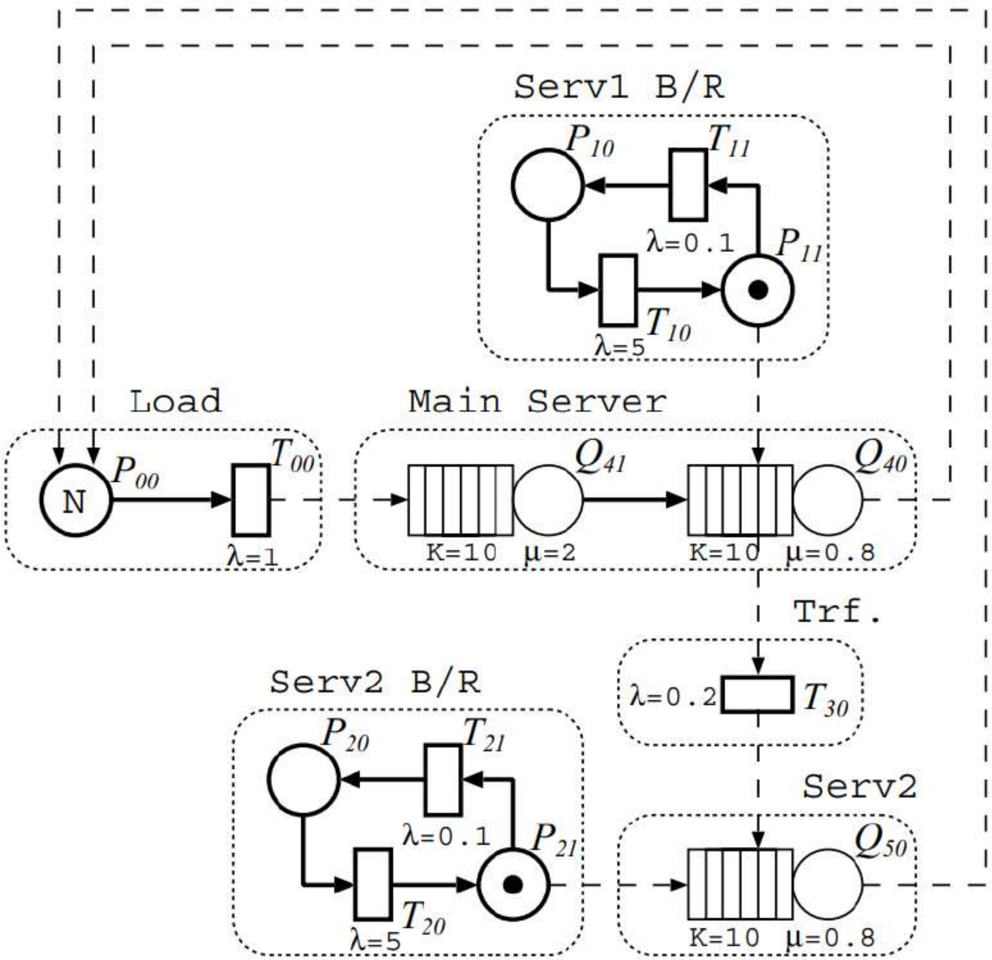

Element Based Semantics in Multi Formalism
Performance Models

“Real” multi-formalism models

- The main idea was finding an object-oriented way of describing the interactions between modelling primitives
- The abstraction offered by OOP could immediately define an interconnection semantic between primitives of different formalisms

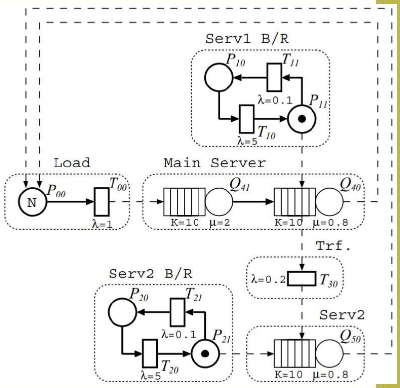
"Real" multi-formalism models

- A closed system, with servers that can break and repair, together with the possibility of transferring a job to a secondary system in case of long waits

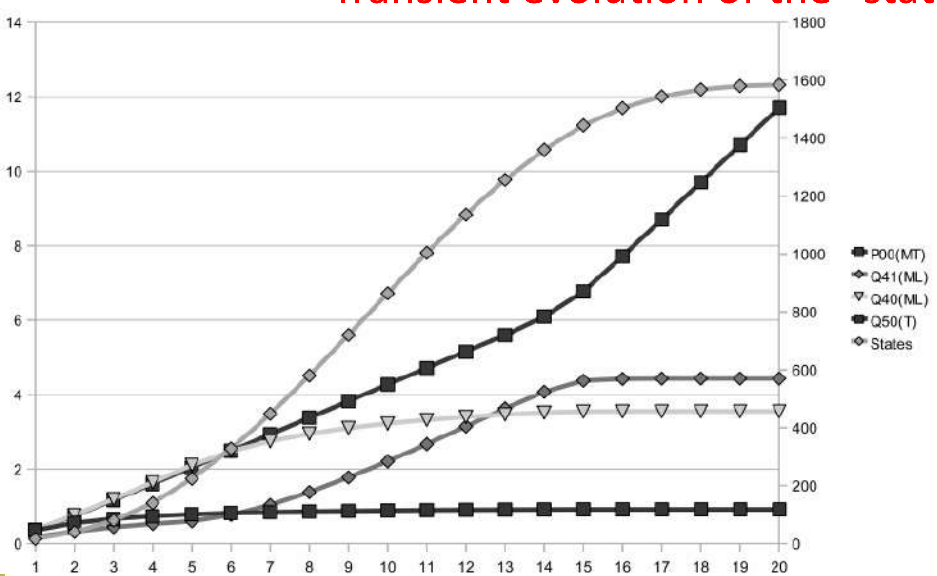


“Real” multi-formalism models

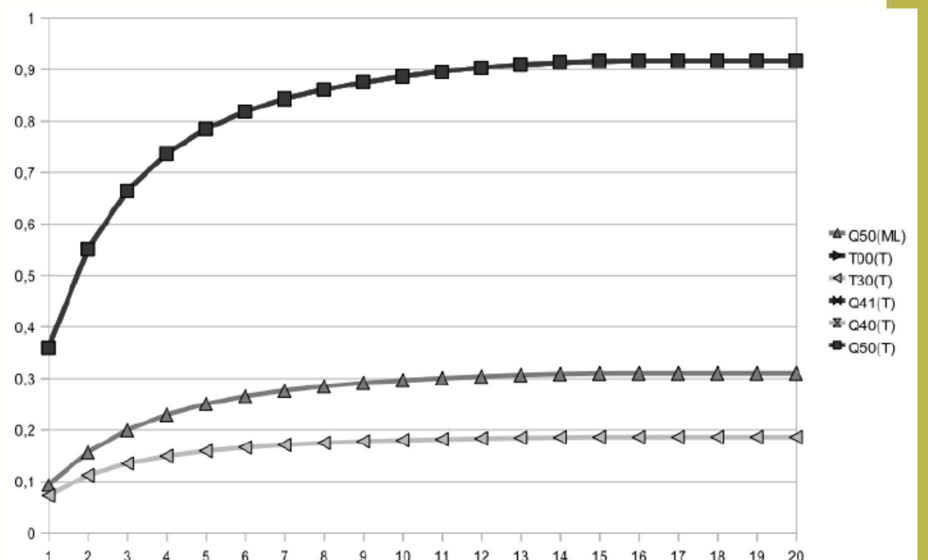
- The proposed rules allow either to produce a Continuous Time Markov Chain for numerical analysis, or to compute a solution using discrete events simulation



Transient evolution of the "states" of the primitives.



N – number of jobs



N – number of jobs

Other interesting cases

- Specifying complex verification properties
  - Petri nets to specify behaviors, Fault Trees to evaluate conditions, automata to assess properties
- Modeling the effects of software rejuvenation policies
  - Combining a traditional modelling language with a *Domain Specific Language* properly defined to represent alternate software rejuvenation policies
- Hybrid systems
  - Modeling systems with continuous time and discrete time components

Exploiting multiformalism models for testing and performance evaluation in SIMTHESys

Multiformalism to support software rejuvenation modeling

Modeling Hybrid Systems in SIMTHESys

Rethinking PN/QN connections


Places, Transitions and Queues: New Proposals for Interconnection Semantics

- We defined non-traditional PN/QN connections
- We started considering possible applicative scenarios of the connections, aiming at making models "*readable*"

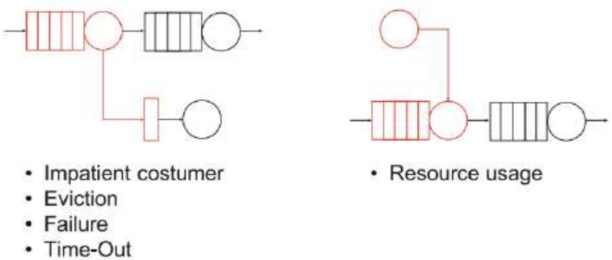


Fig. 2. Advanced connection types: queue to transition and place to queue.

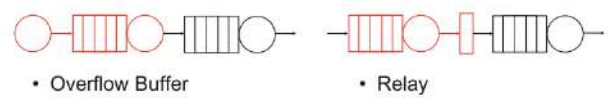


Fig. 4. Advanced connection types: exclusive connection from place to queue and from queue to transition.

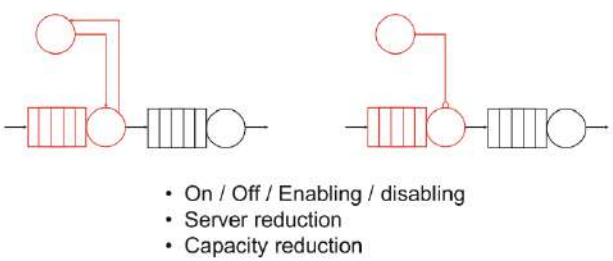


Fig. 3. Advanced connection types: test and inhibitor arcs.

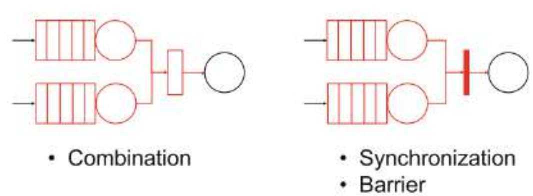
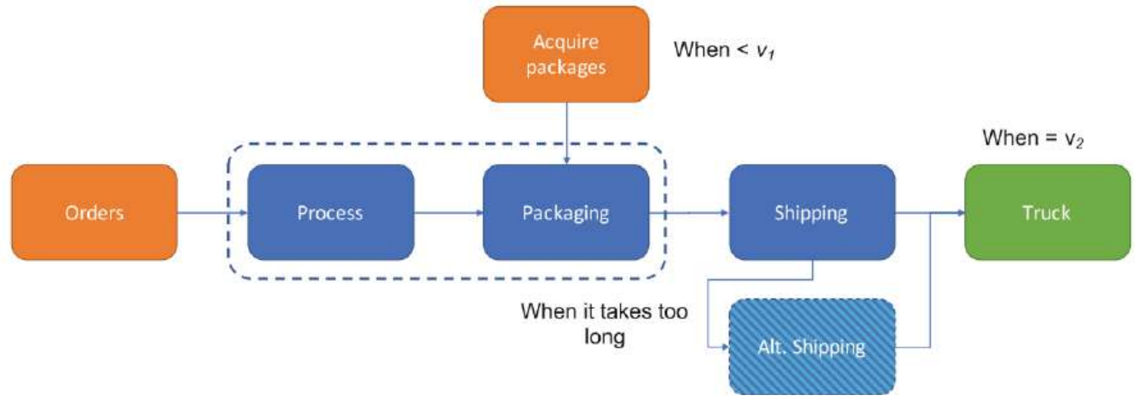
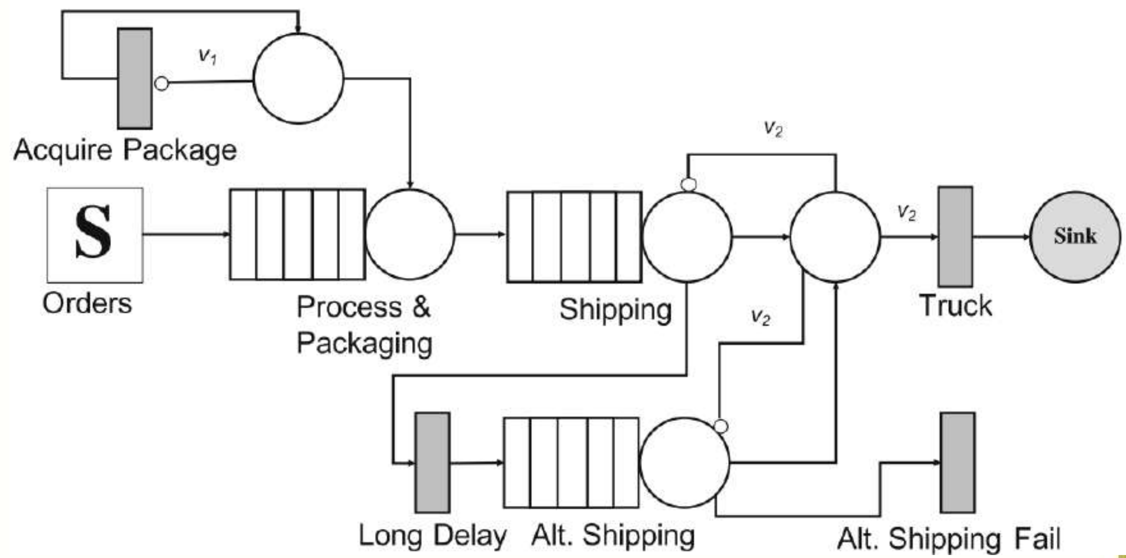


Fig. 5. Advanced connection types: from several queues directly to a timed or immediate transition.

Rethinking PN/QN connections

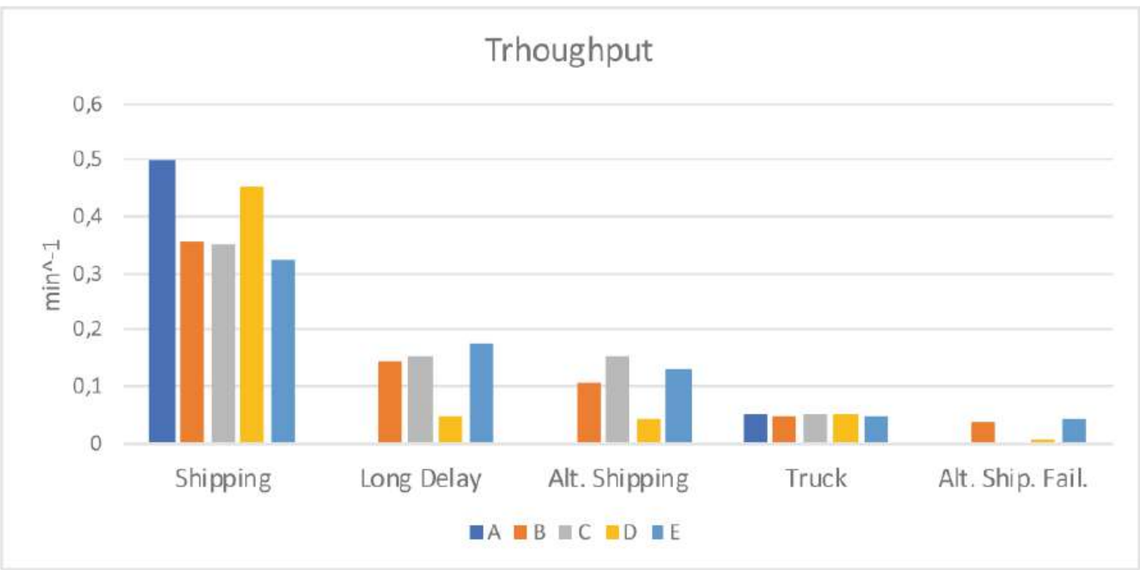
- We tried to test different semantics, in a simple goods delivery model
- These semantic focused on differences between jobs in service and waiting in the queue

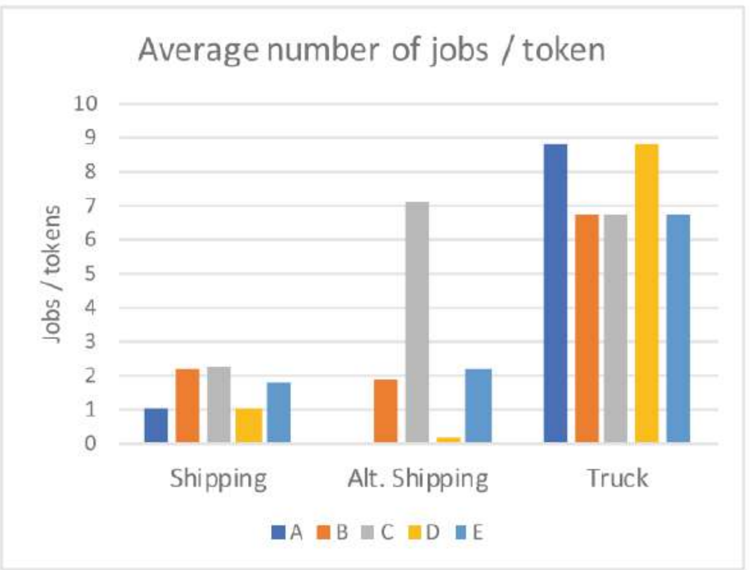




Rethinking PN/QN connections

- We showed that the different semantics have a strong impact on the performance measures that can be computed on the models:
    - they cannot be simply ignored, or left unspecified!





Questions?

The ERTMS/ETCS running example models are from Francesco Flammini, Stefano Marrone, Mauro Iacono, Nicola Mazzocca, Valeria Vittorini, A Multiformalism Modular Approach to ERTMS/ETCS Failure Modelling. International Journal of Reliability, Quality and Safety Engineering, vol. 21, num. 1, pp. 1450001-1-1450001-29, World Scientific , ISSN: 0218-5393, DOI: 10.1142/S0218539314500016

My gratitude goes to my friend prof. Marco Gribaudo for 25 years of work together and for providing part of the materials used in this presentation

Università degli Studi della Campania *Luigi Vanvitelli*

Prof. Mauro Iacono
Professore associato in Sistemi di Elaborazione delle Informazioni
mauro.iacono@unicampania.it