Petri Nets modeling for the schedulability analysis of industrial real time systems

University of Florence, Italy
DINFO[1]

Italian Workshop on
Embedded Systems - IWES 2016

Author:
Stefano Pepi[1]
Alessandro Fantechi[1]

19-20 September, 2016 - Pisa, Italy

INTRODUCTION

This work has been conducted within a long term colaboration between University of Florence and General Electric Transportation Systems (now part of ALSTOM), focused on software development and verifcation techniques for **safety-critical railway signalling systems.**

Inside the Delta project, GETS has developed in the period 2011-2014 its own (certified) Real Time Operating System for the new hardware platform developed to support future more demanding applications, in order not to depend on external RTOS providers.

INTRODUCTION

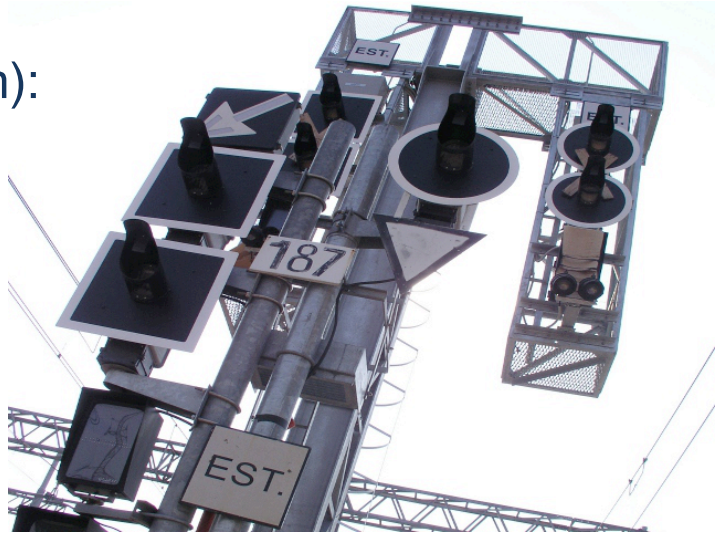
Keywords:

- Safety of Railway Signalling Systems
- Pre-Runtime Scheduling
- Formal modeling
- Formal verification
- Petri Nets

Scope:
We report an approach aimed at substituting possibly unreliable and costly empirical measures with rigorous analysis done by modeling a fixed scheduler algorithm with Petri Nets.

INTRODUCTION

- This study was born from the collaboration with our industrial partner, a railway signaling manufacturing company.
- Previously produced Railway signaling systems did not need a real time operating System:
  - their operations are simply sequenced in a main loop.
- Innovating signaling solution (Operating System):
  - One modular Vital Platform.
  - Same hardware different use.
  - Increased safety and system flexibility.
  - Need to introduce a Scheduler.



REAL-TIME SYSTEMS

Real-Time Systems (RTS): computer-based systems where correct operation does not only depend on the correctness of the results obtained, but also on the time at which the results are produced.

A real-time process is characterized by a fixed time limit, which is called deadline. A result produced after its deadline is not only late, but can be harmful to the environment in which the system operates.

Safety critical systems often require **Hard real-time policy:**
if producing the results after its deadline may cause catastrophic consequences on the system under control.

PROPERTY OF A RTS

Predictability:

- Is the ability to determine in advance if the computation will be completed within the time constraints required.
- Depends on several factors, ranging from the architectural characteristics of the physical machine, to the mechanisms of the core, up to the programming language.
- Can be measured as the percentage of processes for which the constrains are guaranteed.
- The developed RTOS allows both dynamic and fixed scheduling, but **Predictable fixed scheduling is preferred in safety critical systems is preferred**

What experience shown:

- Guaranteeing predictability for the different customizations of the platform takes a considerable portion of the customization effort, if based only on testing every time the newly customized software on the platform.

RESEARCH SCOPE

What has been done:

- The process can be automated by a tool.
- Built a generic model of the scheduling algorithms employed in the platform that is going to be instantiated on the temporal constraints and tasks numbers of the different specific applications.
- Given a task set and a number of constraints, produce a feasible scheduling.
- In this way we can support the validation of predictability by means of proper model simulation tools.

PRE-RUNTIME SCHEDULING

Pre-Runtime scheduling:

- All decision are taken before the process activation on the basis of information known a priori.
- The schedule in stored in a table inside the operating system.
- Better possibility to demonstrate predictability to an assessor.

CENELEC EN50128:

- Generic application: software which can be used for a variety of installations purely by the provision of application-specific data and/or algorithm
- Specific application: defined as a generic application plus configuration data, or plus specific algorithms, that instantiate the generic application for a specific purpose.

- ➢ Platform is a part of a generic application and hence it is validated once for all.
- ➢ For each specific application the satisfaction of real-time constraints must be verified from scratch.
- ➢ Everyday work it is necessary to revise the schedule of some sytems.

PRE-RUNTIME CONFIGURATION

The estimated effort required for the adopted empiric identification and testing of a new configuration of scheduling can be summarized with the following parameters:

- **Offline Identification time:** time needed in order to design the new schedule, it is usually about 30 minutes.
- **Flashing time:** the time needed to load the scheduling on the target, 15 minutes.
- **Startup time:** start-up time of the platform, 1.5 minutes.
- **Running time:** time during which the system must run without exhibiting timing problems, 30 minutes / 1 hour.
- **Attempts:** average number of attempts to get the scheduling, 3.

RESEARCH SCOPE

- The whole process easily reaches 8 hours, which means an entire working day.
- This process can be automated by a tool. This would mean a huge saving in terms of man hours used to refine the scheduling.
- An empirical evaluation of schedulability of a given dataset does not guarantee that the deadlines are met in any case.
- Using a rigorous approach to the analysis of the schedulability will improve hence the conformance, of a specific application, to safety guidelines.

PETRI NETS

Proposed method:
A Petri Net is a mathematical representation of a distributed discrete system that describes the structure of a distributed system as a bipartite graph with annotations.

Formally we can define a Petri Net as a tuple $PN = (P, T, F, W, M_0)$ where:

- P *is a finite set of places*;
- T *is a finite set of transition*;
- $F \subseteq (PxT) \cup (TxP)$ *is a set of arches*;
- $W: F \to \mathbb{N}$ *represents the weight of the flow relation* F.
- $M_0: P \to \mathbb{N}$ *is the initial marking vector (initial state of system)*.
- $P \cap T = \emptyset$ *and* $P \cup T \neq \emptyset$.

PETRI NET DIALECTS AND TOOLS

DIALECTS USED:

- **Timed Petri Net:** is a PN extended with time. The model are generated with the tool *TINA*.
- **Coloured Petri Net:** extension of a PN through which it is possible to use data types and complex data manipulation. The model are generated with the tool *CPN tools 4.0*.

- Due to the limited time available to conduct the experiments, in order to satisfy stringent temporal requirements from our industrial partner, we have chosen not to investigate other temporal modeling formalism.

INTERLOCKING

In the following we use as a running example the case of a real signaling application, an interlocking system.

INTERLOCKING SYSTEM:

- Is a safety-critical system that controls the movement of trains in a station and between adjacent stations.
- Monitors the status of the object in the railway yard and allows or denies the routing of trains in accordance with the railway safety and operational regulations.
- This rules are stored in a part of the system named control table that is specific for the station where the system resides.

TASKSET DEFINITION

Taskset definition:

The application is decomposed into a set of tasks $\tau_i : i = 1, \dots n$ and we only consider periodic tasks . The temporal model is an extension of the model of Liu and Layland where each task τ_i is characterized by the following parameters:

- $R_i : first\ release\ time\ of\ \tau_i;$
- $C_i : run\ time\ of\ \tau_i, which\ is\ its\ worst\ case\ execution\ time;$
- $D_i : relative\ deadline\ of\ \tau_i, the\ maximum\ time\ elapsed\ between\ the\ release\ of\ an\ instance\ of\ \tau_i and\ its\ completation;$
- $P_i : release\ period\ of\ \tau_i.$

TASKSET DEFINITION

System's tasks:

- T1 is in charge of operating on the Ethernet channel;
- T2 is one of the most important threads and it is in charge of the safety of the system;
- T3 implements a protocol stack for the receipt and transmission of messages;
- T4 is in charge of copying the value received in the input of the Business Logic and preparing the output for the trasmission.
- T5 is the application thread that contains the logic of the system.
- T6 is a diagnostic thread;
- T7 is a USB driver used for logging data in a key.

Table 1: TaskSet in first epoch

Epoch1	R_i	C_i	D_i
T_1	0	6	6
T_2	6	5	11
T_3	11	16	27
T_2	27	5	32
T_3	32	4	36
T_4	36	24	60
T_5	60	40	100

Table 2: TaskSet in second epoch

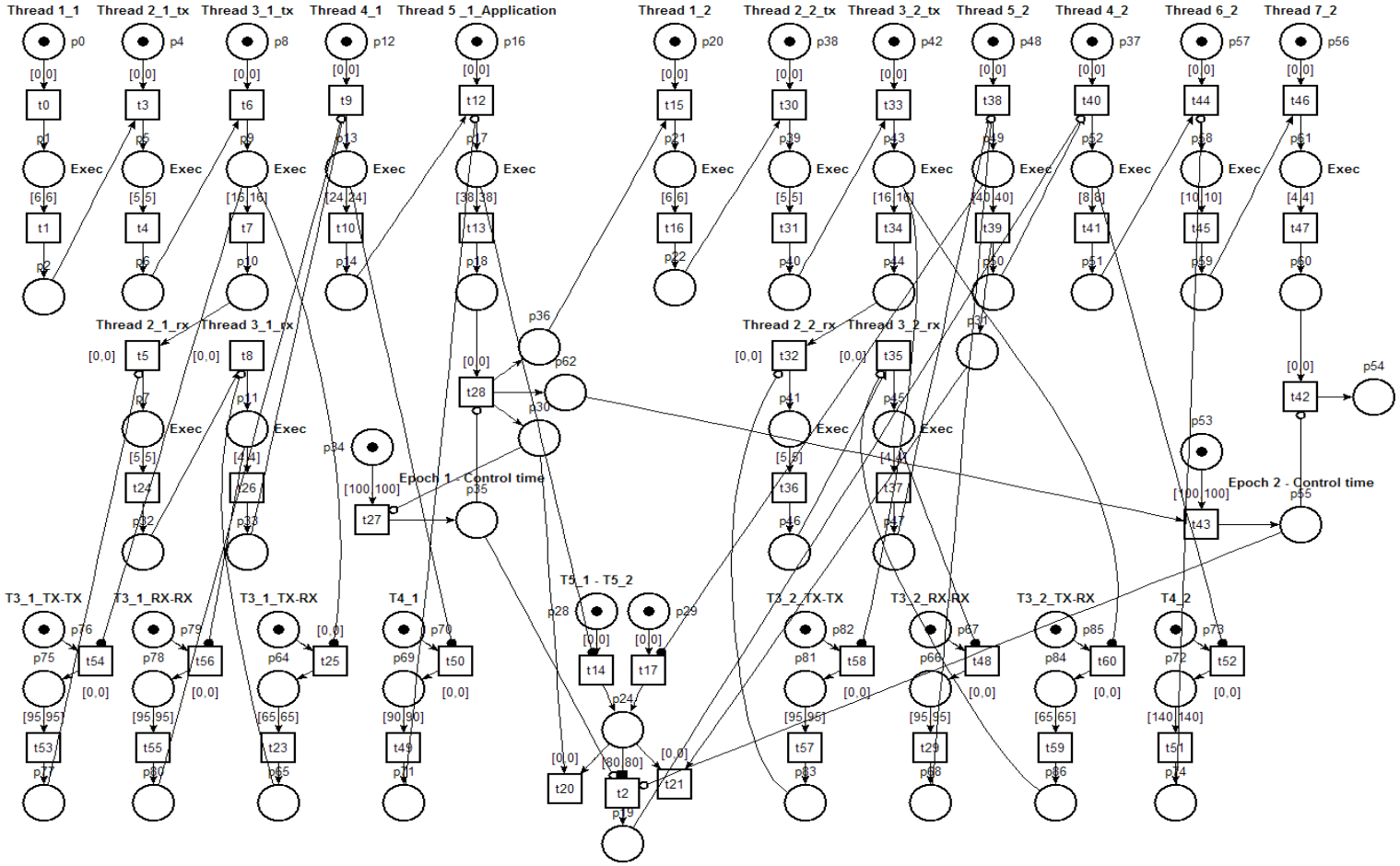
Epoch2	R_i	C_i	D_i
T_1	0	6	6
T_2	6	5	11
T_3	11	16	27
T_2	27	5	32
T_3	32	4	36
T_5	36	40	76
T_4	76	8	84
T_6	84	10	94
T_7	94	6	100

TASKSET CONSTRAINS

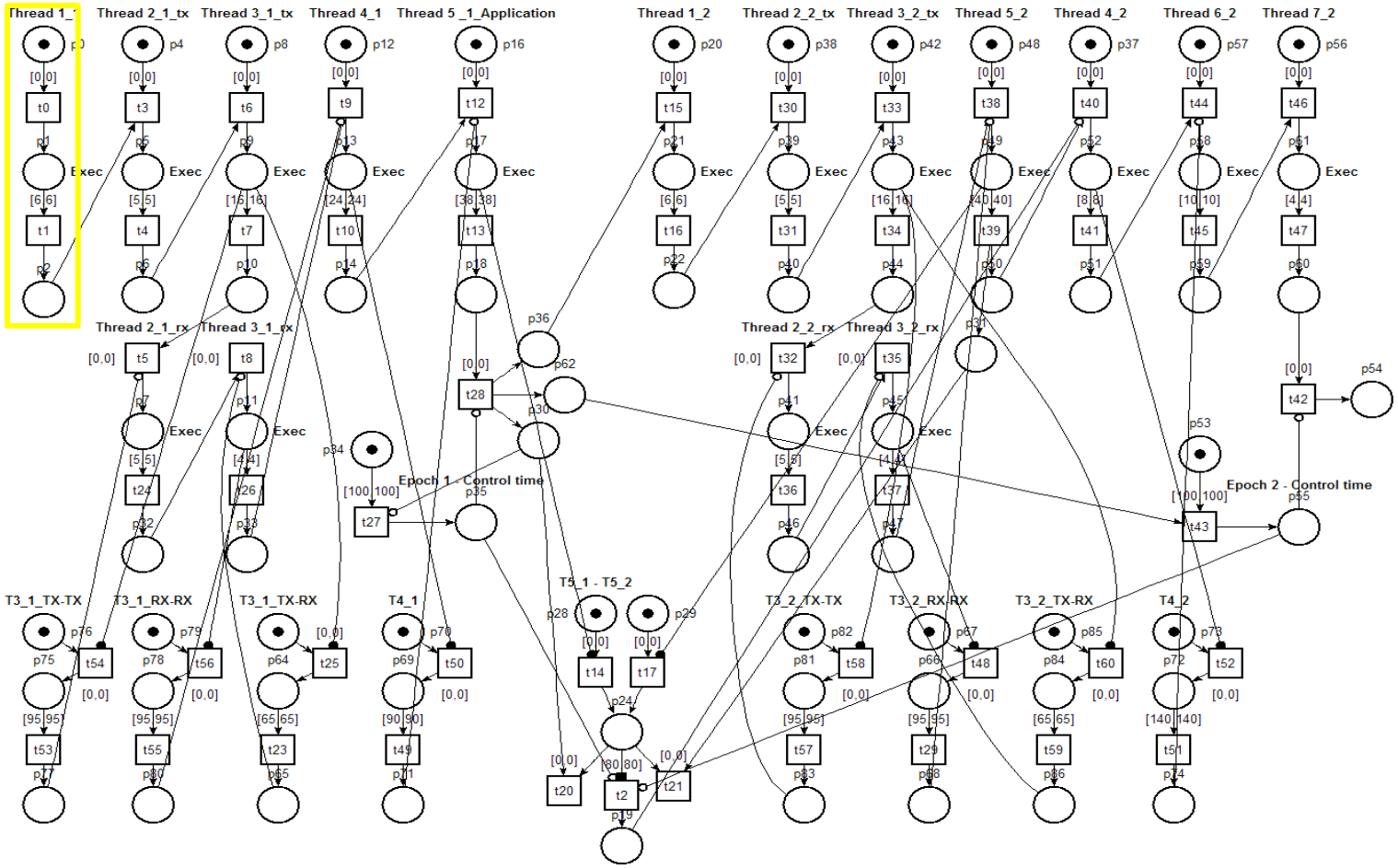
Taskset constrains:

- The total time of scheduling cycle is 200 milliseconds.
- Each epoch needs to last exactly 100 milliseconds.
- The first execution of T3 in the first and second epoch must terminate within 95 milliseconds.
- The second execution of T3 in the first and second epoch must terminate within 95 milliseconds.
- The second execution of T3 in the first and second epoch must execute at least 65 milliseconds after the first one.
- T4 in the first epoch must terminate within 90 milliseconds and in the second epoch in 140 milliseconds.
- The sum of times for T5 must be of at least 90 milliseconds.

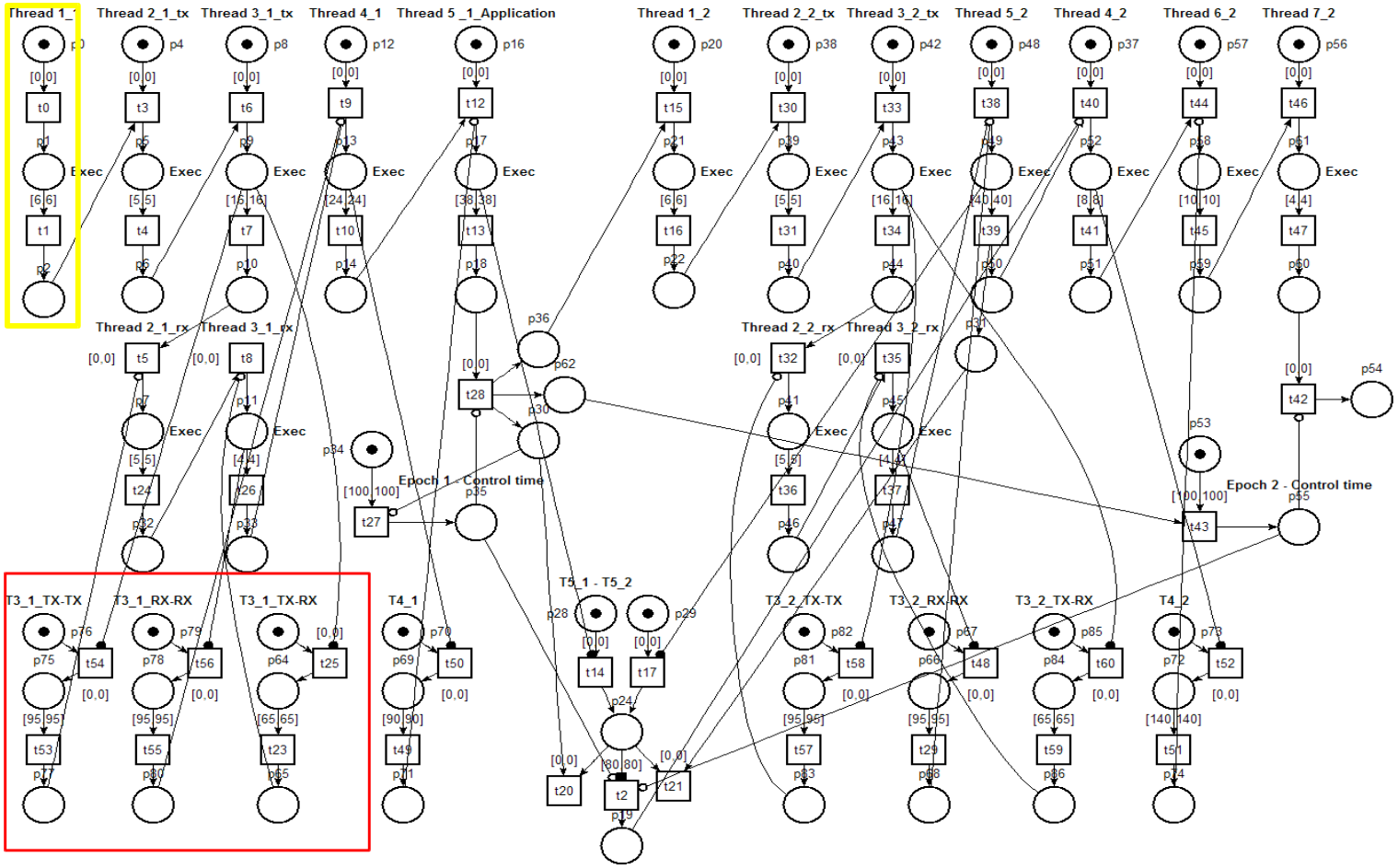
TPN MODELING



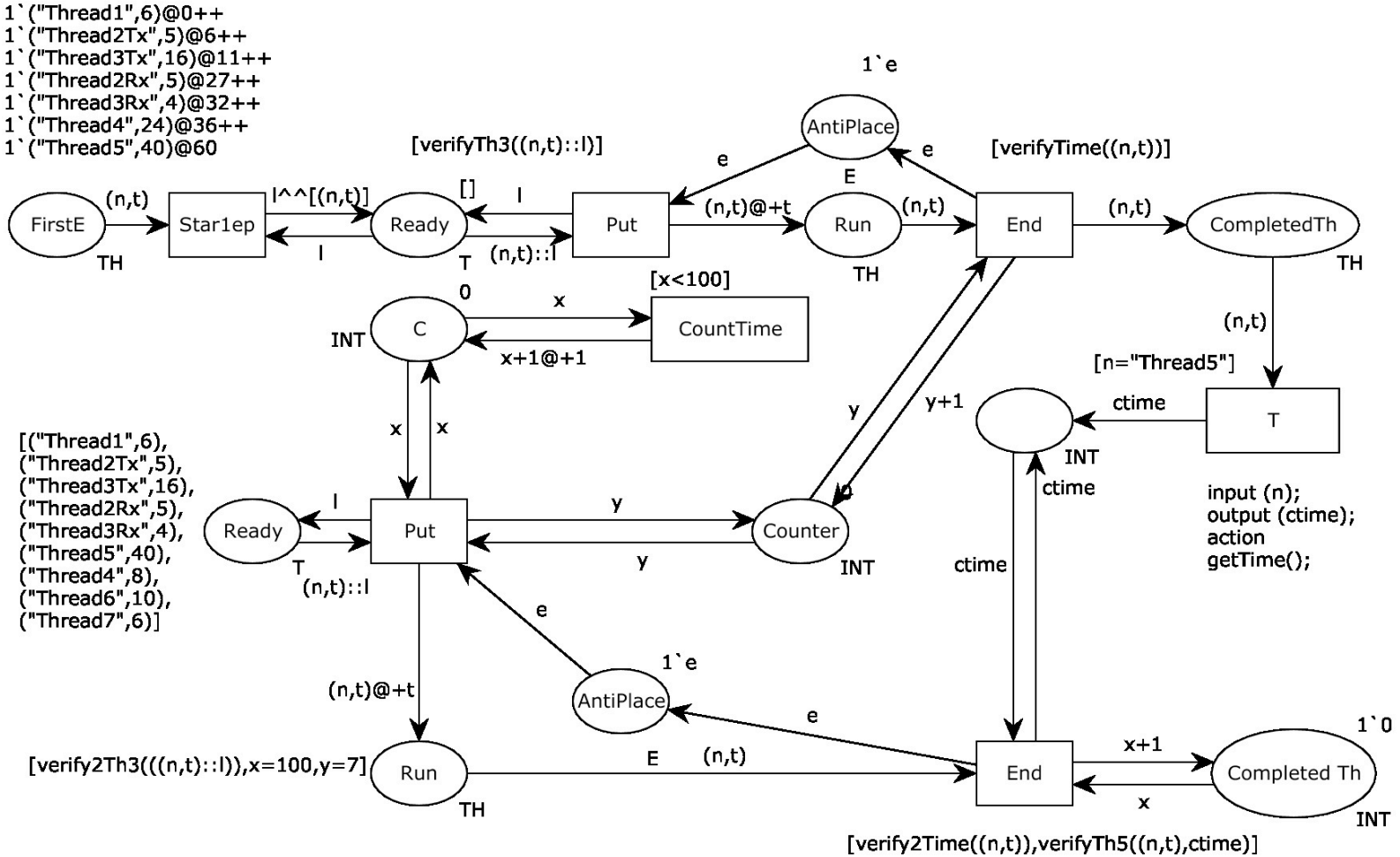
TPN MODELING



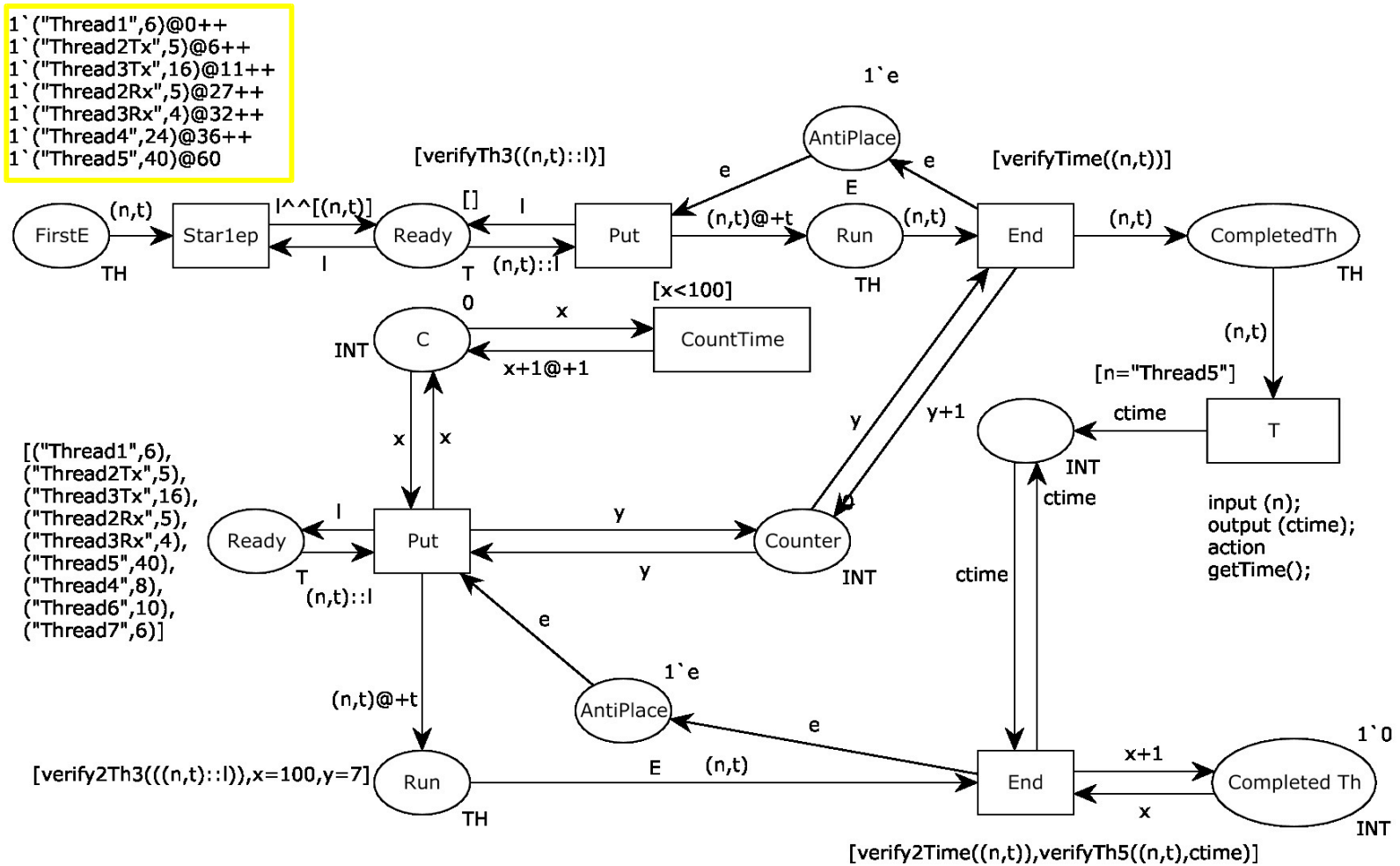
TPN MODELING



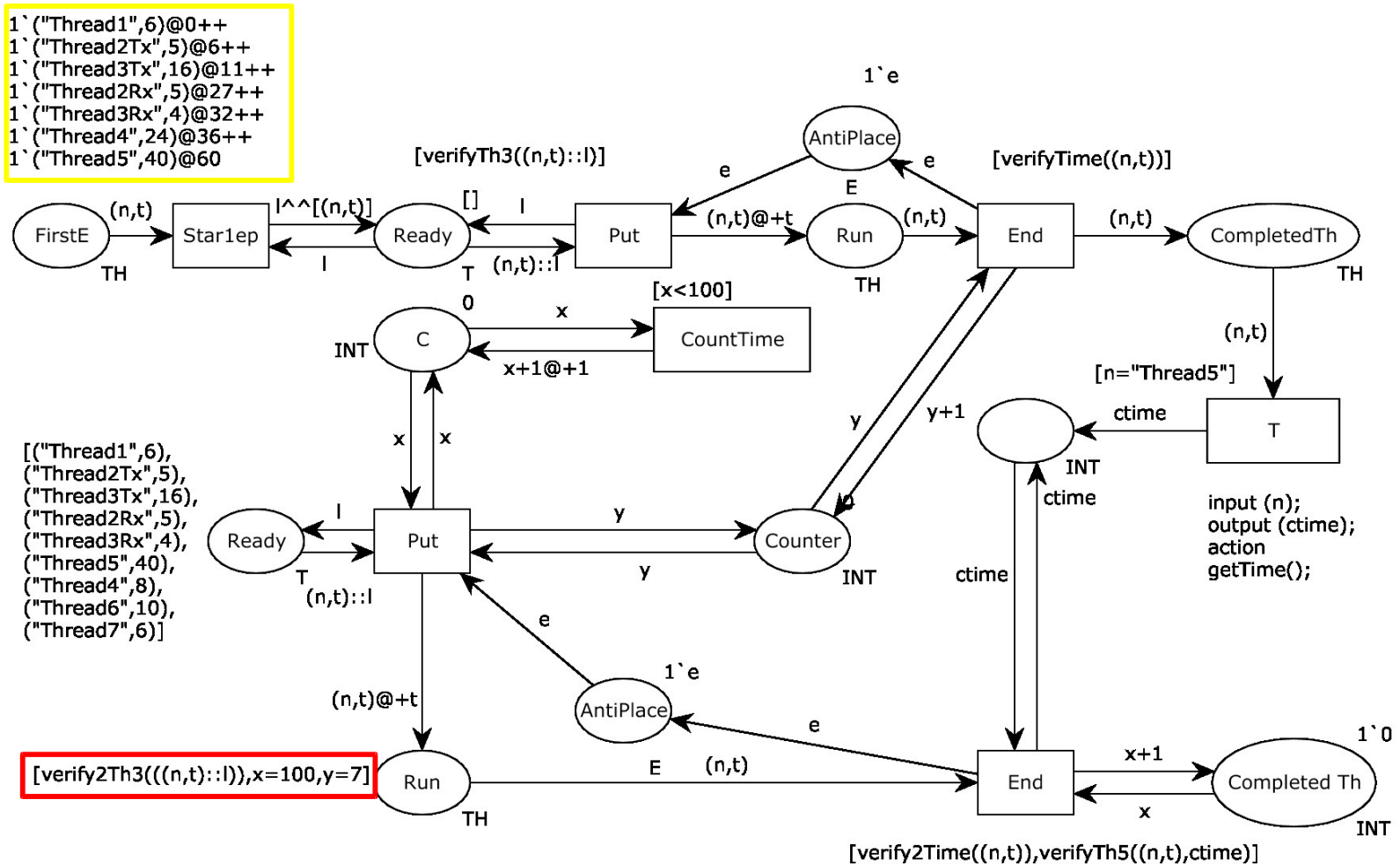
CPN MODELING



CPN MODELING



CPN MODELING



TIMED PETRI NET

TPN modelling cons

- The model is difficult to read (addition of further tasks would result in a huge increase of the net).
- For each thread it is necessary to model a constraint through places and transitions.
- It is not possible to express an attribute that differentiates the identity of a token.
- It is not possible to create aggregate objects: a FIFO queue, for example, can be realized only through checks by inhibitors arcs with a number of places that depends on how many threads should be modeled.

TPN modelling pros

- Control of time during simulation allows to easily understand the global state of the system. The time is increased at any time by a time unit.

COLOURED PETRI NET

CPN modelling pros

- The time constraints can be grouped in auxiliary functions, thus simplifying size and readability of the model.
- Can represent a queue using a single place that contains a token of type list.
- Can represent a large number of tasks by simply adding tokens to the initial marking, leaving the structure of the model unaffected.

CPN modelling cons

- If there are no transitions enabled at the current time, the simulated time count is increased in one step up to the time at which at least one transition is activated.

CONCLUSION

- We have applied the two modelling options to different scheduling algorithms and different sets of task as well.
- CPNs have however resulted to be more advantageous in terms of time spent in model design or in changes, mainly for two reasons:
  - constraints can be simply modeled by a guard on the transition, expressed by a function written in pseudo code, which is easier to express;
  - to populate the model with new tasks it is not necessary to draw new graphic elements but just add an entry to the related place;
- We have experienced that the time spent in CPN modeling is at the end more than half that spent in TPN modeling.
- Only the taskset data need to be changed for a new, or modified, specific application: the overall time to analyse a new taskset, summing up the time to produce a model of the schedule of a new specific application, to run a simulation and to analyse the simulation, is about **two hours (TPN**) or**one hour (CPN)** compared with **eight hours** needed by the previously used empirical approach.

CONCLUSION

- Even if some rework is needed due to a negative response, the information returned by the simulation helps understanding where the problem lies.
- The low cost of the simulation based solution has an obvious positive impact on the costs of the process of instantiating a generic application to a new specific application.
- Next step: automating the modelling from the taskset data
- Approach ready to use in the company

- (Not clear future for this activity)
-