6 décembre 2014

INGI 2143
Concurrent Systems: models and analysis

Assignment 3 - A police department

Professeur : Pecheur Charles

Dederichs François - 49080900
Hardy Simon - 65371100

Introduction

This report answers the different questions of the third assignemnet of the INGI2143 course. This assignment models and analyses a police department using Petri Nets. In addition to this report, we submitted 3 TINA files :

1. `assignment3_first.ndr` (first question).
2. `assignment3_no_deadlock.ndr` (fourth question).
3. `assignment3_asynchronous_messages.ndr` (fifth question).

1 Petri Net model

The Petri Net model can be seen on figure 1 and corresponds to the *TINA* file `assignment3_first.ndr`.

The initial marking is 2 clerks, 2 officers, 1 vehicle, 1 inspector and 1 coroner.

2 Structural analysis

We obtain 5 P-invariants and 3 T-invariants. P- and T-invariants are properties of the graph, independant of the initial marking but P-invariants tend to represent a conservation property (no death or cloning in the police department !) while T-invariants are rather related to periodicity properties.

2.1 Place-invariants

1. `Coroners*5 Investigating` = 5

   There is always one coroner in the place `Coroners`, except when he joins the crime scene, in which case he is in the Investigating place with the patrol and one inspector (and then `Coroners` is empty), so the weighted sum above is always 5.

2. `Investigating*12 On the case*20 On the scene*20 Patroling*20 Recording the report*15 Vehicules*60` = 60

   At the start, we have one vehicle in the `Vehicules` place. The vehicle will pass through the 5 other places listed above, but the number of people varies from one place to the other, so the coefficients vary too : they are 5 on `Investigating` (so the coefficient is 12 because $12 * 5 = 60$), 3 on `On the case`, `On the scene` and `Patroling` and 4 on `Recording the report` (the patrol + one clerck).

   Given that there is only one vehicle, the invariant is as simply as just described.

3. `Clercks*12 Recording phone*12 Recording the report*3 Writing report*4` = 24

   There are initially 2 clercks on the `Clercks` place. Each of them can move to the `Recording phone` place independently. They can also record a report (in which case they are 4 with the patrol) or write write a report (in which case they are three with an officer and an inspector). The coefficients 3 and 4 mean respectively that they are 4 and 3 on the place (because $12 = 3 * 4 = 4 * 3$). The total is 24 because there are 2 clercks initially.

4. `Investigating*12 Officers*30 On the case*20 On the scene*20 Patroling*20 Preparing report*15 Recording the report*15 Writing report*10` = 60

   We have 2 officers that are initially in the `Officers` state and can visit the 7 other states listed above. The coefficents are, again, found regarding the number of tokens in those places when the previous transition is fired. For example, when the

transition `join on crime scene` is fired, 5 tokens (including the 2 officers) arrive in the `Investigating` place, so we put a coefficient of 12 so that $5*12 = 60 = 2*30$.

5. `Inspectors*30 Investigating*6 Preparing report*15 Writing report*10` = 30

   At the start, there is one inspector in the `Inspectors` place. If he goes investigating, there are 5 people on the `Investigating` place, then they are two in the `Preparing report` place and three (with one clerck) on the `Writing report` place. 3 of those 4 states are always empty because there is only one inspector, and he is needed to fire any of those 4 transitions, so the property is always respected.

Notice that the 5 invariants correspond then to the 5 different categories of (human) resources : coroner (1st invariant), vehicles (2nd invariant), clercks (3rd one), officers (4th) and inspectors (last one). That is logic because those resources must stay constant.

2.2 Transition-invariants

1. `come back  go on patrol`

   If two officers and one vehicle go on patrol and then come back to the post (or vice-versa), the initial marking is restored.

2. `answer phone crime call go back to office go on patrol join for report join on crime scene report done`

   It's the cycle representing a crime call. Two officers and a vehicle go on patrol, they are called by a clerck, they join an inspector and a coroner on the crime scene (they are then 5), one officer and the coroner go back home with the vehicle while the 2 others (inspector + one officer) join a clerk to write the report, then the 3 are released when the report is done and we come back to the initial marking (i.e. everyone is at the departure doing nothing).

3. `answer phone contact dpt incident call report recorded`

   It's the cycle representing an incident call : a patrol is called by a clerck to handle an incident. Once it's done, the patrol calls back the clerck (`contact dpt`), they record the report and the clerck is released while the patrol returns to patrol (as it was the case in the initial marking).

3 Reachability analysis

The model is bounded (and more precisely, 5-bounded, the worst case happening when a coroner and an inspector join the patrol on the crime scene).

There are 22 states, 38 transitions and 2 deadlock states.

4 Deadlock state

Here is the deadlock trace :

```
Selt version 3.3.0 -- 10/14/14 -- LAAS/CNRS
ktz loaded, 22 states, 38 transitions
0.000s

- output fullproof;
output mode set
0.016s

```

```
- [] -dead;
FALSE
  state 0: Clercks*2 Coroners Inspectors Officers*2 Vehicules
  -{answer phone}->
  state 1: Clercks Coroners Inspectors Officers*2 {Recording phone} Vehicules
  -{answer phone}->
  state 2: Coroners Inspectors Officers*2 {Recording phone}*2 Vehicules
  -{go on patrol}->
  state 3: Coroners Inspectors Patroling*3 {Recording phone}*2
  -{crime call}->
  state 4: Clercks Coroners Inspectors {On the case}*3 {Recording phone}
  -{answer phone}->
  state 5: Coroners Inspectors {On the case}*3 {Recording phone}*2
  -{join on crime scene}->
  state 6: Investigating*5 {Recording phone}*2
  -{go back to office}->
  state 7: L.dead Coroners Officers {Preparing report}*2 {Recording phone}*2
      Vehicules
  -L.deadlock->
  state 8: L.dead Coroners Officers {Preparing report}*2 {Recording phone}*2
      Vehicules
  [accepting all]
0.016s

-
```

assignment3_first–trace_to_deadlock.txt

Let's analyze it. Two clercks answer the phone one after the other. A patrol receives a crime call and one clerck is realesed, but the other is still blocked waiting for a patrol. Then, the second clerck answers the phone again and the two clercks are then blocked waiting for a patrol.

But there is only one patrol, which joins the crime scene and goes back to office, but they then split : one officer is released while the other waits for a clerck to do the report. Given that the two clerks are waiting for a patrol and that there is one of the two officers waiting for a clerck, we are in a state of deadlock.

To avoid any deadlock, we can put a new transition `hang up the phone` to allow a clerck to escape from the `Recording phone` state and come back to its initial state. Thanks to this new transition, clercks are never blocked on the `Recording phone` state ((because if it was the case, they would just `hang up the phone`) and then no deadlock can occur anymore.

You can see a picture of the model without deadlock on figure 2. The transition added is colored in red.

5 Asynchronous messages

The variant of our Petri Net using asynchronous messages can be seen on figure 3 (main changes are colored in red). Incident and crime calls are transmitted using asynchronous messages so that a clerck doesn't have to wait a patrol when it receives a phone call, he can just record it and then come back to its initial state.

As a consequence, the resulting net is not bounded. More precisely, there is one unbounded place (`Recording phone`, newly named `Buffer`) because clercks can answer the phone, put a token in it and then answer a new phone call, and so on indefinitely, even if there is no patrol available to handle them.

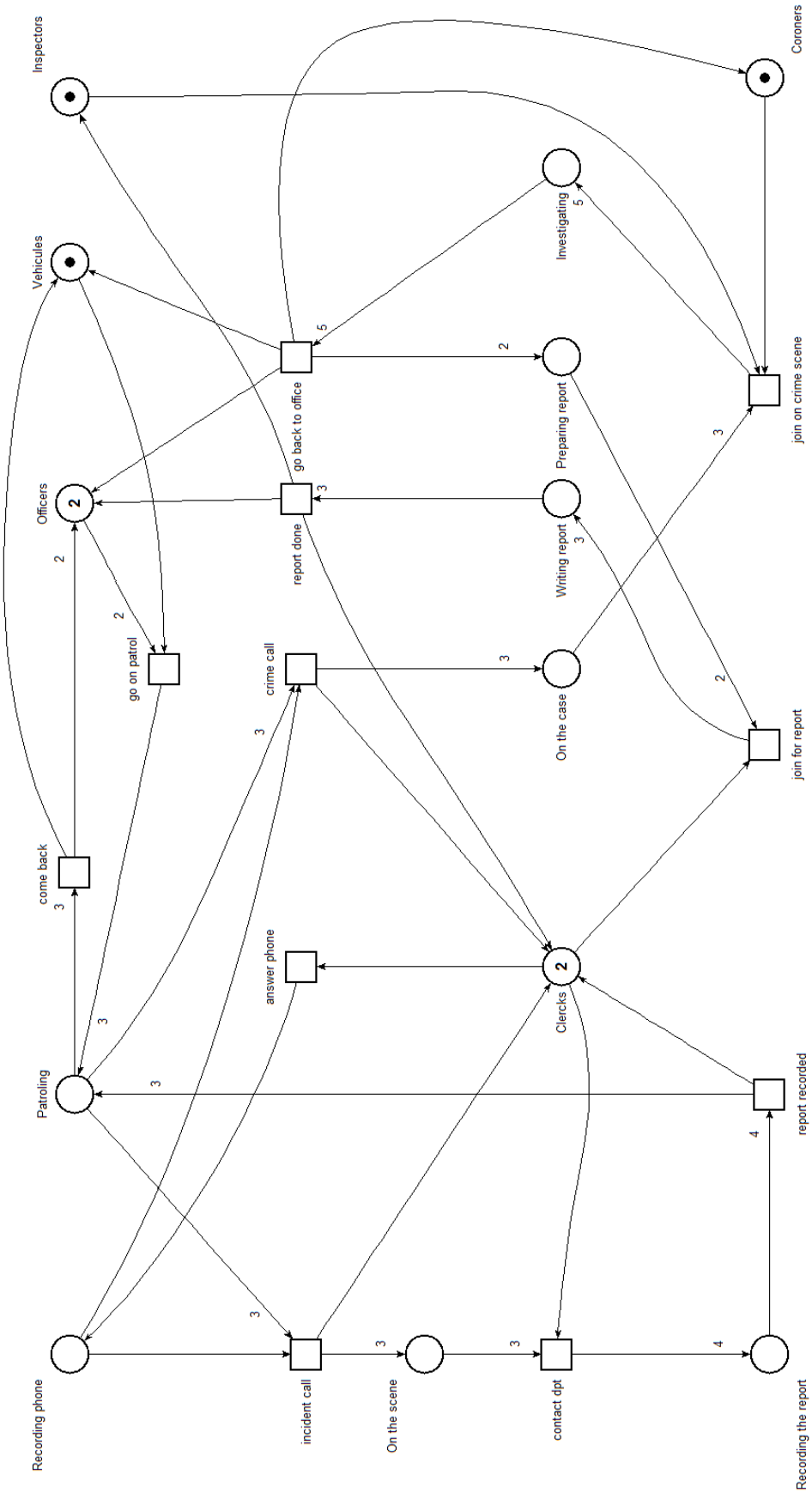

FIGURE 1 – The Petri Net model for the police department

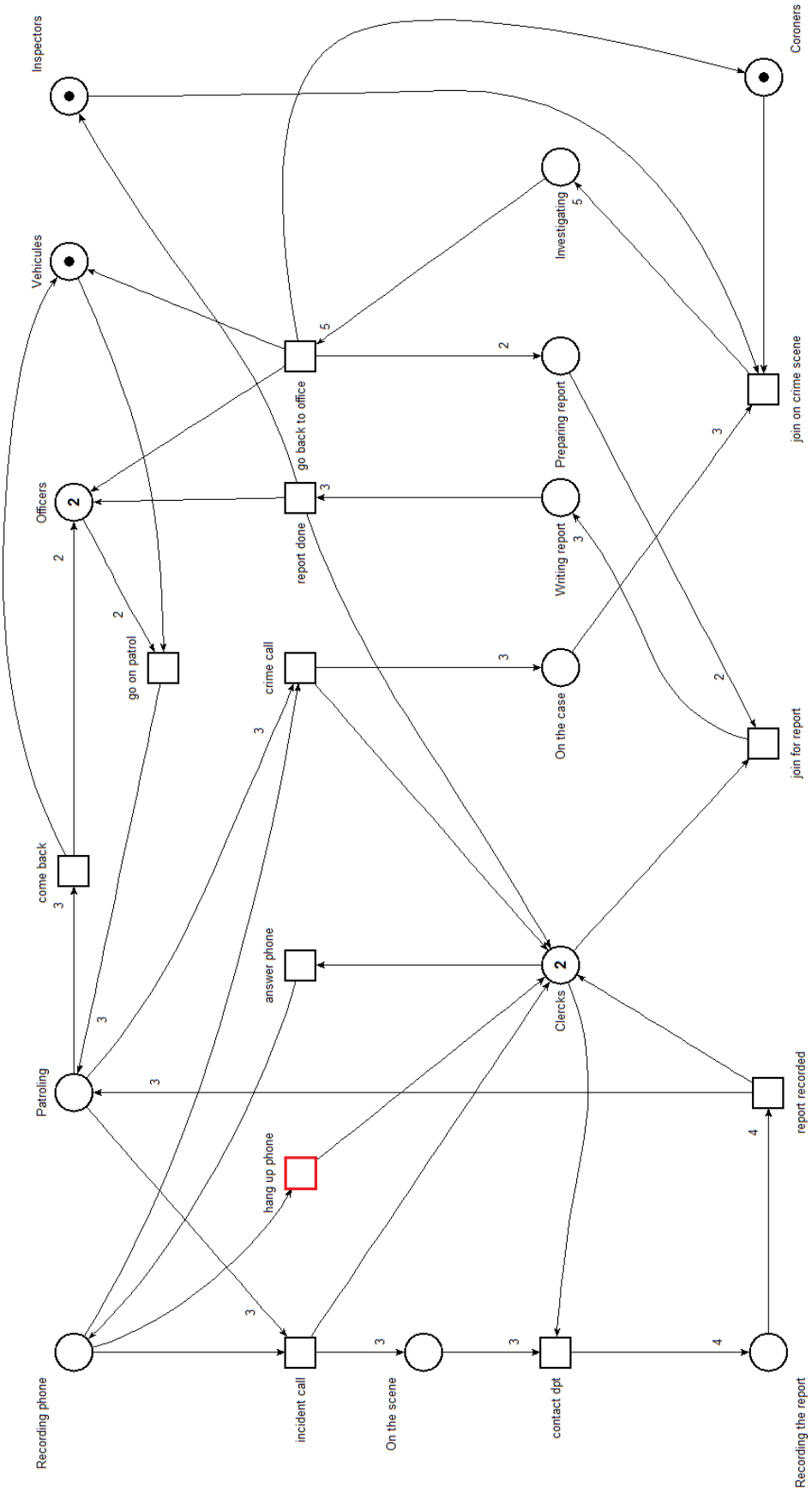

FIGURE 2 – The Petri Net without deadlock using `hang up phone`

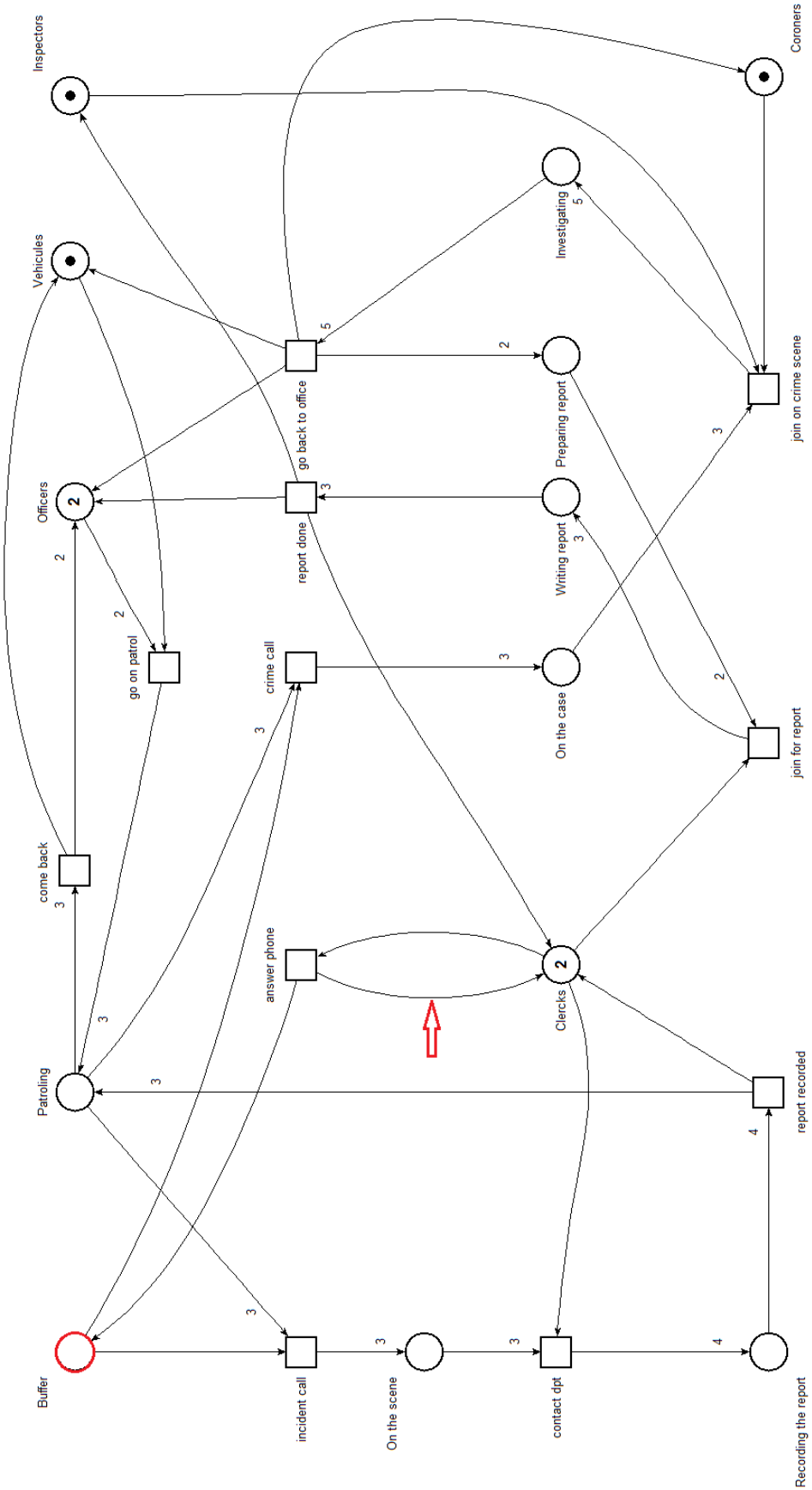

FIGURE 3 – The Petri Net variant with asynchronous messages