Modeling planar signalling in AP axis development in *Xenopus laevis*

using Petri Nets in Higher Level Developmental Biology

Rick van der Zwet, Tiago Borges Coelho
<hvdzwet@liacs.nl>,<borges.coelho@gmail.com>
LIACS - Leiden University, The Netherlands

December 7, 2009

1 Abstract

Planar signaling is the process within the development of the AP axis development of the *Xenopus laevis* [?] in which cells accumulate proteins based on the saturation of nearby cells. If one cell produces n amount of proteins, it will initiate a transferring cascade to cells in the vicinity. This dissemination of proteins will eventually cease, considering that n is a finite variable. There is a gradation in the amount of proteins transferred, meaning that neighbouring cells get n/2 the amount of proteins of the most saturated cell.

We are going to model this into Petri-Nets beeing a mathematical modeling language, which suit well for this purpose as we could nicely model the process in graphical interactive representation and could also be used for automated model tracking and analyze.

2 Approach

First a Petri-Net model will be defined textually and using graphs next the modeling will be taking into practice using the modeling tool*CPNTools* [1].

[1] http://wiki.daimi.au.dk/cpntools/cpntools.wiki

3 Modeling

To model this process we will take a modular approach using coloured Petri-Nets (see Fig 1), since the goal of this assignment is to have a solution that can be applied to any configuration of cells. We start with a building block that is an abstraction of a cell (figure: circle), which can then be coupled to other cells (figure: arrows). The abstraction contains two different types. First the proteins are modelled (figure: red), secondly the proteins (figure: blue) are leading in a second process of the creation of posterisation which also needs modeling. We assume a 1:1 mapping between the amount of proteins and the posterisation -this taken into consideration- ones an `INITIAL` protein is 'used' (e.g. has on posterisation counterpart) in this process it get called `ACTIVATED`. We assume that the proteins to posterisation process is taking place at the same time as the proteins distribution. And in a special format (figure: object B). It tries to matches the posterisation to the same level as the proteins present. But the moment the protein level lowers, the posterisation will remain the same. In pseudo-code:

```
if numPos < numProteins then
    numPos = numPos + 1
endif
```

`numProteins` is the proteins available and `numPos` is the posterisation present.

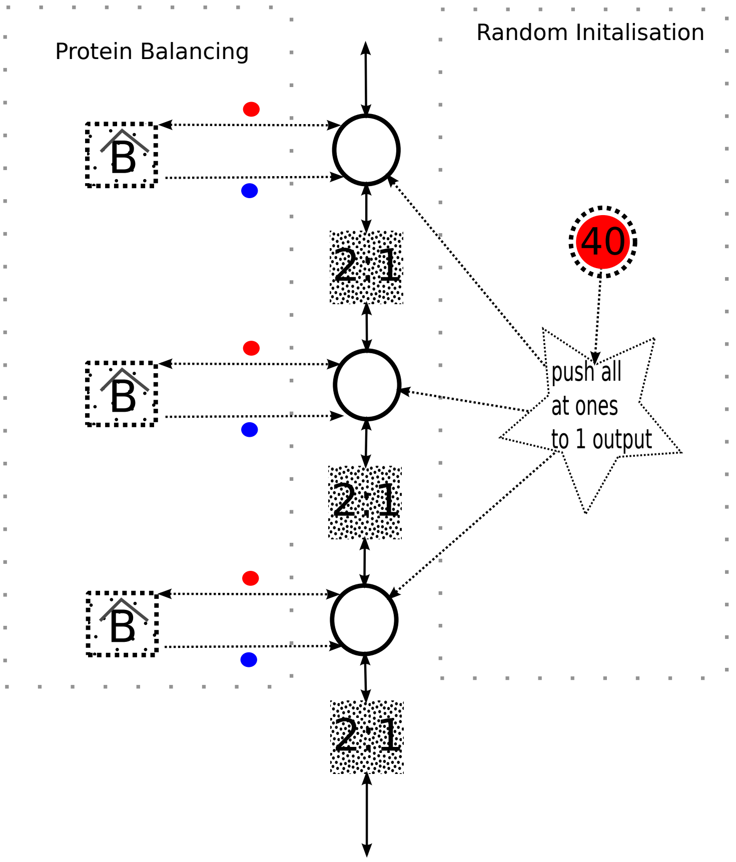
The connectors between the cells (the membranes) has a special properly. One can see them as pressure valves others as siphons (see Fig 2). The moment the 'volume' at complies with the following properly $A/2 < B$ then the pressure closes, else it passes volume from A to B at an certain rate (`flowSpeed`). This rate could depend on the difference, actual value present or something else. Please do mind that negative values could ever appear hence the checking whether the source is bigger or equal then the flowSpeed.

For the case there exists no standard Petri-Net 'component', hence this require the creation of a new property (figure: 2 : 1), with the following properties:

```
flowSpeed = n
if A > 2 * B and A => flowSpeed then
  A = A - flowSpeed
  B = B + flowSpeed
else if B > 2 * A and B => flowSpeed then
  B = B - flowSpeed
  A = A + flowSpeed
endif
```

Planar signaling could theoretically start in every cell, by inserting some amount of proteins. In our model represented as a bunch of INITIAL tokens being put in a random cell.

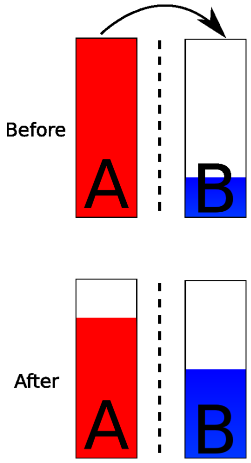
Figure 1: Planar signaling model



4 CPNTools 'implementation'

CPNTools has quite some shortcomings when it comes to modeling (higher level developmental) biology.

Figure 2: Pressure valve example



One it the shortcoming of the 'balancing'. It does not allow reading of how many tokens are present in a certain state and base action upon them. As workaround for this (see Fig 3) we used a 'dump' gradation function. In our case it simply take 3 tokens and pushes 1 forward and converting 2 directly to `ACTIVATED`. This does not take in consideration if the amount get changed in 'further-up', by some external source.

Secondly it is missing a possibility to for easy random initialisation for modeling purposes. As a dirty quirk we 'hacked' it to choose between starting at the head or the tail.

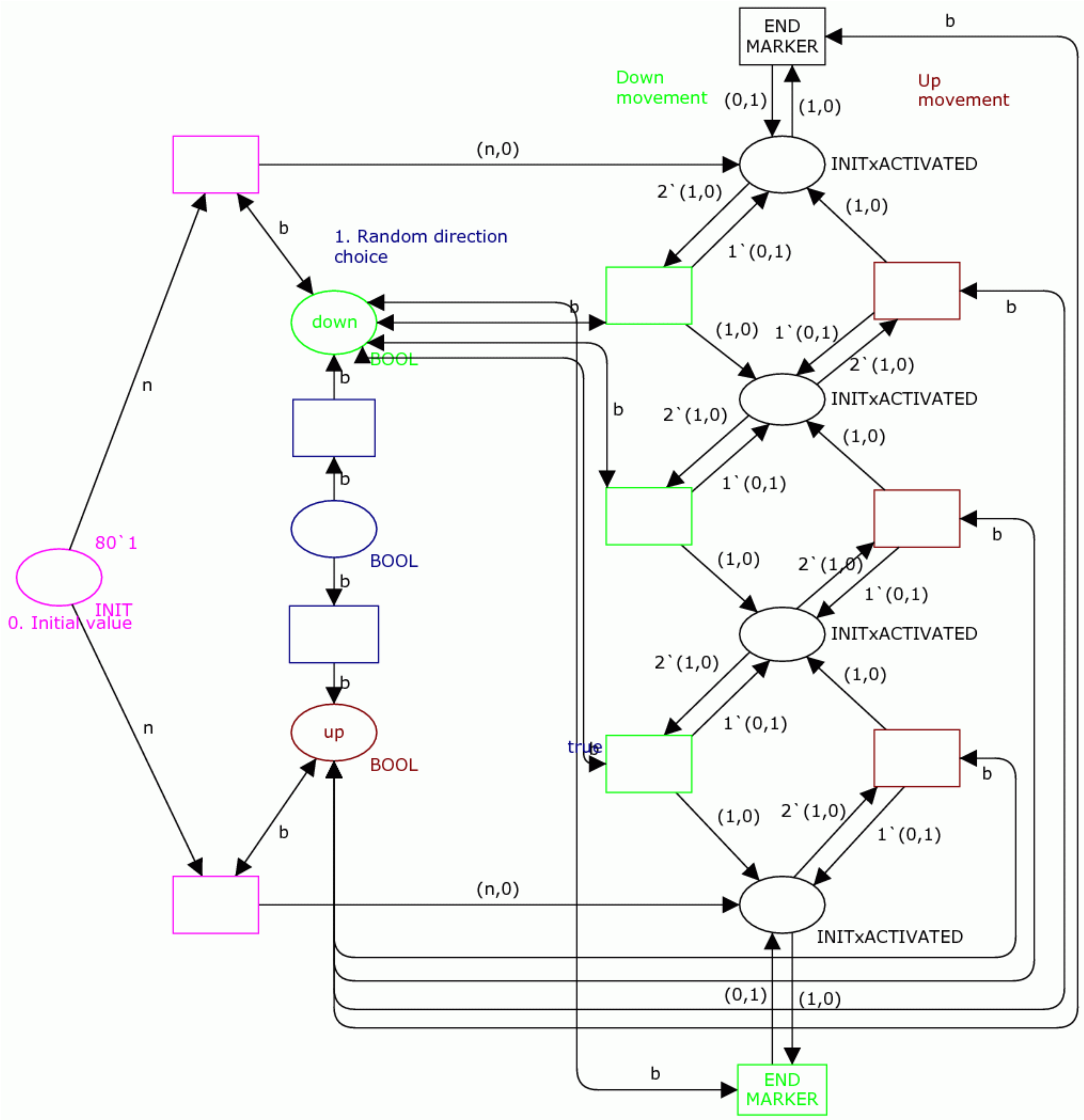
In this implementation the proteins to gradients process is taking place at cell A at the same time that the proteins get transferred from cell A to B.

Also it should be noted that it missing a notion of timed firing sequences; meaning firing sequences which will occur at an certain time. This could for example used to 'trigger' a timed activation of the `INITIAL` to `ACTIVATED` process as modeled in fig 1. An initial idea is shown at fig4 in appendix 1.

5 Conclusion

Using Petri-Nets for modeling biology processes is a powerful framework, which could be well expandable. The Proof Of Concept implementations and visualisations how-ever are lacking. *CPNTools* for example does not provide a powerful enough tool-set for the modeling purposes.

Figure 3: CPNTools implementation

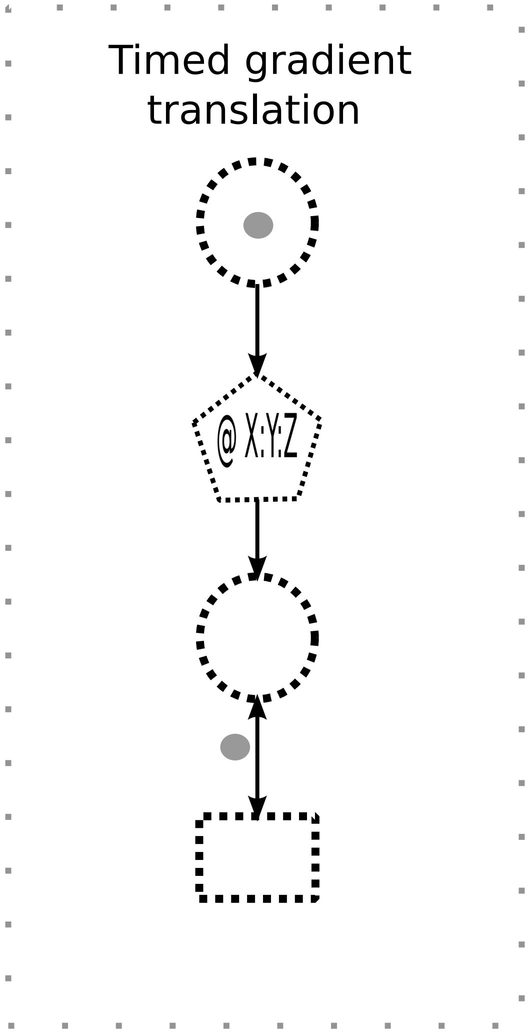


References

[Bertens09]

6 *Appendix

Figure 4: Timed transition idea


Timed gradient
translation
@ X:Y:Z