Agile modelling of cellular signalling

Vincent Danos*

September 19, 2007

Abstract

We illustrate with a simple example how using a rule-based approach to the modelling of cellular signalling allows for quickly putting together models (ease of expression), and quickly modifying them (ease of variation).

1 Introduction

Regev *et al.* have proposed using Milner's π-calculus [19], a minimal language for describing concurrent systems, as a language for modelling biological molecular systems [21, 23, 24]. Variant and sometimes richer languages have been put forward [5, 7, 10, 20, 22]. Rule-based languages such as κ [8, 9] and the closely related BioNetGen language [1–3, 11–13, 17] are specific in that they describe biological interactions by using rules. Both the rule-based and the process-based approaches avoid the combinatorial explosion besetting differential equations. Yet rules have the pratical advantage that they make the building and modification of models easier.

The purpose of this note is to illustrate that modelling flexibility with a small well-understood example, that of the RAS cascade, obtained by refactoring the differential model of Huang and Ferell [18]. We start with a simple example, namely a cycle of enzymatic modification and demodification of a target substrate, to give a feeling of what rules look like. We then give the rules corresponding to the rule-based version of the Huang-Ferell model. (Although this cascade is very simple, it already generates about 600 different species.). Finally we illustrate the ease of changing the model based on a different assumption about some binding mechanism. The variation is chosen in a way that the numerical behaviour of the cascade visibly changes, but this is really subsidiary. The point is that it is simple to modify such models.

2 A simple example

The κ description of a system consists of a collection of *agents* and *rules*. An agent has a certain number of sites, which may have internal states, typically used to denote its phosphorylation status or other post-translational modification, and may also bind to other sites, to represent the formation of biological complexes. Rules provide a concise description of how agents interact. Elementary interactions consist of the binding or unbinding of two agents, the modification of the state of a site, and the deletion or creation of an agent. This seems limited, but closely

*This expository note includes joint work with Jean Krivine (École Polytechnique), Jérôme Feret (École Normale Supérieure), Walter Fontana (Harvard Medical School), and Russell Harmer (CNRS, Université Paris-Diderot).

matches the style of reasoning that molecular biologists apply to mechanistic interactions in cellular signalling.

Suppose three different type of agents: a kinase K, a target T with two phosphorylatable sites x and y, and a phosphatase P. At the rule level, a phosphorylation event can be decomposed in three elementary events: (1) K binds its target T either at site x or y; (2) K may (but need not) phosphorylate the site to which it is bound; (3) K dissociates (unbinds) from T. The rules are shown below; for reference, rules have a name (shown between quotes on the left).

```
'KT@x' K(a),T(x) <-> K(a!1),T(x!1)
'Tp@x' K(a!1),T(x~u!1) -> K(a!1),T(x~p!1)
'KT@y' K(a),T(y) <-> K(a!1),T(y!1)
'Tp@y' K(a!1),T(y~u!1) -> K(a!1),T(y~p!1)
```

Internal states are shown as '`~u`' (unphosphorylated), and '`~p`' (phosphorylated), and physical bonds as '`!`' with shared indices across agents to indicate the two endpoints of a link. The left hand side of a rule specifies a condition to trigger the rule, while the right hand side specifies changes to agents mentioned on the left. A double arrow indicates a reversible rule, the name then refers to the forward version of the rule.

Note that not all sites of an agent need to be present in a rule, eg in the first rule `KT@x`, T does not mention y. Likewise, if a site is mentioned at all, its internal state may be left unspecified, eg in the same first rule one does not say whether site x in T is phosphorylated or not. This is the '*don't care, don't write*' convention: only the information which is conditioning the triggering of a rule needs to be represented. The action of the phosphatase P, which undoes the action of K, is described by a set of entirely similar rules:

```
'PT@x' P(a),T(x) <-> P(a!1),T(x!1)
'Tu@x' P(a!1),T(x~p!1) -> P(a!1),T(x~u!1)
'PT@y' P(a),T(y) <-> P(a!1),T(y!1)
'Tu@y' P(a!1),T(y~p!1) -> P(a!1),T(y~u!1)
```

This simple rule set is a frequent motif that appears in many variants throughout cellular signal transduction [16]. Note that the specification of elementary actions forces us to make our mechanistic choices explicit. The example views phosphorylation of T by K as a 'distributive' mechanism, whereby the kinase must let go of its target in between two modifications, since it cannot remain bound to site x and phosphorylate site y. Other variants of multisite phosphorylation involve a 'processive' mechanism whereby the same kinase acts sequentially on some or all sites of its target.

3 The RAS cascade and a variation

That simple first motive is used three times in a stratified fashion in what is known as the MAPK or RAS cascade. In terms of signal propagation, the cascade involves the primary activator RAS which serves as the incoming signal. Once RAS has activated RAF, RAF plays the role of the kinase for the next tier where it activates MEK, which in turn, when activated, activates ERK. Each tier also has a dedicated phosphatase, in order of appearance PP2A1, PP2A2, and MKP3. Therefore the rules split naturally in three groups all very similar to the rule set considered in the previous section.

To help in the navigation of the rule set one can refer to the rule set *contact map* shown Fig. 1. That is an undirected graph defined from the rule set, where there is one node per agent containing the corresponding agent sites, and any two sites that can be connected by a rule are

connected. In addition sites that are susceptible of a modification are indicated by a colour code (green or grey).

The first group deals with the *activation of RAF by RAS* (which presupposes that RAS is itself active and holds a molecule of GTP):

```
Ras(S1S2~gtp),Raf(x~u) -> Ras(S1S2~gtp!1),Raf(x~u!1)
Ras(S1S2~gtp!1),Raf(x~u!1) -> Ras(S1S2~gtp!1),Raf(x~p!1)
Ras(S1S2~gtp!1),Raf(x!1) -> Ras(S1S2~gtp),Raf(x)
```

and the *inactivation of RAF by PP2A1*:

```
PP2A1(s),Raf(x~p) -> PP2A1(s!1),Raf(x~p!1)
PP2A1(s!1),Raf(x~p!1) -> PP2A1(s!1),Raf(x~u!1)
PP2A1(s!1),Raf(x!1) -> PP2A1(s),Raf(x)
```

The second group deals with the *activation of MEK by activated RAF*:

```
Raf(x~p),MEK(S222~u) -> Raf(x~p!1),MEK(S222~u!1)
Raf(x~p!1),MEK(S222~u!1) -> Raf(x~p!1),MEK(S222~p!1)
Raf(x~p!1),MEK(S222!1) -> Raf(x~p),MEK(S222)
Raf(x~p),MEK(S218~u) -> Raf(x~p!1),MEK(S218~u!1)
Raf(x~p!1),MEK(S218~u!1) -> Raf(x~p!1),MEK(S218~p!1)
Raf(x~p!1),MEK(S218!1) -> Raf(x~p),MEK(S218)
```

and the *inactivation of MEK by PP2A2*:

```
PP2A2(s),MEK(S222~p) -> PP2A2(s!1),MEK(S222~p!1)
PP2A2(s!1),MEK(S222~p!1) -> PP2A2(s!1),MEK(S222~u!1)
PP2A2(s!1),MEK(S222!1) -> PP2A2(s),MEK(S222)
PP2A2(s),MEK(S218~p) -> PP2A2(s!1),MEK(S218~p!1)
PP2A2(s!1),MEK(S218~p!1) -> PP2A2(s!1),MEK(S218~u!1)
PP2A2(s!1),MEK(S218!1) -> PP2A2(s),MEK(S218)
```

The third group has a similar structure and handles the *activation of ERK by activated MEK*:

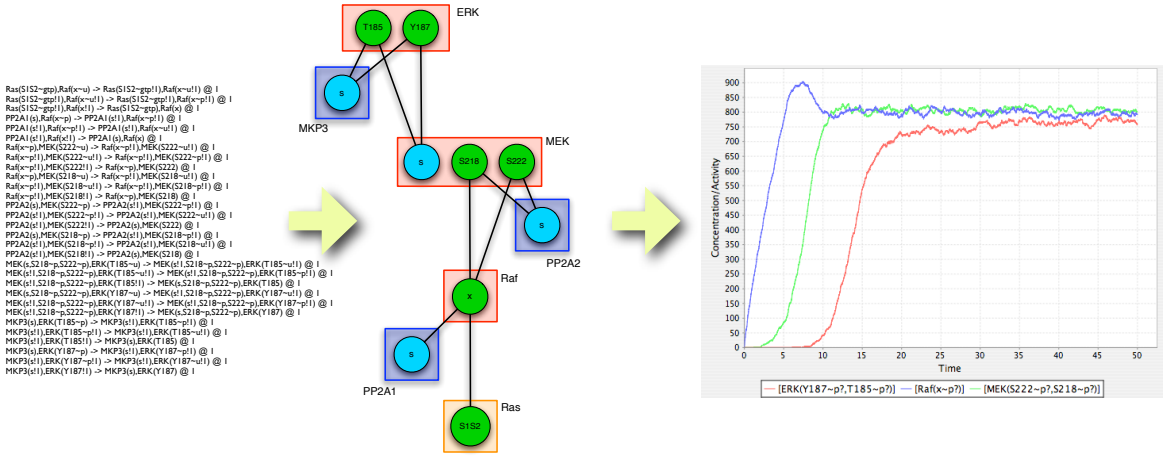
```
MEK(s,S218~p,S222~p),ERK(T185~u) -> MEK(s!1,S218~p,S222~p),ERK(T185~u!1)
MEK(s!1,S218~p,S222~p),ERK(T185~u!1) -> MEK(s!1,S218~p,S222~p),ERK(T185~p!1)
MEK(s!1,S218~p,S222~p),ERK(T185!1) -> MEK(s,S218~p,S222~p),ERK(T185)
MEK(s,S218~p,S222~p),ERK(Y187~u) -> MEK(s!1,S218~p,S222~p),ERK(Y187~u!1)
MEK(s!1,S218~p,S222~p),ERK(Y187~u!1) -> MEK(s!1,S218~p,S222~p),ERK(Y187~p!1)
MEK(s!1,S218~p,S222~p),ERK(Y187!1) -> MEK(s,S218~p,S222~p),ERK(Y187)
```

and the *inactivation of ERK by MKP3*:

```
MKP3(s),ERK(T185~p) -> MKP3(s!1),ERK(T185~p!1)
MKP3(s!1),ERK(T185~p!1) -> MKP3(s!1),ERK(T185~u!1)
MKP3(s!1),ERK(T185!1) -> MKP3(s),ERK(T185)
MKP3(s),ERK(Y187~p) -> MKP3(s!1),ERK(Y187~p!1)
MKP3(s!1),ERK(Y187~p!1) -> MKP3(s!1),ERK(Y187~u!1)
MKP3(s!1),ERK(Y187!1) -> MKP3(s),ERK(Y187)
```

The MEK activation group above has 6 rules (2 bindings, 2 modifications, and 2 unbindings) which specify the manner in which RAF activates MEK. Specifically, one sees that RAF can

Figure 1: *The contact map of the canonical RAS cascade (middle); the time evolution of the level of the fully phosphorylated forms of RAF, MEK, and ERK is shown (right), with with initial state 300 active RAS, a 1000 of each kinase RAF, MEK, and ERK, all unphosphorylated, and 200 of each phosphatase PP2A1, PP2A2, and MKP3.*



bind any of MEK's serine residues S222 and S218 in no particular order (see also contact map, Fig. 1), in the same fashion as our first example.

Given a rate for each rule, a rule set has a natural associated semantics as a continuous time Markov chain. A state x of the Markov chain is a set of agents with various internal states and mutual bindings. The likelihood that a given rule applies is proportional to its total number of instances in x multiplied by its rate, a quantitity known as the *rule activity*, and the associated time advance is given by an exponential distribution with average frequency the total activity of x (ie the sum of the rules's activities). In the particular case where there is no bindings or internal states, ie in the case of flat chemical reactions (aka multiset rewriting, or Petri nets), the above semantics is known as 'Gillespie's method' [4, 14, 15], and is widely used to simulate the kinetics of coupled elementary chemical reactions.

In particular, by assigning a unit rate to all rules, we get a Markov chain. Taking as an initial state the one described in the caption of Fig. 1, one obtains simulations which show the canonical cascade behaviour where each of the successive tiers rise in turn with a slight delay, and little of the incoming signal can activate (ie doubly phosphorylate) ERK almost entirely at steady state (Fig. 1).

3.1 A variation

Now suppose for the sake of the illustration that recent experimental data suggests that in fact RAF binds to MEK at the same site s to which MEK also binds (See the rewiring of RAF to MEK in the variant contact map, Fig. 2). To modify the original model and obtain a model of the new situation, all one has to do is replace the six rules describing the interactions of RAF and MEK (shown above) with the four rules below.

```
Raf(x~p),MEK(s) -> Raf(x~p!1),MEK(s!1)
Raf(x~p!1),MEK(s!1) -> Raf(x~p),MEK(s)
Raf(x~p!1),MEK(S222~u,s!1) -> Raf(x~p!1),MEK(S222~p,s!1)
Raf(x~p!1),MEK(S218~u,s!1) -> Raf(x~p!1),MEK(S218~p,s!1)
```

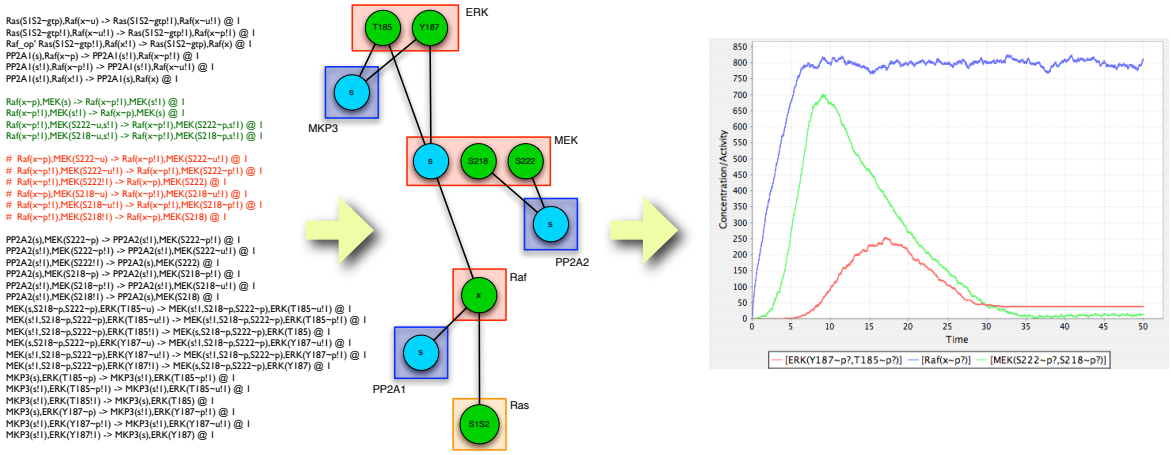


Figure 2: *A variation on the theme: RAF competes now with ERK to bind at site s as expressed by the 4 new rules replacing the 6 older ones. As a result ERK is no longer activated.*

As a result of the modification, the cascade no longer activates ERK in a sustained way (Fig. 2). The likely reason for that behaviour seems to be that the competition between RAF and ERK on MEK's site s is enough for the phosphatases to take over.[1]

So how does a rule-based approach compares in terms of flexibility. Note that the number of generated species in the original model is about 600. Therefore a differential model, were it to take all combinations into account would be already quite large, because such models can only manipulate flat reactions between structure-less entities. So a variation such as the one we performed would probably be very tedious, and in fact rapidly impossible for larger examples. If on the other hand the model was not considering all possibilities, then it could be even harder to modify, since whether what was significant before the variation still is after is not known, and one would have to decompress the model before performing the modification. Process-algebra based notations would not be hit by the same combinatorial explosion, after all they are little distributed programming languages in themselves, and can cope well with contextual interactions. However, by design, they record the behaviour of each agent in the agent itself (the reason why they are called process-centric) and it seems that would make the kind of variation delineated above less of an easy operation. The gain in flexibility is likely to be more pronounced with larger models. It is also likely to be more useful for modelling situations where one knows less and there is greater uncertainty about which specific mechanisms are in place. The interested reader can find a larger example together with further considerations on rule-based modelling in Ref. [6].

[1] In fact the real reason is subtler and has to do with the two specific rules we have chosen to describe how MEK detaches from ERK. Both are asking, perhaps unduly, MEK to be doubly phosphorylated. So if PP2A2 dephosphorylates any of the two MEK sites while MEK is attached to an ERK, MEK and ERK will no longer be able to detach! That is true in both the original model and the variant. However in the original model, it is always possible for PP2A2 to detach, RAF to take its place, and then rephosphorylate its substrate MEK, so that MEK can eventually detach. Not so in the variant where RAF cannot access a MEK attached to an ERK since MEK uses the same site s to bind downstream ERK and upstream RAS. This shows how subtle the behaviour induced by a rule set can be.

References

[1] R. Milner, *Communicating and mobile systems: the π-calculus*, Cambridge University Press, Cambridge, 1999.

[2] A. Regev, W. Silverman, and E. Shapiro, "Representation and Simulation of Biochemical Processes Using the π-Calculus Process Algebra," in *Pacific Symposium on Biocomputing*, edited by R. B. Altman, A. K. Dunker, L. Hunter, and T. E. Klein, World Scientific Press, Singapore, 2001, vol. 6, pp. 459–470.

[3] C. Priami, A. Regev, E. Shapiro, and W. Silverman, *Information Processing Letters* (2001).

[4] A. Regev, and E. Shapiro, *Nature* **419** (2002).

[5] S. Eker, M. Knapp, K. Laderoute, P. Lincoln, J. Meseguer, and K. Sonmez, "Pathway Logic: Symbolic Analysis of Biological Signaling," in *Proceedings of the Pacific Symposium on Biocomputing*, 2002, pp. 400–412.

[6] A. Regev, E. Panina, W. Silverman, L. Cardelli, and E. Shapiro, *Theoretical Computer Science* **325**, 141–167 (2004).

[7] L. Cardelli, "Brane Calculi," in *Proceedings of BIO-CONCUR'03, Marseille, France*, Elsevier, 2003, vol. 180 of *Electronic Notes in Theoretical Computer Science*.

[8] C. Priami, and P. Quaglia, *Proceedings of CMSB* **3082**, 20–33 (2004).

[9] V. Danos, and J. Krivine, "Formal Molecular Biology done in CCS," in *Proceedings of BIO-CONCUR'03, Marseille, France*, Elsevier, 2003, vol. 180 of *Electronic Notes in Theoretical Computer Science*, pp. 31–49.

[10] V. Danos, and C. Laneve, *Theoretical Computer Science* **325**, 69–110 (2004).

[11] V. Danos, and C. Laneve, "Core Formal Molecular Biology," in *Proceedings of the 12th European Symposium on Programming, ESOP'03*, Springer-Verlag, 2003, vol. 2618 of *LNCS*, pp. 302–318.

[12] J. Faeder, M. Blinov, G. B., and W. Hlavacek, *Complexity* **10**, 22–41 (2005).

[13] M. Blinov, J. Yang, J. Faeder, and W. Hlavacek, *Proc. BioCONCUR 2005* (2005).

[14] J. Faeder, M. Blinov, and W. Hlavacek, *Proc. ACM Symp. Appl. Computing* pp. 133–140 (2005).

[15] J. R. Faeder, M. L. Blinov, B. Goldstein, and W. S. Hlavacek, *Systems Biology* **2**, 5–15 (2005).

[16] M. L. Blinov, J. Yang, J. R. Faeder, and W. S. Hlavacek, *Nat. Biotechnol.* **24**, 1–2 (2006).

[17] M. L. Blinov, J. R. Faeder, B. Goldstein, and W. S. Hlavacek, *BioSystems* **83**, 136–151 (2006).

[18] W. Hlavacek, J. Faeder, M. Blinov, R. Posner, M. Hucka, and W. Fontana, *Science's STKE* **2006** (2006).

[19] C. Huang, and J. Ferrell, *Proceedings of the National Academy of Sciences* **93**, 10078–10083 (1996).

[20] A. Goldbeter, and D. Koshland, *Proceedings of the National Academy of Sciences* **78**, 6840–6844 (1981).

[21] A. B. Bortz, M. H. Kalos, and J. L. Lebowitz, *J. Comp. Phys.* **17**, 10–18 (1975).

[22] D. T. Gillespie, *J. Phys. Chem* **81**, 2340–2361 (1977).

[23] D. T. Gillespie, *J. Comp. Phys.* **22**, 403–434 (1976).

[24] V. Danos, J. Feret, W. Fontana, R. Harmer, and J. Krivine, "Rule-based modelling of cellular signalling," in *Proceedings of the 18^{th} International Conference on Concurrency Theory (CONCUR'07)*, edited by L. Caires, and V. Vasconcelos, 2007, Lecture Notes in Computer Science.