

# Process Modeling and Rendering of Biochemical Structures: Actin

Ozan Kahramanoğulları, Andrew Phillips and Federico Vaggi

**Abstract** We propose stochastic process models as a means for studying and rendering unbounded biological structures, involving mechanisms that extend over geometric space. As an example, we discuss a case study of actin polymerization dynamics, which plays a key role in many cellular activities and enjoys a rich structure. We provide a comparative review of various approaches in the literature for modeling actin. We then illustrate on actin models how otherwise challenging structures can be modeled. In these models the complexity of the structures are incrementally increased with respect to the biological data. We present a geometric representation of these models that we use to generate movies reflecting their dynamics while preserving formal cleanliness as well as loyalty to the biological data.

## 1 Introduction

Modeling of biochemical systems with formal methods is now well established. Various disciplines make available a rich ensemble of techniques and tools for addressing biochemical phenomena at different levels, giving rise to otherwise unavailable advantages. While providing the means to conceptualize the biological knowledge by rigorously spelling out the mechanisms of these systems, formal models enable researchers to study various aspects of the considered systems by means of simulations and analyses.

Despite the broad usage of models for studying biological systems, it is still a challenge to draw connections in models between the biochemical phenomena that

---

O. Kahramanoğulları (✉) · F. Vaggi

The Microsoft Research—University of Trento Centre for Computational and Systems Biology, Pizza Manifattura 1, 38068 Rovereto, Italy  
e-mail: ozan@cosbi.eu

A. Phillips

Microsoft Research Cambridge, Cambridge, UK

describe the functioning of a certain system, and its spatial evolution with respect to its behavior. Common approaches towards addressing this challenge often rely on compromises, which either depart from the underlying formal modeling framework to capture the spatial aspects, or make simplifying assumptions about the modeled systems, implying a departure from biological data. Process models propose a partial solution to this challenge, thereby lifting some of the constraints that hinder the above mentioned approaches.

Process algebras are languages, which have originally been designed to formally describe complex reactive computer systems. In these languages, typically, each component of a system is described separately together with its interaction capabilities with other components, that is, the model of a system contains a description of all of its components with respect to their actions and interactions, and how they evolve after performing them. Being defined on formally sound and rigorous grounds, such a setting makes it possible to give an abstract view of a system to study its properties of interest, while fading out others.

Due to the resemblance between computer systems and the biological systems, the consideration above can be carried over to model and study biological systems in a similar fashion [37]. This way, interactions and actions of processes find interpretations as biological phenomena such as association of biochemical species or their transformations. By exploiting the algebraic operators, different components can be put together to build increasingly complex systems and each sub-system model can be altered locally without modifying other components. This constitutes the compositionality as a distinguishing feature.

We give an account of process models for modeling biochemical structures and their graphical rendering, which is challenging by other means. Extending previous work [4], we illustrate these ideas on various examples, and use actin dynamics as a case study, where monomers with complex structures polymerize to form branching filaments. Actin molecules display a rich structure and geometric dynamics in space, which is instrumental in illustrating these ideas. Moreover, actin dynamics, be it the polymerization of actin monomers into filaments or the reverse process, plays a key role in many cellular activities, in particular in those that involve a modification in cell shape such as phagocytosis and cell motility, meshworks of actin filaments form one of the three major cytoskeletal networks in eukaryotic cells.

For the example models we use the **SPiM** language (stochastic pi-calculus machine) [30], which is an implementation of stochastic pi-calculus, the stochastic extension of a broadly studied process algebra. **SPiM** is equipped with a stochastic simulation engine, based on Gillespie algorithm [16]. We propose compositional process models of actin dynamics, which incrementally reflect different levels of complexity in the biochemical mechanisms with respect to the capabilities of actin monomers. In addition to the work in [4], we provide a review of actin dynamics and of other approaches in the literature for modeling actin dynamics. We demonstrate how filaments built from monomers can be modeled compositionally as processes, also when monomers have different states or binding capabilities. We then introduce an extension to **SPiM** language, which makes it possible to obtain simulation traces. By using this extension, we illustrate how **SPiM** language features can be

used to encode geometric data in terms of growth vectors and affine transformations for graphical rendering of the model dynamics to obtain movies. We then extend the two dimensional model in [4] with a three dimensional one to capture not only the branching structure of the filaments, but also the helical shape of the filaments. We use these features to generate movies, reflecting the simulation dynamics without any need for making simplifications to the model or the encoded stochastic dynamics.

This novel form of computational modeling lays the foundation for observing the behavior of the modeled systems in the geometric space. Because process models provide the foundations of the programming language based algorithmic approach to modeling in biology [35], these ideas can be carried over to a family of languages with an algorithmic approach [36].

## 2 Biological Mechanisms as Processes

In process algebra, the basic building blocks of models are processes. A system state can be perceived as a collection of processes that have precisely defined action capabilities: each process in a model is defined as a description of its actions. In this respect, components of the modeled systems are denoted by processes that describe the behavior of each component in terms of the actions it can perform. When a process performs one of these actions, the system evolves to a state, which is also defined in the model. This provides a means to model the computations of a system as the evolution of the system model as it moves between these states. Different process algebra languages provide language constructs that allow modelers to denote various phenomena with respect to the targeted modeling domain.

The stochastic  $\pi$  calculus machine, that is **SPiM**, is an implementation of a stochastic extension of the stochastic  $\pi$  calculus. In the setting of **SPiM**, the actions that result in a state change are either *delay*, or the complementary *input* and *output* actions. **SPiM** actions are stochastic, and its simulation engine implements the Gillespie algorithm, giving rise to a continuous time Markov chain semantics. Given a **SPiM** model, at every simulation step, the simulation engine chooses an action with respect to this semantics from the set of available actions at that state. While a delay action implements a stochastic delay, the complementary input and output actions allow processes to synchronize and exchange information by being executed concurrently with respect to the underlying stochastic semantics. For the formal definitions of **SPiM**, we refer to [5, 30]. Let us now illustrate these ideas on an example which implements the reactions



such that **C** records the number of times it has been bound to **B** and **C** throughout the simulation.

```

directive sample 1.0
directive plot A(); B(); C()
directive debug

new a@1.0:chan(chan)
new b@1.0:chan(chan)

let C(m:int,n:int) =
  do ?a(x); Cb(x,m+1,n)
  or ?b(x); Cb(x,m,n+1)
  or delay@1.0; A() )

and Bb(e:chan) = ?e; B()

and Cb(x:chan,m:int,n:int) =
  !x; C(m,n)

let A() = ( new e@1.0: chan()
  do !a(e); Ab(e)
  or delay@1.0; B() )

and Ab(e:chan) = ?e; A()

and B() = ( new e@1.0: chan()
  do !b(e); Bb(e)
  run 10 of (A() | B() | C(0,0))

```

In the SPiM implementation of this model, each state of each species is described by a process. Processes A, B and C denote the free species, and Ab, Bb and Cb denote their bound forms. In their unbound forms, the processes for the species A and B do not have any parameters, however each process for C is equipped with two integer variables, n and m, that count the number of times it has been bound to A and B, respectively, throughout the simulation. The processes Ab, Bb and Cb representing the bound species are parameterized by a *private channel name* that represents the bond between these processes and their binding partners.

The process C can perform a stochastic choice, denoted with `do ... or`, between various actions. In this case, these are either interacting with A by performing an input action on channel a, denoted with `?a`, or interacting with B on channel b. Process A can perform a stochastic choice between a delay action or an output action on channel a, denoted with `!a`, that complements the input action performed by C. When A and C interact, A sends the private name *e* that becomes the bond between them, and these processes evolve to Ab and Cb, whereby the counter m is incremented by 1. The same description applies to the binding of C and B by interacting on channel b, however in this case n is incremented by 1.

The expression `run ...` indicates the species at the beginning of the simulation that are, in this case, 10 A, 10 B and 10 C, where the counters m and n are set to 0. The expression `directive sample 1.0` indicates that the simulation is run until time point 1.0. The expression `directive plot` provides a list of the species that are recorded during simulation. The optional expression `directive debug` records the simulation trace to a file.

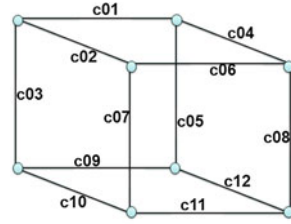
These constructs permit the modeling of systems with rich structures and unbounded length, and cannot be expressed in terms of chemical reactions [6], for instance, models of polymerization and other systems with geometric structures. As an example for this, consider the model in Fig. 1, where neighbors interact with each other over the depicted channels, and each process at each node is parameterized with a counter as in the chemical reaction system example above. However, after an interaction with a neighbor, the process that receives a message increments its counter by 1 and the process that sends a message decrements its counter by 1. The parameterization of the processes by data structures and channels, and the communication of data over these channels make it possible to encode any information that can be

```

let A(a:chan,b:chan,c:chan,k:int) =
  do ?a; A(a,b,c,k+1) or !a; A(a,b,c,k-1)
  or ?b; A(a,b,c,k+1) or !b; A(a,b,c,k-1)
  or ?c; A(a,b,c,k+1) or !c; A(a,b,c,k-1)
...
run ( A(c03,c01,c02,0) | A(c05,c01,c04,0) |
      A(c07,c06,c02,0) | A(c08,c06,c04,0) |

      A(c03,c09,c10,0) | A(c05,c09,c12,0) |
      A(c07,c11,c10,0) | A(c08,c11,c12,0) )

```



**Fig. 1** A model with a cube structure, where neighboring nodes interact with each other over the depicted channels. Each process at each node is parameterized with a counter, which increments or decrements after an interaction

used to alter the dynamics of the model. The encoded information can also be used to analyze the model and render it geometrically. In the following, we first describe how polymerization models can be constructed using the constructs discussed above. We then illustrate these ideas on an actin polymerization model, dynamics of which is otherwise challenging to capture by using alternative techniques.

### 3 A Polymer Model

We consider polymers as filaments resulting from the complexation of single monomers as in [3]. In the process representation, each state of a monomer is given by a process. Because there are two binding sites for each monomer, this results in four states. These states are represented by **Af** for the unbound (free) state, **Al** for the left-bound state, **Ar** for the right-bound state, and **Ab** for the state where it is bound on both sides. In Fig. 2, the graphical representations of this model together with the corresponding SPiM code is given. Their possible interactions are depicted as dashed arrows: the arrows 1 and 2 are the association interactions whereas the arrows 3 and 4 are the disassociation interactions: (1) a process **Af** can interact with

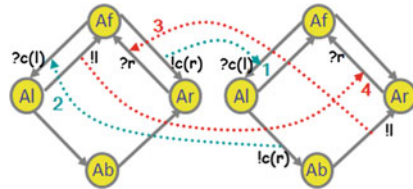
```

new c@1.0:chan(chan)
let Af() = ( new r@1.0:chan
  do ?c(l); Al(l) or !c(r); Ar(r) )

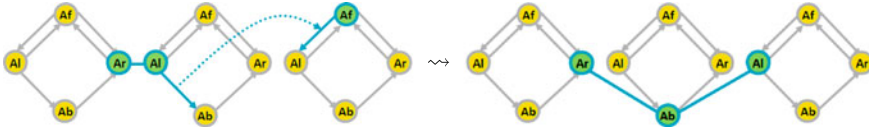
and Al(l:chan) = ( new r@1.0:chan
  do !l; Af() or !c(r); Ab(l,r) )

and Ar(r:chan) = ?r; Af()
and Ab(l:chan, r:chan) = !l; Ar(r)

```



**Fig. 2** The SPiM code of a polymerization model and the graphical representation of the processes that it implements



**Fig. 3** Graphical representation of a polymer, consisting of two monomers, i.e., a dimer, binding with a monomer to form a polymer with three monomers

another process  $Af$ , and as a result of this, one of them evolves to process  $Al$  and the other one evolves to process  $Ar$ . This describes the association of two monomers forming a dimer. (2) A process  $Af$  can also interact with a process  $Al$ , and consequently it evolves to process  $Al$  whereas  $Al$  evolves to process  $Ab$ . This describes the association of monomers to the left end of a polymer (Fig. 3). (3) A process  $Ar$  can dissociate from  $Ab$  by interacting on a name private to both processes, and then  $Ar$  evolves to process  $Af$  whereas  $Ab$  evolves to process  $Ar$ . This describes the disassociation of a monomer from the right end of a filament. (4) A process  $Al$  can dissociate from a process  $Ar$ , and as a result of this both of them evolve to process  $Af$ . This describes the disassociation of a dimer to two monomers.

This model generates polymers, the length of which are bounded only by the number of available free monomers, that is,  $Af$  processes, at the beginning of the simulation. Association of each free monomer to the left end of a filament with length  $n$  generates a filament with length  $n + 1$ . The dissociation of a monomer from the right end of a filament with length  $n + 1$  results in a filament of length  $n$  and a free monomer. In this model, polymers can only grow on one end, and shrink on the other. Thus, a chemical reaction representation of such a polymerization model with  $k$  free monomers would require  $k - 1$  reactions for modeling the generation of  $k$  polymers with different length, and modeling their degradation would also require  $k - 1$  reactions. If the association of filaments with arbitrary lengths would be considered in the model, this would require reactions that cover all the possible combinations. As a consequence ODE models of such systems introduce an ODE for the dynamics of each filament length, while making further simplifying assumptions to overcome the emerging complexity. Adding more structure to each monomer at their bound and unbound states as well as the capability to associate to other species, as we demonstrate below, introduces additional complexity, which is challenging for differential equation models.

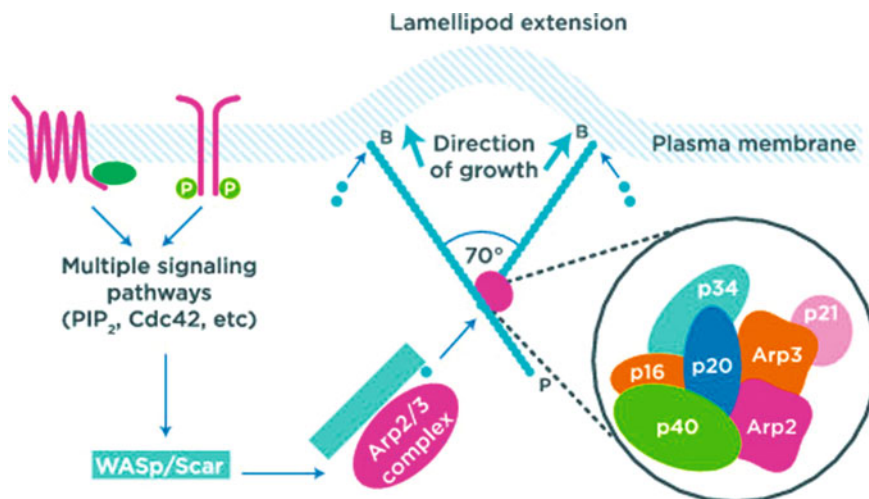
## 4 Actin

Actin is a 43-kDa protein, conserved by evolution among all eukaryotes [41]. Actin is the most abundant protein found in cells, and, together with microtubules, constitutes the principal component of the cytoskeleton of eukaryotes. In a buffer containing physiological salt conditions, actin can be found in two distinct states: either in a

monomeric soluble state called globular actin (or G-actin/Ga) or polymerized in solid filaments, where it is called filamentous actin (or F-actin/Fa).

**The role of actin in cells.** In eukaryotes, actin filaments provide mechanical support to cells and tissues. Dynamic changes in the length of actin filaments, in other words actin polymerization and depolymerization, are essential for many aspects of cell function. Specifically, remodeling of the length and organization of actin filaments is required for all functions that involve changes in cell shape, including cell motility, the division of one cell into two daughter cells and the protrusion of parts of the cells, for example the projection of axon and dendrites by neuronal cells and the capture of microorganisms during phagocytosis [34].

**Polymerization and branching.** The rate-limiting step in actin polymerization in vitro is the formation of the first complex of two to three monomers (the nucleus), also known as the nucleation phase. In eukaryotes, three actin nucleators have so far been identified: spire, formins and the Arp2/3 complex [8]. The Arp2/3 complex was the first actin nucleation factor identified. It is composed of 7 proteins, two of which -the Actin-Related Proteins (Arp) 2 and 3- are thought to interact in such a way that they resemble an actin nucleus. The Arp2/3 complex can speed up the growth of actin filaments in vitro; it can also bind to the side of pre-existing actin filaments and thereby lead to the formation of branches at a  $70^\circ$  angle as depicted in Fig. 4. The nucleating and branching activities of Arp2/3 are tightly regulated intracellularly, to ensure proper spatio-temporal control of actin polymerization. The function of the Arp2/3 complex is activated at the end of intracellular signalling



**Fig. 4** The Arp2/3 complex can bind to the side of pre-existing actin filaments and thereby lead to the formation of branches at a  $70^\circ$  angle [43]



pathways that involve the Rho-family GTP-binding proteins Rac and Cdc42 and their WASP (Wiskott-Aldrich Syndrome Protein)-family interactors [2].

**Treadmilling.** Actin monomers can self-assemble into helical F-actin in vitro in the presence of ATP. These filaments are characterized by two ends with distinct polarities: a fast growing barbed (or plus) end and a slow growing pointed (or minus) end. This asymmetry between the two ends is caused by the intrinsic ATPase activity of actin: under physiological conditions, ATP-G-actin binds to the filament's barbed end polymerizing into ATP-F-actin [31]. ATP-F-actin quickly becomes hydrolyzed into ADP-Pi-F-actin, and then the phosphate group is released leaving behind ADP-F-actin. ADP-F-actin molecules then depolymerize from filament pointed ends, forming ADP-G-actin, which, with the help of molecules such as profilin, exchange their ADP with ATP in the cytosol and are ready to polymerize again. At steady state, this cycle of polymerization, hydrolysis and depolymerization is known as treadmilling. This cycle also allows filaments to 'age'—the fast growing barbed end of filaments has a higher amount of ATP-actin, while the depolymerizing pointed end has an abundance of ADP-actin. This asymmetry is important, as several filament-depolymerizing agents, such as ADF/cofilin, bind preferentially to ADP-actin [7, 12].

**Actin In-Vivo.** In cells, actin polymerization is highly regulated, firstly through the interaction of actin monomers and polymers with a variety of actin-binding proteins (e.g., monomer-trapping proteins, filament capping and severing proteins); and secondly in response to the activation of intracellular signalling pathways by external stimuli (see Fig. 4). In-vivo, actin filaments form important macrostructures such as lamellipodia, filopodia, stress fibers and podosomes [24–26]. The underlying organization of actin filaments in each of those structures varies a lot. In lamellipodia, actin filaments are short and arranged in branched arrays [25]. Conversely, in filopodia and stress fibers, actin filaments are long, un-branched, and arranged in parallel bundles [17, 39]. In-vitro, when actin is allowed to polymerize in the presence of a cross linking agent, depending on the concentration of crosslinkers, actin forms either an isotropic gel or parallel actin bundles [20], while in-vivo, actin filaments are able to form complex structures whose activity is tightly regulated in time and space [33, 40]. Cells control actin polymerization through a large repertoire of actin binding proteins that regulate every step of the treadmilling process as well as the mechanical properties of the actin filament itself. Those actin binding proteins (ABP) are often themselves regulated by signaling pathways downstream of different Rho GTPases. For a full review of the different kinds of actin binding proteins, see [23, 33].

### Actin Models in the Literature

The field of actin dynamics and cellular motility has been the object of intense study for over thirty years. Thanks to the collaboration between theoretical and experimental scientists, it has produced a wealth of mathematical models that have elucidated existing data and produced new research lines.



Before presenting process-algebra based models, we give a short review of some of the commonly used modeling techniques describing actin polymerization both in-vivo and in-vitro. Given the depth of the field, it is impossible to do justice to all the work produced so far: instead, we describe in detail a few representative models and their applications. Models of actin vary from simple models that describe the bulk-polymerization of actin in-vitro, to complex elasticity-theory based models that examine the interaction between flexible actin polymers and the cell membrane.

The simplest models of actin polymerization consist of ODE-based models that treat all polymerized actin as a bulk variable. For example, a model of actin polymerization in the presence of capping proteins might be formalized with the model variables, defined as follows:

- $F$  is the amount of F-actin,
- $G$  is the amount of G-actin,
- $N$  is the amount of actin filaments,
- $CP$  is the amount of capping protein;
- $NCP$  is the amount of capped filament

the following parameters:

- $At$  is the total amount of actin present,
- $CPt$  is the total amount of capping protein,
- $Nt$  is the total amount of filaments,
- $k_{on}$  is the polymerization rate,
- $k_{off}$  is the depolymerization rate,
- $k_{cap}$  is the rate by CP,
- $k_{offcap}$  uncapping rate;

and the following equations:

$$\begin{aligned} dF/dt &= k_{on} \cdot N \cdot G - k_{off} \cdot N, \\ dNCP/dt &= k_{cap} \cdot N \cdot CP - k_{offcap} \cdot NCP, \\ G &= At - F, \\ N &= Nt - NCP, \\ CP &= CPt - NCP. \end{aligned}$$

ODE models are quite convenient when models are built with a small number of species, which often involves a number of simplifying assumptions that result in a depart from the biological data as demonstrated above. This sets barriers for handling filament length distributions or monitoring stochastic fluctuations of individual filaments [15]. The ODE models are however well suited to estimate kinetic parameters for proteins that affect actin dynamics from in-vitro bulk experiments [42].

The partial differential equation (PDE) models are useful in lifting some of the restrictions that are imposed by ODEs. These models are based around arrays of ODEs. A classic example was introduced by Leah Edelstein-Keshet and Bard Ermentrout [13, 14].

$$\frac{dx[3]}{dt} = k_{nuc} \cdot x[0]^3 + k_{off} \cdot x[4] - (k_{on} \cdot x[0] + k_{off}) \cdot x[3] \quad (1)$$

$$\frac{dx[4]}{dt} = k_{on} \cdot x[3] \cdot x[0] + k_{off} \cdot x[5] - (k_{on} \cdot x[0] + k_{off}) \cdot x[4] \quad (2)$$

$$\frac{dx[n]}{dt} = k_{on} \cdot x[n-1] \cdot x[0] + k_{off} \cdot x[n+1] - (k_{on} \cdot x[0] + k_{off}) \cdot x[n] \quad (3)$$

These models can also be extended to include fragmentations, different kinds of boundary conditions (maximum/minimum filament size) and different conditions such as a constant G-Actin concentration,  $x[0] = k$  (corresponding to an in-vivo situation where G-Actin is maintained buffered) or constant actin amount,  $\sum_{n=0}^N n \cdot x[n] = k$ . These models, however, can lead to a huge amount of equations if we wish to consider longer filaments or if we wish to consider side-binding of molecules to actin filaments in an arbitrary location of the filament.

The final category of models we discuss are those where forces are explicitly accounted for. These models can take many forms: some models focus on the chemical reactions between the different types of actin regulating molecules, and treat the effect of force on filaments as a perturbation on the rate of growth. For example, in [44], the rate of growth of an actin filament with a force  $F$  impinging on it, is given by

$$k^* = k_{on} e^{\frac{-f_n}{T \cdot k_b}}$$

where  $k$  is the basal polymerization rate in the absence of forces,  $f_n$  is the variable for the force on a single actin filament, and  $k_b$  and  $T$  are the parameters for the Boltzmann constant and the temperature, respectively. This relationship was originally derived in [29]. Models such as these are very popular to study actin-growth against different cellular membranes (filopodia protrusion, lamellipodia driven motion, etc), and offer a trade-off between molecular detail and an account of physical forces.

For models that aim at addressing larger scale problems, involving structures that often span across the entire cell, different approaches are available. The most well-used implementation is Cytoskin by the Nedelec group, that approximates filament growth using simple equations, but can handle problems involving very complicated cytoskeletal architecture by treating all forces in a Langevin framework.<sup>1</sup>

## A Process Model of Actin

Actin filaments are formed as a result of complex biochemical systems acting in concert. With the aim of capturing some of these biochemical mechanisms, we extend the polymer model with aspects of the actin discussed above.

The polymer model in Sect. 3 can grow at only one end and shrink only at the other, whereas actin monomers can polymerize and depolymerize at both ends. However, the depolymerization at the barbed end and the polymerization at the pointed end are very slow. In order to capture the capability of the monomers to associate and dissociate at both ends, we extend the polymer model above such that state transitions

<sup>1</sup> <http://www.cytosim.org/cytosim/index.html>

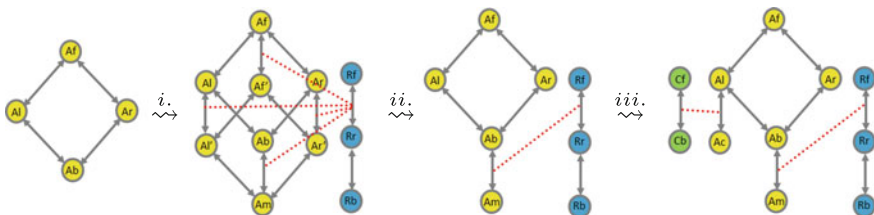
and interactions permit this dynamic behavior. As depicted on the left-hand-side of Fig. 6, we include the interaction of (5) a process  $Af$  with a process  $Ar$  such that it evolves to process  $Ar$  whereas  $Ar$  evolves to process  $Ab$ ; and the interaction of (6) a process  $Al$  to dissociate from  $Ab$  by interacting on a name private to both processes such that  $Al$  evolves to process  $Af$  whereas  $Ab$  evolves to process  $Al$ . While the former interaction describes the association of monomers to the right end of polymers, the latter describes the disassociation of monomer from the left end of filaments.

We can extend this model with severing of filaments such that two bond  $Ab$  monomers dissociate by interacting on the private channel, which models their bond; or the association of two filaments, given the binding of  $Al$  and  $Ar$  monomers.

**Branching.** The localized actin polymerization close to cell membrane depends on the association of actin monomers at the barbed end of the filaments, and also on the generation of daughter branches on mother actin filaments (see Fig. 4). The branching formation is initiated by Arp2/3 (actin related proteins) complex on the sides of existing mother actin filaments. The Arp2/3 complex anchors the pointed end of the future daughter filament to the mother filament as the free barbed end of the daughter grows away from the complex.

In order to model the branching in the filaments, we extend the model with an additional binding site such that a process that models the Arp2/3 complex, that is  $Rf$ , can bind at this site. The binding of process  $Rf$  would then evolve this process to  $Rr$ , and this way nucleate the polymerization of the daughter filament as other monomers can bind to  $Rr$ . This results in 3 binding sites for each actin monomer, thereby increasing the number of possible states for the monomers from 4 to 8. This is because each of the previously available 4 states of the monomer are extended with a capability of binding to the process  $Rf$ . This extension is depicted with  $\overset{i.}{\rightsquigarrow}$  in Fig. 5. There, an actin monomer, bound on all its three sites, is denoted with  $Am$ .

Although such a structure for the model is plausible, a simplifying assumption that reduces the number of states can be made. That is, we can rule out the association of the Arp2/3 molecule to free actin monomers and to the monomers at the ends of the filaments. This allows us to consider in the model only the binding of  $Rf$  processes to  $Ab$ , the bound actin monomer, and omit the binding of  $R$  processes to  $Af$ ,  $Al$  and



**Fig. 5** Graphical representation of incremental construction of actin models with increased complexity. (i.) Branching structure with the addition of a binding site. (ii.) Simplification of the branching structure. (iii.) Inclusion of filament capping

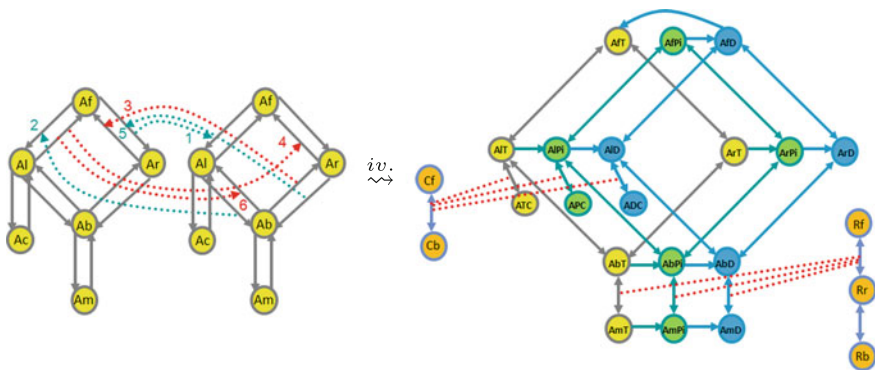
**Ar.** We thus introduce an alternative model by restricting the binding of the process **Rf** only to the monomer in the bound state **Ab**, This extension is denoted with  $\overset{ii.}{\rightsquigarrow}$  in Fig. 5.

**Capping.** In actin dependent events such as cell motility and phagocytosis, mechanisms of control for the actin assembly are essential. The barbed end of an actin filament is the site for rapid actin polymerization in cells, so altering the availability of free actin filament barbed ends provides a regulation mechanism for the actin assembly. Capping of the barbed ends by capping proteins is a mechanism, which reduces the rate of drawdown on the pool of unpolymerized actin. The free end of the new filament elongates until a capping protein becomes available. Then, capping proteins bind with a high rate to barbed ends and terminate the growth. As a result of this, each filament grows only transiently [22, 38].

We model the capping protein as a process which can bind to **Al** representing the barbed end of the filament. This extension is denoted with  $\overset{iii.}{\rightsquigarrow}$  in Fig. 5. There, **Cf** and **Cb** denote the free and bound capping protein.

**The role of ATP/ADP.** A model that explains the mechanisms that are involved in actin assembly can be further extended with the role of ATP/ADP. Filamental or monomeric actin are bound to ATP molecules which can hydrolyze to ADP- $P_i$ -actin which can then evolve to ADP-actin by dissociating the phosphate. Actin subunits in branched network hydrolyze their bound ATP quickly, but dissociate the phosphate slowly. Dissociation of phosphate initiates disassembly reactions, which then promote severing and dissociation of ADP-actin monomers from filament ends [32, 34].

In order to reflect the role of ATP/ADP, we construct a three layered model of actin monomers as depicted in Fig. 6. For this purpose, we use the model above as a single layer of an actin monomer. Each layer of this model, denotes one of the 3 states in which a monomer is either ATP bound or ADP- $P_i$  bound or ADP bound.



**Fig. 6** Graphical representation of the construction of an actin model with multiple layers modeling ATP-actin, ADP- $P_i$ -actin and ADP-actin

This extension is denoted with  $\overset{iv}{\rightsquigarrow}$  in Fig. 6. There, AfT, AlT, ArT, AbT, AmT and ATC denote the ATP-actin in its free and bound forms. AfPi and AfD denote the free ADP-P<sub>i</sub>-actin and ADP-actin, respectively. We denote their bound forms similarly. In this model, ADP-actin can hydrolyse to ADP-P<sub>i</sub>-actin and ADP-P<sub>i</sub>-actin can dissociate its phosphate to become ADP-actin. We assume that the exchange to ATP actin is quick in the free monomer and we reflect this assumption also in the structure of our model by not allowing the hydrolysis of free ATP-actin.

## 5 Rendering of Process Models

The output of the simulations of process models are often used to display the change in the number of the processes representing the biochemical species over the course of the simulation. However, when models of biochemical entities with a geometric structure are considered as in actin dynamics, their geometric representation gains importance in analyzing these systems (see, e.g., [21, 28]). In this respect, the trace resulting from the simulations can be used to display a geometric structure that emerges as a result of the simulation. This is done by encoding the geometric information as process parameters and the dynamics as functions that alter this information, by recording the emerging dynamics as the simulation evolves. In the following, we demonstrate these ideas on branching actin models discussed above.

### Encoding the Geometric Data

In order to visualize the actin filaments which are constructed by the process models, we extend our process models with coordinate parameters. The free actin processes (Af) do not have coordinate parameters, because they are assumed to be free in the cytosol. However, all the bound actin monomers are equipped with a coordinate parameter. When a free monomer binds to a filament, the free monomer evolves to a bound state, while receiving the coordinate information from the filament that it binds to. As an example for this, consider the following SPiM code in the three dimensional coordinate system.

```
let Af() = ?c(x,y,z,left_bond); Al(x, y, z + 1.0, left_bond)

and Al(point_x:float, point_y:float, point_z:float, lft:chan) =
  ( new rht@lam:chan !c(point_x, point_y, point_z, rht);
    Ab(point_x, point_y, point_z, lft, rht) )
```

When the processes Af and Al interact over the channel *c*, the process Af receives the *x*, *y*, and *z* coordinates of the process Al. Then, Af evolves into the state Al, while recording its coordinates as (*x*, *y*, *z* + 1). Al evolves into the state Ab, keeping its coordinates unchanged, because its position in space does not change, while modifying its state. Here, incrementing the *z* coordinate by 1 implements the growth vector (0, 0, 1).

When we are modeling branching filaments, we adopt this idea to include the rotation of the filaments with respect to the axis. This is because the angle between a mother actin filament and a daughter filament is measured as  $70^\circ$  [43]. Moreover, actin filaments have a helical shape with a rotating structure, repeating every 13 subunits [18]. In order to model this, we equip each monomer in a filament with three vectors. The first vector denotes the position of the monomer in the coordinate space. The second vector denotes the growth direction of the filament. The third vector denotes the growth direction of a possible daughter filament at that monomer. Let us consider the following SPiM code of the process Af.

```
let Af() = do ?c(x,y,z, x1,y1,z1, x2,y2,z2, lft);
           Al(x + x1, y + y1, z + z1, x1,y1,z1,
             x2 * 0.8853 - y2 * 0.4648,
             x2 * 0.4648 + y2 * 0.8853,
             z2, lft)
       or ?r(x,y,z, x1,y1,z1, x2,y2,z2, lft);
           Al(x + x1, y + y1, z + z1, x1,y1,z1,
             x2 * 0.8853 - y2 * 0.4648,
             x2 * 0.4648 + y2 * 0.8853,
             z2, lft)
```

Af can interact with an Al process over the channel *c*, whereby Al sends its three vectors  $(x,y,z)$ ,  $(x1,y1,z1)$ , and  $(x2,y2,z2)$  to Af. Following this, Al evolves to Ab without altering its parameters, whereas Af evolves to Al by updating its parameters in the following manner. The first vector denoting the position coordinates becomes  $(x + x1, y + y1, z + z1)$ . The second vector denoting the direction of the filament growth remains unaltered as  $(x1, y1, z1)$ . The third vector is updated by being multiplied with a rotation matrix of  $27.7^\circ$  about the *z*-axis. This results in the vector  $(x2 * 0.8853 - y2 * 0.4648, x2 * 0.4648 + y2 * 0.8853, z2)$ .

Af can interact with a process that implements the Arp2/3 molecule over the channel *r*, which can have a different rate. Here, in order to simplify the representation for the purpose of rendering, we model the binding of the Arp2/3 molecule by the transformation of the Ab into the process Arb, which is idle. The transformation of Ab upon interaction on channel *r* is implemented in SPiM as follows:

```
and Ab(x:float,y:float,z:float,
      x1:float,y1:float,z1:float,
      x2:float,y2:float,z2:float, lft:chan,rht:chan) =
  ( new e@1.0:chan !r(x,y,z, x2,y2,z2,
    x2, (0.3420 * y2) - (0.9396 * z2),
    (0.9396 * y2) + (0.3420 * z2), e);
    Arb(x,y,z, x1,y1,z1, x2,y2,z2, e,lft,rht) )
```

When Ab interacts with an Af process over the channel *r*, Ab sends its position vector  $(x,y,z)$  and its daughter filament direction vector  $(x2, y2, z2)$  as filament growth vector. The third vector is sent after being multiplied with a rotation matrix of  $70^\circ$  about the *x*-axis. This results in the vector  $(x2, 0.3420 * y2 - 0.9396 * z2, 0.9396 * y2 + 0.3420 * z2)$ .

In the simulations the monomers are assumed to be freely diffusing until they become bound to polymers. Each bound monomer is parameterized by a set of coordinates that represent its current location. These coordinates do not have any effect on the rate of interaction of the monomers. Thus, the geometric model remains consistent with the hypotheses of the stochastic simulation. It is important to note that in the structure of the model, we do not allow the interaction of the monomers at the two ends of a filament. This way, we prevent loops that would result in wrong coordinates. However, by means of additional functions, the geometric information can be used to interfere with the simulation dynamics.

## 5.1 SPiM Extensions

In order to enable geometric plotting of the models, we extend the SPiM tool. This extension consists of two parts, one for outputting the simulation results as event traces and one for filtering and plotting these events.

The original algorithm for choosing the next reaction at each step of a simulation is described in detail in [30]. At each step, this algorithm chooses a reaction from the set of possible reactions that can be one of the two types: a delay or an interaction between an output and an input. We modified the simulator such that it outputs these reactions. Then, each reaction denotes an event to be plotted: the reactants of an event denote those processes that need to be removed from the plot, while the products denote the processes that need to be added to the plot at each time step.

The following event, which can be seen as a ground instance of rewriting rules similar to those used in rule-based modeling approaches [9–11], is an example to the output of the simulator for geometric plotting. From these events, we extract the processes that are relevant for the geometric plotting together with their first vector, which gives their coordinates. By treating the left-hand-side and right-hand-side of an event as negative and positive effects of an event, we then plot the evolution of the system on the coordinate system.

```
0.1132243655 Af() Al(0.5,0.5,0.24,0.,0.,0.02,0.0086,0.0166,0.0068,rht~38)
-->
    Ab(0.5,0.5,0.24,0.,0.,0.02,0.0086,0.0166,0.0068,rht~38,rht~44)
    Al(0.5,0.5,0.26,0.,0.,0.02,-3.4304e-005,0.0187,0.0068,rht~44)
```

$$() \oplus L \triangleq L \quad (4)$$

$$X(n) \oplus L \triangleq X(n) :: L \quad \text{if } X(m) = \text{do } P1 \text{ or } \dots \text{ or } PN \quad (5)$$

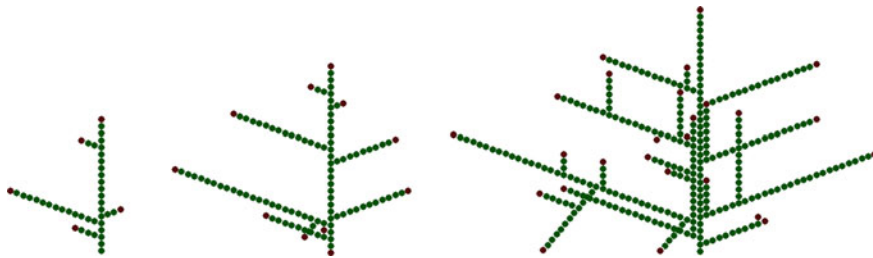
$$X(n) \oplus L \triangleq P \oplus L \quad \text{if } X(m) = P \neq \text{do } P1 \text{ or } \dots \text{ or } PN \quad (6)$$

$$P1 | \dots | PN \oplus L \triangleq P1 \oplus \dots \oplus PN \oplus L \quad (7)$$

$$\text{new } x@r:t \ P \oplus L \triangleq P\{x:=y\} \oplus L \quad \text{if } y \text{ is fresh} \quad (8)$$

**Fig. 7** Expanding a process  $P$  into a list of products  $X1(m1) :: \dots :: XN(mN)$





**Fig. 8** Screen shots from the movie generated by the actin model which demonstrates the growth of an actin filament in time in two dimensions

These events are generated as follows. In the case of a delay  $\text{delay}@r; P$  executed by a process  $X(m)$ , there is a single reactant  $X(m)$  with products  $P$ . Since  $P$  can be an arbitrary process, an additional function is needed to convert this process to a list of products  $X1(m1) :: \dots :: XN(mN)$ . The conversion is done using an expansion function  $P \oplus L$ , which adds a process  $P$  to a list  $L$ , as defined in Fig. 7.

The rules assume that each choice of actions  $\text{do } P1 \text{ or } \dots \text{ or } PN$  is associated with a corresponding process definition  $X(m)$ . This constraint is enforced by the original simulator as described in [30]. It is straightforward to prove that the expansion is compatible with the structural congruence rules of the calculus, and therefore preserves the correctness of the simulator. The expanded list of products is then added to the simulator by adapting the simulation algorithm of [30].

In the case of an interaction between an output  $!x(n); P1$  executed by a process  $X1(m1)$  and an input  $?x(m); P2$  executed by a process  $X2(m2)$ , there are two reactants  $X1(m1)$  and  $X2(m2)$  with product  $P1 | P2 \{m := n\}$ . The same expansion rules of Fig. 7 are used to convert this into a list of products.

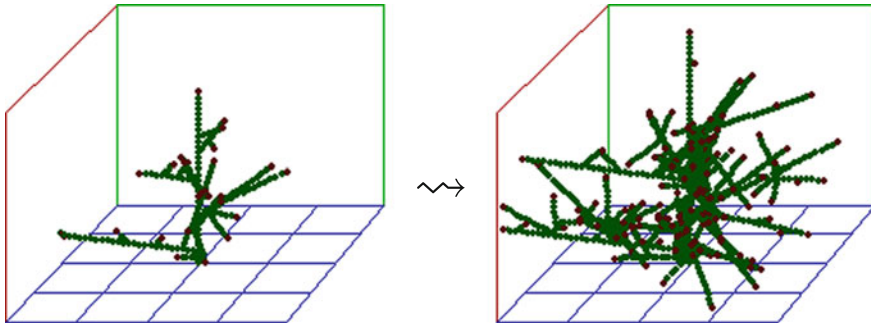
## An Example

Figures 8 and 9 display screen-shots from movies generated by this model by using the extension of the SPiM tool. These movies, obtained from the simulations, reflect the experimentally observed helical rotating structures of the filaments that repeat every 13 subunits [18] and the  $70^\circ$  angle between the mother and the daughter filaments [43]. The movies are available online as well as the code of the model used.<sup>2</sup> Here, for simplicity, we model only the association of the monomers, but not their dissociation.

## 6 Discussion

There is a growing number of experimental and theoretical studies on the components of biochemical systems with rich structures. However, it is still a challenge to develop models that can test hypotheses regarding the mechanisms of these systems,

<sup>2</sup> <http://sites.google.com/site/ozankahramanogullari/software/graphical>



**Fig. 9** Screen shots from the movie generated by the actin model which demonstrates the growth of an actin filament in time in three dimensions

especially when the models need to be composed with models established for other biological components. In particular, models of complexation as in actin dynamics by other means is challenging. This is because of the necessity in these models to rely on simplifying assumptions [1, 19, 27], for example, to treat every possible filament length as a different species in order to be able to describe its behavior with a distinct equation. The situation becomes even more complicated when different states of biochemical species, such as being bound to ATP or ADP, are considered. In this respect, process models provide an ease in experimenting with various structures without taking a depart from formal rigor or the biological data being modeled.

Because process models reflect the interaction of individual components with each other, they provide the means to lift the restrictions that are imposed on differential equation models, and thereby allow the modelers to concentrate on the interactions of each component at a higher-level setting. This becomes instrumental in choosing the right level of abstraction, since the behavior of the system arises as the emergent behavior of the components interacting with each other, as in the actual biological systems being modeled, rather than the encoded equations. However, process models face barriers when aspects of the models with a physical nature, such as force generation, are considered.

Due to the compositionality of the algebraic operators that process models employ, these models bring about a flexibility and expressivity in the construction of the models. The models presented above also benefit from compositionality, as they describe a single component of the system as a module of internal states, and employ mechanisms to run multiple instances of this component in parallel in order to simulate the emergence of structures of arbitrary lengths, which requires a Turing-complete language [6].

It is important to note that the rendering machinery discussed above does not affect the simulation dynamics, and is therefore mainly an extension to the simulator in order to deliver simulation traces. By encoding the geometric information and the vectors of growth together with affine transformations that modify them, it becomes possible to exploit these traces to render the dynamics of the model graphically.

In such a setting, the main challenge is determining how to program local changes in coordinates such that the global geometric properties of a system are accurately reproduced.

**Acknowledgments** We thank Luca Cardelli for providing the initial filament model and his helpful suggestions.

## References

1. Alberts JB, Odell GM (2004) In silico reconstitution of listeria propulsion exhibits nano-saltation. *PLOS Biol* 2:2054–2066
2. Bompard G, Caron E (2004) Regulation of WASP/WAVE proteins: making a long story short. *J Cell Biol* 166(7):957–962
3. Cardelli L (2009) Artificial biochemistry. In: *Algorithmic Bioprocesses*. LNCS, Springer, Heidelberg
4. Cardelli L, Caron E, Gardner P, Kahramanoğulları O, Phillips A (2008) A process model of actin polymerisation. In: *From biology to concurrency and back, satellite workshop of ICALP'08, ENTCS vol 229*, Elsevier, Reykjavik, Iceland, Amsterdam, The Netherlands, pp 127–144
5. Cardelli L, Caron E, Gardner P, Kahramanoğulları O, Phillips A (2009) A process model of Rho GTP-binding proteins. *Theor Comput Sci* 410/33–34:3166–3185
6. Cardelli L, Zavattaro G (2010) Turing universality of the biochemical ground form. *Math Struct Comput Sci* 20(1):45–73
7. Carlier M-F (2010) *Actin-based motility: cellular, molecular and physical aspects*. Springer, New York
8. Chhabra ES, Higgs HN (2007) The many faces of actin: matching assembly factors with cellular structures. *Nat Cell Biol* 9:1110–1121
9. Danos V, Feret J, Fontana W, Harmer R, Krivine J (2007) Rule-based modelling of cellular signalling. In: Caires L, Vasconcelos VT (eds) *Concurrency theory, proceedings of the 18th international conference on CONCUR 2007*, LNCS vol 4703. Springer, Heidelberg, pp 17–41
10. Danos V, Feret J, Fontana W, Krivine J (2007) Scalable simulation of cellular signaling networks. In: Shao Z (ed) *Proceeding of 5th asian symposium on APLAS 2007*, LNCS vol 4807. Springer, Heidelberg, pp 139–157
11. Danos V, Feret J, Fontana W, Krivine J (2008) Abstract interpretation of cellular signalling networks. In: F. Logozzo, D. Peled, L. D. Zuck, (eds), *Verification, model checking, and abstract interpretation, proceedings of the 9th international conference, VMCAI 2008*, volume 4905 of LNCS, pp 83–97. Springer, 2008
12. Dominguez R, Holmes KC (2011) Actin structure and function. *Ann Rev Biophys* 40:169–186
13. Edelstein-Keshet L, Ermentrout GB (1998) Models for the length distributions of actin filaments: I. simple polymerization and fragmentation. *Bull Math Biol* 60(3):449–475
14. Edelstein-Keshet L, Ermentrout GB (2001) A model for actin-filament length distribution in a lamellipod. *Math Biol* 355:325–355
15. Fass J, Pak C, Bamberg J, Mogilner A (2008) Stochastic simulation of actin dynamics reveals the role of annealing and fragmentation. *J Theor Biol* 252(1):173–183
16. Gillespie DT (1977) Exact stochastic simulation of coupled chemical reactions. *J Phys Chem* 81(25):2340–2361
17. Gup-ton SL, Gertler FB (2007) Filopodia: the fingers that do the walking. *Science's STKE Sig Transduct Knowl Environ* 400(400):re5
18. Holmes KC, Popp D, Gebhard W, Kabsch W (1990) Atomic model of the actin filament. *Nature* 347:44–49
19. Hu J, Matzavinos A, Othmer HG (2007) A theoretical approach to actin filament dynamics. *J Stat Phys* 128(1/2):111–138

20. Ideses Y, Brill-Karniely Y, Haviv L, Ben-Shaul A, Bernheim-Groswasser A (2008) Arp2/3 branched actin network mediates filopodia-like bundles formation in vitro. *PLoS One* 3(9):e3297
21. Iwasa JH, Mullins RD (2007) Spatial and temporal relationships between actin-filament nucleation, capping, and disassembly. *Curr Biol* 17:395–406
22. Jaffe AB, Hall A (2005) Dynamic changes in the length distribution of actin filaments during polymerization can be modulated by barbed end capping proteins. *Cell Motil Cytoskeleton* 61:1–8
23. Kovar DR, Harris ES, Mahaffy R, Higgs HN, Pollard TD (2006) Control of the assembly of ATP- and ADP-actin by formins and profilin. *Cell* 124(2):423–435
24. Mattila PK, Lappalainen P (2008) Filopodia: molecular architecture and cellular functions. *Nat Rev Mol Cell Biol* 9(6):446–454
25. Mejillano MR, Kojima S, Applewhite DA, Gertler FB, Svitkina TM, Borisy GG (2004) Lamellipodial versus filopodial mode of the actin nanomachinery: pivotal role of the filament barbed end. *Cell* 118(3):363–373
26. Mogilner A (2006) On the edge: modeling protrusion. *Curr Opin Cell Biol* 18(1):32–39
27. Mogilner A, Oster G (2003) Force generation by actin polymerization II: the elastic ratchet and tethered filaments. *Biophys J* 84:1591–1605
28. Mogilner A, Oster G (2008) Cell motility driven by actin polymerization. *Biophys J* 71:3030–3045
29. Mogilner A, Rubinstein B (2005) The physics of filopodial protrusion. *Biophys J* 89(2):782–795
30. Phillips A, Cardelli L (2007) Efficient, correct simulation of biological processes in the stochastic pi-calculus. In: *Computational methods in systems biology, LNCS vol 4695*. Springer, Heidelberg, LNBI vol 4695. Springer, Berlin, pp 184–199
31. Pollard TD (1986) Rate constants for the reactions of ATP- and ADP-actin with the ends of actin filaments. *J Cell Biol* 103(6):2747–2754
32. Pollard TD (2007) Regulation of actin filament assembly by Arp2/3 complex and formins. *Ann Rev Biophys Biomol Struct* 36:451–477
33. Pollard TD, Blanchoin L, Mullins RD (2000) Molecular mechanisms controlling actin filament dynamics in nonmuscle cells. *Ann Rev Biophys Biomol Struct* 29:545–576
34. Pollard TD, Borisy GG (2003) Cellular motility driven by assembly and disassembly of actin filaments. *Cell* 112:453–465
35. Priami C (2009) Algorithmic systems biology. *Commun ACM* 52(5):80–88
36. Priami C, Quaglia P, Zunino R (2012) An imperative language of self-modifying graphs for biological systems. In: *Proceedings of the 27th Annual ACM Symposium on Applied Computing, SAC'12 ACM New York, NY, USA*, pp 1903–1909
37. Priami C, Regev A, Shapiro E, Silverman W (2001) Application of a stochastic name-passing calculus to representation and simulation of molecular processes. *Inform Process Lett* 80:25–31
38. Schafer DA, Jennings PB, Cooper JA (1996) Dynamics of capping protein and actin assembly in vitro: uncapping barbed ends by polyphosphoinositides. *J Cell Biol* 135:169–179
39. Svitkina TM, Bulanova EA, Chaga OY, Vignjevic DM, Kojima S, Vasiliev JM, Borisy GG (2003) Mechanism of filopodia initiation by reorganization of a dendritic network. *J Cell Biol* 160(3):409–421
40. Vaggi F, Disanza A, Milanesi F, Di Fiore PP, Menna E, Matteoli M, Gov NS, Scita G, Ciliberto A (2011) The Eps8/IRSp53/VASP network differentially controls actin capping and bundling in filopodia formation. *PLoS Comput Biol* 7(7):e1002088
41. Vandekerckhove J, Weber K (1978) At least six different actins are expressed in a higher mammal: an analysis based on the amino acid sequence of the amino-terminal tryptic peptide \*1. *J Mol Biol* 126(4):783–802
42. Wear MA, Cooper JA (2004) Capping protein: new insights into mechanism and regulation. *Trends Biochem Sci* 29(8):418–428
43. Weeds A, Yeoh S (2001) Action at the Y-branch. *Science* 294:1660–1661
44. Zhuravlev PI, Papoian GA (2009) Molecular noise of capping protein binding induces macroscopic instability in filopodial dynamics. *PNAS* 106(28):11570–11575



<http://www.springer.com/978-94-007-5889-6>

Biomechanics of Cells and Tissues  
Experiments, Models and Simulations  
Lecca, P. (Ed.)  
2013, VII, 168 p., Hardcover  
ISBN: 978-94-007-5889-6