# A SystemC Platform for Signal Transduction Modelling and Simulation in Systems Biology

Rosario Distefano[1] Franco Fummi[1] Carlo Laudanna[2] Nicola Bombieri[1]
Rosalba Giugno[3]

[1]Dept. Computer Science, University of Verona, Italy
[2]Dept. Patology and Diagnostics, University of Verona, Italy
[3]Dept. Clinical and Molecular Biomedicine, University of Catania, Italy
name.surname@univr.it, giugno@dmi.unict.it

## ABSTRACT

Signal transduction is a class of cell's biological processes, which are commonly represented as highly concurrent reactive systems. In the Systems Biology community, modelling and simulation of signal transduction require overcoming issues like discrete event-based execution of complex systems, description from building blocks through composition and encapsulation, description at different levels of granularity, methods for abstraction and refinement. This paper presents a signal transduction modelling and simulation platform based on SystemC, and shows how the platform allows handling the system complexity by modelling it at different abstraction levels. The paper reports the results obtained by applying the platform to model the intracellular signalling network controlling integrin activation mediating leukocyte recruitment from the blood into the tissues.

## Categories and Subject Descriptors

Applied computing [**Life and medical sciences**]: [Systems biology]; Applied computing [**Life and medical sciences**]: [Biological networks]; Computing methodologies [**Modeling and simulation**]: [Model development and analysis]

## Keywords

Signal transduction, Modeling and simulation, SystemC.

## 1. INTRODUCTION

Modeling and simulation signal transduction systems is a key requirement for integrating in-vitro and in-vivo experimental data. In-silico simulation allows testing different experimental conditions, thus helping in the discovery of the dynamics that regulate the system. These dynamics include simulating errors in the cellular information processing that are responsible for diseases such as cancer, autoimmunity, and diabetes [8].

In System Biology, different Software are available for simulation and analysis of biochemical processes or pathways [11, 9, 4, 2]. They can be classified into two categories: those that rely on mathematical models such as ordinary differential equations [3], and those that rely on computational models, like Boolean networks [13], Petri nets [5], interactive state machines [12], and $\pi$-calculus [10]. Tools based on mathematical models have the highest potential to accurately describe and simulate the system but they are difficult to apply in case of large systems. Tools based on computational models best apply if precise quantitative relationships of the system are unknown, if the system involves many different variables, or if it changes over time [7].

Despite the adopted tool and model, a common way to explain such complex dynamical systems is to view them as highly concurrent reactive systems, whose design requires (i) techniques for composition and encapsulation starting from building blocks, (ii) methods for modeling at different abstraction levels, and (iii) efficient validation methodologies [6, 4]. All these issues related to concurrent reactive systems have been largely addressed in the past years in the electronic design automation (EDA) field and many methodologies and tools have been proposed to design and verify system-on-chips as well as embedded systems.

This paper presents a platform for modeling and simulation of signal transduction networks. The platform relies on SystemC (*www.systemc.org*), a standard language for modelling and simulation of Hardware/Software systems at different abstraction levels. The paper shows how a generic protein network representing signal transduction can be modelled at different abstraction levels, where each level distinguishes for the accuracy degree of the protein and co-factor models. The platform has been applied for modeling and simulation of the signaling network controlling LFA-1 beta2 integrin activation mediating leukocyte recruitment from the blood into the tissues. Simulation has been conducted to understand how the concerted action of the signaling proteins generate a concurrent modular mechanism of regulation of integrin activation, which is characterized both by topological and dynamic properties such as oscillations and hysteresis. The paper underlines the benefit of modeling the system at different abstraction levels, by showing how the model simulation provides information on the system properties with an accuracy degree proportional to the simulation time.

# 2. THE SYSTEMC PLATFORM

In System Biology, a signal transduction network consists of a set of biological elements, such as, *proteins* or *co-factors*. Such elements behave as concurrent objects and interact each other through activation or inhibition actions to form signal transduction chains. An element can be activated (or inhibited) by an *upstream element*, and it can activate (or inhibit) a *downstream element*. In the proposed platform, each element behaviour is formally modelled through finite state machines (FSMs) and implemented as a SystemC module through processes. The element modules are finally connected and simulated at system level.

## 2.1 Modeling of biological elements

The elements of a signal transduction network share a common behavior, which, through a FSM model, is represented by three states: *Inactive*, *Activated* and *Behaving* (see upper side of Figure 1). In the *Inactive* state, the element does not perform any biological function neither interact with other elements. The element becomes *Activated* as soon as an upstream element starts a reaction, which may consist of an activation (e.g., steric activation, phosphorilation, co-factor synthesis, etc.) or inhibition. Once activated, the element is ready to execute its biological function, that is, to react with a downstream element of the chain. Nevertheless, this can happen only after a *delay time*, which represents the time spent by the element to reach the target. Thus, after the delay time, the element state moves to *Behaving*, in which the element executes its biological function. The delay time depends on several factors, such as, the molecular concentrations of the element and of the target. Any element returns to the inactive state either if it receives an inhibition signal by an upstream protein or if the element *lifetime* expires. *t* represents the time elapsed, which is constantly updated during simulation, while *lifetime* represents the maximum lifetime from the activation instant in which the protein carries out its biological function. In the proposed FSM model, the transition guards (i.e., the conditions controlling the state transitions) are expressed in terms of variables (e.g., delay_time, lifetime) as well as activation or inhibition events raised by upstream elements. To model such a behavior, we define three classes of input/output signals:

- *Unknown inputs (Input_Ui)*: They are inputs whose values depends on the environment characteristics and status, which are unknown at modelling time. Some examples are the *delay time* (i.e., time spent by the protein to reach a protein target), the molecular concentrations, and the element lifetime. For each unknown input, the platform generates different values with the aim of observing, via simulation, how such values affect the biological system dynamics.
- *Topological inputs (Input_Ti)*: They are inputs whose values depend on the topological interactions of the modelled element with upstream elements, such as activation via phosphorylation, steric, co-factor, or inhibition. During simulation, topological inputs may be dynamically set to a value representing an activation or an inhibition action.
- *Topological outputs (Output_Ti)*: They are outputs whose values are set at simulation time and depend on the role of the modelled element towards downstream elements.

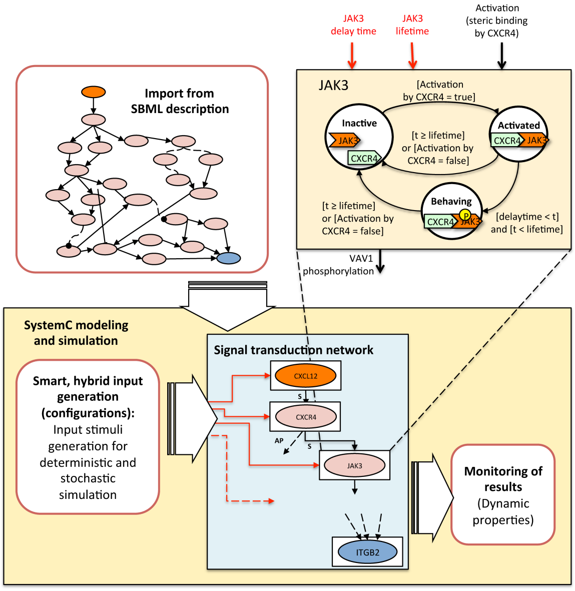

**Figure 1: The SystemC platform.**

## 2.2 System-level Simulation Platform

Figure 1 shows the SystemC-based platform, in which the FSM model of the JAK3 protein is reported as example among the elements of the system. All the element modules are connected through SystemC signals, to form the networks. The system of proteins and co-factors is connected to a stimuli generator, which automatically generates patterns of values for each unknown variable. We refer to a pattern of values as *configuration* of the system. A configuration consists of values for each unknown input of the network elements. The platform generates a new configuration and runs (i.e., execute) a dynamic simulation of the system for a given simulation time. During such a run, the platform monitors the system properties (Monitoring of results) by observing the behavior of one ore more network elements (e.g., it monitors the state and the molecular concentration of a given protein). After the simulation time, the platform generates a new configuration and starts a new run. The whole simulation ends when all the possible configurations have been run. The simulation aims at identifying those configurations that lead the system to specific behaviors.
The platform generates the configurations by combining deterministic and probabilistic approaches. The main goal of the input generation is to explore and, at the same time, to handle the solution space through an hybrid approach (i.e., deterministic and probabilistic), by exploiting the the simulation results to drive the generation of a new configuration.
The proposed FSM model, which is shared by each network element, allows the corresponding SystemC implementation to be automatically generated from a Systems Biology Markup Language (SBML) description [1]. SBML is a representation format, based on XML, for communicating and storing computational models of biological processes. It is a free and open standard with widespread software support and a community of users and developers. SBML can represent many different classes of biological phenomena, including metabolic networks, cell signaling pathways, regulatory networks, infectious diseases, and many others. It is the de facto standard for representing computational models in systems biology today.

# 3. ABSTRACTION LEVELS AND SIMULATION ACCURACY

At system-level, concurrency and interaction of signal transduction elements can be modelled and simulated at different levels of accuracy. Consider, for example, the element interaction represented in Figure 2. Protein *P1* may bind with (and, thus, activate) three different target elements (i.e., proteins), *P2*, *P3* and *P4*. In turn, *P2* or *P3* may bind with *P5*, while *P4* may inhibit *P6*. As a result, *P1* may be involved into three different pathways, two of them forming protein complexes (*P1-P2-P5*, *P1-P3-P5*), while the third one to inhibit *P6* (*P1-P4-P6*). Such a dynamic interaction can be viewed at different abstraction levels, as described in the following sections.

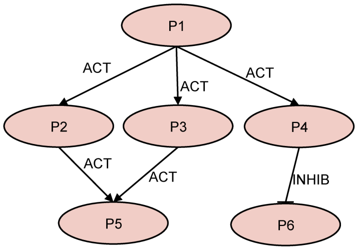

**Figure 2: An example of protein network**

## 3.1 High Level Model of Elements

At this level, each network element can establish only one interaction at a time (e.g., *P1* may activate either *P2*, *P3*, or *P4*) and each element may be part of one complex at a time (e.g., P5 may be part of either *P1-P2-P5* or *P1-P3-P5*). Each element behavior is modeled through a single FSM and concurrency is viewed at protein/co-factor level. The element behavior is implemented by a single SystemC process, which is sensible to the input signals coming from upstream elements and writes on output ports to activate/inhibit one of the downstream elements at a time. The element interaction is modeled through boolean signals (i.e., activation/inhibition true or false). The downstream target is chosen by following a probability distribution, which takes into account the molecular concentrations of the target elements. The higher the molecular concentration of a target, the higher the interaction probability.

## 3.2 Intermediate Level Model of Elements

At this level, each element of the network can establish more than one interactions at a time. (e.g., a subset of molecules of *P1* may activate molecules of *P2*, and a different subset of molecules of P1 may activate *P3*, or *P4*). Each element may be part of one or more complexes at a time (e.g., some molecules of P5 may be part of *P1-P2-P5* and some others may be part of *P1-P3-P5*). Each element behavior is modeled through one or more FSMs and the description granularity allows viewing concurrency among molecular sets of protein/co-factors.

The element behavior is implemented by more SystemC processes, and the number of implemented FSMs is handled dynamically at run time. As soon as a set of molecules of the upstream element reaches the modeled element, a new FSM is created to implement the behavior of the subset of the element molecules interacting with them. The downstream targets are chosen by following a probability distribution, which takes into account the molecular concentrations of the target elements. Differently from the HLM model, a set of molecules can be split over different targets.

This level allows signal transduction networks to be modeled more accurately than HLM and, thus, it allows analysing their dynamic properties more in detail. On the other hand, it requires implementing many FSMs (which number depends on the system topology) and handling a larger solution space, with a direct impact on the simulation time.

## 3.3 Low Level Model of Elements

At this level, concurrency is viewed and implemented at molecular level. Each element behavior is modeled through a FSM per molecule. The element behavior is implemented by more SystemC processes, and the number of implemented FSMs corresponds to the molecular number of the system. The downstream target of each element molecule is chosen by following a probability distribution that, still, takes into account the molecular concentrations of the target elements. This level allows signal transduction networks to be modeled with the maximum accuracy. Nevertheless, such a modeling style may lead to the a prohibitive number of FSMs and to an intractable solution space to explore in case of complex systems. Thus, it fits to model subsets of signal transduction networks or to model networks with a reduced molecular number of each element.

# 4. EXPERIMENTAL RESULTS

The SystemC platform has been applied to model and simulate the leukocyte recruitment system at different abstraction levels, as proposed in Section 3. The model simulations have been conducted to identify the system properties (i.e., the system *configurations*) that lead to oscillating behaviors, and to compare the results obtained at each abstraction level in terms of accuracy and simulation time. The main goal was identifying the configurations that lead to oscillations of ITGB2 with a period of 30-40 ms, which represents the average stopping time of a cell when it interacts with the blood vessel epithelium.

Table 1 reports the characteristics of the system, which have been provided as input (i.e., input assumptions deduced by in-vitro experimental data) to restrict the solution space. Each protein and cofactor (reported in the table with (P) and (C), respectively) have been simulated with different molecular concentrations, within the range reported in column *MConcentration*. The delay time of each protein has been generated within the range reported in column *delay time* and biased as function of the molecular concentration of the target element. The lifetime value in each configuration has been sampled in the range reported in column *lifetime*. For each configuration, the system dynamics have been simulated and monitored for a total time of 250 ms.

For each implementation of the system (i.e., HLM, ILM, LLM), we run 41,943,040 configurations. Table 2 reports the results in terms of simulation time, amount of configurations that lead to oscillations of ITGB2 (in percent over the total amount of configurations), and amount of *useful* rather than *redundant* configurations. We refer as useful those configurations that lead to oscillations and differ each other by at least one input value. A configuration is redundant (and thus it does not represent any new property) when it is a duplicate of any useful configuration, and occurs

**Table 1: The protein network characteristics**

| | Unknown inputs | | |
|---|---|---|---|
| | **MConcentration (# molecules)** | **delay time (ms)** | **lifetime (ms)** |
| CXCL12 (P) | [1,400] | - | [250,250] |
| CXCR4 (P) | [1,325] | [2,3] | [250,250] |
| JAK3 (P) | [1,300] | [2,5] | [250,250] |
| JAK2 (P) | [1,175] | [2,5] | [42,42] |
| ABG (P) | [1,200] | [2,5] | [31,37] |
| VAV1 (P) | [1,168] | [2,2] | [45,51] |
| RAC1 (P) | [1,235] | [2,6] | [34,40] |
| RHOA (P) | [1,146] | [2,6] | [29,35] |
| CDC42 (P) | [1,256] | [2,2] | [35,41] |
| PLC (P) | [1,210] | [2,4] | [33,33] |
| IP3 (C) | [1,115] | [2,5] | [51,57] |
| CA (C) | [1,140] | [2,5] | [44,50] |
| DAG (C) | [1,123] | [2,5] | [56,62] |
| RASGRP1 (P) | [1,127] | [2,4] | [32,38] |
| PLD1 (P) | [1,67] | [2,4] | [28,28] |
| PIP5K1C (P) | [1,234] | [2,4] | [27,33] |
| PA (C) | [1,322] | [2,2] | [63,69] |
| RAP1A (P) | [1,364] | [2,2] | [34,40] |
| PIP2 (C) | [1,243] | [2,3] | [55,61] |
| RIAM (P) | [1,435] | [2,4] | [39,39] |
| RASSF5 (P) | [1,134] | [2,5] | [32,38] |
| FERMT3 (P) | [1,123] | [2,5] | [31,31] |
| TLN1 (P) | [1,364 | [2,5] | [36,36] |
| ITGB2 (P) | [1,125] | - | [43,49] |

**Table 2: Experimental results obtained at each abstraction level.** $^*MConcentration \in [1, \lceil max/100 \rceil]$

| Level | Config. (#) | Sim.time (min) | Oscill. (%) | Useful config(%) | Redound. config(%) |
|---|---|---|---|---|---|
| HLM | 4,194,304 | 10 | 1.1 | 0.3 | 0.7 |
| ILM | 4,194,304 | 150 | 5.2 | 2.9 | 2.1 |
| LLM* | 4,194,304 | 850 | 1.0 | 0.96 | 0.04 |

when a configuration value is randomly chosen twice, due to the hybrid deterministic/stochastic approach adopted for the input generation (see Section 2.2). To avoid the state explosion in the LLM model, we implemented the system as explained in Section 3.3, but we reduced its complexity by limiting the maximum molecular concentration of each protein. Table 2 shows that the HLM simulation provides results in less time, even though the quality of such results is lower (i.e., the number of observed oscillations is low and the majority of them are given by redundant configurations). The table shows that the quality of results and the simulation time increase by refining the model. In the LLM simulation, the amount of oscillations is low w.r.t. the higher level implementations since the simulated model is reduced (MConcentration). The table shows that, at this level, the accuracy of the obtained results is the highest. In general, the obtained results confirm the benefit of modeling the system at different abstraction levels, that is, at each level, the model simulation provides information on the system property with an accuracy degree proportional to the simulation time.

Finally, Table 3 reports a more detailed analysis of the obtained results (which, for the lack of space, are reported for the HLM simulation only). In particular, Table 3 reports the amount of configurations that lead to the ITGB2 oscillation (*# config.*), periodic and aperiodic, by grouping them into sets depending on the number of oscillations (*# ITGB2 osc.*) over the simulated period of 250 ms (i.e., from 2 to 8).

## 5. CONCLUSIONS

**Table 3: HLM implementation**

| | Periodic osc. | | Aperiodic osc. | |
|---|---|---|---|---|
| **# ITGB2 osc.** | **# config.** | **%** | **# config.** | **%** |
| 2 | 51,241 | 52.06 | 11,740,831 | 41.79 |
| 3 | 29,583 | 30.06 | 9,568,149 | 34.06 |
| 4 | 13,172 | 13.38 | 4,822,496 | 17.17 |
| 5 | 4,000 | 4.06 | 1,550,865 | 5.52 |
| 6 | 381 | 0.39 | 349,761 | 1.24 |
| 7 | 34 | 0.03 | 55,829 | 0.2 |
| 8 | 7 | 0.01 | 5,662 | 0.02 |

The paper presented a signal transduction modeling and simulation platform based on SystemC. The platform allows the complexity of such a biological system to be handled by modelling it at different abstraction levels. Experimental results obtained by applying the platform to model the intracellular signalling network controlling integrin activation mediating leukocyte recruitment from the blood into the tissues have been presented and analysed.

## 6. REFERENCES

[1] The Systems Biology Markup Language. `http://www.sbml.org`.

[2] N. Bombieri, R. Distefano, G. Scardoni, F. Fummi, C. Laudanna, and R. Giugno. Dynamic modeling and simulation of leukocyte integrin activation through an electronic design automation framework. In *Proc. of the Int. Conf. on Computational Methods in Systems Biology (CMSB) - Springer*, pages 143–154, 2014.

[3] J. C. Butcher. *Numerical Methods for Ordinary Differential Equations*. Wiley, Chichester, UK, 2003.

[4] L. Calzone, F. Fages, and S. Soliman. Biocham: An environment for modeling biological systems and formalizing experimental knowledge. *Bioinformatics*, 22(14):1805–1807, 2006.

[5] C. Chaouiya. Petri net modelling of biological networks. *Brief. in Bioinformatics*, 8(4):210–9, 2007.

[6] J. Fisher, D. Harel, and T. A. Henzinger. Biology as reactivity. *Commun. ACM*, 54(10):72–82, Oct. 2011.

[7] J. Fisher and T. A. Henzinger. Executable cell biology. *Nature Biotechnology*, 25:1239 – 1249, 2007.

[8] E. Gonçalves, J. Bucher, A. Ryll, J. Niklas, K. Mauch, S. Klamt, M. Rocha, and J. Saez-Rodriguez. Bridging the layers: Towards integration of signal transduction, regulation and metabolism into mathematical models. *Molecular BioSystems*, 9(7):1576–1583, 2013.

[9] N. Le Novère and T. Shimizu. Stochsim: modelling of stochastic biomolecular processes. *Bioinformatics*, 17(6):575–576, 2001.

[10] C. Priami. Stochastic pi-calculus. *The Computer Journal*, 38(7):578–589, 1995.

[11] D. Resasco, F. Gao, F. Morgan, I. Novak, J. Schaff, and B. Slepchenko. Virtual cell: computational tools for modeling in cell biology. *Wiley Interdiscip Rev Syst Biol Med*, 4(2):120–140, 2012.

[12] A. Sadot, J. Fisher, D. Barak, Y. Admanit, M. J. Stern, E. J. Hubbard, and D. Harel. Toward verified biological models. *IEEE Trans. on Computational Biology and Bioinformatics*, 5(2):223–34, 2008.

[13] S. Srihari, V. Raman, H. W. Leong, and M. A. Ragan. Evolution and controllability of cancer networks: A boolean perspective. *IEEE/ACM Transactions on*

*Computational Biology and Bioinformatics*, 11(1):83–94, 2013.