An Agent-oriented Conceptual Framework for Biological Systems Simulation

Nicola Cannata[1], Flavio Corradini[2], Emanuela Merelli[2], Andrea Omicini[3], and Alessandro Ricci[3]

[1] CRIBI Biotechnology Centre, Università di Padova
via Ugo Bassi 58/B, 35131 Padova, Italy
nicola@cribi.unipd.it
[2] Dipartimento di Matematica e Informatica, Università di Camerino
Via Madonna delle Carceri 62032 Camerino, Italy
{flavio.corradini,emanuela.merelli}@unicam.it
[3] Dipartimento di Elettronica, Informatica e Sistemistica, Università di Bologna
Via Venezia 52, 47023 Cesena, Italy
andrea.omicini@unibo.it, aricci@deis.unibo.it

Abstract. Recently, a collective effort from multiple research areas has been made to understand biological systems at the system level. On the one hand, researchers working on Systems Biology aim at understanding how living systems perform routinely complex tasks. On the other, people involved in Pharmacogenomics strive to study how an individual's genetic inheritance affects the body's response to drugs. Research in all the above disciplines requires the ability to simulate particular biological processes (i.e. metabolic pathways) which characterize biological systems as cells, organs, organisms and communities. Biological processes are complex systems, i.e. a set of components that interacts with each other and with an external dynamic environment.
In this work, we aim at providing an alternative way to specify complex systems based on behavioral modelling. We consider a biological process as an activity-based application performed by actors in a dynamic and sometime unpredictable environment; each actor plays his role in relation to the process it is involved, but in general they are part of a more complex system. We propose a conceptual framework to engineering an agent society which simulate the behavior of a biological process. The agents society and its social rules are described through a coordination model specified for a biological process with semiformal languages based on System Biology Modelling Languages for the static-structural and functional views and on UML-like diagrams for the dynamic (control flow) behavior.

1 Introduction

In the last years, biology has made many steps to establish a strong relation with computer science. Several new research fields arose, sometimes covered by the term "Bioinformatics", which has been defined by the National Institutes of Health Committee as: "Research, development, or application of computational tools and approaches for expanding the use of biological, medical, behavioral

or health data, including those to acquire, store, organize, archive, analyze, or visualize such data" [4]. Therefore the term Bioinformatics includes research fields from biological data management to biological processes simulation.

In this work we will consider that part of modern biology regarding modelling and simulation of biological processes indicated as Systems Biology, which aims at system-level understanding of biological systems [1]. The rapidly growing huge amount of knowledge recently acquired at molecular level (i.e. Genomes, Transcriptomes, Proteomes, Metabolomes, Interactomes) now is giving for the first time the opportunity to constitute the solid ground upon which to create an understanding at the system level of the living organisms (the Systeome project suggested in [1]). This effort is intended not only to describe in detail the system structure and behavior but also to comprehend its reaction in response to external stimuli or disruptions. Examples of molecular networks already well studied comprise gene regulation networks (how genes and their products, proteins, are regulating gene expression), metabolic pathways (the chains of reaction connecting metabolites) and signal transduction cascades (the molecular interactions activating a genetic answer to a signal received from the cell).

Because of the scale, nature and structure of the data, this new scientific challenge is much more demanding than ever and is intended to involve computer scientist, mathematicians, physicists, biochemists, engineers and automatic control systems experts working in close partnership with life scientist. Fundamental, in this optics, are the information management framework and the model construction, analysis and validation phases [2]. Efforts are being made to provide a common and versatile software platform for Systems Biology research and in the Systems Biology Workbench [5] project critical issues regard the exchange of data and the interface between software modules; SBML [6] and CELLML [7] description languages are emerging standard for simulation in system biology.

Although Systems Biology has no clear end point, the prize to be attained is immense. From in silico drug design and testing to individualized medicine, which will take into account physiological and genetic profiles, there is the potential to profoundly affect health care and medical science generally [2].

Our intention is to provide a general framework for biological processes modelling, to support life scientist in the building and verification of their hypotheses. We believe that use of an agent coordination infrastructure [3] will allow dominating the complexity of the biological domain by delegating to intelligent software agents learning of behavior of the bio-entities they are expected to emulate.

2 Biological Systems Understanding

A biological system is an assembly of biological components; to understand a system it is not sufficient to describe its components in details, it is necessary to

[4] http://www.nih.gov
[5] http://sbw.sourceforge.net
[6] http://www.sbml.org
[7] http://www.cellml.org

describe the behavior of molecules in relation to the characteristics of the system and also to comprehend what happens when certain stimuli or disruptions occur. In order to understand biological systems as a system, Kitano in [1] suggests to accomplish four steps: 1. *System structure identification*, 2. *System behavior analysis*, 3. *System control*, 4. *System design*.

System structure identification leads to the specification of the topological relationship of the network of components as well as parameters for each relation. For example to identify a metabolic pathway, one must identify all components of the pathway, the function of each component, interactions and all associated parameters by also using experimental data to infer new results as prediction of new gene and interactions. This process, (we have experimented on modelling malaria process and cell behavior in previous works [4, 5], is divided into two tasks: 1) network structure identification, and 2) parameter identification.

System behavior analysis means to understand the mechanisms that are behind the robustness and stability of the system, and functionalities of the interactions among components. *Simulation is an essential tool both to understand the behavior and to design the biological process.* There is a necessity to develop a simulator system tool that is user-friendly, highly functional, accurate and modular. The simulator need to be coupled with parameter optimization tools, an hypothesis generator and a group of analysis tools.
We need to develop a common platform that integrates many modules useful to assist systems biology research as database for storing experimental data, a cell and tissues simulator, parameter optimization software, etc.

System control in the simulation of a complex system, usually allows to increase the stability of the system. In particular, in biological systems we can frequently identify the two most used control schemas, the feedforward and the feedback control. In fact the feedforward control represents the an open-loop control where a set a predefined reaction sequences is triggered when a certain stimulus is present. The feedback control is a close-loop control which allows to process the signal in output as one of the input of the system and therefore to control the desired behavior of the system. Many examples can be found for demonstrating the utility of feedback control. For the purpose of this work we choose the simulation of the growing process of human cells. We know that the protein P53, each time a cell is undergone by environmental stress, is activated and it sends a signal to P66 which in turns produces oxidant substances that if they are in excess will induce the cell to kill itself (apopthosis); the close-loop will control the cell behavior.
Beyond the stability, a system must be robust, therefore redundancy (i.e. duplication of components and clustering of components with similar functions) is a widely used method to ensure the correct functioning of the system.

System design The last, but non least important aspect, is the design of biological system for instance to support the pharmacogenomics (the study of

how an individual's genetic inheritance affects the body's response to drugs) [8] Pharmacogenomics holds the promise that drugs might one day be tailor-made for individuals and adapted to each person's own genetic makeup. Environment, diet, age, lifestyle, and state of health all can influence a person's response to medicines, but understanding an individual's genetic makeup is thought to be the key to creating, through a biological system design, personalized drugs with greater efficacy and safety. Since biological systems are so complex and the biological knowledge is still so poor, we need a technology which supports the design of small and modular systems and then their composition and integration.

The system structure identification and the system behavior analysis are activities concerning the biologist work, whereas the system simulation and control are activities that can be supported by a suitable conceptual framework. The system design phase would benefit from the experience matured in the software engineering research environment, for instance by using *reverse engineering* methodologies and considering the biological system as a black box.

In general a conceptual framework should allow biologists, given the systems structure and system behavior of a biological system as input, to create the corresponding model and to simulate the system by studying its behavior and by verifying the properties of each component. Of course, a more sophisticated scenario is that in which a biologist does not know the system structure and behavior, (he/she only can access the huge amount of data) but aims at inferring a new model by mining biological data and knowledge dispersed all over the world.

3 Related Works

In the literature we can find different approaches to biological processes simulation with some, more or less limited, successful modelling examples.

ODE based. This is the classical approach arising from the biochemical point of view. A network of interactions (chemical reactions) between molecules (metabolites, proteins, genes) is established and Ordinary Differential Equations describe numerically the continuous variations in the concentration of substances. The GEPASI program [6], even with a limited number of metabolites and reactions is for sure a milestone in this area. The group of Mendes at Virginia Bioinformatics Institute, in collaboration with the Kummer group at EML is now accomplishing the development of COPASI, the successor of GEPASI, capable of carrying out more sophisticated analysis (stochastic integration, non linear dynamic analysis such as bifurcation). Another ODE-based software environment is E-CELL [7], in which the user can define protein functions, protein-protein and protein-DNA interactions and regulation of gene expression and observe the dynamic changes in the concentrations of proteins, protein complexes and other chemical compound in the cell. The authors simulated also a hypothetical cell

[8] The term comes from the words pharmacology and genomics and is thus the intersection of pharmaceuticals and genetics.

with a 127 genes genome sufficient for the processes of transcription, translation, energy production and phospholipids synthesis.

LISP based. QSIM [8] represents a Qualitative Reasoning LISP-based approach to the problem.

PI-Calculus based. Bio-Calculus [9] tries to bridge the gap between the biochemical approach and the formal and symbolic handling capacity requested from the Computer Science. A syntax similar to conventional expressions in Biology is provided to describe the system structure but at the same time is giving the information needed for simulation and a mathematical background. The authors introduced a biosyntax and multisemantic system. In their opinion is almost impossible to define a unique absolute simulation model valid at all the level and for all the processes. They showed the practicality of Bio-Calculus by describing and simulating some molecular interactions. In [10, 11] is remarked the necessity of describing biomolecular processes using a formal computer language. The chosen formalism is PI-Calculus, the process algebra originally introduced for describing computer processes and a simulation system (PiFPC) for execution and analysis of PI-Calculus program was developed and experimented with a model for the RTK-MAPK signal transduction pathway. Another example of PI-Calculus application is the VICE virtual cell [12]. The authors observe that cell's mechanism and global computing applications are closely related and that biological components can be thought as processes while organisms as networks. The interactions between biological components are then represented by the communications between processes. They also proposed a very basic cell with a 180 genes genomes and the essentials metabolic pathways.

Petri-nets based. In [13], it is observed that all the existing approaches have some disadvantages, lacking for example of valid GUI interfaces (bio-pathway editors) or with some implicit weakness in the architecture itself. To overcame this situation they suggest to use an architecture based on Petri nets because of their intuitive graphical representation and their capabilities for mathematical analysis. Several enhanced Petri nets (for example colored Petri nets and stochastic Petri nets) have been used to model biological phenomena but a more suitable approach is constituted by hybrid Petri nets that take in account both the discrete and continuous dynamics aspects.

MAS based. Cellulat [14] and Stem-Cell [15] are two examples of MAS application to modelling and simulation of cell behavior. Cellulat is an intracellular signalling model based on the fusion of agent-based approach and the blackboard architecture. In Stem-Cell the authors propose new formal models and simulations of new theories of stem cell behaviors by using functional programming language

4 Motivation

From the analysis of the literature, the need for formal frameworks for the modelling and simulation of a Biological System clearly emerge, in order to get the chance of applying the wide range of existing methods and tools provided by Computer Science (property design and verification, automated reasoning, ...). Another important remark regards the importance of the intuitive graphical representation of systems, along with the network of all the interactions among the entities, and the possibility to define it in an easy way (user friendly GUI). The choice of the computational paradigm affects the possibility to perform more complicated tasks like stochastic integration or non-linear dynamic analysis like bifurcation, and to describe discrete and continuous hybrid systems.

What we found illuminating was the molecular-as-computation abstraction presented in [16], in which a system of interacting molecular entities is described and modelled by a system of interacting computational entities. Abstract computer languages, originally developed for the specification and study of systems of interacting computations are now actively used to represent biomolecular systems, including regulatory, metabolic and signalling pathways as well as multicellular processes. Processes, the basic interacting computational entities of these languages have an internal state and interaction capabilities. Process behavior is governed by reaction rules specifying the response to an input message based on its content and on the state of the process. The response can include state change, a change in interaction capabilities and/or sending messages. Complex entities are described hierarchically. Using this abstraction opens up new possibilities for understanding molecular systems. Computers and biomolecular systems both start from a small set of elementary components from which, layer by layer, more complex entities are constructed with evermore sophisticated functions. While computers are networked to perform larger and larger computations, cells form multicellular organs and organisms and organisms build societies.

As we cannot and need not to recreate the world as an isomorphic in silico image of itself [2] it has no sense to start the modelling from the atomic level. At the molecular level we can consider the cell chemical components: water, inorganic ions, sugars, aminoacids, nucleotides, fatty acid and other small molecules. They can interact with each others and be used to build up more complex macromolecules, like polysaccharides, composed of sugars, nucleic acids (DNA and RNA), composed of nucleotides, and proteins, composed of aminoacids. Macromolecules can then generate macromolecular aggregates, for example the ribosome is made out of proteins and RNAs. Molecules can have an internal state, for example proteins can have different conformational states, they can be phosphorilated, the DNA can be methylated and the RNA can form secondary structure. Another level of modularity can be found in proteins domains (autonomous functional subunit of the proteins) and in nucleic acids signals, for example transcription factor binding sites in the DNA, protein binding sites in the RNA, which usually are involved in the molecular interactions. The interactions at the various levels are often modelled by chemical covalent bonds, either very strong and stable (for example the peptidic bond between aminoacids) or

weak and temporary (for example the binding of proteins to DNA or between proteins). The composition of a lot of simpler interactions makes up the cellular processes that are fundamental for the physiology of the cell, for example the DNA replication (to propagate the genetic inheritance to the following generations) and the "gene" expression, that is composed of more phases, the most important being the RNA transcription and the protein translation (to produce all the perturbed life of the cell and the molecular machinery necessary for the normal). For each of these processes soon would be possible to exactly individuate which are the actors involved, which is their role, which are the interactions between them, which is the result of the interactions, which new entity are produced by the composition of simpler entities.

5 An Agent-oriented Framework for Systems Biology

Generally speaking, multiagent systems (MASs) are considered the right level of abstraction for modelling and engineering complex systems, characterized by organization structures and coordination processes that are more and more articulated and dynamic [17, 18]. Also, they are considered a promising approach for engineering simulations of complex systems, as one can see from the series of the Multi-Agent Based Simulation (MABS) workshops – held since 1998 – or from the Journal of Artificial Societies and Social Simulation (JASSS).

In particular, MAS-based models are often used for the simulation of systemic and social phenomena [19, 20]. Recently, however, their effectiveness has been remarked also beyond social simulation in domains where traditional techniques are typically adopted [21], such as parallel and distributed discrete event system simulation, object oriented simulation, and dynamic micro simulation. In general, simulations based on the agent paradigm integrate aspects that can be found both in *micro* and *macro* techniques to simulation.

On the one side, in the same way as micro techniques, agent-based approaches model specific behaviour of individual entities or components. This can be contrasted to macro simulation techniques which are typically based on mathematical models where the characteristics of a *population* are averaged together and the model attempts to simulate changes in the averaged characteristics of the whole population. Thus, in macro simulation the set of individuals is viewed as a structure that can be characterized by a number of variables, whereas in micro simulations the structure is viewed as emergent from the interactions between the individuals. Parunak et al. [22] recently compared the approaches and pointed-out that "... agent-based modelling is most appropriate for domains characterized by a high degree of localization and distribution and dominated by discrete decision. Equation-based modelling is most naturally applied in systems that can be modelled centrally, and in which the dynamics are dominated by physic laws rather that information processing...". Here, we promote a conceptual framework for simulating biological systems heavily based on localization, distribution and interaction (communication) of the system components, where an agent-based approach seems to be more effective. On the other side, in the same way as in macro techniques, agent-based approaches promote the investigation of systemic properties that cannot be understood at the individual

component level, but require the introduction of new categories for their description. In other words, agent-based approaches make it possible to simulate and analyze *emergent* properties, which can be understood as properties of the ensemble of the components in the overall.

The rest of the section first recalls the basic properties of the agent and MAS paradigm and how do they fit within the biological system simulation domain.

5.1 Agents, Societies and Mediating Artifacts for Modelling Biological Systems

The notion of agent has been described in several ways in literature [23–25], with different acceptation according to the research field where it has been considered – from distributed artificial intelligence (DAI) to software engineering and concurrent/distributed systems, from social/psychological/economic sciences to computer supported cooperative work (CSCW). In this context we stick to a weak definition of agency, where agents feature two basic foundational properties: *autonomy* and *situatedness*.

As an autonomous entity, an agent encapsulates the execution of independent activities or tasks within the overall system/environment. In the software engineering context, autonomy is characterized by the encapsulation of behavior control: as for the object abstraction, agents encapsulate a state and a behavior; differently from objects, agents have full control of both (objects only of their state). So agents work by autonomously executing their tasks, concurrently to the work of the other agents.

As a situated entity, an agent is a persistent entity immersed within and interacting with an environment, which is typically open and dynamic. Interaction – in its most wide characterization [26] – is a fundamental dimension of the agent paradigm: generally speaking, an agent interacts with its environment by means of actions and perceptions, which enable the agent to partially observe and control the environment.

In the literature, further characterizations have been attributed to the agent abstraction: examples are pro-activity (as the capability of taking initiative) and social ability (the adoption of high level languages for inter-agent communication). Heterogeneous computational/behavioural models have led to different forms of agent classification: examples are *intelligent* agents – when the agent behavior is defined in terms of high level cognitive/mentalistic structures and processes, with an explicit symbolic representation of knowledge, interaction and related reasoning processes –, and *reactive* agents – typically characterized by sub-symbolic (such as neural networks) or imperative computational models.

So, how can the agent abstraction be exploited for modelling biological systems? As a first, obvious point, biological systems are typically characterized by a number of complex and concurrent activities. So, in biological system simulation agents can be suitably adopted for modelling at the appropriate level of abstractions such activities (tasks), or better the biological components that are responsible of them.

Also, MASs are adopted to model (non trivial) systems composed by an ensemble of autonomous entities and of the resources involved in their interaction.

Here we focus in particular on approaches that explicitly model the space of agent interaction through *mediating artifacts*, that is, those first-class entities used by agents to engage different forms of interaction (including inter-agent communication). A mediating artifact can range from a simple communication channel to a shared data structure, from a shared blackboard to a scheduler, useful for agents to synchronize their tasks. The latter ones, in particular, are examples of *coordination artifacts*, i.e. mediating artifacts providing specific coordination functionalities [27].

Thus, MASs provide the appropriate abstraction level to model a biological system in the overall: if an agent represents an individual component of the systems, the overall MAS including the mediating artifacts captures the overall set of the biological components including also the structures involved in their interaction. Mediating and coordination artifacts in particular can be adopted to model the various patterns of interaction that can be found in biological processes, many examples can be found in the KEGG Pathway database [9].

It is worth noting that the model of mediating artifacts can be crucial for creating a simulation where the overall emergent and stable behavior can be reproduced. For instance, recent works on complex system simulation [28, 29] described a case study where the correct behaviour of the simulated system could be obtained only by properly modelling knowledge structures shared by agents, which here can be framed as sorts of mediating artifacts. In particular, that research pointed out the benefits of such structures in simulations in terms of the dramatic enhancement of the probability of reproducing stable and acyclic overall behaviours. Generalising this case, we expect to have analogous benefits in adopting first class abstraction to model and control interaction in biological systems.

Finally, the *agent society* notion can be used here for defining an ensemble of agents *and* the mediating/coordination artifacts involved in the *social task* characterizing the society: a social task accounts for the coordinated execution and interaction of agent individual tasks, toward the achievement of an overall (society) objective. The notion of agent society can be suitable adopted for scaling with complexity, identifying different levels of descriptions of the same system: what can be described at one level as an individual agent, at a more detailed level can be described as a society of agents (zooming in) – so an ensemble of agents plus their mediating artifacts – and vice-versa (zooming out). These modelling features can be exploited then for simulation of biological systems involving different description levels, each one characterized by different kind of emerging phenomena.

5.2 Engineering Biological System Simulations

MAS paradigm can be effective not only for technologies to build simulations, but first of all for devising a *methodology* for covering the whole simulation engineering spectrum, from design to development, execution and runtime (dynamic) control. Critical points of biological systems – concerning structures, activities,

[9] http://www.genome.jp/kegg/pathway.html

interactions – can be captured directly by abstractions that *are kept alive from design to runtime*, supported by suitable infrastructures.

The simulation then can be framed as an online experiment, where the scientist can observe and interact dynamically with the system and its environment, both by changing its structure – by introducing for instance new agents representing biological components or removing existing ones – and global biological processes – by acting on the mediating/coordination artifacts. In particular, the *control* of mediating artifacts at runtime is the key for supporting the analytical and synthetical processes, promoting system behavior analysis and system control as defined by Kitano in [1]. In this case controlling the mediating artifacts means the possibility of *(a)* inspecting the dynamic state of the artifact; *(b)* adapting the artifacts, by changing the state and – more radically – changing its mediating behavior. On the one side, inspecting the dynamic state of system interactions is fundamental for supporting the analysis of system behavior, enabling the identification, observation and monitoring of (emerging) patterns of interaction. On the other side, the possibility of dynamically adapting the state and behaviour of the mediating artifacts can be useful, for instance, to observe system reaction – in term of stability and robustness – to unexpected events concerning the component interactions; but also to enact some forms of feedback, analogously to feedbacks in control system theory: coordination processes gluing the components can be adapted dynamically according to (unexpected) events occurring in the system or in its environment.

The capability of controlling mediating and coordination artifacts is meant to be provided both to humans (scientists and simulation engineers) and artificial cognitive agents – the latter typically provided with reasoning capabilities capable to support automatic forms of system behavior analysis.

In this overall scenario MAS *infrastructures*, and agent coordination infrastructures in particular, play a fundamental role. Agent infrastructures provide basic services to sustain the agent life-cycle, supporting dynamic agent spawning, death, (possibly) mobility and also some form of native communication services. Well-known examples are JADE [30], a FIPA-compliant platform, and RETSINA [31]. Coordination infrastructures provide instead specific services to support agent interaction and coordination [32]. These services are typically concerned with the access and management of different kind of mediating/coordination artifacts, in order to be shared and used concurrently by agents. In particular, they can provide support (tools, interfaces) for artifacts control [33].

TuCSoN is an example of MAS coordination infrastructure [34]. TuCSoN coordination artifacts are *tuple centres*, i.e. programmable tuple spaces [35]: agents interact by inserting, retrieving and reading logic tuples – information chunks structured as Prolog terms – to and from shared information spaces (tuple centres). Tuple centres can be dynamically programmed in order to react to interaction events and enact a specific coordinating behavior, described in a logic based language called ReSpecT. Finally, tuple centres are distributed among the nodes of the infrastructure, and they can be accessed and controlled both from agents (and humans) residing on the same nodes and remotely, from different Internet nodes. Recently the infrastructure has been extended to support also organization and security [36].

So, generally speaking adopting TuCSoN as infrastructure for engineering simulations (and following an agent and coordination oriented methodology like SODA [37]) involves: the identification of biological subsystems, which can be mapped into agent societies; for each society, agents – which can be developed in heterogeneous computational languages – represent biological system components and tuple centres the coordination artifacts embedding and enacting the coordination laws gluing the components. Tuple centres are exploited also for enabling weaker interaction among different societies (biological subsystems). According to the specific simulation, logic tuples can represent different kind of signals or chemical materials exchanged by the components, carrying both qualitative and quantitative information. Finally, from a topological point of view, multiple nodes can be used to distributed agents and tuple centres according to the topology of the simulated biological system.

5.3 Formal and Semiformal Language for Agent-based Systems Specification and Verification

To design systems for simulating, we need suitable models (i) to represent peculiar aspects of the biological system itself and (ii) to analyze the system from different view points, for instances: static/structural, dynamic and functional. The introduction of models to describe a biological system helps to understand of biological system itself (by identifying the system structure, critical roles and responsibilities, functions and interactions, not well identified). Of course, to create models we need languages and/or suitable notations. In the literature, we can find a wide range of formal and semi-formal languages and notations depending on the considered level, on the properties we are interested in, and on tools available to make analysis and properties verification.

Following the views classification in [38], consider three different views of a biological system:

1. the **static-structural** view of a bio-molecular knowledge as complexes, chemical, and biopolymers that participate in the system, their properties and their relationships. Several languages are already available; SBML [39] and references therein;
2. the **dynamic** view shows how the system's components behave, react to the environment and how their activities are distributed over time (control flow). The languages used to represent dynamic modelling should support sequential, parallel, conditional and iterative behaviors; Petri Nets [38, 13], UML-AD [4], SB-UML [40] and process algebra [10, 11, 41] are suitable tools for such a purpose.
3. the **functional** view that shows the functions performed by the different actors (e.g enzyme, ...) involved in the system; the substrates (input) of each function, and the product of the functions (output).

Furthermore a model should include a biological ontology that will define biological concepts and arrange them in classification hierarchies [42]. Ontologies provide consistent definitions and interpretations of biological concepts and enable software application (agents) to reuse knowledge consistently.

A possible scenario, taken into account in a previous work [4, 43], uses a semi-formal notation based on UML Activity Diagrams to describe the activities workflow describing a biological process (malaria parasites invading human host erythrocytes). The resulting description, on the one hand can be translated in a formal notation (process algebra like) to verify suitable properties as those functional and structural of the resulting system [4], and on the other hand, can be translated in a low level description (implementation) to simulate the biological process. The implementation part of the process itself makes use of agent-oriented technologies [43], to support composition, amalgamation, dynamicity and mobility.

6 Conclusion and Future Directions

In this paper, we have presented preliminary results of our investigation of how a general formal framework could be defined for biological process modelling and to support life scientists in the building and verification of their hypotheses. In the future, we will investigate the use of the proposed conceptual framework to perform modelling of real cases in systems biology.

Acknowledgement

We would like to thank Fulvio Esposito, Guido Favia, Russ Altman, Roberto Ciccocioppo for valuable speeches on malaria and pharmacogenomics process modelling.

References

1. Kitano, H.: Foundations of Systems Biology. MIT Press (2002)
2. Finkelstein, A., Hetherington, J., Li, L., Margominski, O., Saffrey, P., Seymour, R., Warner, A.: Computational challenges of systems biology. IEEE Computer **37** (2004) 26–33
3. Denti, E., Ricci, A., Rubino, R.: Integrating and orchestrating services upon an agent coordination infrastructure. In Omicini, A., Petta, P., Pitt, J., eds.: Engineering Societies in the Agents World IV. LNCS. Springer-Verlag (2004) 4th International Workshop (ESAW 2003), London,UK, 29–31 October 2003. Revised Papers.
4. Amici, R., Corradini, F., Merelli, E.: A process algebra view of coordination models with a case study in computational biology. In: Proceedings of 1st International Workshop on Coordination and Petri Nets, PNC'04. (2004)
5. Merelli, E., Luck, M.: Technical report of the 1^{st} tfg bioagent (agents in bioinformatics). Technical report, AgentLink III (2004)
6. Mendes, P.: GEPASI: a software package for modelling the dynamics, steady states and control of biochemical and other systems. Comput Appl Biosci **9** (1993) 563–571
7. Tomita, M., Hashimoto, K., Takahashi, K., Shimizu, T.S., Matsuzaki, Y., Miyoshi, F., Saito, K., Tanida, S., Yugi, K., Venter, J.C., III, C.A.H.: E-cell: software environment for whole-cell simulation. Bioinformatics (1999) 72–84

8. Kuipers, B., Shults, B.: Reasoning in logic about continuos systems, principles of knowledge representation and reasoning. In Kaufmann, M., ed.: Fourh International Conference KR94. (1994) 391–402
9. Nagasaki, M., Onami, S.: Bio-calculus: Its concept, and an application for molecular interaction. Genome Informatics (1999) 133–143
10. Regev, A., Silverman, W., Shapiro, E.: Representation and simulation of biochemical processes using the pi-calculus process algebra. In: Pacific Symposium of Biocomputing (PSB2001). (2001) 459–470
11. Regev, A.: Representation and simulation of molecular pathways in the stochastic pi-calculus. In: Proceedings of the 2nd workshop on Computation of Biochemical Pathways and Genetic Networks. (2001)
12. Chiarugi, D., Curti, M., Degano, P., Marangoni, R.: Vice: a VIrtual CEll. In: Computational Methods in Systems Biology. (2004)
13. Nagasaki, M., Doi, A., Matsuno, H., Miyano, S.: A versatile petri net based architecture for modeling and simulation of complex biological processes. Genome Informatics **16** (2004) in press.
14. Gonzalez, P., M.Cardenas, D.Camacho, A., O.Rosas, J.L.Otero: Cellulat: an agent-based intracellular signalling model. BioSystem (2003) 171–185
15. d'Inverno, M., Theise, N.D., Prophet, J.: Mathematical modelling of stem cells: a complexity primer for the stem cell biologist. In Potten, C., Watson, J., Clarke, R., , Renehan, A., eds.: Tissue Stem Cells: Biology and Applications. MIT Press (2004)
16. Regev, A., Shapiro, E.: Cell as computations. Nature (2002) 419:343
17. Zambonelli, F., Omicini, A.: Challenges and research directions in agent-oriented software engineering. Autonomous Agents and Multi-Agent Systems **9** (2004) 253–283
18. Jennings, N.: An agent-based approach for building complex software systems. Communication of ACM **44** (2001) 35–41
19. Conte, R., Edmonds, B., Moss, S., Sawyer, K.: Sociology and social theory in agent based social simulation: A symposium. Computational and Mathematical Organizational Theory **7** (2001)
20. Gilbert, N., Conte, R., eds.: Artificial Societies: the computer simulation of social life. UCL Press (1995)
21. Davidsson, P.: Agent based social simulation: A computer science view. Journal of Artificial Societies and Social Simulation **5** (2002)
22. Parunak, V.D., Savit, R., Riolo, R.L.: Agent-based modelling vs. equation based modelling: A case study and users' guide. In Sichman, J.S., Gilbert, N., Corte, R., eds.: Multi-Agent Systems and Agent-Based Simulation. Springer-Verlag (1998) 10–26
23. Jennings, N., Wooldridge, M., eds.: Agent Technololgy: Foundations, Applications, and Markets. Springer-Verlag (1998)
24. Wooldridge, M.J., Jennings, N.R.: Intelligent agents: Theory and practice. The Knowledge Engineering Review **10** (1995) 115–152
25. Jennings, N.R.: On agent based software engineering. Artificial Intelligence **117** (2000) 277–296
26. Wegner, P.: Coordination as constrained interaction. In Ciancarini, P., Hankin, C., eds.: Coordination Languages and Models – Proceedings of the 1st International Conference (COORDINATION'96). Volume 1061 of LNCS. Springer-Verlag, Cesena (I) (1996) 305–320
27. Omicini, A., Ricci, A., Viroli, M., Castelfranchi, C., Tummolini, L.: Coordination artifacts: Environment-based coordination for intelligent agents. In Jennings, N.R., Sierra, C., Sonenberg, L., Tambe, M., eds.: 3rd international Joint Conference on

Autonomous Agents and Multiagent Systems (AAMAS 2004). Volume 1., New York, USA, ACM (2004) 286–293
28. Joslyn, C., Rocha, L.M.: Towards semiotic agent-based models of socio-technical organizations. In Sarjoughian, H.e.a., ed.: AI, Simulation and Planning in High Autonomy Systems (AIS 2000), Tucson, Arizona, USA (2000) 70–79
29. Richards, D., Richards, W.A., McKey, B.: Collective choices and mutual knowledge structures. Advances in Complex Systems **1** (1998) 221–236
30. Bellifemine, F., Poggi, A., Rimassa, G.: Developing multi-agent systems with a fipa-compliant agent framework. Software Practice and Experience **31** (2001) 103–128
31. Sycara, K., Paolucci, M., van Velsen, M., Giampapa, J.: The RETSINA MAS infrastructure. Autonomous Agents and Multi-Agent Systems **7** (2003) 29–48
32. Omicini, A., Ossowski, S., Ricci, A.: Coordination infrastructures in the engineering of multiagent systems. In Bergenti, F., Gleizes, M.P., Zambonelli, F., eds.: Methodologies and Software Engineering for Agent Systems: The Agent-Oriented Software Engineering Handbook. Kluwer Academic Publishers (2004) 273–296
33. Denti, E., Omicini, A., Ricci, A.: Coordination tools for the development of agent-based systems. Applied Artificial Intelligence **16** (2002)
34. Omicini, A., Zambonelli, F.: Coordination for Internet application development. Autonomous Agents and Multi-Agent Systems **2** (1999) 251–269 Special Issue: Coordination Mechanisms for Web Agents.
35. Omicini, A., Denti, E.: From tuple spaces to tuple centres. Science of Computer Programming **41** (2001) 277–294
36. Omicini, A., Ricci, A.: Reasoning about organisation: Shaping the infrastructure. AI*IA Notizie **XVI** (2003) 7–16
37. Omicini, A.: SODA: Societies and infrastructures in the analysis and design of agent-based systems. In Ciancarini, P., Wooldridge, M.J., eds.: Agent-Oriented Software Engineering. Volume 1957 of LNCS., Springer-Verlag (2001) 185–193 1st International Workshop (AOSE 2000), Limerick, Ireland, 10 June 2000. Revised Papers.
38. Peleg, M., Yeh, I., Altman, R.: Modeling biological process using workflow and Petri nets. Bioinfomatics **18** (2002) 825–837
39. Hucka, M., Finney, A., et al., H.S.: The systems biology markup language (SBML): a medium for representation and exchange of biomedical network models. Bioinformatics **19** (2003) 524–531
40. Roux-Rouquie, M., Caritey, N., Gaubert, L., Auffray, C.: A systemic reference metamodel for semantic data integration in systems biology. In: International Conference "Pathways, Networks, and Systems: Theory and Experiments". Volume 11., Aegean Conferences, Santorini, Greece (2003) 32–34
41. Regev, A., Panina, E.M., Silverman, W., Cardelli, L., Shapiro, E.: Bioambients: An abstraction for biological compartments. Theoretical Computer Science. Special Issue on Computational Methods in Systems Biology. Elsevier (2004) To appear.
42. Gruber, T.R.: Toward principles for the design of ontologies used for knowledge sharing. Int. J. Human-Computer Studies **43** (1995) 907–928
43. Corradini, F., Mariani, L., Merelli, E.: An agent-based approach to tool integration. Journal of Software Tools Technology Transfer (2004) To appear.