ALMA MATER STUDIORUM — UNIVERSITÀ DI BOLOGNA

DEIS - Department of Electronics, Computer Science and Systems
PhD Course in Electronics, Computer Science and Telecommunications

Cycle XXIII
Disciplinary Sector: ING-INF/05

MULTI-LEVEL MODELS AND INFRASTRUCTURES FOR SIMULATING BIOLOGICAL SYSTEM DEVELOPMENT

Candidate

Dott.ssa SARA MONTAGNA

Coordinator:
Chiar.ma Prof. Ing. PAOLA MELLO

Supervisor:
Chiar.mo Prof. Ing. ANDREA OMICINI

Tutor:
Chiar.mo Prof. Ing. ANTONIO NATALI

Advisors:
Ing. ANDREA ROLI
Ing. MIRKO VIROLI

FINAL EXAMINATION YEAR 2011

To Sofia who is my life and my love.
To her father who made my dreams truth.
To my great parents and brothers for everything.

Acknowledgements

A special thank goes to my supervisors. I would like to express my deep and sincere gratitude to my supervisor Prof. Andrea Omicini because you first believed in me and gave me the chance and the privilege to begin my PhD. I am deeply grateful to Mirko Viroli and Andrea Roli, for all the important discussions we had together looking for the results we finally reached, and for your friendship.

Sara Montagna, March 2011

Contents

Abstract

The hierarchical organisation of biological systems plays a crucial role in the pattern formation of gene expression resulting from the morphogenetic processes, where autonomous internal dynamics of cells, as well as cell-to-cell interactions through membranes, are responsible for the emergent peculiar structures of the individual phenotype. Being able to reproduce the systems dynamics at different levels of such a hierarchy might be very useful for studying such a complex phenomenon of self-organisation. The idea is to model the phenomenon in terms of a large and dynamic network of compartments, where the interplay between inter-compartment and intra-compartment events determines the emergent behaviour resulting in the formation of spatial patterns.

According to these premises the thesis proposes a review of the different approaches already developed in modelling developmental biology problems, as well as the main models and infrastructures available in literature for modelling biological systems, analysing their capabilities in tackling multi-compartment / multi-level models.

The thesis then introduces a practical framework, MS-BioNET, for modelling and simulating these scenarios exploiting the potential of multi-level dynamics. This is based on *(i)* a computational model featuring networks of compartments and an enhanced model of chemical reaction addressing molecule transfer, *(ii)* a logic-oriented language to flexibly specify complex simulation scenarios, and *(iii)* a simulation engine based on the many-species/many-channels optimised version of Gillespie's direct method. The thesis finally proposes the adoption of the agent-based model as an approach capable of capture multi-level dynamics.

To overcome the problem of parameter tuning in the model, the simulators are supplied with a module for parameter optimisation. The task is defined as an optimisation problem over the parameter space in which the objective function to be minimised is the distance between the output of the simulator and a target one. The problem is tackled with a metaheuristic algorithm.

As an example of application of the MS-BioNET framework and of the agent-based model, a model of the first stages of *Drosophila Melanogaster* development is realised. The model goal is to generate the early spatial pattern of gap gene expression. The correctness of the models is shown comparing the simulation results with real data of gene expression with spatial and temporal resolution, acquired in free on-line sources.

Keywords: *Morphogenesis*, *Developmental Biology*, *Self-Organisation*, *Computational Biology*, *Multi-level Models*, *Formal Methods*, *Agent-based Model*, *Stochastic Simulation*, *Metaheuristic*, *Parameter Optimisation.*

1 Introduction

1.1 Motivation and Research Context

Developmental biology is an interesting branch of life science that studies the process by which organisms develop, focussing on the genetic control of cell growth, differentiation and movement. A main problem in developmental biology is understanding the mechanisms that make the process of vertebrates' embryo regionalisation so robust, making it possible that from one cell (the zygote), the organism evolves acquiring the same morphologies each time. This phenomenon involves at the same time the dynamics of – at least – two levels including both cell-to-cell communication and intracellular phenomena: they work together, and influence each other in the formation of complex and elaborate patterns that are peculiar to the individual phenotype. How do local interaction among cells and inside cells gives rise to the emergent self-organised patterns that are observable at the system level? This happens according to the principles of *downward* and *upward* causation, where the behavior of the parts (down) is determined by the behavior of the whole (up), and the emergent behaviour of the whole is determined by the behaviour of the parts [UDZ05].

Given the fact that the overall dynamics underlying these phenomena is extremely complex and very few biological data are readily available, the help of modelling techniques seems to acquire more and more importance—and this holds true especially when mixing together different embryogenetic mechanisms and their relations. These scenarios require tools that can support multi-scale models, where different cells form large-scale, dynamic networked systems – as e.g. in tissues of cells, organs, and even full embryos – and where both the biochemical reactions that occur inside each cell and the molecules diffusion across membrane (mediating the interaction among cells) can be captured.

Modelling embryo- and morphogenesis presents big challenges: *(i)* there is lack of biological understanding of how intracellular networks affect multicellular development and of rigorous methods for simplifying the correspondent biological complexity: these make model definition a very hard task; *(ii)* there is a significant lack of multi-level models of vertebrate development that capture spatial and temporal cells differentiation and the consequent heterogeneity in these 4 dimensions; *(iii)* on the computational framework side there is a need for tools able to integrate dynamics at different hierarchical levels and spatial and temporal scales, and to simulate across them.

Such scenarios have already been addressed with different approaches, including mathematical and computational ones. Mathematical models, on the one side, are continuous and use differential equations, in particular partial differential equations which describe how the concentration of molecules varies in time and space. A main example is the reaction-diffusion model developed in Tur52 and applied to the *Drosophila* development in PJRG06. The main drawback of mathematical models is the incapability of realising multi-level model so that to reproduce dynamics at different levels.

Computational models, on the other side, are discrete and model individual entities of the system (cells, proteins, genes). Formal methods, such as different Calculi and Petri Nets, are rarely be used for modelling morphogenesis. SPIM or Bio-PEPA [Phi06, CDG09] for instance, although can in principle tackle such large network scenarios, would require additional tools (e.g. for automatic code creation) to make modelling and simulation more practical.

The agent-based approach (ABM) is an other example of computational approach that can be used to explicitly model a set of entities with a complex internal behaviour and which interact with the others and with the environment generating an emergent behaviour representing the system dynamics. It is widely used for modelling complex systems in general. Some work has already been done in the application of ABMs in morphogenesis-like scenarios. A good review is proposed in TBDP08. Most of these models generate artificial pattern – French and Japanese flags [BMF06] – realising bio-inspired models of multicellular development in order to obtain predefined spatial structures. But at the best of our knowledge there are not many results already obtained in the application of ABM for analysing real phenomena of morphogenesis.

Finally also hybrid frameworks have been developed, such as COMPUCELL 3D [CHC$^+$05] which combine discrete methods based on cellular-automata to model cell interactions and continuous model based on reaction-diffusion equation to model chemical diffusion. COMPUCELL 3D looks a very promising framework whose main limitation is the lack of a suitable model for cell internal behaviour, gene regulatory network in particular.

1.2 Overview and Contribution

According to the above premises, the work presented in this thesis is an effort in the direction of finding suitable models for developmental biology research.

To this end, the first contribution of this thesis is given by a computational model and related framework, MS-BioNET, developed for modelling and then simulating large networks of biological compartments (e.g. several hundreds of cells), as required by the study of phenomena like morphogenesis and embryogenesis. On top of the framework, a logic-oriented specification language is used to flexibly specify simulation scenarios: on the one side it tightly focusses on biochemistry, by providing constructs to directly express biochemical reactions, compartments, compartment link topology, and reactions involving selective transfer through membranes; on the other side it relies on logic-based goal resolution and unification, achieving the expressiveness needed to easily handle size and complexity of the biochemical network. Behind the hood, such a specification is turned into an intermediate language (a sort of bytecode) that feeds a simulation engine implemented by adapting the optimised version (described in [GB00]) of Gillespie's stochastic simulation algorithm [Gil77] to our computational model of biochemical cell networks.

To show the applicability of this framework a scenario of morphogenesis in embryo formation is discussed, where I conceive a biochemical system that – by the interplay of intra-compartment chemical reactions and inter-compartment chemical transfer – manifests the ability of regionalising a tissue of cells. In particular the *Drosophila Melanogaster*'s embryo development is modelled, with the goal of reproducing the gene regulatory network that causes the early (stripes-like) regionalisation of gene expression in the anteroposterior axis [YN08, PJRG06].

The same scenario is modelled with ABMs, providing a model for capturing morphogenesis in general.

Working with models of such a complexity for the number of actors and of interactions involved gives rise to a further problem to face: the *parameter tuning*. Given the model structure and a set of target data, the goal is to find the values for model parameters so as to reproduce the system behaviour. As in general it is not possible to capture the influence of parameters by theoretical models, this is in fact a resource-intensive task that requires many repeated simulation-analysis runs and the introduction of ad-hoc optimisation algorithms. To this aim, the task is in this thesis formulated as an optimisation problem. The optimisation module makes use of *metaheuristics* to find a parameter configuration such that the simulated system has the desired behaviour. Metaheuristic algorithms combine diverse concepts for exploring the search space and they also apply learning strategies in order to find (near-)optimal solutions efficiently [BR03]. Metaheuristics are successfully applied to optimisation problems since decades and are particularly effective in tackling problems in which the objective function is rather complex or even an approximation of the actual optimisation criterion. Other works have been published which bring support to the important contribute that the use of metaheuristics can give for solving this optimisation problem [RFEB06, Ban08], but they mainly refer to mathematical and deterministic models. In order to nimbly perform the studies on the biological system, a framework has been built which integrates a *simulator* executing the model, an *evaluator* estimating the quality of the simulation results, and an *optimiser* finding an optimal parameter setting. Such a framework has been used for identifying the parameters of the MS-BioNET and ABM models.

As an additional contribution the thesis proposes a recent work within the FP7 project SAPERE (Self-Aware Pervasive Service Ecosystems)[1], FET Program of the EU. The work is inspired to the studies on morphogenesis described so far and exploits the possibility of using the basic mechanisms of morphogenesis for designing Pervasive Systems given some similarity between the two processes: they both are in fact intrinsically distributed in space and composed of a network of interacting entities which also sense their physical environment and accordingly

[1] `http://www.sapere-project.eu/`

adapt their behaviour.

1.3 Organisation of the Thesis

The remainder of this thesis is organised as follows.

Part I: Background

Chapter 2 - The Biological Domain This chapter provides the necessary background to understand the motivation of this work and the case study used in the rest of thesis. In particular it defines the biological domain of Developmental Biology and it gives a description of the main mechanisms involved into the embryogenesis and morphogenesis processes. It finally describes the *Drosophila Melanogaster* morphogenesis, introducing the main actors involved and their role into development.

Chapter 3 - Computational Systems Biology The meaning of Computational Systems Biology (CSB) as well as its role in biological sciences is here presented. An overview of the models and simulation techniques developed in CSB is then given so as to provide an idea of related work and to contextualise the idea and tools described in this thesis.

Chapter 4 - Models of Development Different models of development found in literature are cited in this chapter and briefly described.

Part II: Motivation

Chapter 5 - Towards Multi-level Models The role of hierarchy in the spatial self-organisation of gene expression during morphogenesis is here highlighted. The characteristics of a model able to capture such a property are then illustrated.

Part III: Contribution

Chapter 6 - MS-BioNET and the Application at Drosophila Development This chapter presents – in terms of a stochastic calculus – the formal computational model, the logic specification language and the simulator engine that grounds the developed framework MS-BioNET. A model of the pattern formation in the *Drosophila Melanogaster* embryo is then described.

Chapter 7 - ABM and the Application at Drosophila Development The agent-based approach is here presented with the modelling abstractions it provides. The ABM of *Drosophila melanogaster* morphogenesis is then fully described, showing how cell internal behaviour, cell interacting capability and environment dynamic can be modelled with ABM, a possible model of morphogenesis in general.

Chapter 8 - The Parameter Optimisation Module In this chapter the role of metaheuristics in parameter tuning is first explained, then the general framework architecture is illustrated. It includes the model/simulator module together with a module for evaluating the simulation results and a module for optimise the solution. Different optimisation techniques evaluated are then described.

Part IV: Experiments

Chapter 9 - Simulation and Results Conditions and results of the simulations performed with MS-BioNET and ABM are reported. Results are compared with experimental data of *Drosophila melanogaster* gene expression acquired from a free on-line source.

Part V: Further Application

Chapter 10 - Towards the Design of Pervasive Systems In this chapter a preliminary work done in the direction of designing Pervasive Systems is described, adopting as inspiration the main mechanisms of morphogenesis, so to realise computational systems able to self-organise in elaborated structures.

Part VI: Conclusion

Chapter 11 - Conclusion and Upcoming Work This chapter provides concluding remarks and discusses future work.

Part I

Background

2
The Biological Domain

The biological domain in which this thesis is placed is the *Developmental biology*, one of the most challenging fields in biology [Gil06, Dav05, AJL^{+}02].

Developmental biology studies how a multicellular organism develops and grows, from the early forms of zygote, larva, embryo, into an adult. It investigates the mechanisms that control cell growth, differentiation and morphogenesis. Therefore the main questions of Developmental biology are:

1. Which are the processes and the genetic mechanisms by which cells grow with the process of duplication, *i.e.*, doing mitosis;

2. Which are the processes and the genetic mechanisms by which cells differentiate, *i.e.*, become of different types by expressing different sets of genes:

3. How it is possible that cellular differentiation is spatially organised, *i.e.*, each cell type sort out into its region, in a robust and precise way creating patterns characteristic of each organism, that is, making *morphogenesis*.

Developmental Biology is a great field for scientists who wants to integrate different levels of biology. We can take a problem and study it on the molecular and chemical levels; on the cellular and tissue levels; on the organ and organ-system levels; and even at the ecological and evolutionary levels.

Emergence is one of the most important concepts in the study of morphogenesis, and indeed in developmental biology. The aspect of emergence that is most relevant to understand embryonic development is that very complex behaviours can emerge from the action of very simple operations, and very complex forms can emerge from the action of comparatively simple machines.

2.1 The Question of Developmental Biology

The study of animal development between fertilisation and birth has traditionally been called *embryology*. The steps of the process of development of a multicellular organism can be summarised as follow: it begins with a single cell – the fertilised egg, or zygote – which divides mitotically to produce all the cells of the body. The resulting blob of cells starts the process of differentiation which is caused by the presence of unique factors, called *maternal factors*: they are products of the maternal genome and are located in specific areas of the organism's egg. The spatial organisation of this diversity is then caused by the interactions among cells. After that tissues have been created, and the formation of organs begins so to originate the final shape of the organism. Details of the overall process are given in Section 2.2.

The question of differentiation

A single cell, the fertilised egg also called the *zygote*, gives rise to a lot of different cell types—muscle cells, epidermal cells, neurons, blood cells and so on. The process which generate cellular diversity is called *differentiation*. Since every cell of the body (with very few exception) contains the same genome, how can this set of genetic instructions produce cells of different types? How can a single cell, the fertilised egg, generate so many different types of cells?

The question of morphogenesis

Differentiated cells are spatially distributed and organised into a precise pattern which is at the basis of tissues and organs generation. During development different processes happen, such as cell migration, division an death. Tissues fold and separate. But the resulting organs are precisely and robustly formed. The overall phenomenon which create such a spectacular order is called *morphogenesis*. How can the differentiated cells form such organised structure? How do migrating cells follow the correct direction to reach their destination?

The question of growth

Cells divide a precise number of time: If each cell would undergo just one more cell division, the resulting organ would not have the same shape so that the organ would not be able to carry on its function. How do our cells know when to stop dividing? How is cell division regulated?

These are the main questions of developmental biology whose amazing complexity requires decades of studies, efforts, experiments and analysis.

2.2 Stages of Animal Development

Each animal passes through similar stages of development.

Fertilisation The life of animals begins with the fertilisation of the egg—the fusion of gametes (sperm and ovum) to produce a new organism. The fertilised egg cell is known as the zygote. The egg cell is always asymmetric, *i.e.*, the distribution of maternal proteins inside is not uniform.

Cleavage The fertilisation stimulates the egg to begin development. Cleavage is a series of extremely rapid and synchronous divisions with no significant growth, producing a cluster of cells usually formed a sphere – called *blastula* – that is almost the same size as the original zygote. The end of cleavage is known as midblastula transition and coincides with the beginning of zygotic transcription, *i.e.* cells start the process of gene transcription and of protein synthesis.

Gastrulation During gastrulation the rate of mitotic division slows down and the cells undergo dramatic movements and change their positions and neighbourhood. The embryo is in the gastrula stage. As a result of gastrulation the embryo contains three germ layers: the ectoderm, the mesoderm and the endoderm. The cell that will form the endodermal and mesodermal organs are brought to the inside of the embryo, while the cells that will form the skin and nervous system are spread over its outside surface.

Organogenesis Once the three germ layers are established, the cells interact with one another and rearrange themselves to produce tissues and organs. In most animals organogenesis along with morphogenesis will result in a larva. The hatching of the larva, which must then undergo metamorphosis, marks the end of embryonic development.

2.3 The Biology of Development

2.3.1 Cell division

During the initial phase of development, when cleavage rhythms are controlled by maternal factors, the cytoplasmatic volume does not increase. Rather the enormous volume of zygote cytoplasm is divided into increasingly smaller cells. First the zygote is divided in half, then quarters, then eighths, and so forth. This division of cytoplasm without increasing its volume is accomplished by abolishing the gap periods of the cell cycle (the G_1 and G_2 phases), when growth can occur. So blastomeres generally progress through a biphasic cell cycle consisting of just two steps. Meanwhile nuclear division occurs at a rapid rate never seen again.

The regulation of cell cycle during cleavage is mainly due to factors stored in the egg cytoplasm. Therefore, the cell cycle remains independent of the nuclear genome for a number of cell divisions. These early division are rapid and synchronous. However, as the cytoplasmic components are used up, the nucleus begins to synthesise the. The embryo enters a mid-balstula transition, in which several phenomena are added to the biphasic cell divisions of the embryo. First the gap stages are added. Second the synchronicity of cell division is lost. Third, new mRNAs are transcribed.

2.3.2 Cell migration

Cell migration takes place at least at some point in the life cycle of almost all animals. It is involved in the morphogenesis of most parts of the body and it is used for the translocation of cells from the place of their birth to the place of their final use. Cell migration is an almost universal attribute of sexual reproduction, for the obvious reason that two gametes produced in different places and usually in different individuals have to unit but there are other spectacular example such as the development of the vertebrate neural crest.

The sorting of cells is also due to migrations of cells produced by chemotactic response, adhesion and motility.

Chemotaxis

Via a process called chemotaxis [EBLB10], living cells are able to communicate emitting chemicals and responding to chemicals released by other cells that diffuse into the environment. Chemotaxis is called positive if movement is in the direction of a higher concentration of the chemical in question, and negative if the direction is opposite

Cells must be therefore able to sense local chemoattractant concentrations.

The chemotactic gradient is normally generated from a localised source where the chemoattractant is produced and then allowing it to diffuse away. The rate of diffusion is determined by the chemoattractant's diffusion constant and by the local concentration gradient. The change of concentration over time at any point in the field is given by the Fick's law, which state for one-dimensional systems that

$$\frac{dm}{dt} = -D(\frac{d^2m}{dx^2}) \tag{2.1}$$

where m is the local concentration of the molecule and D its diffusion constant. Real chemotactic field also involve a sink that destroys the chemoattractant. It might be localised or distributed for example as an enzyme activity scattered homogeneously throughout the field.

Chemotaxis does not always involve chemoattraction, and some morphogenetic events depend on the ability of tissues to repel specific migratory cells.

Cells can move also in response of electric fields.

Cell adhesion

Cell migration is also determined by the adhesion phenomenon. Cells do not sort randomly rather can move to self-organise in tissues. Somehow cells are able to sort out into their proper embryonic position.

Some evidences suggest that cells interact so as to form an aggregate with the smallest interfacial free energy, *i.e.*, a thermodynamically stable pattern. According to this hypothesis an embryo can be viewed as existing in an equilibrium state until some changes in gene activity changes the cell surface molecules.

Other evidences suggest that many of the answer regarding morphogenesis involve the properties of the cell surface. Each type of cell has a different set of proteins in its cell membrane, and some of these differences are responsible for the formation of tissues. Selective affinity is the name given to the phenomenon which emerges from this protein diversity. It can changes during development resulting in a not stable relationship with othe cell types. Different cell types can have a positive affinity between each other or a negative one.

The major cell adhesion molecules appear to be the *cadherins*. They are crucial for connecting cells together and maintaining them closely connected, and they appear to be critical to the spatial segregation of cell types. Cadherins join cells together by binding cadherins on another cell. Different cell types synthesise different cadherins and / or different amounts of the same cadherins. Cells that express more cadherins have a higher surface cohesion and migrate internally to the lower-expressing group of cells. Moreover the surface tension of these aggregates are linearly related to the amount of cadherin they are expressing on surface. Finally both quantity and type of cadherins expressed determine which cells sort out from one another.

In [Ste07] it is proposed the theory of differential adhesion hypothesis (DAH) according to which cell populations tend to maximise the strength of mutual binding within them, minimising the adhesive free energy of the system.

Selective adhesion in embryonic cells allows the cells to move and self-organise into the patterns that develop into tissues and organs.

2.3.3 Cell differentiation

A specialised cell is called differentiate, and the process by which a cell become differentiate is called *differentiation*. Each cell of an organism normally owns an identical genome; thus the differentiation process is *not* due to different genetic information, but to diverse gene expression in each cell. The set of genes expressed in a cell controls cell proliferation, specialisation, interactions and movement, hence it corresponds to a specific cell behaviour and role in the entire embryo development. The commitment of the cell into a certain fate is not immediate, rather it happens in two main stages: the specification and determination processes. The fate of a cell is specified if the cell is capable of differentiate even if it is placed in neutral environment, but it is still reversible. A cell is determined when it differentiates according to its original fate even if placed in non neutral different environment that normally drives the cells into different fates. This stage is irreversible.

One possible way for creating cells diversity during embryogenesis is to expose them to different environmental conditions, normally generated by signals from other cells, either by cell-to-cell contact, or mediated by cues that travel in the environment.

In the first case only adjacent cells are involved. The most common phenomenon is *lateral inhibition* where initially similar cells start a competition that drives them to become different from one another: one cell or a group of cells emerges as the “winner”, specialising into one cell type and inhibiting the neighbouring cells from doing likewise. Lateral inhibition is therefore a process by which precise patterns of distinct cell types are generated [AJL$^+$02, GR92].

Another strategy for driving cells toward a specific fate is *inductive interaction*, where cells are pushed into a developmental line through signal molecules that can act either as short-range inducers – normally via cell-to-cell contacts – or long-range inducer—mediated by molecules that diffuse in the extracellular medium. Such molecular inducers are also called *morphogens*.

Morphogens are not uniformly distributed along the embryo. Morphogens are synthesised in specific region of the embryo, deposited in the environment, and then diffuse over long distances and form concentration gradients where the higher concentration is at the point of synthesis and gets lower as the morphogene diffuses away from its sources and degrade over time.

On the side of intracellular dynamics, signalling pathways and gene regulatory networks are the means to achieve cell diversity. Signalling pathways are the ways through which an external signal is converted into information travelling inside the cell and, in most cases, affecting the expression of one or more target genes. The signalling pathways are activated as a consequence of binding between *(i)* a cue in the environment and a receptor in the cell's membrane or *(ii)* two membrane proteins belonging to different cells. The binding causes the activation of downstream proteins until a transcription factor that activates or inhibits the expression of target genes is produced.

During embryogenesis or morphogenesis few pathways are active. They work either as mutual inhibitors, or as mutual enhancers. The idea is that there are regions where the mutual enhancers are active and interact giving rise to positive feedback. Pathways active in different regions generally work as mutual inhibitors. Then, there are boundary regions where we can observe a gradient of activity of the different sets of pathways, due to the inhibitory effect of the pathways belonging to neighbouring regions.

An other possible classification for differentiation is syncitial spetification and conditional specification.

Conditional specification

This process depends on stimulatory or inhibitory interactions that either take place among neighbouring cells through membrane proteins or are mediated by the gradients of signals in the environment. Interaction of a cell with other cells restrict the fate of one or more of the participant: the fate of a cell depends upon the conditions in which the cell finds itself. In numerous cases specific types of proteins are secreted by adjacent cells to instruct the target cell as to its fate. In other cases, a given cell is committed to become one type of cell if it receives a certain protein in a specific concentration, but can be committed in a different direction if it receives the same protein but at a higher or lower concentration.

Syncytial specification

In early embryo of the insects, such as *Drosophila Melanogaster* cell membranes do not surround nuclei which divide within the egg cytoplasm creating many nuclei within one large egg cell—a syncitium. The egg cytoplasm is not homogeneous and the posterior part is differently composed then the anterior one. This method involves the action of particular cue gradients within the syncytium. As there are no cell boundaries in the syncytium, these signals can influence nuclei in a concentration-dependent manner.

In the first case morphogens are secreted by a set of cells and received by other nearby cells. In the latter case the diffusion of morphogens happens within the cytoplasm of a single cell.

2.3.4 The Role of the Environment

The developing embryo is not isolated from its environment: environmental cues are a fundamental part of the organism's life cycle. In many instances it was found that the genome did not necessarily encode a particular phenotype rather it encoded the information for a repertoire of possible phenotypes. The environment was then the determining factor for the reaction expressed.

2.3.5 Pattern formation

Self-assembly is one of the simplest possible mechanisms for morphogenesis. Its defining feature is that the components that assemble together contain, in their shapes and binding characteristics, sufficient information to determine the structure they produce: there is no need for any special prior spacial arrangements.

Self-assembly requires that the coming together of components in right way is not too improbable. This partly reflects the energy requirements of the process (particularly is the transition of a subunit from an unbound state to a bound one involves the crossing of an energy barrier).

Changes in cell shape that occur during development drive the morphogenesis of tissues.

2.4 On Drosophila Melanogaster Development

Drosophila is one of the best known multicellular organism. The egg of *Drosophila* is about 0.5 mm long and 0.15 mm in diameter. It is already polarised by differently localised mRNA molecules which are called *maternal effects*. In this section I first discuss the main stages during embryogenesis, and then I detail the genetics of *Drosophila* development as we have come to understand it over the past two decades. In particular I will focus on how the segments of gene expressions are formed along the anterior-posterior axis.

Cleavage

The early nuclear divisions are synchronous and fast (about every 8 minutes): the first nine divisions generate a set of nuclei, most of which move from the middle of the egg towards the surface, where they form a monolayer called *syncytial blastoderm*. All the dividing nuclei share

Cleavage Cycle	Length [min]
1 → 10	8
11	10
12	10
13	25
14	75 → 175

Table 2.1: Cleavage cycle length

a common cytoplasm, and material can diffuse throughout the embryo. After other four nuclear divisions, after the thirteen nuclear division, plasma membranes grow to enclose each nucleus, converting the syncytial blastoderm into a *cellular blastoderm* consisting of about 6000 separate cells. After the first ten divisions the time required to complete each of the next four divisions becomes progressively longer: cycle 13 takes for instance 25 minutes. Cycle 14 in which the cells are formed is asynchronous and cells conclude their mitosis in rather different time: some cells take 75 minutes, other 175 minutes to complete this cycle. The rate of division is then constant in the first hours of development (9.05 min^{-1}), then decreases until a low value (0.2 min^{-1}), as it appears in Figure 2.2. The transcription of RNA massively begins during cleavage cycle 14 so that the embryo enters in the mid-blastula transition.

In Figure 2.1 is shown how the number of cells varies in the first four hours of development, and a graphical representation of each cleavage cycle length is given. In Table 2.1 is given a numeric view of such temporal division. The numbers provided are not to be consider exact as soon as slightly different values are given from different literature sources [PPSR09, Gil06, AJL$^+$02, WBB$^+$98]. What I propose here is an average of them.

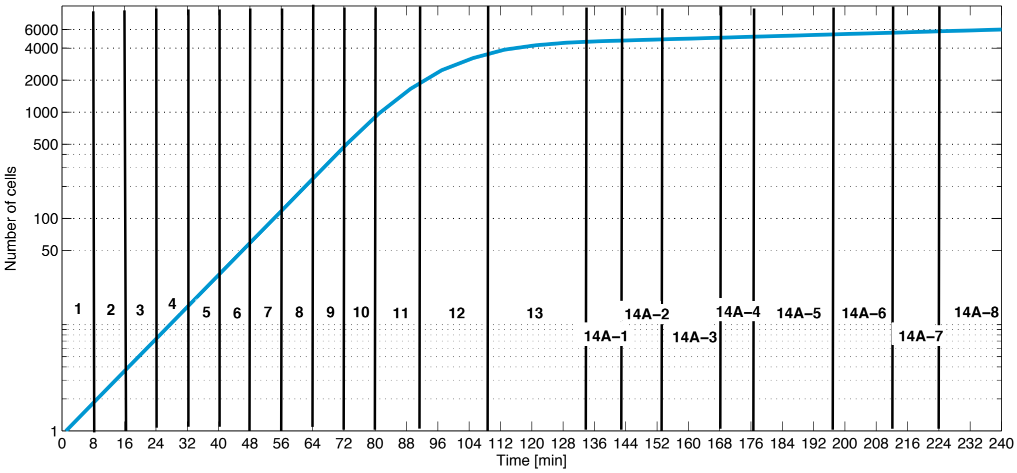

Figure 2.1: Number of cells varying from one to 6000 in the first 14 cleavage cycles

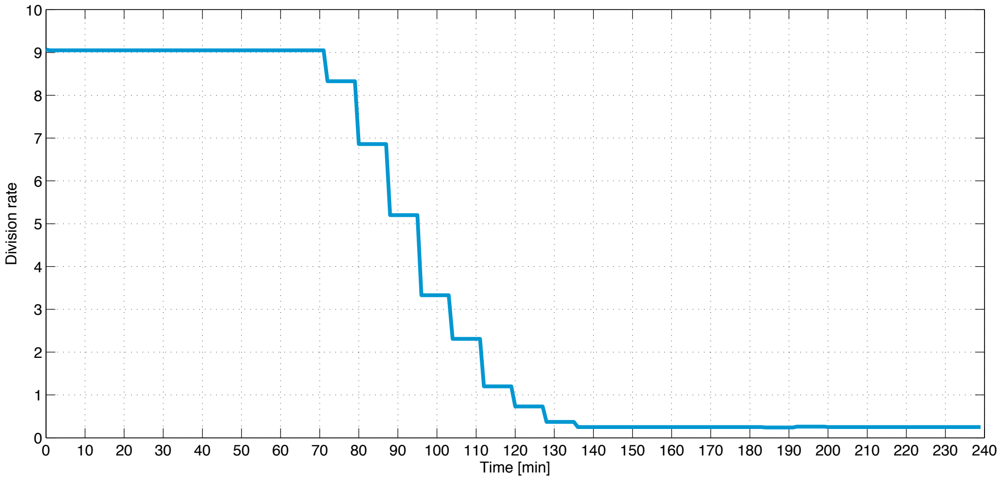

Figure 2.2: Rate of division in the first 14 cleavage cycles

Up to the cellular blastoderm stage, development depends largely -although not exclusively– on maternal mRNAs and proteins that are deposited in the egg before fertilisation. After cellularisation, cell division continues asynchronously and at a slower rate, and the transcription increases dramatically. Once cellularisation is completed the gene expression regionalisation is already observable.

Gastrulation

With the blastoderm stage the territories of embryos are defined. With gastrulation morphogenetic movements are orchestrated to bring the cells into their final positions inside the embryo. Gastrulation begins shortly after the mid-blastula transition. Gastrulation is the process that transforms a blastula or a blastoderm into a multilayered embryo with three germ layers. The first movements of Drosophila grastrulation segregate the embryo into three germ layers: mesoderm, endoderm, and ectoderm. The choreography of gastrulation movements is determined by region-specific transcription factors which turn on a set of downstream targets whose products mediate the successive steps of gastrulation. These transcription factors pre-exist in the embryo, being supplied to the egg by the mother, while others are synthesised at the time when gastrulation begins, specifically in those cells that initiate gastrulation. For a review of the molecular mechanisms regulating such process see [Lep99].

One of the earliest events in the development of a multicellular organism is the process of gastrulation, i.e. the segregation of the primordia of the future internal tissues, the mesoderm and the endoderm, into the interior of the developing embryo. For gastrulation to occur, the territories to be internalized must first be defined, and morphogenetic movements then orchestrated to bring the cells into their final positions inside the embryo. Gastrulation thus involves both pattern formation and morphogenesis, as well as the coordination of cell behaviour. Not surprisingly, molecules known to be important for gastrulation range from transcription factors through cytoskeletal components to signalling molecules. Some of these pre-exist in the embryo, being supplied to the egg by the mother, while others are synthesized at the time when gastrulation begins, specifically in those cells that initiate gastrulation.

2.4.1 Setting up the body axes

The insect body is bilaterally symmetrical. Organisation along the antero-posterior and dorso-ventral axes of the early embryo develops more or less simultaneously, but is specified by independent mechanisms and by different sets of genes in each axis.

The early development of *Drosophila* is peculiar to insect, as patterning occurs within a multinucleate syncytial blastoderm. Only after beginning of segmentation does the embryo become truly multicellular. At the syncytial stage of segmentation many proteins, including those that are not normally secreted from cells such as transcription factors, can diffuse throughout the blastoderm and enter other nuclei. Concentration gradients of transcription factors can thus set up in the syncitial blastoderm.

Genes that pattern the anterior-posterior axis of the Drosophila body plan

The studies on the genetics at the basis of the segmentation in the anterior-posterior body plan identified a hierarchy of genes that establish anterior-posterior polarity and divide the embryo into a specific number of segments with different identities, as shown in Figure 2.3.

The building blocks of anterior-posterior axis patterning are laid out during egg formation thanks to the maternal effects. The maternal effect genes that are most important for patterning the anterior-posterior plan of the embryo in this early stage are *bicoid* and *hunchback* – forming an anterior-to-posterior gradient – and *nanos* and *caudal* —forming a posterior-to anterior gradient. The mRNAs of such genes are placed in different regions of the egg and initiate the hierarchy. They are transcription factors that drive the expression of *gap genes*, which are the first zygotic genes to be expressed, such as *hunchback* (hb), *Krüppel* (Kr), *knirps* (kni) and *giant* (gt). These genes are expressed in certain broad (about three segments wide, partially overlapping domains. Differing combinations and concentrations of the gap gene proteins then regulate the expression of downstream targets, *i.e.*, the *pair-rule genes*, which divide the embryo into periodic units. The transcription of the different pair-rule genes results in a striped pattern of seven transverse bands perpendicular to the anterior-posterior axis. The most important pair-rule genes are *even-skipped* (eve) and *fushi-tarazu* (ftz). The pair-rule gene proteins activate the transcription of the *segment polarity genes*, whose mRNA and protein products specify 14 parasegments that are closely related to the final anatomical segments [AJL$^+$02].

Gene regulatory network

Early in development, the fate of a cell depends on cues provided by protein gradients. This specification of cell fate is flexible and can still be altered in response to signals from other

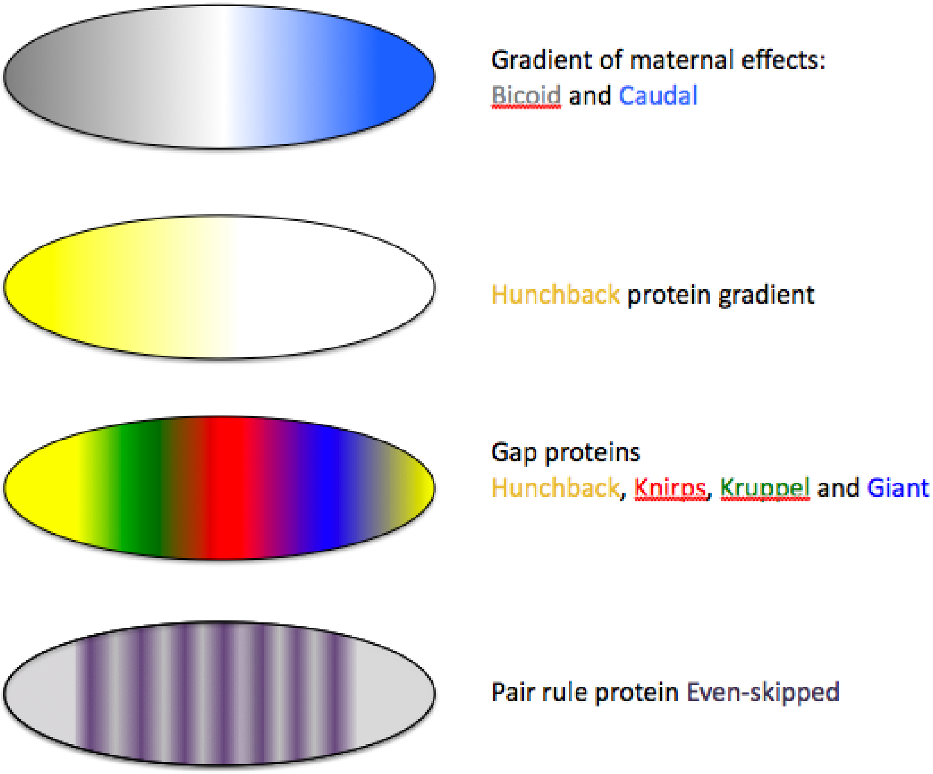
Gradient of maternal effects:
Bicoid and Caudal
Hunchback protein gradient
Gap proteins
Hunchback, Knirps, Kruppel and Giant
Pair rule protein Even-skipped

Figure 2.3: Hierarchy if genes establishing the anterior-posterior body plan

cells. Eventually, however, the cell undergo a transition from this loose type of commitment to an irreversible determination. The transition from specification to determination is mediated in *Drosophila* by the segmentation genes (gap, pair-rule, segment polarity genes).

In Figure 2.4 is shown the network of interactions among maternal effectors and gap genes. An other gap gene, *tailess* (tll), also appears as input of the network whose regulation is not clear and we do not represent in the remind of the work.

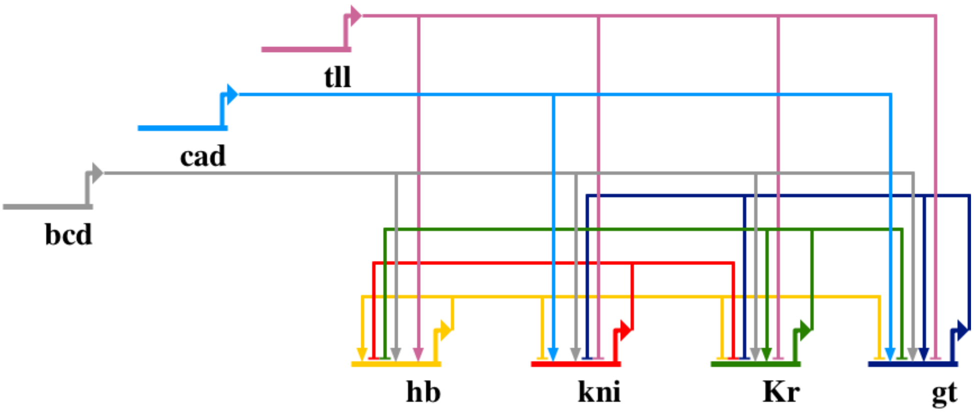

Figure 2.4: Gene regulatory relationships as in the model presented in [PJRG06, RPJ96, GJK$^+$04]. The diagram is realised with BioTapestry [LDB09]. The type of link is pointed out by its shape: links with arrowhead are enhancers, while the repressor links have a foot.

3
Computational Systems Biology

As extensively reported in Chapter 2, the ultimate goal of developmental biology is to understand biological systems during development in sufficient detail to enable accurate, quantitative predictions about their behaviour, including predictions on the effects of modifications of the systems. This is a big challenge, given the complex properties of biological systems.

Whereas the experimental techniques of developmental and molecular biology fail, modelling and simulating tools have been introduces in the last decades. In this Chapter a review of the main works developed is proposed within a classification which distinguish the models into mathematical and computational ones.

3.1 On the role of models in biology

3.1.1 Molecular biology

With the successes obtained by the genetics during the 1950s, which brought at the understanding of the physical structure of DNA, the biological perspective significantly shifted: whereas before cells were regarded as the basic building-blocks of living systems, the attention shifted from cells to molecules. Advancing in their explorations of the phenomena of life, biologists understood the role of chromosomes whose code determine the characteristic of the living being and they codified the chromosomes code.

The edges of molecular biology

This triumph of molecular biology resulted in the belief that all biological functions can be explained in terms of molecular structures and mechanism. Thus most biologist have become fervent reductionists, concerned with molecular detailed. At the same time, the problems that resist the reductionistic approach of molecular biology become ever more apparent: while biologist know the precise structure of a few gene, they know very little about the ways in which genes communicate and cooperate. This means that, while molecular biology made progresses in understanding the structures and functions of many of the cell's subunits, it remained largely ignorant about the coordinating activities that integrate those operations into the functioning of the cell as a whole. Biological systems, in fact, have obvious both structure and organizational principles, and their behaviour cannot be understood either by "reading the DNA" (even though in principle all the information is there) or by studying the biological components one by one or one level at a time. In few summarizing world: "*the whole is more then the sum of the parts*".

At now, even if the molecular biologists have been unravelling the functions of cellular components and networks, and the amount of molecular-level knowledge accumulated so far is absolutely amazing, yet we cannot say that we understand how a cell work: the process of understanding cellular components is far from finished, but it is becoming clear that simply obtaining a full part list will not tell us how a cell works.

The complexity barrier between components and systems prevents us from predicting the behaviour of biological systems, and therefore from repairing them reliably. New concepts and new tools are clearly necessary to describe nature, tools for modelling and simulating biological complex systems. It enters in this context *Systems biology*, and later on *Computational systems biology*, emergent disciplines with the goal of understanding biological systems as a whole.

3.1.2 Computational systems biology

Around the year 2000, Systems biology emerged as a movement in its own right with the pioneer work of Kitano [Kit02b]. The goal of Systems biology is to achieve a systems-level understanding of biological processes and ultimatively whole cells and organisms: from the huge amounts of data that biologists collected, Systems biology is building a science of the principles of operation of biological systems, based on the *integration and interaction between components*, *i.e.*, on that interactions which are ultimately responsible for an organism's form and functions. It is so a discipline that, instead of analysing individual components or aspects of an organism, focuses on all the components, and on the interactions among them, all as part of one system. To address the question, Systems biology chooses modelling methods, which are implemented and then simulated through computational tools. Simulation is the process of using a developed model to analyse and predict the behaviour of the original system, doing experiment with this model. Because off the use of computational technique, in both modelling and simulating phase, Systems biology is often called *in-silico* biology.

Although Systems biology believes that the essence of systems lies in dynamics and it cannot be described merely by enumerating components of the system, at the same time it does not believe that only systems dynamic and structure is important without paying sufficient attention to diversities and functionalities of the structure of the components. It means that modelling and experimental techniques has to work in conjunction for integrating system dynamic with functions and data.

Therefore Systems biology is a new way of doing biology, starting with experimental knowledge, passing through modelling, and finally returning to biological experiments with the simulated results: it is so an approach that works if integrated with experimental biology.

With the introduction of computational models, theories and tools in the discipline of systems biology the field has been more generally called *Computational systems biology* [Kit02a] or *Computational biology* as an “interdisciplinary field that applies the principles and techniques of computer science, chemistry, applied mathematics, statistics and engineering to address biological problems”. Currently there is not a perfect agreement on the boundaries between the classical systems biology and computational systems biology. As soon as in this work we will present a computational model hereafter I will refer to Computational systems biology (CSB) as the reference field.

3.2 On the concept of model

A model is an abstract representation, a schematic description of a system, theory, or phenomenon, that allows the investigation of the properties of the system and, in some cases, prediction of future outcome or studies of its characteristics. It is usually in the form of a set of objects and the relations among them.

Models normally represent only a portion of the biological phenomena of interest, including a subset of mechanisms opportunely chosen for the goal of the simulations. *It is a skeleton, but not a replica of the real system*, built with key components based on a mix of assumptions and known knowledge. It involves simplification, aggregation and omissions of details. The key for modelling is then to identify the elements that can reflect global properties with incomplete information. This is one of the hardest task when designing a model.

3.3 An overview of modelling approaches

Modelling lies at the hart of CSB. They can be used at different levels of the biological scales, from DNA and gene expressions to intracellular networks, to cell-to-cell and transmembrane signals, and through to the organ level. Models developed in the discipline of CSB might be classified first of all into the two categories of *Mathematical* and *Computational* Models [FH07]. [BB01, SSP06, DJ02] are comprehensive reviews of the most stable modelling approaches.

3.3.1 Mathematical models

Mathematical models are typically based on differential equations, which are the mathematical language used to specify them and to describe how the state variables change mainly over time – Ordinary Differential Equation (ODE) – or time and space —Partial Differential Equation (PDE).

Ordinary Differential Equation

To write a system of ODE, normally it is necessary to begin with a structural model, in which the reactions and the effectors are known. To get from a set of reactions, to a set of ODEs, the set of variables $X_i(t)$ describing the state of the system has to be identified. $X_i(t)$ is a collective variable of such a system and is, in the most of cases, the concentration of specie i. The collection of values of all these state variables $\{X_1, X_2, ..., X_n\}$ denote a complete set of variables to define the *instantaneous state* of the system $\mathbf{X}$. The time evolution of $X_i(t)$ will take the form, through a mathematical expression:

$$\frac{dX_i}{dt} = F_i(X_1, X_2, ..., X_n; \gamma_1, \gamma_2, ..., \gamma_m) \tag{3.1}$$

where F_i may be complicated functions of the state variables, and $\gamma_1, \gamma_2, ..., \gamma_m$, are some parameters present in the problem whose variation influences the evolution of the system and which can be modified by the external world.

The system of differential equations defined describes how the concentrations of the molecules involved in the process under study, changes over time due to its interactions with the other species in the network. It does not tell us the value of X at any specific time t. Solving the differential equations is to find these functions, $X_i(t)$, for each variable i of the system. In order to solve Equation 3.1 we must first prescribe a set of initial conditions $\{X_1(0), X_2(0), ..., X_n(0)\}$.

Partial Differential Equation

ODE describes processes while abstracting from spatial dimensions so under the assumption of modelling a "well-stirred" system: the systems of interest are assumed, implicitly, to be spatially homogeneous.

There are situations in which these assumptions are not appropriate. From cells to tissues and organisms, biological systems own spatially inhomogeneous structures. All processes, in fact, develop in time and space. It might be necessary, for instance, to distinguish between different compartments of a cell, say the nucleus and the cytoplasm, and to take into account the diffusion of regulatory proteins or metabolites from one compartment to another. Again, gradients of protein concentrations across cell tissues are a critical feature in development processes.

In those cases in which spatial inhomogeneities has to be explicitly modelled, PDE can be used.

In the simplest case of only one spatial dimension, given a continuous variable $l \in [0, \lambda]$, where λ represents the size of the system, let's define the state of the system X as functions of both t and l. The time variation of the concentration of each substance X_i, is computed trough a partial differential equation, in the form:

$$\frac{\partial X_i}{\partial t} = F_i(\mathbf{x}) + \delta_i \frac{\partial^2 X_i}{\partial l^2} \tag{3.2}$$

where δ_i is the diffusion constant for the species i.

In order to solve Equation 3.2 it is needed other then a set of initial conditions $\{X_1(0), X_2(0), ..., X_n(0)\}$, also a set of boundary conditions in $l = 0$ and $l = \lambda$.

ODE and PDE generate deterministic models, but a further class of differential equation can be adopted to capture stochastic phenomena: the Chemical Master Equation, *i.e.*, ordinary differential equation describing the time evolution of the probability of a system to be in one of a discrete set of states. The main difference between these two class of mathematical models is that from the same initial state in the latter case there is only one possible successor state, while in the former one it is possible to get to multiple possible successor states.

Chemical Master Equation

The temporal behaviour of a spatially homogeneous mixture of molecular species can be described by a Chemical Master Equation. Chemical Master Equation is a form of mathematical formalism that describes the transition of the system from one state to another state using probabilistic methods. Before introducing the Master Equation, we first define the following notations:

- **M** = number of reactions
- **N** = number of species
- $\mathbf{X} = [X_1, X_2, ..., X_i, ...X_N]$ = number of molecules of species i in the system – $i = [1, 2, ..., N]$
- $p(\mathbf{X}, t)$ = probability of the system in state X at time t
- c_j = stochastic kinetic constant for reaction j – $j = [1, 2, ..., M]$
- R_j = reaction j – $j = [1, 2, ..., M]$
- $\alpha_j \Delta t$ = probability of R_j happening in time (t, Δt) given that the system is in the state **X** at time t.
- $\beta_j \Delta t$ = probability that the system is one R_j reaction removed from the **X** and undergoes the R_j reaction in time (t, Δt).

Given the notations, we can describe the evolution of $p(\mathbf{X}, t)$ in terms of the rates α and β as follows:

$$p(\mathbf{X}, t + \Delta t) = p(\mathbf{X}, t)\left(1 - \sum_{j=1}^{M} \alpha_j \Delta t\right) + \sum_{j=1}^{M} \beta_j \Delta t \tag{3.3}$$

The first term on the right hand side of Equation 3.3 represents the probability at which **X** remains its state, whereas the second term is the probability at which **X** undergoes one reaction in time (t, $t + \Delta t$). Reorganizing Equation 3.3, and taking the limit as $\Delta t \rightarrow 0$, gives the final form of Master Equation 3.4. Notice that the transition of the state of the system is described through changes of the probability of the system being in a certain state, $p(\mathbf{X},t)$. Hence, the inherent stochasticity of the system is mathematically formalized in this context:

$$\frac{\partial p(\mathbf{X},t)}{\partial t} = \sum_{j=1}^{M} (\beta_j - \alpha_j p(\mathbf{X},t)) \tag{3.4}$$

The Master Equation approach tries to write a system of equations for every possible transition state and solve them simultaneously. Solving that equation gives the complete probability distribution at any point in time. Even better, it is linear differential equations with constant coefficients, so one can actually solve it.

The number of tools for designing mathematical models and performing simulations is huge and is continuously growing. In [AAS06] is presented a survey of 12 software packages based on mathematical models, evaluating their functionality – such as which model they support, if they own the compartment abstraction, steady state analysis– reliability, efficiency, user-friendliness and compatibility. To cite few: CellDesigner, CellWare, COPASI, Dizzy, GEPASI, JDesigner, Virtual Cell. Species are produced and consumed by reactions which happen with a speed given by kinetic functions. Tools normally provide a set of kinetic functions that model the most common mechanisms such as enzymatic activation, phosphorilation...

3.3.2 Computational models

In contrast computational models are based on computational formalisms and approaches, *i.e.*, approaches developed in the field of Computer Science. The approaches range from Boolean Networks to Formal Methods to Cellular Automata and Agent-based models. The model is normally specified as an algorithm through a programming language such as Java, C, C++, Python, or as an high level code written in modelling languages such as Statecharts, Reactive Modules, different Calculi (π, λ...), Petri Nets or languages defined ad-hoc for creating a biological model, which are then compiled into a machine-readable language.

Boolean Networks

The simplest approach to characterising the dynamics of biological networks is a Boolean model, which is often applied to studying molecular interaction networks inside a cell, as a method that allow predictions of *qualitative* properties of such systems, i.e. dynamical properties that are invariant for a range of reaction mechanism and values of kinetic constants.

In [Kau93] are presented some Boolean models of biological systems. To most notable examples of Boolean models are those of gene regulatory networks. As a first approximation, the state of a gene can be described by a Boolean variable expressing that it is active (**on, 1**) or inactive (**off, 0**) and hence that its products are present or absent. The change in gene expression can be described by making the assumption that the change in activation state of a gene is determined in a combinatorial fashion by the activation of other genes, in particular genes encoding for regulatory proteins. Interactions between elements can be represented by *Boolean functions* which calculate the state of a gene from the activation of other genes.

Let the n-vector $\hat{\mathbf{x}}$ of variables in a Boolean network represent the state of a regulatory system of n elements. Each $\hat{x}_i$ has the value 1 or 0, so that the state space of the system consists of 2^n states. The state $\hat{x}_i$ of an element at time-point $t+1$ is computed by means of a Boolean function or rule $\hat{b}_i$ from the state of k of the n elements at the previous time-point t. (Notice that k may be different for each $\hat{x}_i$) The variable $\hat{x}_i$ is also referred to as the output of the element and the k variables from which it is calculated the inputs. In summary, the dynamics of a Boolean network describing a regulatory system are given by

$$\hat{x}_i(t+1) = \hat{b}_i(\hat{\mathbf{x}}(t)), 1 \leq i \leq n. \tag{3.5}$$

Transitions between states in a network are *deterministic*, with a single output state for a given input, and *synchronous*, in the sense that the outputs of the elements are updated simultaneously.

Extensions towards Probabilistic Boolean Network are also known [SDKZ02].

Formal Methods

A good deal of work in the research field of Computational Systems Biology has been done in the field of Formal Methods.

Process Algebra The first works saw the application of process algebra – namely π-calculus [RSS01], stochastic π-calculus [Pri95, PRSS01], and more recently Beta-Binders [PQ04], κ-calculus [DL04, DFF$^+$07], Bio-PEPA [CH09] – at the simulation of various intracellular systems, including gene regulatory networks, metabolic pathways, and signal transduction networks. The growing interest recently observed in the community is mainly due at the key characteristics of this approach such as: *(i)* compositionality, *i.e.*, the whole system can be defined starting from the definition of its subcomponents, *(ii)* formal representation of the model so to avoid ambiguity, *(iii)* different kinds of analysis, such as sensitivity analysis and probabilistic model checking, can be done on a process algebra model and *(iv)* from a model perspective a biological system can be naturally seen as a system composed by concurrent interacting processes so that with process algebra biological entities such as molecules are modelled as computational processes, and their complementary structural and chemical determinants correspond to communication channels. Chemical interaction and subsequent modications coincide with communication and channel transmission. Almost all of the cited process algebra are supported with a platform on top of which coding the models and then performing simulations. The most-known are SPIM [Phi06] for stochastic π-calculus, the BetaWB [DPRS08] for Beta-binders (and its extension BlenX cited below), the Bio-PEPA Eclipse Plug-in or the Bio-PEPA Workbench. They all perform simulation according to the Gillespie's stochastic simulation algorithm

– or more efficient variants – whose details are given in the following, and eventually provide a graphical user interface for editing the model, visualising the simulation dynamic and analysing the results of a stochastic simulation run.

Petri Nets Also Petri Nets have recently been introduced as a potential tool for modelling, composing and analysing biological systems [HGD08]. The simplest kind of Petri Net is a bipartite directed graph with two types of nodes – *place nodes*, represented as circles, indicate the local availability of resources and *transition nodes*, represented as boxes, are active components that can change the state of the resources – and weighted arcs which connect nodes of different types —there cannot be arcs between places or between transitions. Places may hold an integer number of tokens represent time events. The distribution of tokens over the places (the *marking*) of a petri net is the overall state of the system. The transition between states is implemented as a transition of tokens from one place to the other passing through the arc and the transition node: when the transition is fired a number of tokens from each input place of the arc is removed and an equal number of new tokens is created in the output node. Petri Nets are well-suited for modelling biochemical processes as they own key properties such as concurrency – several transitions may happen in parallel as in cells all reactions can happen in parallel and most are independent of each other – and stochastic – the choice among different possible transitions is made following a probability distribution. They are mainly used to model intracellular networks. The places in a Petri net can represent genes, protein species and complexes. Transitions represent reactions or transfer of a signal. Arcs represent reaction substrates and products. Firing of a transition is execution of a reaction, such as consuming substrates and creating products.

Spatial extensions More recent extensions have been developed for modelling biological compartment. BioAmbients [RPS$^+$04] is one of the first effort in this direction: recognised the key role of compartments in the biological organisation, it proposes a model which represents various aspects of molecular localisation and compartmentalisation, including the diffusion of molecules between compartments and the interaction between molecules in a location of the overall network of compartments. In [VB08] the $S\pi@$ process calculus is introduced to deal with the notion of compartments (possibly with variable volumes), by adding to the stochastic π calculus the idea that process-molecules are situated into a location. In [CDG09] Bio-PEPA has been extended for expressing hierarchies of locations with different sizes so that to model compartments, membrane and cell intra-compartment and inter-compartment reactions. In *Beta-binders* and its extension called BlenX, systems are modelled as a set of boxes representing biological entities at different levels —proteins, cells. A model in *Membrane computing* [Pau02], formally called P systems, consists of a membrane that contains a multi-set of objects (representing chemical substances) that evolve according to given evolution rules (representing reactions).

The Gillespie's Stochastic Simulation Algorithm

Simulations of models developed with formal methods are usually simulated upon the Gillespie's stochastic simulation algorithm (SSA) [Gil77]. In 1976 Gillespie developed a discrete stochastic simulator algorithm to solve the Chemical Master Equation based on the assumptions that the system is homogeneous and well mixed.

The Gillespie algorithm makes time steps of variable length; in each time step, based on the rate constants and population size of each chemical species, one random number is choose which reaction will occur, and another random number determines how long the time step will last.

At each time step, the chemical system is exactly in one state. The idea is to directly simulate the time evolution of the system. Basically, the algorithm determines the nature and occurrence of the next reaction, and consequently of the next transition, given that the system is in state α at time t. Given a system with total number of reaction channels **N** and total number of species **M**, there are at most **N** possible transitions from a given state. The key is to choose random number using a computer random number generator, and use those to pick transitions.

More schematic details of the algorithm are presented above. Given the following premises:

- *Well-stirred* system of molecules of N chemical species $\{S_1, ..., S_N\}$
- The molecules interact through M reactions $\{R_1, ..., R_M\}$
  - only *unimolcular* and *bimolecular* reactions are considered
  - *trimulecolar, reversible...* are modelled as a sequence of reactions
- Let $\mathbf{X}(t) = (X_1(t), ..., X_N(t)$ be the state of the system
  - $X_i(t)$ is the number of S_i molecules in the volume at time t
  - v_j is the state change vector
- Let c_j be the reaction probability rate constant for R_j
- Let $a_j(\mathbf{x})dt$ be the *propensity function* for R_j as the probability, given $\mathbf{X}(t) = \mathbf{x}$, that on R_j will occur in $[t, t+dt)$

$$S_1 \xrightarrow{c_j} \text{products: } a_j(\mathbf{x})dt = c_j x_1$$

$$S_1 + S_2 \xrightarrow{c_j} \text{products: } a_j(\mathbf{x})dt = c_j x_1 x_2$$

$$2S_1 \xrightarrow{c_j} \text{products: } a_j(\mathbf{x})dt = c_j \frac{1}{2} x_1(x_1 - 1)$$

the Gillespie's *Direct Method* is implemented as follows:

1. Initialise the time $t = t_0$ and the system's state $\mathbf{x} = \mathbf{x}_0$
2. With the system in state $\mathbf{x}$ at time t, evaluate
   - all the $a_j(\mathbf{x})$
   - their sum $a_0(\mathbf{x}) \equiv \sum_{j'=1}^{M} a_{j'}(\mathbf{x})$
3. Draw two random numbers r_1 and r_2 from the uniform distribution in the unit interval and take

$$\tau = \frac{1}{a_0\mathbf{x}} ln\left(\frac{1}{r_1}\right)$$

$$j = \text{the smallest integer satisfying} \sum_{j'=1}^{j} a_{j'}(\mathbf{x}) > r_2 a_0(\mathbf{x})$$

4. Effect the next reaction by replacing $t \leftarrow t + \tau$ and $\mathbf{x} \leftarrow \mathbf{x} + \nu_j$
5. Record ($\mathbf{x}$, t) as desired
6. Return to step 2, or else stop

Cellular Automata and Agent-based models

Cellular Automata (CA) and Agent based models (ABM) (the latter diffusely discussed in Chapter 7) are quite similar computational models organised around the same abstraction of individual entities called *cells* in CA, and agents in ABM, and around the concept of space which defines, in different ways, the neighbourhood of each entities, *i.e.* a set of entities with which can happen a communication / interaction. Cells in CA are rather simpler then agents in ABM. Their behaviour is defined with update rules that change their states, switching between boolean values in the elementary case. Such rules has as input the state of the cell itself and of its neighbours. The grid of cells is normally static so that movement or cell replication is allowed.

This simple idea has been extended with the theory of agents and multi-agents systems where entities have a more elaborated autonomous behaviour and live, move, replicate in an eventually dynamic environment with a complex behaviour itself.

4 Models of Development

Several models of development have already been realised. They can be divided into two categories: mathematical and computational models. They either model real biological systems involving real biological entities and processes, or reproduce an abstract process that resembles in some way the morphogenetic processes and that results in the formation of a spatial pattern.

4.1 Mathematical Models of Development

Mathematical models are continuous in time and space, and use families of differential equations.

Among the mathematical models the eligible work of Alan Turing, extended in multiple way into the class of *reaction-diffusion* models, plays a crucial role. Inspired at the reaction-diffusion model, real system models have then been developed, capturing the gene interactions and the transcription factors diffusion in the embryo. In Section 4.1.2 is presented the model of spatial organisation in Drosophila Melanogaster, adopted in this thesis as reference case study.

4.1.1 The reaction-diffusion models

One of the most important mathematical model in developmental biology was formulated by Alan Turing [Tur52]. He proposed the reaction-diffusion model as the basis of the development of patterns during morphogenesis. His models is able to self-organise into spatially heterogeneous patterns of chemical concentrations beginning with species homogenously distributed over space.

The central idea of the model follows. A chemical system contains two morphogens, P and S. P is able to activate his own synthesis as well as the synthesis of S while S inhibits the synthesis of both P and S. Morphogens are able to diffuse: the cell-to-cell diffusion constant for P will be called μ, and that for S will be called ν and $\mu < \nu$ so that P is a slow-diffusing activator and S is a fast-diffusing inhibitor. The dynamic of such a chemical system is described

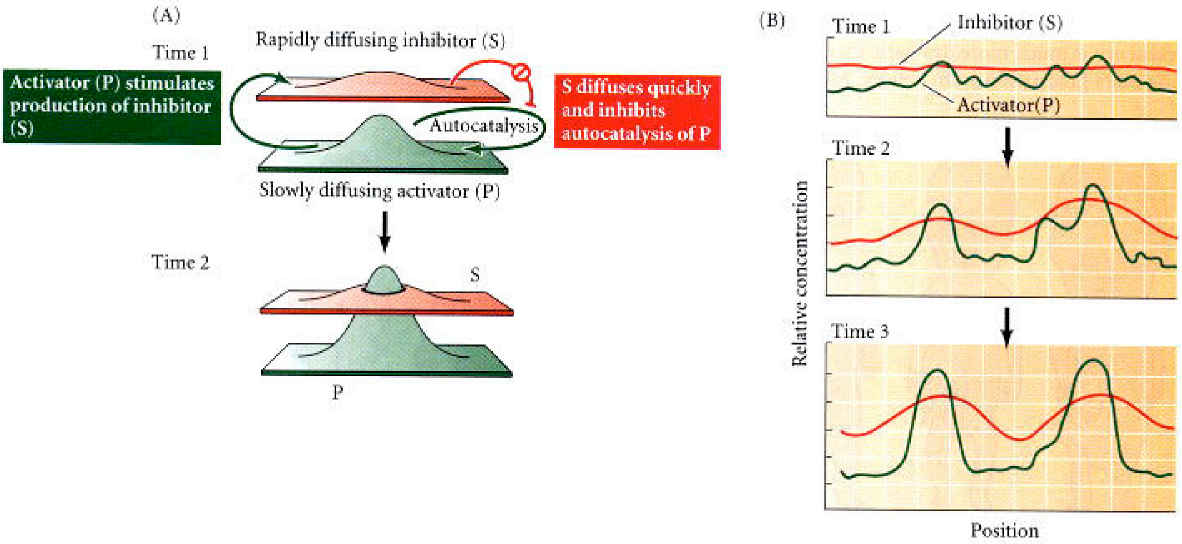
(A)
Time 1
Rapidly diffusing inhibitor (S)
Activator (P) stimulates production of inhibitor (S)
Autocatalysis
S diffuses quickly and inhibits autocatalysis of P
Slowly diffusing activator (P)
Time 2
S
P
(B)
Time 1
Inhibitor (S)
Activator(P)
Time 2
Time 3
Relative concentration
Position

Figure 4.1: The Turing model dynamic from [Gil06]

by the partial differential equations:

$$\frac{\delta P}{\delta t} = f(P,S) + \mu \overset{2}{\nabla} P \tag{4.1}$$

$$\frac{\delta S}{\delta t} = g(P,S) + \nu \overset{2}{\nabla} S \tag{4.2}$$

where $f$and g are the rate of synthesis for P and S respectively.

Turing's model shows that if S diffuses more rapidly then P, sharp waves of concentration difference will be generated for substances (see Figure 4.1). When the concentration of the morphogen is above a certain threshold cells then may be instructed to differentiate.

Along this line several *reaction-diffusion* models have been developed. They normally extend theoretical models for autocatalytic reactions, such as the Brusselator, proposed by Ilya Prigogine in [PG71] or the Oregonator, developed by Field and Noyes in [FN74], with a term of diffusion of the reactants in the model. These extensions show that the diffusion in two or three dimension is able to generate spatial structures whose shape and dimension is highly dependent on the model parameters [BFDP10, GF92].

Spatial structures in the Brusselator model

The Brusselator model is a typical example of nonlinear chemical oscillator. Varying the model parameters the system can stabilise into a fixed point or a limit cycle. The reactions of the model are:

$$\begin{aligned} r1 &: \overline{A} \xrightarrow{k_1} X \\ r2 &: \overline{B} + X \xrightarrow{k_2} Y \\ r3 &: 2X + Y \xrightarrow{k_3} 3X \\ r4 &: X \xrightarrow{k_4} \emptyset \end{aligned}$$

where $\overline{A}$ and $\overline{B}$ are catalysers which for assumption are always available, while the dynamics of X and Y are studied. The spatial extension adds two reactions:

$$\begin{aligned} r5 &: X_i \xrightarrow{\delta_x} X_j \\ r6 &: Y_i \xrightarrow{\delta_y} Y_j \end{aligned}$$

where i and j are the indexes of the spatial position. The overall system is modelled in terms of PDE as follows:

$$\begin{aligned} \frac{\delta X}{\delta t} &= k_1 A - k_2 BX + k_3 X^2 Y - k_4 X + D_x \overset{2}{\nabla} X \\ \frac{\delta Y}{\delta t} &= k_2 BX - k_3 X^2 Y + D_y \overset{2}{\nabla} Y \end{aligned}$$

The results of such a kind of model and the analysis of the parameters is presented in [YDZE02] and a fantastic demo is reproduced in [Cro].

4.1.2 A Drosophila Melanogaster model

On the line of Equations 4.1 and 4.2 an interesting model of Drosophila Melanogaster morphogenesis have been developed in [GJK$^+$04]. The activity of each gene, *i.e.*, the rate of protein synthesis, is regulated locally by three processe: cross-regulation of the protein synthesis, decay and diffusion. The state variables are the concentrations of gene products (proteins). Only one dimension of space on the AP axis is considered introducing the continuous variable x. Denoting the concentration of the ath gene product in a nucleus at position x at time t by $v^a(x,t)$, the set of reaction-diffusion equations for N zygotic genes is:

$$\frac{\delta v^a(x,t)}{\delta t} = P^a(v^a(x,t)) - \lambda^a v^a(x,t) + \frac{\delta^2 v^a(x,t)}{\delta x^2}$$

P^a is the term that controls the activation of each gene as a function of the concentration of its regulatory proteins. For a more extensive description of the function P^a see [GJK$^+$04, PJRG06].

4.2 Computational Models of Development

Computational models, on the other side, are normally discrete in space, time or state and model individual entities of the system (cells, proteins, genes).

4.2.1 Formal Methods for biological system development

The approaches developed in the field of formal methods applied in computational biology still miss the notion of compartments – that will be shown in Chapter 5 to be crucial in such a kind of scenario – or they have it but mainly with respect to intracellular compartments, so that they do not scale with the system's size. For this reason there are not significant examples of formal methods applied at morphogenetic phenomena.

An interesting exception is the application of Petri Nets at the model of *Caenorhabditis elegans* vulval development [BKF$^+$09]. There the large intracellular network is divided into modules, each one corresponding to different biological functions, such as gene expression, protein activation and protein degradation. Cells itself are modules that can eventually be reused if there is the need to model the same function and behaviour. The entire multicellular systems is then modelled as interconnected identical modules of a multipotent cell.

4.2.2 ABM for biological system development

The agent-based approach is instead an example of computational model extensively used in the field of system development. Agent-based modelling (ABM) can be used to explicitly model a set of entities with a complex internal behaviour and which interact with the others and with the environment generating an emergent behaviour representing the system dynamics. More details on such a kind of approach will be given in Chapter 7. Some work has already been done in the application of ABMs in morphogenesis-like scenarios. A good review is proposed in TBDP08. Most of these models generate artificial pattern – French and Japanese flags [BMF06] – realising bio-inspired models of multicellular development in order to obtain predefined spatial structures. But at the best of my knowledge there are not much results already obtained in the application of ABM for analysing real phenomena of morphogenesis, even though some interesting idea have been developed. One of the best example is presented in [CZNA07] where an ABM of developing limbs is reported.

4.3 Hybrid Models of Development

Also hybrid framework has been developed. They combine aspects of mathematical models with computational ones.

COMPUCELL 3D [CHC$^+$05] is one of those frameworks, which combine discrete methods based on cellular-automata to model cell interactions and continuous model based on reaction-diffusion equation to model chemical diffusion. COMPUCELL 3D looks a very promising framework. It provides a friendly interface for the 2D or 3D visualisation of the simulation dynamics, showing either the cells behaviour (movement, growth) or the environment composition and dynamic in terms of diffusing molecules. Unfortunately the framework has some important limitations: *(i)* the documentation is a bit poor, and the user can be in trouble once he has to extend the available models, or more, to create a new model from scratch; *(ii)* there is a lack of an high level language or a user interface to easily define the model and initialise the simulation, so that it requires to implement the model directly in XML or even Python; *(iii)* there are still not available, even though it is known that there are extensions at work, suitable tools and abstractions for modelling cell internal behaviour, gene regulatory network in particular, so to explicit each gene involved in the pattern formation, and how it interacts with other genes. The cell differentiation is however still a rough concept in COMPUCELL 3D.

Part II

Motivation

5 Towards Multi-level Models

According to one of the main properties of biological systems as complex systems, all biological systems are organised in a hierarchy of levels: they are amenable to be represented as organised on different layers, ranging from genes and cells up to tissues, organs and organisms.

Such a property particularly influences multi-cellular phenomena, where the autonomous behaviour of the cell is highly influenced by the interactions taking place at the level of tissue, while such interactions are dependent by the internal behaviour of each cell. One of the best examples of this fundamental interplay among different levels of the biological hierarchy is the morphogenesis, whose main biological mechanisms are extensively described in Chapter 2.

In Chapter 3 the central role of models and simulation is explained. When we create a model of such a kind of complex organization, we may not, and in most cases we can't, analyse all hierarchical levels to understand the functioning of the biological system. We can so focus our attention at one level or more, depending on the problem. In this Chapter I will analyse different modelling perspective which can be adopted once modelling biological systems in general – as proposed in [UDZ05] – with a particular focus at the morphogenesis, identifying the best way for capturing such a phenomenon.

5.1 The Role of Hierarchy in Biological Systems

Complex systems in general exhibit a hierarchical organisation that divide the system into levels composed by many interacting elements whose behaviours are no rigid but are self-organised according to a continuous feedback between levels. Each level is essential to the general understanding of the system emergent behaviour, and it is autonomous with its own laws, pattern and behaviour. At the same time, no level can be understood in isolation independently of all the other levels, and the system as a whole can be understood only through the understanding and representation of all of its levels. Hierarchy has therefore a crucial role in the static and dynamic characteristics of the systems themselves. This happens according to the principles of *downward* and *upward* causation, where the behaviour of the parts (down) is determined by the behaviour of the whole (up), and the emergent behaviour of the whole is determined by the behaviour of the part [UDZ05]. An example is given by biological systems: an outstanding property of all life is the tendency to form multi-levelled structures of systems within systems. Each of these forms a whole with respect to its parts, while at the same time being a part of a larger whole. Biological systems have different level of hierarchical organisation – *(1)* sequences; *(2)* molecules; *(3)* pathways (such as metabolic or signalling); *(4)* networks, collections of cross-interacting pathways; *(5)* cells; *(6)* tissues; *(7)* organs – and the constant interplay among these levels gives rise to their observed behaviour and structure. This interplay extends from the events that happen very slowly on a global scale right down to the most rapid events observed on a microscopic scale. A unique molecular event, like a mutation occurring in particularly fortuitous circumstances, can be amplified to the extent that it changes the course of evolution. In addition, all processes at the lower level of this hierarchy are restrained by and act in conformity to the laws of the higher level.

In this contest, an emblematic process is morphogenesis, which takes place at the beginning of the animal life and is responsible for the formation of the animal structure. Morphogenesis phenomena includes both cell-to-cell communication and intracellular dynamics: they work together, and influence each other in the formation of complex and elaborate patterns that are peculiar to the individual phenotype.

5.2 On Different Modelling Perspective

Modelling approaches might be classified in three main classes – macro-, micro- and multilevel-models – according to the perspective they adopt once designing the model.

5.2.1 Macro-level, continuous models – Macroscopic view

In a macro model and subsequent simulation, a complete system is tackled as one entity whose state variables are updated during simulation. Modelling, simulation and observation happens on one global level. The system is described by a set of state variables with their interdependencies, which can be expressed as rules, equations, constraints etc. All the simulations based on the macroscopic view are deterministic in nature. As a result, the system evolves along a fixed path from its initial state.

Typical representatives of this class are *differential equation* models which describe the time-dependent changes of the state variables, e.g. a biochemical system based on concentrations and reaction rates.

Focusing only on the population, we lose the representation of the individual and its locality, with the conditional and adaptive behaviour of each entity in its local environment.

Despite these limitations, macro simulation is used. Its advantages results from their relative simplicity and from their formal aspect. First of all, in fact, differential equations are a really well understood and established framework, in which the complete model is documented concisely through formulas, and in which low number of parameters construction based on global input/output behaviour. With this approach, moreover, simulation experiments can be very fast (depending on the integration step).

5.2.2 Micro-level, discrete models – Microscopic view

Micro models are models that represent systems as composed by huge numbers of rather homogeneously structured entities. Only the behaviours of the individuals is explicitly modelled. The macro level of the system exists only as it aggregates results of the activities at micro level and is used for reflecting emergent phenomena, e.g. the development of specific spatial patterns. It does not have any behaviour on its own.

Typical representatives of this class are *cellular automata*, where individual behaviours are modelled through a set of rules. The behaviour of the whole system is then captures by the states of its entities.

5.2.3 Multi-level models

Micro models often form only a transition to multi-level models, which describe a system at least at two different levels. Interactions are taking place within and between these levels: not only interdependencies at one organizational level but between different ones become of interest examining the links between macro- and micro-levels.

The importance of multi-level models has been emphasized for biological and social systems in particular, due to the great interplay that takes place between different levels of hierarchical organization. Only this kind of approach guarantees in fact to explicitly model the effects of the macro dynamics on the individual behaviours.

Interpreting the interaction that occurs across the layers, from genes and proteins to cells, tissues, and so on, requires a multi-level model. Moreover, the description of systems at different levels and different time scales facilitates taking spatial and temporal structured processes into consideration [MG05].

5.3 A critical analysis of related work

For designing a multi-level model, the three main components of a multi-level model has to be represented, *i.e.*,

1. the reactions that take place inside a cell
2. the release of molecules in the environment, and the stimuli from the environment
3. the direct communication between cells, especially through membrane proteins

Moreover morphogenetic scenarios presents an other cause of complexity given by the huge number of cells that it normally involves, so that, even with proper simplifications, the system cant be reduced at few compartments. As already reported in Chapter 3, the ability of modelling biochemical networks is firstly offered by simulation tools for kinetic modelling of biochemistry. They tackle the problem mostly at the level of Graphical User Interfaces, with little support to flexibility in expressing large-scale and dynamic networks at the language level. They are based on mathematical models (ODE, PDE), that, for construction, are not able to capture the individual behaviour so that they are mainly focused on intracellular networks. However there are few exceptions such as CellDesigner, CellWare, COPASI, Dizzy, GEPASI, JDesigner, Virtual Cell, MesoRD which support the representation of intracellular compartments, inside which the dynamic is modelled by means of differential equations, but hardly can model big networks of compartments specifically because the specification of their behaviour and structure is normally done by hand through the User Interface.

Formal methods have also been described in Chapter 3, where the problem of multi - compartimentalisation seems to be more felt. A good deal of work has been moving towards multi-compartimentalisation, as we witness a trend moving from the single, global solution idea of e.g. stochastic π-calculus [Pri95] and κ-calculus [DL04, DFF^{+}07], to mechanisms and constructs tackling the multi-compartment scenario.

One of the best examples are the techniques in *Membrane computing* [Pau02], formally called P systems, which consist of a membrane that contains a multi-set of objects (representing chemical substances) that evolve according to given evolution rules (representing reactions). Each membrane encloses a compartment that can itself be enclosed in an other compartment so that to form a hierarchy of compartments. In [SMC^{+}08] the P systems formalism has been extended with stochastic semantics integrating the Gillespie's SSA with P systems. In that work reactions taking place inside each compartment are modelled with the SSA, while system's topology is given by the P system compartment and the interaction among compartments are modelled inspiring at the P system rules.

In [VB08] the $S\pi@$ process calculus is introduced to deal with the notion of compartments (possibly with variable volumes), by adding to the stochastic π calculus the idea that process-molecules are situated into a location. In [CDG09] Bio-PEPA has been extended for expressing hierarchies of locations with different sizes so that to model compartments, membrane and cell intra-compartment and inter-compartment reactions. In *Beta-binders* and its extension called BlenX, systems are modelled as a set of boxes representing biological entities at different levels – proteins, cells – and are simulated on top of BetaWB [DPRS08].

It is worth reminding that the mentioned languages and frameworks are not conceived to address systems composed by a huge number of interacting cells. In particular, why it is still possible to use e.g. SPIM or Bio-PEPA to model scenarios like the one studied in this paper (featuring a network of 1000 cells), it would require a huge specification that is simply impractical to produce by hand.

Languages ad-hoc for scenarios of spatial patterning have also been identified in the spatial computing research thread, such as Proto [BB06] and MGS [SMG04]. Proto is a language for programming the behaviour of a possibly large distributed systems in which any node performs local computations and interacts with a limited neighbourhood. By using Proto the designer programs in terms of functional computation over spatial computational fields. The main differences with our language are: *(i)* differentiation is achieved in Proto by a mechanism of "sensing" (by intercepting a given signal, a device can perform a tailored computation), while in our framework we can even specify different behaviours in each cell; *(ii)* Proto language abstracts from network topology issues (they are handled at the level of platform), while our language has features to flexibly specify this topology); *(iii)* the only form of inter-device interaction in Proto is a mechanism of automatic, asynchronous broadcasting of messages to all neighbours, while our transfer of molecules is unidirectional and probabilistic. `MGS` is a programming language that supports the modelling and simulation of system with a complex structure which might also change over time, then supporting also dynamic topology. In [SMC^{+}08] `MGS` is used for implementing a stochastic P system. The syntax of the language looks somewhat awkward to people with not specific background in computer science so that we claim that it can hardly be used by a broader user composed, among the others, by biologists.

Part III

Contribution

6 MS-BioNET and the Application at Drosophila Development

In this chapter I first describe the computational model that grounds our work, and then I describe MS-BioNet (MultiScale-Biochemical NETwork), the surface language, and finally the main features of the implemented simulation engine.

The calculation is very simple, yet it also allows to model complex simulation scenarios. They could involve large networks of compartments (one can easily think of biological scenarios containing several thousands of them), with specific topological structures (a lattice, a torus, a scale-free network), and where certain compartments might have specific characteristics, such as certain chemical laws, initial concentrations of chemical substances, outgoing link rates, and so on. Therefore, as far as the practical aspect of the simulation methodology is concerned, it is rather important to devise a surface language that can, on the one hand, provide suitable constructs to easily express the above cases, and on the other hand, intuitively "compile" into the computational model above—much in the same way as, e.g., the SPIM language is an extended version of the stochastic π-calculus. Finally an efficient simulation engine able to execute such computational model is necessary for performing simulations in reasonable time.

The capability of the framework are then tested on the *Drosophila* development, and the obtained results compared with experimental data.

6.1 The computational model

The proposed computational model of multi-compartment biochemical networks is a direct extension of the Gillespie's basic chemical model [Gil77], which realise a discrete and stochastic simulation of a system. In the Gillespie's model a chemical system is view as a unique well-mixed solution. Every molecule is explicitly modelled and every reaction they can participate to explicitly simulated on the basis of a stochastic algorithm, the Gillespie's *stochastic simulation algorithm* (SSA). Once the system has been initialised, i.e., molecules, reactions and reaction rates are defined, the simulation proceeds choosing the next reaction to occur on the basis of a random number and its *propensity function* that is calculated on the reaction rate and on the number of reactants. The time interval to update the simulation time is also computed step by

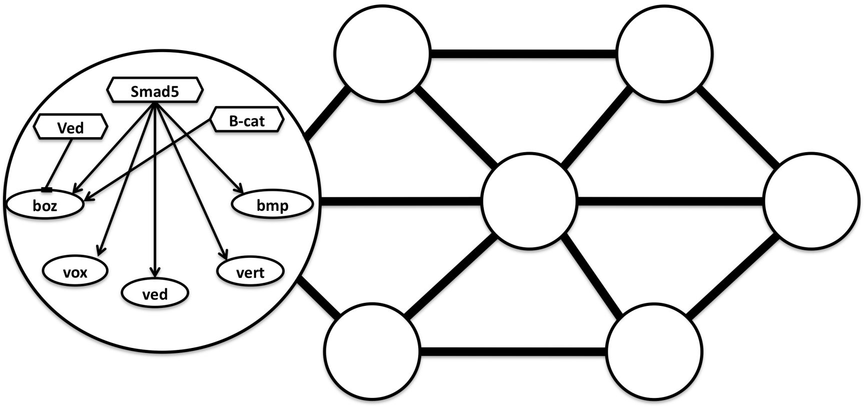

Figure 6.1: A multi-compartment version of Gillespie's chemical model.

step depending on a random number and on the sum of propensity functions of all reactions. The iteration of these steps constitutes the simulation.
This model is enriched with two additional concepts:

- As shown in Fig. 6.1 instead of a single compartment we actually have a graph-like network of compartments. In each node of the graph is contained a chemical solution of molecules that react according to Gillespie's stochastic model; proximity is modelled by a concept of *link*: the connection between neighbouring and communicating compartments is established by a link, i.e., an edge between two nodes of the graph. A node of the network can model each entity that has an autonomous behaviour, often bounded by a membrane, and able to interact with other entities, such as, in biological systems, intracellular compartments (nucleus, mitochondria, the Golgi complex and so on) or cells.

- Chemical laws can display on their right-hand side (as "products" of the chemical reaction) a *firing molecule*, namely, a molecule that is crossing the boundary of the compartment (such as the cell membrane), and thus is ready to be transferred through a link towards another compartment. As such, each link is characterised by a rate dictating the velocity of molecule transfer and modelling proximity, and defines a set of molecules that are allowed to transit through it. Fig. 6.2 depicts this behaviour.

To keep the parallel with biochemistry as exact as possible, we rely on mechanisms mimicking chemical transport through biological membranes. We use a refined chemical transfer model, based on Nernst equation [AJL$^+$02] of electrochemical gradient, which basically states that transfer is proportional to the logarithm of c_t/c_s, where c_t and c_s are concentrations of a given chemical substance in the target and source compartments. On the other hand, in networks of compartments we observe that the transfer rate cannot grow indefinitely: a maximum transfer bandwidth involving two neighbouring compartments is to be considered due to their physical characteristics. As a result, we introduce a characterisation of chemical transfer as follows:

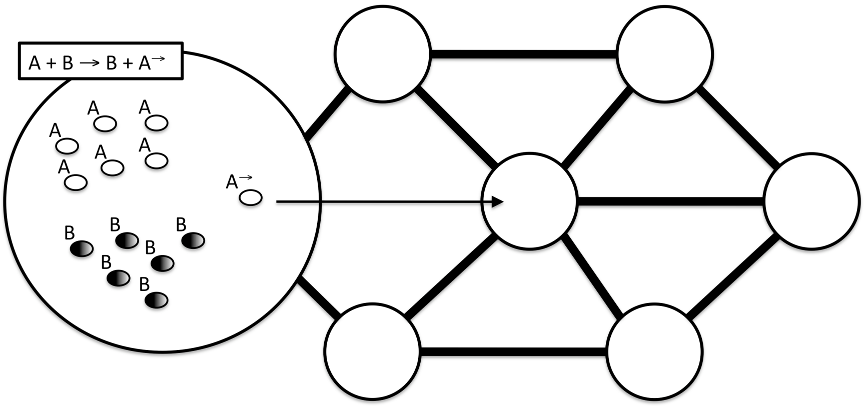

Figure 6.2: Transfer Model. Some chemical reactions can produce *firing molecules*.

(i) a topological connection between two compartments is reflected by a (unidirectional) *link* concept, characterised by a link rate r that measures the maximum transfer of molecules per time unit; *(ii)* the actual transfer rate may be affected by a *gradient substance* G (namely, by the ratio between the concentrations of substance G in the source and in the target compartment); *(iii)* gradient strength and direction can vary and are described by a *gradient factor* f: when f is 0 the transfer rate is not influenced by any gradient and remains equal to r, when f is positive a molecule would tend to ascend the gradient created by G, when f is negative it would instead move down the gradient—the actual gradient slope increases with the absolute value of f.

The computational model described is therefore able to integrate what happens inside a compartment with what happens outside being in such a way a multi-level model. For instance, modelling a multi-cellular organism, we are able to reproduce intracellular genetic networks and signalling pathways as reactions inside the compartments, but, at the same time, we can capture also the phenomena of cell-to-cell communications that happen at the level above through the exchange of molecules among compartments.

For the scope of this chapter we do not provide here the formal details of our model. Interested reader can refer to [MV09a].

6.2 MS-BioNet: the surface language

The language adopted is basically a description language on top of Prolog. While it basically amounts to describing facts that specify relevant aspects of molecules, laws, compartments, networks, and so on, it can flexibly and coherently rely on preconditions of facts (namely, rules) where Prolog's goal resolution and variable unification can be leveraged when expressive power is needed.

Let `T` be any first-order term, `R` any real number, `N` any natural number, and `L` any list of terms (with usual syntax `[T1,..,Tn]`, that is, a comma-separated list of terms within square brackets), the language provides the declaration constructs shown in Table 6.1, each option-

```
constant T.                        % asserts T as a fact
molecule T.                        % declares molecule with id T
reaction T : Li --> Lo rate R.     % declares chemical
                                          reaction with id T
compartment T.                     % declares compartment with id T
link Ts >>> Td rate R molecule Tm. % declares link for molecule
                                          Tm, from comp Ts to Td
concentration N of Tm in Tc.       % sets initial concentration
                                     of Tm into comp Tc equal to N
place Tr into Tc.                  % places reaction Tr in comp Tc
final_time R.                      % sets overall simulation time
final_steps N.                     % sets overall simulation steps
sample_time R.                     % sets time between
                                          two observations
sample_steps R.                    % sets number of steps between
                                          two observations
out L.                             % prints a list of items
                                          as of below
out molecule(Tc,Lm).               % prints a molecule conc
out time.                          % prints elapsed time
out step.                          % prints number of steps so far
out end_of_line.                   % prints a carriage return
out string(T).                     % prints a string
```

Table 6.1: Surface language constructs for specifying the model components and their behaviour, the initial conditions, and the desired output of the simulation.

ally providing a `where` clause by which additional conditions can be specified that bind logic variables used in the declarations. Such conditions can be any Prolog goal, with an additional syntactic sugar such that "`X in [1,2,3,4]`" and " `X in 1 .. 4` " unify `X` with 1, 2, 3, 4, iteratively.

For the sake of space, we explain the semantics of these declarations by one simple, yet interesting example whose meaning and results are explained later on in Section 10.2.1 when describing the application of our framework. We consider a square grid of 70x70 compartments, named `c(1,1),..,c(70,70)`, where each compartment is connected to the 4 adjacent neighbours (except for compartments in boundary positions of the grid); the compartment in the central position pumps a chemical substance that diffuses around and is subject to decay; we want to simulate 500000 steps and print for each timestep a matrix visualising the diffusion of the substance, followed by the elapsed time. This system is specified in Table 6.2. The first line declares the grid size to be 70. Molecules `pump` and `field` are then declared. Notice that

```
constant size(70).

molecule M where (M in [pump,field]).
reaction r(pump)  : [pump] --> [pump,field] rate 10.0.
reaction r(diff)  : [field] --> [field,firing(field)] rate 0.2.
reaction r(decay) : [field] --> [0] rate 0.1.

compartment c(X,Y) where (size(N), X in 1..N, Y in 1..N).
link c(X,Y) >> c(X,Y+1) rate 10000.0 molecule field.
link c(X,Y) >> c(X,Y-1) rate 10000.0 molecule field.
link c(X,Y) >> c(X-1,Y) rate 10000.0 molecule field.
link c(X,Y) >> c(X+1,Y) rate 10000.0 molecule field.
concentration 1 of pump in c(M,M) where (size(N), M is N/2).
place _ into _.

final_steps 500000.
sample_steps 50000.
out [time,step,end_of_line].
out S where ( compartment c(X,Y), S1 = molecule(c(X,Y),field)),
            ((X=N,S=[S1,end_of_line]);S=S1)
).
```

Table 6.2: Code for a 70x70 field diffusion scenario (see Section 10.2.1 for details).

the internal interpreter invokes goal "`molecule M`" to inspect molecules' type, which yields two solutions, binding `M` to `pump` and then to `field`—this mechanism is used in all other declarations, and is key in our language semantics. Three chemical reactions are defined, one that pumps molecules of `field` if molecule `pump` is present, one that diffuses a copy of `field` in some neighbouring compartment, and one to decay `field` substance.

The `compartment` declaration defines the 70x70 grid: note that, due to Prolog resolution, they are ordered as follows:

```
c(1,1),..,c(1,70),c(2,1),...,c(70,70)
```

Then, links are declared for `field`—note that the interpreter automatically excludes links escaping the grid. First, the `concentration` declaration is used to place one molecule of `pump` in the centre of the grid, while all other concentrations are set to 0 by default. The `place` declaration is then used to place all defined reactions in all compartments. Finally, we define a total number of simulation steps, the number of steps between two observations, and printing commands: the last `out` declaration emits the value of `field` in each compartment in the right order, also producing an end-of-line at the end of each row—recall that in Prolog, “;” stands for logical disjunction.

The reader can notice that the resulting specification combines two key aspects: *(i)* a tight connection between programming constructs and biological aspects of the network (which ultimately improves readability and simplicity of use), and *(ii)* preconditions that can be flexibly structured to incorporate any complex scenario—one could for example specify a scale-free network of cells, or adopt any other algorithmic graph construction.

6.3 The simulator engine

In this section we provide a brief outline of our simulator’s internal functioning. It is basically structured into two standalone, command-line tools: a front-end compiler for the surface language, and a back-end simulation engine producing output results in the form of a text file.

The front-end is a Java application embedding tuProlog [DOR05], an open-source Prolog engine with built-in Java programming. This compiler receives a specification file as described in the previous section, parses it, checks for basic correctness, and generates an intermediate file containing a list of commands to initialise the back-end engine. Currently, such an intermediate file is obtained by simply compiling the Prolog, namely, turning each universal declaration (a declaration with variables, and possibly preconditions) into a list of ground commands, one per correct instantiation of the declaration. For instance, link declaration in the example of the previous section gets compiled into the commands:

```
link c(1,1) c(1,2) 10000.0 field
link c(1,1) c(2,1) 10000.0 field
...
```

As this file can grow to more than a few hundred kilobytes in large networks, future work will make this intermediate file more compact, namely, deferring some parts of the Prolog resolution process to the engine initialisation module.

The back-end engine is a C++ program that receives the intermediate file and accordingly initialises all the proper data structures. The simulation process is based on the optimised Gillespie’s algorithm described in [GB00] and used in [VB08] which is called the *Next Reaction Method*, particularly efficient because it takes computational time proportional to the logarithm of the number of reactions, not to the number of reactions itself. It mainly consists in the computation, step by step, of only a small portion of the reaction propensity functions using proper data structures which store the dependency between propensity functions and reactions. Moreover it suggests the reuse, under certain condition, of the time interval to update the simulation time previously computed.

This work has been modified and adapted to tackle our computational model; namely:

Real entity	Protein	Gene Inactive	Gene Active
Bicoid	pBcd		
Caudal	pCad		
Tailless	pTll		
Hunchback	pHb	gHb0	gHb1
Knirps	pKni	gKni0	gKni1
Giant	pGt	gGt0	gGt1
Krüppel	pKr	gKr0	gKr1

Table 6.3: Proteins.

- A list of available actions is maintained over time, each representing a possible transition in the system, i.e., one chemical reaction inside each compartment of the network, and additionally, all the actions involving transfers through links.
- A dependency graph is built that links each action *a* to those whose rate should change when *a* is executed. This extends the basic definition in [GB00] with the idea that a link action possibly also affects chemical reactions in the target compartment.

The engine produces an output text file according to `out` declarations in the specification, which can be used to produce charts using standard tools like spreadsheets, `gnuplot`, or `Matlab`, as in the command-line version of SPIM [Phi06].

6.4 A model of Drosophila development

The model aims at reproducing the expression pattern of the gap genes so that the main actors of the model and their interactions are those of Figure 2.4.

6.4.1 Intracellular reactions

The reactions that model the intracellular behaviour directly implement the graph of Fig. 2.4, which provides a snapshot of the regulatory interaction among gap genes and among gap genes and maternal genes in each cell. We use 0 or 1 as suffix in the genes name to express the genes' activity—inactive and active respectively—and "p" or "g" prefix for representing the protein or gene form of the molecule type. The entities involved are listed in Table 6.3 and are expressed by the declaration:

```
molecule M where (M in [pBcd, pCad, pHb, pKr, pGt,
pKni, pTll, gHb0, gKr0, gGt0, gKni0, gHb1, gKr1, gGt1, gKni1]).
```

Protein synthesis is assumed to be an atomic event, so that transcription and translation are modelled with one reaction only. Gene activation is modelled through a reaction that changes

```
% gene hb regulation
reaction r(gHbAct00)  : [pBcd, gHb0] --> [pBcd, gHb1] rate 0.1678.
reaction r(gHbAct01)  : [pTll, gHb0] --> [pTll, gHb1] rate 0.0159.
reaction r(gHbAct02)  : [pHb, gHb0] --> [pHb, gHb1] rate 0.0317.
reaction r(gHbDeAct00)  : [pKni, gHb1] --> [pKni, gHb0] rate 0.39.
reaction r(gHbDeAct01)  : [pKr, gHb1] --> [pKr, gHb0] rate 0.0342.

% gene hb synthesis and degradation
reaction r(pHbSynth) : [gHb1] --> [gHb1, pHb] rate 41.032.
reaction r(pHbDegr) : [pHb] --> [] rate 0.165.
```

Table 6.4: Model's reactions for *hb* dynamic written using the MS-BioNET language.

the state of the gene from 0 to 1 once the activating protein is available. The reverse change causes gene inhibition. Gene deactivation as a consequence of activating protein detachment is modelled with a specific reaction.

An example of such reactions is given by the model of *hb* dynamics (and similarly for the others) which is specified in Table 6.4 with MS-BioNET language and formally described by the following chemical laws:

$$\begin{array}{lrcl}
(\texttt{gHB\_ACTIVATION\_00}) & \texttt{pBcd}\,|\,\texttt{gHb0} & \xmapsto{0.1678} & \texttt{pBcd}\,|\,\texttt{gHb1} \\
(\texttt{gHB\_ACTIVATION\_01}) & \texttt{pTll}\,|\,\texttt{gHb0} & \xmapsto{0.0159} & \texttt{pTll}\,|\,\texttt{gHb1} \\
(\texttt{gHB\_ACTIVATION\_02}) & \texttt{pHb}\,|\,\texttt{gHb0} & \xmapsto{0.0317} & \texttt{pHb}\,|\,\texttt{gHb1} \\
(\texttt{gHB\_DEACTIVATION\_00}) & \texttt{pKni}\,|\,\texttt{gHb1} & \xmapsto{0.39} & \texttt{pKni}\,|\,\texttt{gHb0} \\
(\texttt{gHB\_DEACTIVATION\_01}) & \texttt{pKr}\,|\,\texttt{gHb1} & \xmapsto{0.0342} & \texttt{pKr}\,|\,\texttt{gHb0} \\
(\texttt{pHB\_SYNTHESIS}) & \texttt{gHb1} & \xmapsto{41.032} & \texttt{gHb1}\,|\,\texttt{pHb} \\
(\texttt{pHB\_DEGRADATION}) & \texttt{pHb} & \xmapsto{0.165} & 0
\end{array}$$

where the gene inactive `gHb0` is activated by the proteins `pBcd`, `pTll` and its product `pHb`, while the gene active `gHb1` is repressed by the inhibitors `pKni` and `pKr`. The parameter search is driven by the results published in [PJRG06].

6.4.2 Cell graph and cell-to-cell communication

We performed experiments with a 10x100 grid built as shown in Fig. 6.2, which allowed molecules to diffuse along both the *x* and *y* axes. The horizontal axis represents the A-P position, while the vertical axis represents a portion of the D-V position, ranging from 45% to 55% of embryo's width. This last simplification is possible assuming that the activity of the cells along the *y* axis is quite similar for a fixed *x* coordinate.

Molecules diffuse crossing the cell's membrane and going into one of the four neighbouring cells chosen at random. An example of such reactions is:

$$\texttt{(pHB\_MOVE)} \qquad \texttt{pHb} \xmapsto{2.25} \texttt{pHb}^{\leadsto}$$

that coded in MS-BioNET appears as:

```
reaction r(pHbMove)  : [pHb] --> [firing(pHb)] rate 2.25.
```

7
ABM and the Application at Drosophila Development

In literature, agent-based systems, in particular Multi-Agent Systems (MAS), are considered an effective paradigm for modelling, understanding, and engineering *complex systems*, and biological systems in particular, providing a basic set of high level abstractions that make be possible to directly capture and represent the main aspects of such complex systems, such as interaction, multiplicity and decentralisation of control, openness and dynamism [MFD09, MAC$^+$07, MN10]. In this chapter I first describe the main characteristics of the agent-based model (ABM) and then the model I created for the morphogenesis of multi-celluar organisms, with the particular application at the *Drosophila* development.

7.1 The Computational Model

A MAS can be characterised by three key abstractions: *agents*, *societies* and *environment*. Agents are the basic *active* components of the systems, executing pro-actively and autonomously. Societies are formed by set of agents that interact and communicate with each other, exploiting and affecting the environment where they are situated. Such an environment plays a fundamental role, as a context enabling, mediating and constraining agent activities [WOO07].

By adopting MAS, biological systems can be modelled as a set of interacting autonomous components, i.e. a set of agents, and their chemical environment can be modelled by suitable agent environment abstractions, enabling and mediating agent interactions. In particular, MAS provide a direct way to model: *(i)* the individual structures and behaviours of different entities of the biological system as different agents (*heterogeneity*); *(ii)* the heterogeneous – in space and time – environment structure and its dynamics; *(ii)* the local interactions between biological entities/agents (*locality*) and their environment. A MAS-based simulation means executing the MAS and studying its evolution through time, in particular: *(i)* observing individual and environment evolution; *(ii)* observing global system properties as emergent properties from agent-environment and inter-agent local interaction; *(iii)* performing in-silico experiments. The approach is ideal then for studying the systemic and emergent properties that characterise a biological system, which are meant to be reproduced *in virtuo*. In the context of biological system, agent-based models can therefore account for individual cell biochemical mechanisms – gene regulatory network, protein synthesis, secretion and absorption, mitosis and so on – as well as the extracellular matrix dynamic – diffusion of morphogens, degradation and so on – and their dynamic influences on cell behaviour.

7.2 Agent-based Simulation Platforms

An ABM can be implemented upon specific platforms that already provide the main requirements of an ABM and its simulation, or using a general purpose programming language, such as Python, Java, and C++.

The development from scratch can be unwisely expensive given that this would require the autonomous development of many of the services already available in specialised agent modelling tools.

Several platforms have been developed in order to overcome this problem, providing the basic functionality for building correctly and easily an ABM and for automatically performing the simulation. Some of the more common capabilities included in such tools are: project specification services; agent specification services; input data specification and storage services; model execution services; results storage and analysis services. The most known platforms are: NetLogo [WC], MASON [LCRP$^+$05, Uni], Repast [NHCV07, Teaa], Swarm [Teab]. An interesting review is proposed in [RLJ06] where the same model is implemented upon each platform and the platforms are compared in terms of the quality of the documentation, the execution speed, the presence of tools for executing and observing simulation experiments and eventually other crucial features.

Generally speaking these platform provide a set of libraries to specify the project. Modellers create models by making a series of calls to the various functions within the modelling toolkit. It is the responsibility of modellers to ensure that the correct call sequences are used and that all of the required files are present. In exchange, modellers have great flexibility in the way that they define their models. In addition, or as a replacement of the previous approach, some platforms provide IDEs for supporting the modellers in the project description. The model can be edited using a graphic interface. IDEs also provide a built-in mechanism to compile or interpret and then execute models. The IDE supports are often quite easy to be used, but they are less flexible and do not always scale well to larger and more complex models as compared to the other project specification approaches: the main difficulty is in the organisation of the model code as it grows.

7.2.1 REPAST

Repast is the open-source, agent-based modelling and simulation toolkit I finally used in the exploration described in the following. It is a Java-based tool that supports also the description of the model through an IDE where agents behaviour is specified through a set of blocks composing a flowchart. Each block has a meaning such as Decision, Task, Loop, Join, End and so on. At the best of my experience the use of the flowchart is not adequate for building a model where the internal behaviour of the agent is quite complex and the interaction with a dynamic environment is constant and rich.

The simulation is executed upon a multithreaded discrete event simulator provided with a scheduler of actions. In this sort of simulation, the same set of agent behavior gets executed every tick, and what gets scheduled then is an event or, as it is called in Repast, an action which executes each phase of the agent behaviour.

7.3 An ABM of Drosophila Development

The model consists of a set of agents that model the cells and of a grid-like environment that model the extra-cellular matrix. Agent internal behaviour reproduces the gene regulatory network of the cell and agent interactive capability with the environment models the process of cell-to-cell communication mediated by the signalling molecules secreted in and absorbed by the extra-cellular matrix. The model aims at reproducing the expression pattern of the gap genes, before the pair-rule genes are activated.

7.3.1 Model of the cell

We model different cell processes: secretion-absorption diffusion of chemicals from and towards the environment, cell growth and cell internal dynamic—gene regulatory network in particular.

Chemical diffusion

Until cleavage cycle 13, there are not cell membranes surrounding cell cytoplasm and nucleus and the transport of material mainly interests the nuclear membrane. It involves also cell membranes once they grow. We do not make differences between these two stages, and we model the process of molecule secretion and absorption as facilitated diffusion— the literature lacks of information about the transport mechanisms of such transcription factors and about the rate of diffusion.

Gene regulatory network

Gene transcription begins with the binding at the gene promoter of one or more transcription factors. Gene transcription might also be repressed once transcription factors bind to other control regions called silencers. This activation/inhibition is stochastic [KEBC05] and highly depends on the concentration of transcription factors. For those genes whose transcription is regulated by a set of other gene products we define a probability of transcription as a sum of positive and negative contributions from the concentration of enhancers and silencers, respectively. The probability of transcription of *hunckback*, according to the graph of Fig. 2.4, is then calculated as:

$$\begin{aligned} P_{hb} = f([Bicoid]) + f([Hunkbuck]) + f([Tailess]) \\ -f([Knirps]) - f([Kruppel]) \end{aligned}$$

where f is a linear function with the proportionality constant representing the strength of interaction. Then if $P_{hb} > 0$ the protein is synthesised, otherwise the gene remains silent.

No distinction has been done in the model between anterior (a) and posterior (p) *hunckback* and *giant*, whose different expression only deals with the spatial distribution of maternal products.

Mitosis

According to Fig. 2.1 where we show how the number of cells varies in the first four hours of embryo development – until the cleavage cycle 14, temporal class 8 – we computed the rate of division as a function of time: cell division is fast and synchronous until cleavage cycle 9 and then it slows down and becomes asynchronous. The rate of division is then constant in the first hours of development (9.05 min^{-1}) and then it decreases until a low value (0.2 min^{-1}) , as it appears in Figure 2.2.

7.3.2 Model of the environment

The 3D tapered structure of the embryo, as in Figure 7.1, is modelled as a 2D section of the embryo along the antero-posterior axis (c) with the hypothesis that the dynamics along the other two axis, a and b, do not influence what happens along the c axis. The space scale is 1:3.33 according to the real dimension of the embryo where the antero-posterior axis is almost three times the dorso-ventral one a. Space is not continuous but grid like, and each location might be occupied both by a set of morphogenes and by a cell.

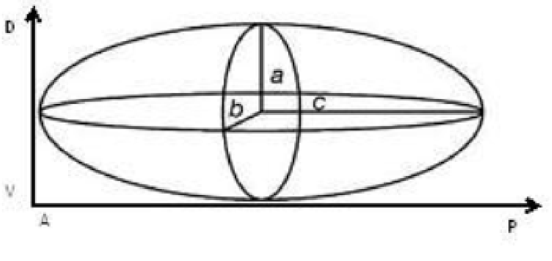

Figure 7.1: 3D structure of real embryo

The environment has its own dynamic, which mainly consists in the diffusion of morphogenes from region with bigger concentration to region with lower concentration, according to the *Fick's low* that the diffusive flux is proportional to the local concentration gradient [SH05]. This law is used in its discretised form.

8 The Parameter Optimisation Module

In this chapter I address the problem of tuning parameters of a biological model, in particular a simulator of stochastic processes, as MS-BioNET presented in Chapter 6. The task is defined as an optimisation problem over the parameter space in which the objective function to be minimised is the distance between the output of the simulator and a target one. We tackle the problem with different metaheuristic algorithms: *trajecoty methods* – such as iterated local search – on one side, and *population-based methods*—such as particle swarm optimisation and Covariance matrix adaptation evolution strategy –on the other.

8.1 Metaheuristics for parameter tuning

Metaheuristics are search strategies upon which approximate algorithms for continuous and combinatorial optimisation problems can be designed. Notable examples of metaheuristics are simulated annealing, tabu search, genetic algorithms and ant colony optimisation [BR03]. The first two are examples of so-called *trajectory methods*, which are search processes that can be defined as (stochastic) trajectories over a search graph [HS05]. Conversely, the latter two are prominent cases of *population-based search algorithms*, which perform a search process characterised by an iterative sampling of the search space. The probabilistic model defining the sampling distribution is adapted during search so as to concentrate search around regions containing promising solutions.

The problem of parameter tuning of a system can be cast into an optimisation problem once an error measure (or a performance measure) is defined. Thus, the problem is to finding an assignment to the parameters such that the error is minimised. Depending on the parameter domains (i.e., continuous or categorical parameters), the problem can be continuous, discrete or mixed-integer. The objective function, that we suppose is to be minimised, defines an error landscape which is explored by the search algorithm.

Metaheuristic algorithms, both for discrete and continuous variables, are usually effective in tackling this kind of problems because they can exploit the information provided by the objective function and learn local properties of the error landscape.Metaheuristics have been used to tune parameters in search algorithms. Recent results on this subject are presented in [HHS07] and references therein. In [RFEB06], a scatter search algorithm is used to calibrating parameters of a mathematical model based on ordinary differential equations and in [Ban08] a survey on optimisation techniques in computational systems biology is provided.

The adopted approach is based on the assumption that the model is a black-box that can be controlled by providing as the input a set of parameter values and whose output can be observed. The optimisation process evaluates the output and uses this piece of information to guide search in the parameter space. The choice of the metaheuristic algorithm depends on the problem characteristics and mainly on the parameter domains and the objective function evaluation. In the next section we introduce the framework of our optimisation system and we detail its usage in a case study.

8.2 A general framework architecture

In this section I illustrate the framework that defines the approach for tackling parameter tuning in biological models. The framework describes any situation in which parameters of a system are to be tuned. As from Figure 8.2, the main entities of the framework are the *model/simulator*, the *evaluator*, the *target* and the *optimiser*.

The *model* can be, in general, any model of a system, possibly stochastic. The *simulator* is responsible for the model execution. If the model designed is a stochastic model of a biological system, the performance is evaluated by collecting statistics on sample executions. For brevity, in the following I will simply refer to the *model/simulator* component as *simulator*.

The *evaluator* is the component in charge of evaluating the performance of the system at a given parameter setting. This component is crucial and it is problem dependent because it has the responsibility of measuring the performance of the model when a particular parameter configuration is chosen. This measure depends on the kind of the system that has to be modelled and it provides the primary information that guides the optimisation/search process. Thus, the designer has to define a proper distance function to evaluate the dynamics of the simulated system in comparison with a target behaviour. For example, the main characteristics of the final attractor can be considered, such as the average values of the output variables, their periods and the oscillation amplitude. Other characteristics can be also extracted in the frequency domain, for example by applying a Fourier transform. In addition, any method for comparing multi-dimensional data series can be used. The *evaluator* possibly makes use of a *target* behaviour that the simulator has to reproduce. For example, the target may be an attractor whose characteristics the system has to reproduce in its steady-state.

The last component, the *optimiser*, has the goal of finding an optimal parameter setting. It is important to remark that the goal is to find any parameter setting that produces in the simulator the desired behaviour, therefore the concept of ‘proven optimal solution’ is meaningless because the objective of the designer is simply to achieve a model calibration satisfying the requirements. The optimiser can be any optimisation algorithm, but in my researches I consider only metaheuristic algorithms.

As for the simulator, details of two possible alternative are given in Chapter 6 and Chapter 7,

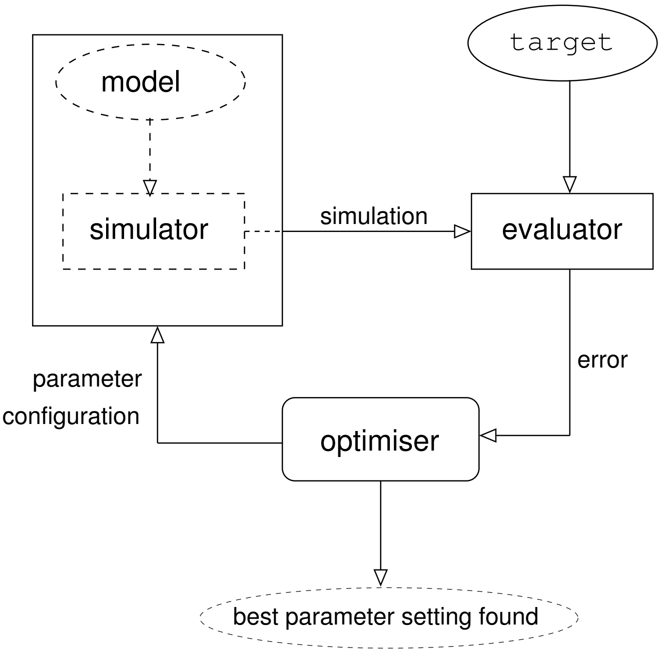

Figure 8.1: Framework describing the parameter tuning process of a biological model.

while the evaluator component of the framework, which is very specific for the problem, is described in Chapter 9 once the model of the case study adopted in this work is provided.

An other example application of the overall framework is given in [MR09], where it is applied at a well-known biological case study —the MAPK cascade, a unicellular system, whose discussion is a bit outside of this work, but quite useful as an example of how the evaluator can be designed according to the specific dynamic of the case study.

In the following I finally illustrate the metaheuristics implemented in the optimiser component.

8.3 The metaheuristics in the optimiser

In this section I will give some details more on the optimisation methods used in the optimiser. More details can be found in [BR03].

8.3.1 First and Best Improvement

First and Best Improvement are the extremes of the more general basic method called iterative improvement, belonging to the class of trajectory methods. Each move inside the search space is allowed only if there is an improvement in the value of the objective function. The search stops when a local minimum is found. With the First Improvement, while scanning the neighbourhood $N(s)$ of the actual solution s, the first solution found better then the actual one is returned. With the Best Improvement the solution with the lowest objective function value in the neighbourhood is returned. The Best Improvement pseudo-code is reported in Algorithm 1.

Algorithm 1 Best Improvement high-level description

```
$s \leftarrow$ Initialise population
while no improvement is possible do
  $s \leftarrow$Improve(N(s))
end while
output: local optimum found
```

8.3.2 Iterated Local Search

The basic mechanism of Iterated Local Search is the Iterated Improvement but in addition once it finds a local optimum it perturbs the solution and it restarts local search. The goal of the perturbation is to enable the system to escape from local minimum just found so to finally realise a trajectory along local minima. Obviously the perturbation has to be sufficient for escaping from the basin of attraction of the actual local optimum as well it has not to be too large so to implement a random restart local search.

To be this method really effective, so that a trajectory along local optima is performed, the following scheme has to be applied iteratively (also shown with the pseudo-code is reported in Algorithm 2):

1. Execute a local search (such as Best Improvement) from an initial state $\hat{s}$ so to find a local minimum
2. Perturb $\hat{s}$ to obtain s'
3. Execute again the first step to obtain a new local minimum $\hat{s}'$
4. Given an *acceptance criterion* replace or not $\hat{s}$ with $\hat{s}'$

The acceptance criterion gives feedback to the perturbation action deciding whether or not accept the $\hat{s}'$ found. It can accept it only if there is an improvement with respect to $\hat{s}$ or it can accept it always. In between there are different possibilities that can be designed.

8.3.3 Particle swarm optimisation

Particle swarm optimisation (PSO) is a population-based metaheuristic particularly effective when dealing with problems defined over continuous variables [Cle06]. Population-based algorithms are suitable for this kind of applications because they are very robust against rugged

Algorithm 2 Iterated Local Search high-level description

```
$s_0 \leftarrow$ Initialise population
$\hat{s} \leftarrow$ Improve(N($s_0$))
while termination condition not met do
    $s' \leftarrow$ Perturbation($\hat{s}, history$)
    $\hat{s}' \leftarrow$ Improve(N($s'$))
    $\hat{s} \leftarrow$ ApplyAcceptanceCriterion($\hat{s}, \hat{s}', history$)
end while
output: local optimum found
```

search landscapes, in which there are many local minima. PSO is an optimisation technique inspired by the metaphor of social interaction, for example bird flocking and fish schooling. Besides the metaphor, PSO is defined by formal mathematical models and has been proven to be very effective in solving optimisation problems, mainly continuous ones. The algorithm iteratively samples the search space by a *population* of samples, called *particles*. Particles have their own position and velocity that are updated each iteration by a rule that takes into account the quality of solutions represented by the particles. Positions and velocity are updated trying to gather the swarm toward good solutions, while keeping a form of exploration so as to balance search intensification and diversification. The PSO pseudo-code is reported in Algorithm 3.

Algorithm 3 PSO high-level description

```
Initialise population
while termination conditions not met do
    for $i = 1$ to pop_size do
        if $f(\overrightarrow{x_i}) < f(\overrightarrow{p_i})$ then $\overrightarrow{p_i} = \overrightarrow{x_i}$
        $\overrightarrow{p_g} = \mathsf{best}(\{p_j | j \text{ is a neighbour of } i\})$
        for $d = 1$ to $n$ do
            $\phi_1 = \mathsf{randomFloat}(0, \phi_{max})$
            $\phi_2 = \mathsf{randomFloat}(0, \phi_{max})$
            $v_{id} = \omega v_{id} + \phi_1(p_{id} - x_{id}) + \phi_2(p_{gd} - x_{id})$
            $v_{id} = \mathsf{sign}(v_{id}) \cdot \mathsf{sign}(|v_{id}|, v_{max})$
            $x_{id} = x_{id} + v_{id}$
        end for
    end for
end while
output: best solution found
```

Particle positions are denoted by n-dimensional vectors $\overrightarrow{x_i}$ and velocities by vectors $\overrightarrow{v_i}$, for $i = 1, \ldots,$ pop_size. The best solution found by particle i since the beginning of the search is denoted by $\overrightarrow{p_i}$, while the best solution found so far by its neighbouring particles (possibly all the other particles) is denoted by $\overrightarrow{p_g}$. Positions and velocity are stochastically adjusted in such a way that each particle searches in the surroundings of the best solution it found and the best solution found by its neighbours. The objective function to be minimised is denoted by $f(\cdot)$. ϕ_{max} and ω are parameters of the algorithm.

8.3.4 Covariance matrix adaptation evolution strategy

The covariance matrix adaptation evolution strategy (CMA-ES) is an evolutionary algorithm for difficult non-linear non-convex (i.e. there may be several local minima and maxima) optimisation problems in continuous domain. The CMA-ES is typically useful in those cases in which other more common methods fail due to a rugged search landscape. Technical details on the algorithm are given in [HO01].

Part IV

Experiments

9 Applications on Drosophila Melanogaster Morphogenesis

In this Chapter the results of the simulation performed with both the MS-BioNET model and the ABM are shown. Results are compared with experimental data acquired in on-line sources.

9.1 Simulation and results

The models aims at reproducing the expression pattern of the gap genes, before the pair-rule genes are activated.

The model aims at reproducing the expression pattern of the gap genes, before the pair-rule genes are activated. In [PJRG06] the same phenomenon is modelled through a mathematical model – a reaction-diffusion partial differential equation (see Section 4.1.2 – and experimental data are used in order to estimate the model parameters.

We used the experimental data available online in the FlyEx database [PPB$^+$04, SKK$^+$08][1]. "The data include quantitative wild-type concentration profiles for the protein products of *bcd*, *cad*, *hb*, *Kr*, *kni*, *gt*, and *tll* during cleavage cycles 13 and 14A, which constitute the late syncytial blastoderm stage of *Drosophila* development" [PJRG06]. These data are used as initial condition and to validate the model dynamic. Expression data from cleavage cycle 11 are used as the initial condition—see Figure 9.1. The concentration of proteins are unitless, ranging from 0 to 255, at space point *x*, ranging from 0 to 100 % of embryo length.

9.1.1 Simulation and results with MS-BioNET

Qualitative results

Results charted in the 2D grid are shown in Figure 9.2 (left) for expression of *hb, kni, gt, Kr* at the eighth time step of cleavage cycle 14A (14tc8 for short). Experimental data are also provided in Figure 9.8 (right) with 2D Atlas reconstructing the expression level of the four genes in A-P sections of the embryo. These results are published in [MV10]. A qualitative

[1] `http://urchin.spbcas.ru/flyex/`

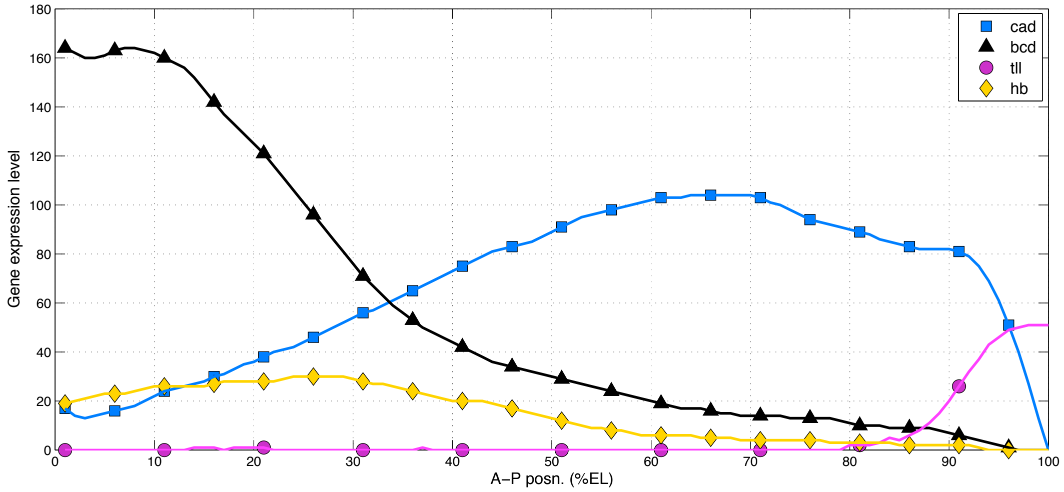

Figure 9.1: Experimental data at cleavage cycle 11 of genes with non-zero concentration: maternal genes *bcd, cad, tll* and the gap gene *hb*.

comparison shows that the expression pattern of genes *hb, kni, gt* and *Kr* nicely fit the spatial distribution shown in the experimental data: *hb* is expressed in the extreme left pole until about 45% of embryo length and on the right between about 85% and 95%; *kni* is expressed mostly between 65% and 75%; *gt* is correctly reproduced on the left, but it loses precision on the right where its expression slightly overlaps *hb*; while *Kr* properly appears between 40% and 60%. This shows that the proposed framework smoothly allows to check the qualitative validity of our working model against the sought embryogenesis phenomenon.

First quantitative results

Quantitative results showing the expression of genes *hb, kni, gt, Kr* at the eighth time step of cleavage cycle 14A, averaged on the y-DV axis of the embryo, where the cells activity is assumed to be quite similar (as already explained in the model description in Section 6.4.2) are charted with respect to the x coordinate in Figure 9.3.

On the model parameters fitting

The model parameters are: *(i)* diffusion constants of morphogenes motion; *(ii)* rates of gene interactions; *(iii)* rates of protein synthesis. In [PJRG06] a set of all these parameters is published. The results shown in Figure 9.2 and Figure 9.3 are obtained imposing the parameters initial values at the ones estimated for the Unc-GC model there, and then a bit modified by hand for better fitting the experimental data.

If qualitative results are quite satisfactory showing the formation of a correct pattern where

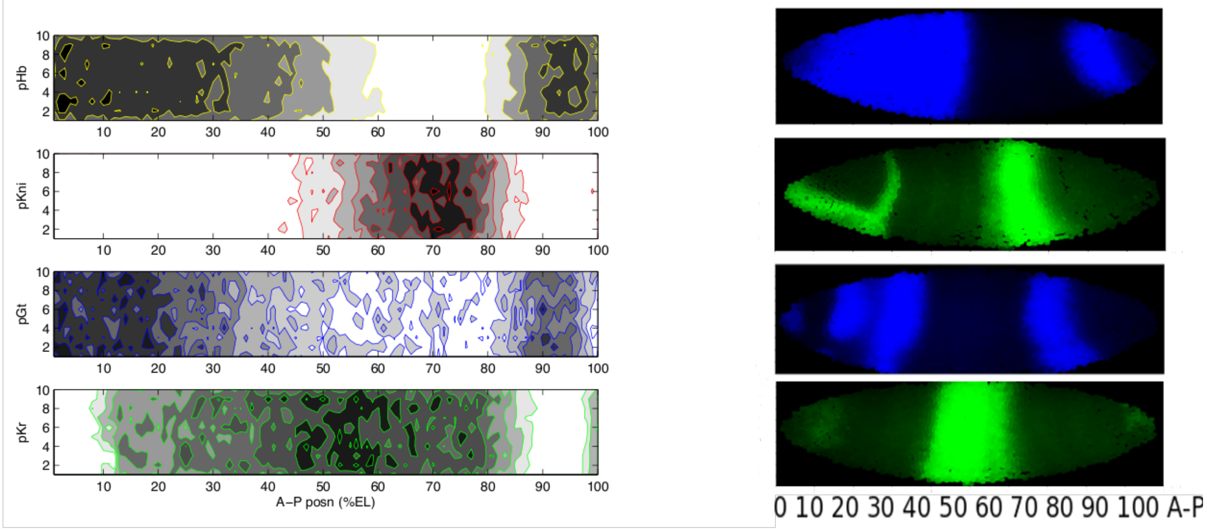

Figure 9.2: Simulation results for the four gap genes *hb, kni, gt, Kr* at a simulation time equivalent to the eighth time step of cleavage cycle 14A (left) and the corresponding experimental data (right)—% A-P length on the *x* and % D-V width on the *y*. Reconstructed images from [PPSR09].

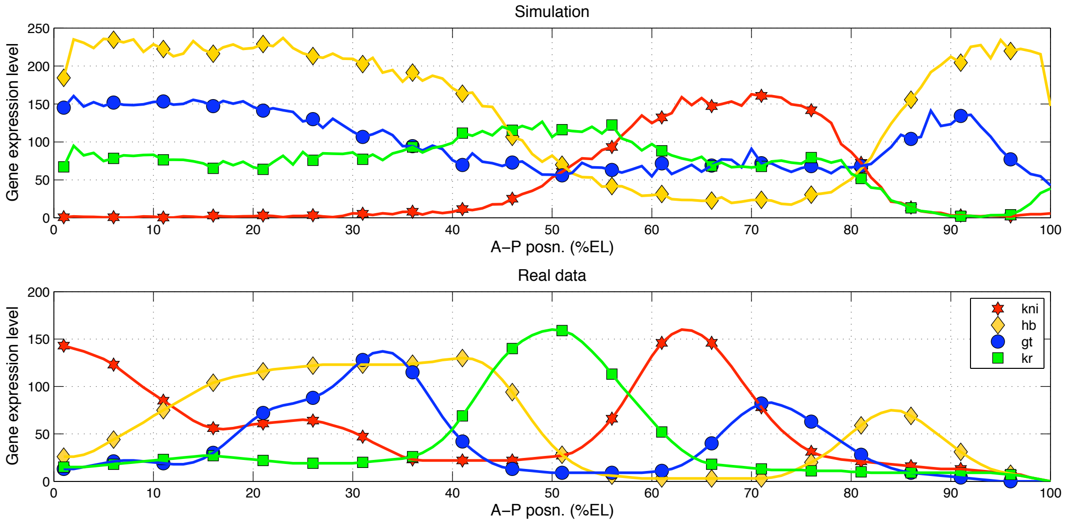

Figure 9.3: Quantitative simulation results obtained with MS-BioNET for the four gap genes *hb, kni, gt, Kr* at a simulation time equivalent to the eighth time step of cleavage cycle 14A (top) and the corresponding experimental data (bottom)

genes are expressed in their proper regions, the quantitative ones are not completely satistactory as soon as the level of gene expression is, in some point of the space, too high or too low. For this reason the module for automatic parameter fitting have been used.

Designing the evaluator component of the optimisation module means to compute a measure of error for the simulator output. The simulator output is a 40x100 matrix, where in each row is reported the expression value of a particular gene in a fixed y and for all the length of the embryo. In particular the first ten rows are devoted to the *hb* gene, followed by *kni, gt, Kr*. In order to compute the error, for each gene is computed a vector, whose length is 100, which represents the average value along the y coordinate for each x, so that for instance:

$$hb_i = \frac{\sum_{j=1}^{10} hb_{j,i}}{10} \qquad where \;\; i \in [1,100]$$

being i the index of column, and j the index of row. The process is graphically represented in Figure 9.4, where for the seek of space the output matrix is shortened in a 20 X 100 matrix while the experimental data shown are the ones at cleavage cycle 13. The 10x100 matrix is thus elaborated to produce a 4x100 matrix of mean values.

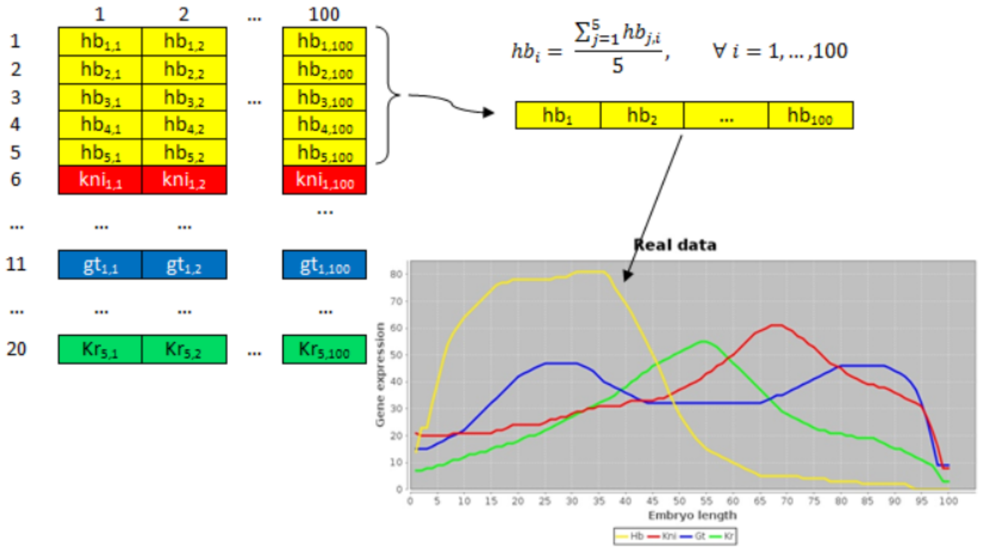

Figure 9.4: Pre-processing the simulator output to compute the error with respect to real data

Such a resulting matrix is used for computing the error, *i.e.* the euclidean distance with the – same dimension – matrix of experimental data. In particular given the elaborated simulator output O, whose elements are $o_{j,i}$, and the target matrix T of experimental data, whose elements are $t_{j,i}$ – where j correspond to a specific gene and i to the x coordinate – the total error E_{TOT} is computed as the sum of the euclidean distances of each gene with respect to their desired behaviour, with the formula:

$$E_{TOT} = \sum_{j=1}^{4} \sqrt{\sum_{i=1}^{100} (o_{j,i} - t_{j,i})^2}$$

The goal is therefore to minimise such error measure. The optimisation algorithm choose is the CMA-ES which has finally been identified as the one giving best results by performing different experiments.

Quantitative results after the optimisation process

In order to address the problem the overall window from cleavage cycle 11 to 14tc8 is split in two windows: *(i)* from 11 to 13 and *(ii)* from 13 to 14tc8.

Quantitative results obtained from 11 to 13 before and after the optimisation process are shown in Figure 9.5 and Figure 9.6 respectively. A different display is given so to better emphasise the differences between simulation results and experimental data and the improvements obtained with the application of the optimisation.

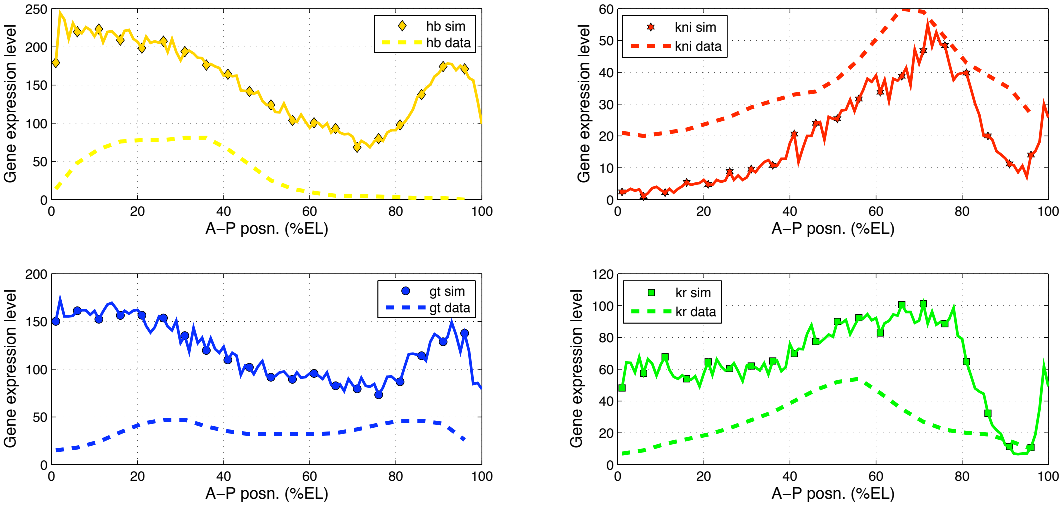

Figure 9.5: Quantitative simulation results obtained with MS-BioNET at cleavage cycle 13 **before** optimisation

For what it concerns the second time window still not satisfactory results are obtained, and much more work has to be done for improving the results of Figure 9.3.

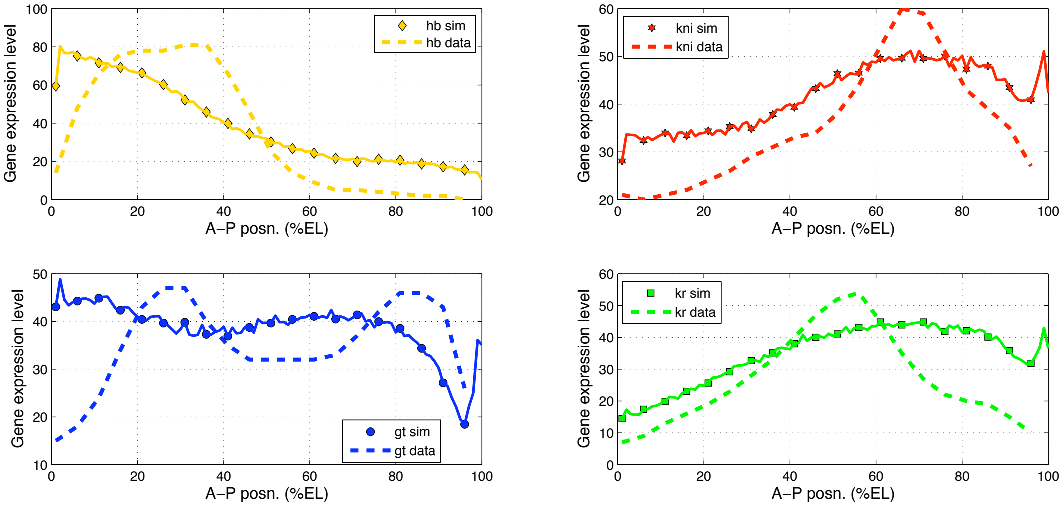

Figure 9.6: Quantitative simulation results obtained with MS-BioNET at cleavage cycle 13 **after** optimisation

9.1.2 Simulation and results with ABM

Qualitative results

Qualitative results charted in the 2D grid are shown in Fig. 9.7 (top) for expression of *hb, kni, gt, Kr* at the eighth time step of cleavage cycle 14A. The image shows for each cell of the embryo the genes with higher expression. It clearly displays the formation of a precise spatial pattern along the A-P axis but it does not give any information about gene expression level. Experimental data are also provided in Fig. 9.7 (bottom) with 2D Atlas reconstructing the expression level of the four genes in A-P sections of the embryo.

Quantitative results

More precise information about simulation behaviour are given with the quantitative results provided in Fig. 9.8. A comparison shows that the expression pattern of genes *hb, kni, gt* and *Kr* nicely fit the spatial distribution shown in the experimental data: *hb* is expressed in the left pole until about 45% of embryo length and while it doesn't appear on the right as it should between about 85% and 95%; *kni* is correctly expressed on the extreme left and between 65% and 75% but it is slightly over-expressed on the right; *gt* is reproduced in the correct regions but over-expressed in the extreme left and slightly under-expressed between 20% and 30%; while *Kr* properly appears between 40% and 60%.

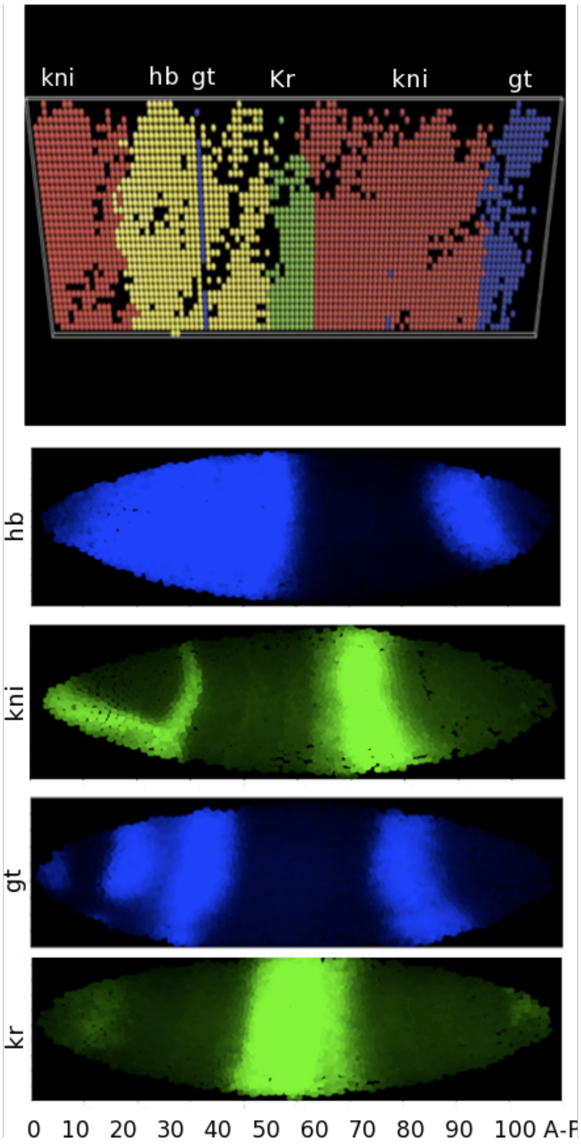

Figure 9.7: Qualitative results

On the model parameters

Model parameters are: *(i)* diffusion constants of morphogenes motion; *(ii)* rates of gene interactions; *(iii)* rates of protein synthesis. Few data are available in literature for inferring the diffusion constants. The work of GWM$^+$07 that calculates the diffusion rate for *Bicoid* is took as inspiration for imposing the values of diffusion for all the morphogenes at 0.3 $\mu m^2/sec$. The rates of gene interactions and of protein synthesis are determined by hand. As future work it will be possibly apply the module for automatic parameter search.

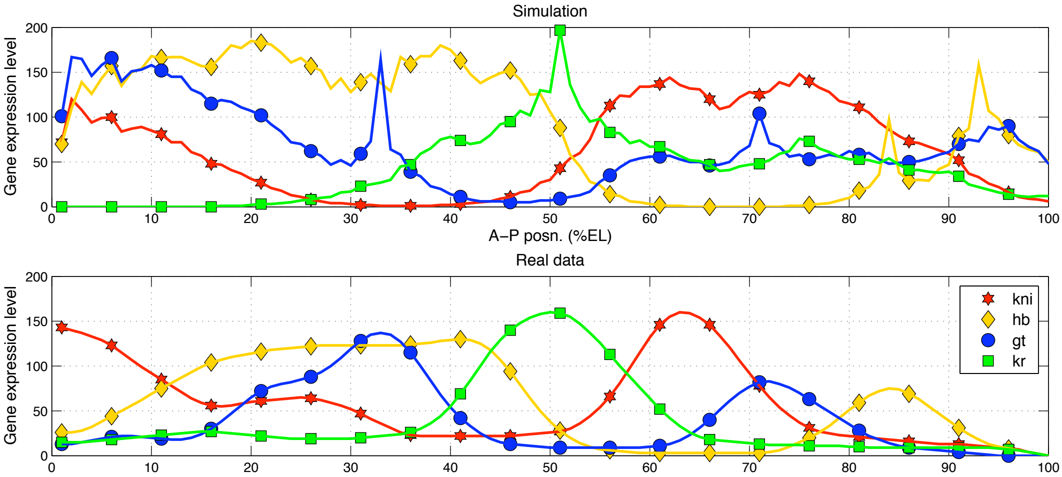

Figure 9.8: Quantitative simulation results for the four gap genes *hb, kni, gt, Kr* at a simulation time equivalent to the eighth time step of cleavage cycle 14A (top) and the corresponding experimental data (bottom)

Part V

Further Application

10
Towards a Model for Designing Pervasive Systems

Inspired at the mechanisms and emergent results of the morphogenesis, and at the idea discussed in the previous chapters, a new research trend has been faced with the goal of designing Pervasive Systems so to realise computational systems able to self-organised in elaborated structures. The key biological phenomenon of the diffusion in the environment of morphogens, that create fields of substances and react with other substances to generate precise spatial structures, is a useful metaphor for artificial systems where different entities, such as services, move and autonomously distribute in the network to reach different areas of the infrastructure, where they can interact with other entities, such as requests. A more detailed description of the general context and ideas presented in this chapter is given in VCMZ11.

This work is developed within the EU-funded project SAPERE: Self-aware Pervasive Service Ecosystems (Grant No. 25873).

> The objective of SAPERE is the development of a highly-innovative theoretical and practical framework for the decentralized deployment and execution of self-aware and adaptive services for future and emerging pervasive network scenarios. The framework will be grounded on a foundational re-thinking of current service models and of associated infrastructures and algorithms. In particular, getting inspiration from natural ecosystems, the project will demonstrate and experiment the possibility of modelling and deploying services as autonomous individuals in an ecosystem of other services, data sources, and pervasive devices, and of enforcing self-awareness and autonomic behaviours as inherent properties of the ecosystem, rather than as peculiar characteristics of its individuals only [1].

In particular in this chapter is described the application of MS-BioNET (see Chapter 6) for designing “self-organising morphogenetic systems”, *i.e.*, artificial systems that organise themselves exploiting bio-inspired morphogenetic processes.

[1] `http://www.sapere-project.eu/`

10.1 Pervasive service systems and spatial computation

Today's network scenarios face an increasing availability of pervasive sensing and actuating devices (RFID tags, PDAs, localisation devices), which are able to densely populate our everyday environments with digital information of users and locations, and tightly integrate with the Web—seen as a shared space for providing business as well as social services. This will eventually lead to the emergence of very dense, spatially distributed infrastructures providing general-purpose digital services: traditional Web services (social networks, video broadcasting) enriched with new means of interaction (e.g. smaller displays and cameras) and exploiting contextual information; pervasive location-based information services (finding the nearest post-office, market goods, etc.); services to coordinate activities of situated users (intelligent lights and signs, user-adaptive advertisement displays), services that encapsulate pervasive devices and make them accessible and retrievable in the network (e.g. by computational fields that facilitate retrieval in case of network mobility [MZ09]), and so on.

A coordination infrastructure supporting such highly decentralised and dynamic scenarios should act as a sort of "distributed space" reifying the existence of services and supporting their interaction and evolution. Proper coordination laws are to be designed and deployed over such a space that can guarantee properties of self-organisation, self-adaptation and self-management, in order to tolerate openness in service functionality, user data and needs.

In this framework, the notion of *pervasive service* can be used to uniformly represent concepts such as software functionality, pervasive devices, data, knowledge, and signals. A pervasive service gets injected in one location and possibly diffuses around, interacting and competing in a context-dependent way in each node of the network. Accordingly, each pervasive service will happen to effectively work in a region of the whole space, composed of one or more "niches" where the context is favourable. Namely, the state and behaviour of each service is to be understood as a spatial concept, while the dynamics of the whole system of pervasive services is naturally seen as a form of spatial computation.

Hence the possibility to adopt in the framework rules inspired by morphogenetic processes, which can be useful to control the emergence of a proper spatial structure, such as "niches" into which services can autonomously be distributed.

Such services, forming a sort of "ecology" as argued in [ZV08], should be managed by an infrastructure addressing the following issues:

- *Spatial Distribution* — The networks over which the infrastructure is to be deployed can span different sizes (from large-scale sensor networks such as a traffic control system, to smaller-scale networks such as the intranet of a company, a market, or a museum), topologies (fixed for traffic sensors, or dynamic when PDAs enter the picture), and metrics (the actual distance in space, number of hops to reach one another, communicating bandwidth of network links, and so on). The infrastructure should then generally handle a possibly dense "space" of localities (nodes) where services live (run, move, diffuse).

- *Context-Awareness* — The infrastructure should naturally match the inherent spatially-

and socially-situated nature of services, their users, and their environment. In some regions of the network, services might find a suitable context to operate, while in others they might simply be unnecessary and thus be removed for the sake of optimisation. Hence, the infrastructure should be able to adapt its coordination behaviour to the contextual information available, and to the logical and physical characteristics of each location and its proximity.

- *Adaptivity* — Properties of self-organisation, self-adaptation and self-manage-ment are to be inherently enabled, including the possibility of services to automatically bind to each other (to compose, or to match request and response), decay when not being used, move to more favourable regions of space, compete for the use of common resources, adaptively retrieve each other, and so on. Since the network is highly decentralised and highly dynamic, such behaviour should be obtained without a global manager monitoring the overall network, but rather by emergence from the local interaction of services in each node.

- *Openness* — The infrastructure should be able to tolerate – even in the long-term – open models of service production and usage, addressing unforeseen new classes of services provided, new user needs, and changes in network topology. In principle, any repair of system applications should be seen within the model, namely, through proper services injected in the network to improve or change on-the-fly existing ones, moving the overall system behaviour towards a new equilibrium.

Accordingly, we can envision the infrastructure for pervasive services as a sort of "spatial computer", running in a possibly large and mobile network, and providing a ground for spatially-distributed services to be injected, globally or partially diffuse, interact and compete with each other, spatially and temporally diminish and decay, and contextualise in each subregion of the space.

The problem of finding good metaphors for designing and implementing this kind of infrastructure has been typically tackled in the literature by relying on nature-inspired approaches: physical metaphors [Cro08, MZ06], chemical metaphors [BD06, VCO09], biological metaphors [BCD$^+$06, BB06], together with metaphors focusing on higher-level ecological/social metaphors (e.g. trophic networks [Agh08, VRVZ08]). In this work, we propose the morphogenesis and biochemical working laws as proper metaphors that can suitably support the required self-* features that pervasive computing calls for. In particular the computational model described in Section 6.1 and the overall framework presented look particularly promising for addressing such scenarios.

In the resulting biochemical inspiration, pervasive services are associated with an activity/relevance numeric value resembling a chemical concentration, and measuring the extent to which the services can influence the coordination state. For example, services with low concentration would be rather inert, taking part in coordination at a very low frequency. By relying on proper chemical-like laws, to be established into each location of the distributed infrastructure, i.e., in each node/compartment of the graph constituting the whole distributed system, concentration of services is automatically changed over time so as to make interesting self-* coordination properties emerge.

10.2 Spatial coordination patterns

In the following it is explained by two examples how appropriate chemical laws can support interesting scenarios of spatial coordination using key morphogenetic mechanisms –namely, field formation and gradient ascent.

10.2.1 Fields

Computational fields – also known as gradients [BBVT08] – constitute a key building brick for pervasive computing systems [MZ09]. They are distributed data structures that are initiated (i.e. "pumped") from a source node and diffuse in their surrounding until each node of the network reaches a stable field amplitude (called the field value in that node): such a value is meant to depend only on the estimated distance of the node from the source. This spatial data structure is primarily used to project some data from the source to a neighbouring region, so that all "agents" in that region can not only be aware of such data, but also retrieve the source by simply moving on a step-by-step basis towards the source, namely, choosing at each step the neighbouring node whose field value indicates the node nearer to the source. This mechanism can be primarily seen as physically inspired, for it mimics the way gravitational fields work [MZ09], but it also resembles diffusion mechanisms either at the chemical level, or at biological level such as in stigmergy. In the pervasive display infrastructure, fields can be pumped by displays with specific characteristics (e.g. a very wide screen may be willing to attract suitable visualisation services), by user profiles (a user interested in sports attracts services visualising sport news), and even by visualisation services (attracting profiles of users in a given proximity, so as to reason about which visualisation policy should be used in the near future).

In our framework there can be many ways of creating a field through diffusion laws; we focus here on one way described by the following chemical laws:

$$\begin{array}{lrcl}
\texttt{(PUMP)} & \texttt{PUMP} & \xmapsto{r\_pump} & \texttt{PUMP}\,|\,\texttt{FIELD} \\
\texttt{(DECAY)} & \texttt{FIELD} & \xmapsto{r\_decay} & 0 \\
\texttt{(DIFFUSE)} & \texttt{FIELD} & \xmapsto{r\_diff} & \texttt{FIELD}\,|\,\texttt{FIELD}^{\rightsquigarrow}
\end{array}$$

and specified in Table 6.2 with MS-BioNET language.

Initially, assume a PUMP token with concentration 1 is inserted into the source node. The (PUMP) rule starts spawning units of the FIELD service, such that the FIELD concentration starts rising. On the other hand, the (DECAY) rule makes any single FIELD unit disappear after an average $1/r\_decay$ time units. As discussed in previous section, decay balances

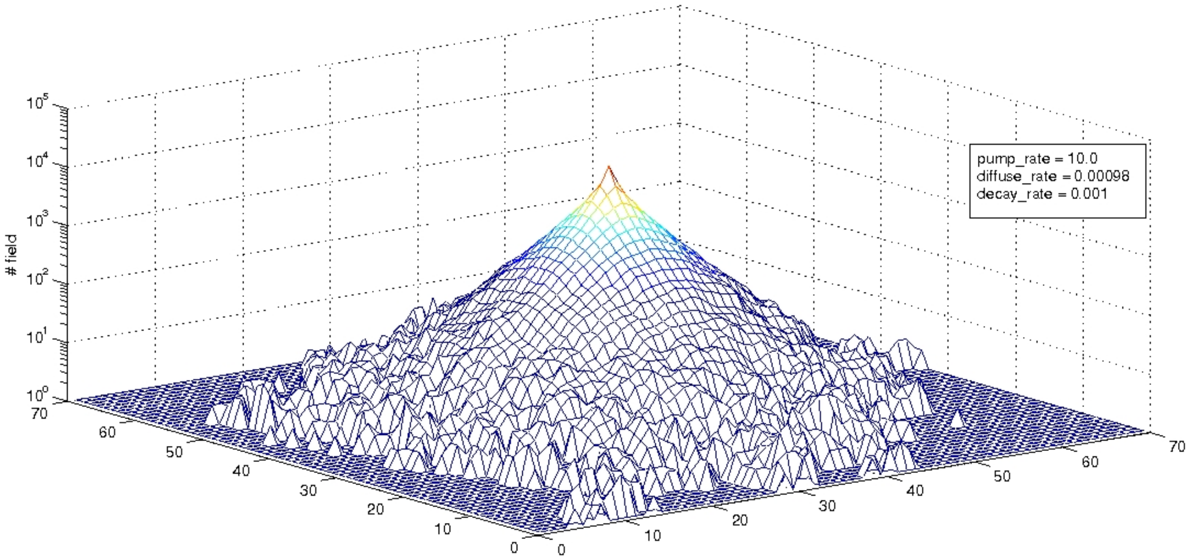

Figure 10.1: The stabilised spatial structure induced by a field (z-axis reports concentration of field service).

pumping, until they reach an equilibrium where the FIELD concentration stabilises to about r_pump/r_decay. Additionally, and mimicking the behaviour of population dynamics as in [Ber92], rule (DIFFUSE) is used to spawn new "firing tuples", representing FIELD units that will be moved to a neighbouring node. As a result, the computational field starts diffusing in the neighbourhood where it is also subject to decay: the nearest it is to the source, the higher is the influence of the "pumping force", hence the higher is the field value.

An estimation of the overall "size" of the field – namely the sum of field values in all network nodes – can be given considering that new field units are added with fixed rate *r_pump*, and that each of them either decays with rate *r_decay*, or creates a clone copy with rate *r_diffusion*; accordingly, the overall size can be shown to stabilise to:

$$field\text{-}size = \frac{r\_pump}{r\_decay - r\_diffusion} \qquad (\text{if } r\_decay > r\_diffusion)$$

Now, if $r\_decay >> r\_diffusion$ then we expect decay to be too high, and the field horizon to remain too small, so that no useful field is actually created; otherwise, if $r\_decay \leq r\_diffusion$ the field diverges and concentrations grow everywhere indefinitely—a situation we shall avoid in general. With this consideration in mind, we can set up a first experiment: over a uniform 70 × 70 network grid a field is pumped using parameters $r\_pump = 10, r\_decay = 0.001, r\_diffusion = 0.00098$. The result of the simulation is reported in Fig. 10.1, which shows how the field value decreases until becoming zero at about distance 30 from the source, i.e., 30 is the field horizon. The size of the resulting stabilised field is about $500,000$ units, and the observed amplitude in the source is about $20,000$, which is twice the expected quantity due to pumping and decaying

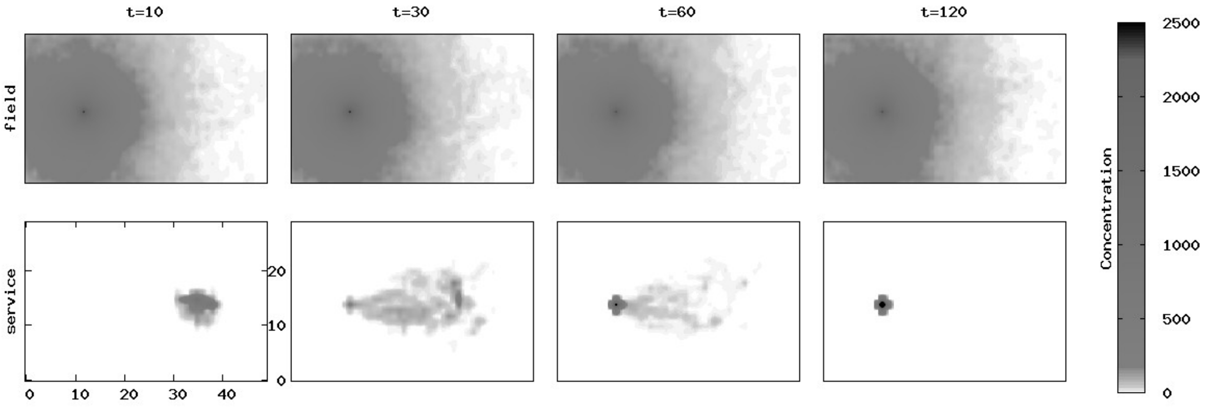

Figure 10.2: Field-based attraction: a service enacts a field (top) used by requests to reach a device (bottom).

(*r_pump*/*r_decay*): this effect is due to the field units that, by diffusion, go back to the source from neighbouring nodes.

10.2.2 Ascending a field

We now consider a typical retrieval scenario of spatial computing (see e.g. [MZ06]), which is the main application of computational fields. Considering the pervasive display infrastructure, suppose a display service *d* located in a node pumps a field, so as to be retrieved in a suitable region. A visualisation service *r* (requester) situated in that region is attracted by the display, because it has compatible visualisation parameters (e.g. the display is sufficiently wide): *r* should reach the location of *d* in order to use it, and it can do so by ascending the field. Once the field structure is generated, ascent of a requester is obtained by the simple chemical rule:

$$\texttt{(ASCENT)} \qquad \texttt{REQ} \xmapsto{r\_asc} \texttt{REQ}^{\leadsto}(\texttt{FIELD}, |f|)$$

`REQ` matches any service that should retrieve some target, `FIELD` matches the field signal generated by the target, $|f|$ is a positive gradient factor measuring sensitivity to gradient slope, and finally *r_asc* is the ascent rate—namely, the maximum speed of ascension (corresponding to the case of perfect match as usual). Note that, unlike the diffusion law, ascent law is a *relocation* law, which does not spawn a new unit of service.

Fig. 10.2 provides a visualisation of the dynamics of a simulation conducted as follows. A field with pump rate 100, decay rate 0.1 and diffusion rate 0.0997 is started on the left of a 50×30 grid. When it is established, at time $t = 0$, a service with initial concentration $10,000$ is injected in one shot on the right, which ascends ($r\_asc = 1.0$) the field moving towards the target (the gradient factor is assumed to be 10 henceforth). The first column in Fig. 10.2 shows a snapshot of the system's spatial configuration at time $t = 10$. At the top, the field shape is visualised (darker colours represent higher field values), which has an approximate horizon of 30—note that the shape of the field does not change in different snapshots. At the bottom, the service is visualised: at time $t = 10$ it is a small cloud around coordinates $(35, 15)$, since it started diffusing around; at time $t = 30$ the effect of gradient ascent due to the `(ASCENT)` law becomes clearly visible, in that the cloud has not only expanded, but several service units have moved towards the target; at time $t = 60$ a significant portion of the service already reached the target, though several units are still diffused along the path towards the target; finally at time $t = 120$ the service has entirely relocated in the target proximity.

Part VI

Conclusion

11 Conclusion

This chapter summarises the work presented in the thesis, highlighting the corresponding contributions and shortcomings, and finally tracing a feasible path for future work.

11.1 Contributions

Developmental biology calls for modelling and simulation tools that can effectively and efficiently support the analysis of biochemical systems featuring multiply-nested, dynamic networks of compartments. The process of spatial organisation resulting from the morphogenesis process is demonstrated to be highly dependent by the interplay between the dynamics at different levels of the biological systems hierarchical organisation. In modelling and simulating the phenomena of morphogenesis it might be appropriate to reproduce such a hierarchy.

Accordingly, this thesis firstly proposes a new framework, MS-BioNET structured as follows: *(i)* a simple and coherent computational model to structure biochemical networks of compartments, *(ii)* a language to express articulated systems in a simple and flexible way, *(iii)* a corresponding simulation engine based on known optimisation techniques [GB00]. Although this framework is limited to the case of static networks, it makes it possible to experiment on rather large scenarios of embryo- and morphogenesis in a way which we believe is more expressive and flexible with respect to existing tools. The thesis secondly presents the application of ABM at the scenario, as an approach well known in literature for modelling complex systems and given its built-in ability of supporting multi-level dynamics. This thesis finally proposes the adoption of metaheuristic for parameter tuning, defining this problem as an optimisation problem over the parameter space.

In order to demonstrate the approaches and frameworks applicability, the phenomenon of pattern formation during *Drosophila* embryo development is studied, modelling the interactions between maternal factors and gap genes that originate the early regionalisation of the embryo. The possibility to model both the reactions taking place inside the cells that regulate the gene expressions, and the molecules diffusion that mediates the cell-to-cell communication, allows the reproduction of the interplay between these two levels so to verify its fundamental role in the spatial self-organisation characteristic of such a kind of phenomenon. The results presented show the formation of a precise spatial pattern which have been successfully compared with observations acquired from the real embryo gene expressions.

The thesis finally introduced a new trend of research which is in part freely inspired at the mechanisms of morphogenesis. It shows how the chemical metaphor – along with the semantic character of chemical laws and the possibility of modelling chemical diffusion – plays a crucial role to enable coordination models to tackle the requirements of engineering adaptive pervasive services. The preliminary ideas presented show that with very simple chemical reactions, it is possible to model reaction/diffusion behaviour that amounts to useful spatial patterns: indeed, they promote a view of Pervasive Systems as spatial computers, and of adaptive services in it as spatial structures that compete and diffuse in a context-dependent way.

11.2 Main Shortcomings

The extreme complexity of developmental biology scenarios often requires to study systems where the actors involved are in the worst case unknown, or if known it is actually not completely clear how they interact, or in the best case the gene regulatory network is widely known but the number of components involved is so huge that it is simply unfeasible to build a model where each reaction is explicitly modelled —the number of parameters would be so huge, and the biological knowledge on that so low, that it would be difficult to determine them automatically; moreover the computational cost for simulating such a big model can be unacceptable. Given that a model that directly implements the gene regulatory network of the organism, as the ones proposed in this thesis, could probably not be used. In these cases, certainly a huge work of abstraction has to be done. An abstract model which resembles in some way the behaviour of the biological system, and whose meaning is clear and useful to answer biological questions, has to be built. This is one of the biggest difficulties in Computational systems biology.

Another big problem when tackling such scenarios is the lack or incompleteness of biological data. The acquisition of data of gene expressions in multicellular organisms is a big challenge of experimental biology, and given that the elaboration to obtain proper data with spatial and temporal resolution is a very hard task. For this reason few organisms have actually been sufficiently investigated, such as *Drosophila* and *C. elegans* embryos which also are the more known one. For those organisms where big questions are still unsolved, few and incomplete data are available. This makes the process of model initialisation and validation quite hard.

11.3 Future Work

There are several works that would be compelling to pursue in future research activities. Mainly future work will be devoted to extend the exploration to other stages of embryogenesis, up to the long-term goal of simulating/predicting larger portions of embryogenesis. It means to introduce new phenomena on the side of both intracellular dynamics and cell-to-cell interacting capabilities. Gene regulatory network will be enlarged with other sets of genes which are downstream to gap genes such as the pair rule genes, *even-skipped* as first, whose expression gives rise at the characteristic segments of *Drosophila* embryo. Mechanisms regulating cell movements will then be added, cell adhesion and chemotaxis in particular, as soon as they are known to play a crucial role in cell sorting during morphogenesis.

On the side of MS-BioNET, moving towards larger time windows will require a dynamic topology of the cell network: the assumption made in this work that few and mostly non-relevant topological changes occur during the considered period – so that it is reasonable to assume a fixed topology – will in fact decay. Therefore future works will go in the direction of making it possible to simulate large scale dynamic network of cells, where they can move, divide and die. Unfortunately even if the language is ready for such an extension thanks to its flexibility [MV09b], implementing dynamic topologies is quite a complex task at the engine level, and will require an important effort. As the model will become more and more complex, it will then be necessary to consider other extensions of the basic Gillespie's SSA – such as tau-leaping [Gil01] – in order to maintain the good performance of the engine. Finally future works will be devoted to develop our tool up to a public release.

To conclude as the work presented in Chapter 10 is aimed at describing an executable model and spatial pattern emergence via simulation, it paves the way towards several research activities. Along the work presented in this thesis in particular, the main research direction concerns the identification of different chemical reactions that can be designed in order to generate different self-organised spatial patterns useful for capturing scenarios of Pervasive Systems.

Part VII

Bibliography

Bibliography

[AAS06] Rui Alves, Fernando Antunes, and Armindo Salvador. Tools for kinetic modeling of biochemical networks. *Nature Biotechnology*, 24(6):667–672, June 2006.

[Agh08] Gul Agha. Computing in pervasive cyberspace. *Commun. ACM*, 51(1):68–70, 2008.

[AJL$^+$02] Bruce Alberts, Alexander Johnson, Julian Lewis, Martin Raff, Keith Roberts, and Peter Walter. *Molecular Biology of the Cell*. Garland Science Textbooks. Garland Science, Fourth edition, June 2002.

[Ban08] J.R Banga. Optimization in computational systems biology. *BMC Bioinformatics*, 2(7), 2008.

[BB01] James M. Bower and Hamid Bolouri. *Computational Modeling of Genetic and Biochemical Networks (Computational Molecular Biology)*. The MIT Press, January 2001.

[BB06] Jacob Beal and Jonathan Bachrach. Infrastructure for engineered emergence on sensor/actuator networks. *IEEE Intelligent Systems*, 21(2):10–19, 2006.

[BBVT08] Jacob Beal, Jonathan Bachrach, Dan Vickery, and Mark Tobenkin. Fast self-healing gradients. In *SAC '08: Proceedings of the 2008 ACM symposium on Applied computing*, pages 1969–1975, New York, NY, USA, 2008. ACM.

[BCD$^+$06] Ozalp Babaoglu, Geoffrey Canright, Andreas Deutsch, Gianni A. Di Caro, Frederick Ducatelle, Luca M. Gambardella, Niloy Ganguly, Márk Jelasity, Roberto Montemanni, Alberto Montresor, and Tore Urnes. Design patterns from biology for distributed computing. *ACM Trans. Auton. Adapt. Syst.*, 1(1):26–66, 2006.

[BD06] Alistair P. Barros and Marlon Dumas. The rise of web service ecosystems. *IT Professional*, 8(5):31–37, 2006.

[Ber92] Alan A. Berryman. The origins and evolution of predator-prey theory. *Ecology*, 73(5):1530–1535, October 1992.

[BFDP10] Tommaso Biancalani, Duccio Fanelli, and Francesca Di Patti. Stochastic turing patterns in the brusselator model. *Phys Rev E Stat Nonlin Soft Matter Phys*, 81(4 Pt 2):046215, 2010.

[BKF$^+$09] Nicola Bonzanni, Elzbieta Krepska, K. Anton Feenstra, Wan Fokkink, Thilo Kielmann, Henri E. Bal, and Jaap Heringa. Executing multicellular differentiation: quantitative predictive modelling of *C. elegans* vulval development. *Bioinformatics*, 25(16):2049–2056, 2009.

[BMF06] Gregory Beurier, Fabien Michel, and Jacques Ferber. A morphogenesis model for multiagent embryogeny. In Luis Mateus Rocha, Larry S. Yaeger, Mark A. Bedau, Dario Floreano, Robert L. Goldstone, and Alessandro Vespignani, editors, *Artificial Life X: Proceedings of the Tenth International Conference on the Simulation and Synthesis of Living Systems*, pages 84–90. MIT Press, Cambridge, MA, 2006.

[BR03] Christian. Blum and Andrea Roli. Metaheuristics in combinatorial optimization: Overview and conceptual comparison. *ACM Computing Surveys*, 35(3):268–308, September 2003.

[CDG09] Federica Ciocchetta, Adam Duguid, and Maria Luisa Guerriero. A compartmental model of the cAMP/PKA/MAPK pathway in Bio-PEPA. *CoRR*, abs/0911.4984, 2009.

[CH09] Federica Ciocchetta and Jane Hillston. Bio-PEPA: A framework for the modelling and analysis of biological systems. *Theoretical Computer Science*, 410(33-34):3065–3084, August 2009.

[CHC$^+$05] Trevor M. Cickovski, Chengbang Huang, Rajiv Chaturvedi, Tilmann Glimm, H. George E. Hentschel, Mark S. Alber, James A. Glazier, Stuart A. Newman, and Jes?s A. Izaguirre. A framework for three-dimensional simulation of morphogenesis. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 2:273–288, 2005.

[Cle06] M. Clerc. *Particle Swarm Optimization*. ISTE, 2006.

[Cro] Michael Cross. `http://crossgroup.caltech.edu/Patterns/Demo4_5.html`. Last seen on February 2011.

[Cro08] Jon Crowcroft. Toward a network architecture that does everything. *Commun. ACM*, 51(1):74–77, 2008.

[CZNA07] Scott Christley, Xuelian Zhu, Stuart A. Newman, and Mark S. Alber. Multiscale agent-based simulation for chondrogenic pattern formation in vitro. *Cybernetic and Systems*, 38:707–727, September 2007.

[Dav05] Jamie A. Davies. *Mechanisms of Morphogenesis*. Academic Press, First edition, October 2005.

[DFF$^+$07] Vincent Danos, Jérôme Feret, Walter Fontana, Russell Harmer, and Jean Krivine. Rule-based modelling of cellular signalling. In *CONCUR*, pages 17–41, 2007.

[DJ02] Hidde De Jong. Modeling and Simulation of Genetic Regulatory Systems: A Literature Review. *Journal of Computational Biology*, 9(1):67–103, 2002.

[DL04] Vincent Danos and Cosimo Laneve. Formal molecular biology. *Theoretical Computer Science*, 325(1):69–110, 2004.

[DOR05] Enrico Denti, Andrea Omicini, and Alessandro Ricci. Multi-paradigm Java-Prolog integration in tuProlog. *Science of Computer Programming*, 57(2):217–250, August 2005.

[DPRS08] Lorenzo Dematté, Corrado Priami, Alessandro Romanel, and Orkun Soyer. Evolving blenx programs to simulate the evolution of biological networks. *Theoretical Computer Science*, 408(1):83–96, 2008.

[EBLB10] Manolya Eyiyurekli, Linge Bai, Peter I. Lelkes, and David E. Breen. Chemotaxis-based sorting of self-organizing heterotypic agents. In *SAC '10: Proceedings of the 2010 ACM Symposium on Applied Computing*, pages 1315–1322, New York, NY, USA, 2010. ACM.

[FH07] Jasmin Fisher and Thomas A. Henzinger. Executable cell biology. *Nature biotechnology*, 25(11):1239–1249, November 2007.

[FN74] Richard J. Field and Richard M. Noyes. Oscillations in chemical systems. v. quantitative explanation of band migration in the belousov-zhabotinskii reaction. *Journal of the American Chemical Society*, 96(7):2001–2006, 1974.

[GB00] Michael A. Gibson and Jehoshua Bruck. Efficient exact stochastic simulation of chemical systems with many species and many channels. *The Journal of Physical Chemistry A*, 104(9):1876–1889, March 2000.

[GF92] John Guslander and Richard J. Field. Spatial structures in an oregonator model with diffusion in two dimensions. *The Journal of Physical Chemistry*, 96(25):10575–10580, 1992.

[Gil77] Daniel T. Gillespie. Exact stochastic simulation of coupled chemical reactions. *The Journal of Physical Chemistry*, 81(25):2340–2361, 1977.

[Gil01] Daniel T. Gillespie. Approximate accelerated stochastic simulation of chemically reacting systems. *Journal of Chemical Physics*, 115(4):1716–1733, 2001.

[Gil06] Scott F. Gilbert. *Developmental Biology, Eighth Edition*. Sinauer Associates Inc., Eighth edition, March 2006.

[GJK$^+$04] Vitaly V. Gursky, Johannes Jaeger, Konstantin N. Kozlov, John Reinitz, and Alexander M. Samsonov. Pattern formation and nuclear divisions are uncoupled in drosophila segmentation: comparison of spatially discrete and continuous models. *Physica D: Nonlinear Phenomena*, 197(3-4):286–302, October 2004.

[GR92] Iva Greenwald and Gerald M. Rubin. Making a difference: the role of cell-cell interactions in establishing separate identities for equivalent cells. *Cell*, 68:271–281, 1992.

[GWM$^+$07] Thomas Gregor, Eric F. Wieschaus, Alistair P. McGregor, William Bialek, and David W. Tank. Stability and nuclear dynamics of the bicoid morphogen gradient. *Cell*, 130(1):141–152, July 2007.

[HGD08] Monika Heiner, David Gilbert, and Robin Donaldson. Petri nets for systems and synthetic biology. In Marco Bernardo, Pierpaolo Degano, and Gianluigi Zavattaro, editors, *Formal Methods for Computational Systems Biology*, volume 5016 of *Lecture Notes in Computer Science*, pages 215–264. Springer Berlin / Heidelberg, 2008.

[HHS07] F. Hutter, H.H. Hoos, and T. Stützle. Automatic algorithm configuration based on local search. In *Proceedings of the 22nd Conference on Artificial Intelligence (AAAI-07)*, pages 1152–1157, 2007.

[HO01] Nikolaus Hansen and Andreas Ostermeier. Completely Derandomized Self-Adaptation in Evolution Strategies. *Evolutionary Computation*, 9(2):159–195, June 2001.

[HS05] H.H. Hoos and T. Stützle. *Stochastic Local Search Foundations and Applications*. Morgan Kaufmann Publishers, San Francisco, CA (USA), 2005.

[Kau93] Stuart A. Kauffman. *The Origins of Order: Self-Organization and Selection in Evolution*. Oxford University Press, USA, First edition, June 1993.

[KEBC05] Mads Kaern, Timothy C. Elston, William J. Blake, and James J. Collins. Stochasticity in gene expression: from theories to phenotypes. *Nature reviews. Genetics*, 6(6):451–464, June 2005.

[Kit02a] Hiroaki Kitano. Computational systems biology. *Nature*, 420(6912):206–210, November 2002.

[Kit02b] Hiroaki Kitano. Systems Biology: A Brief Overview. *Science*, 295(5560):1662–1664, March 2002.

[LCRP$^+$05] Sean Luke, Claudio Cioffi-Revilla, Liviu Panait, Keith Sullivan, and Gabriel Catalin Balan. Mason: A multiagent simulation environment. *Simulation*, 81(7):517–527, 2005.

[LDB09] William J.R. Longabaugh, Eric H. Davidson, and Hamid Bolouri. Visualization, documentation, analysis, and communication of large-scale gene regulatory networks. *Biochimica et Biophysica Acta (BBA) - Gene Regulatory Mechanisms*, 1789(4):363–374, 2009.

[Lep99] Maria Leptin. Gastrulation in drosophila: the logic and the cellular mechanisms. *The EMBO Journal*, 18(12):3187–3192, April 1999.

[MAC$^+$07] Emanuela Merelli, Giuliano Armano, Nicola Cannata, Flavio Corradini, Mark d'Inverno, Andreas Doms, Philip W. Lord, Andrew Martin, Luciano Milanesi, Steffen Möller, Michael Schroeder, and Michael Luck. Agents in bioinformatics, computational and systems biology. *Briefings in Bioinformatics*, 8(1):45–59, 2007.

[MFD09] Fabien Michel, Jacques Ferber, and Alexis Drogoul. Multi-Agent Systems and Simulation: a Survey From the Agents Community's Perspective. In Adelinde Uhrmacher Danny Weyns, editor, *Multi-Agent Systems: Simulation and Applications*, Computational Analysis, Synthesis, and Design of Dynamic Systems, pages 3–51. CRC Press - Taylor & Francis, May 2009.

[MG05] Roeland M.H. Merks and James A. Glazier. A cell-centered approach to developmental biology. *Physica A: Statistical Mechanics and its Applications*, 352(1):113 – 130, July 2005.

[MN10] Charles M Macal and Michael J. North. Tutorial on agent-based modelling and simulation. *Journal of Simulation*, 4:151–162, 2010.

[MR09] Sara Montagna and Andrea Roli. Parameter tuning of a stochastic biological simulator by metaheuristics. In *AI*IA*, volume 5883 of *Lecture Notes in Computer Science*, pages 466–475. Springer, 2009.

[MV09a] Sara Montagna and Mirko Viroli. A computational framework for modelling multicellular biochemistry. In *2009 IEEE Congress on Evolutionary Computation (CEC 2009)*, pages 2233–2240, Trondheim, Norway, 18-21 May 2009. IEEE Computer Society Press.

[MV09b] Sara Montagna and Mirko Viroli. A computational framework for modelling multicellular biochemistry. In *IEEE CEC09 Preceedings*, Trondheim, Norway, May 2009.

[MV10] Sara Montagna and Mirko Viroli. A framework for modelling and simulating networks of cells. *Electronic Notes in Theoretical Computer Science*, 268:115–129, December 2010. Proceedings of the 1st International Workshop on Interactions between Computer Science and Biology (CS2Bio'10).

[MZ06] Marco Mamei and Franco Zambonelli. *Field-based Coordination for Pervasive Multiagent Systems*. Springer Verlag, 2006.

[MZ09] Marco Mamei and Franco Zambonelli. Programming pervasive and mobile computing applications: The tota approach. *ACM Trans. Softw. Eng. Methodol.*, 18(4):1–56, 2009.

[NHCV07] Michael J. North, Tom R. Howe, Nicholson T. Collier, and Jerry R. Vos. A declarative model assembly infrastructure for verification and validation. In Shingo Takahashi, David Sallach, and Juliette Rouchier, editors, *Advancing Social Simulation: The First World Congress*, pages 129–140. Springer Japan, 2007.

[Pau02] Gheorghe Paun. *Membrane Computing: An Introduction*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2002.

[PG71] Ilya Prigogine and Paul Glansdorff. *Thermodynamic Theory of Structure, Stability and Fluctuations*. John Wiley & Sons Ltd, New York, 1971.

[Phi06] Andrew Phillips. The Stochastic Pi Machine (SPiM), 2006. Version 0.042 available online at http://www.doc.ic.ac.uk/~anp/spim/.

[PJRG06] Theodore J Perkins, Johannes Jaeger, John Reinitz, and Leon Glass. Reverse engineering the gap gene network of *Drosophila Melanogaster*. *PLoS Computational Biology*, 2(5):e51, 05 2006.

[PPB$^+$04] Ekaterina Poustelnikova, Andrei Pisarev, Maxim Blagov, Maria Samsonova, and John Reinitz. A database for management of gene expression data in situ. *Bioinformatics*, 20(14):2212–2221, 2004.

[PPSR09] Andrei Pisarev, Ekaterina Poustelnikova, Maria Samsonova, and John Reinitz. Flyex, the quantitative atlas on segmentation gene expression at cellular resolution. *Nucleic Acids Research*, 37(Database-Issue):560–566, 2009.

[PQ04] Corrado Priami and Paola Quaglia. Beta binders for biological interactions. In *CMSB*, volume 3082 of *Lecture Notes in Computer Science*, pages 20–33. Springer, 2004.

[Pri95] Corrado Priami. Stochastic pi-calculus. *The Computer Journal*, 38(7):578–589, 1995.

[PRSS01] Corrado Priami, Aviv Regev, Ehud Shapiro, and William Silverman. Application of a stochastic name-passing calculus to representation and simulation of molecular processes. *Information Processing Letters*, 80:25–31, October 2001.

[RFEB06] M. Rodriguez-Fernandez, J.A. Egea, and J.R Banga. Novel metaheuristic for parameter estimation in nonlinear dynamic biological systems. *BMC Bioinformatics*, 7(483), 2006.

[RLJ06] Steven F. Railsback, Steven L. Lytinen, and Stephen K. Jackson. Agent-based simulation platforms: Review and development recommendations. *Simulation*, 82(9):609–623, 2006.

[RPJ96] Rolando Rivera-Pomar and Herbert Jackle. From gradients to stripes in drosophila embryogenesis: filling in the gaps. *Trends in Genetics*, 12(11):478 – 483, 1996.

[RPS$^+$04] Aviv Regev, Ekaterina M. Panina, William Silverman, Luca Cardelli, and Ehud Y. Shapiro. Bioambients: an abstraction for biological compartments. *Theoretical Computer Science*, 325(1):141–167, 2004.

[RSS01] Aviv Regev, William Silverman, and Ehud Shapiro. Representation and simulation of biochemical processes using the pi-calculus process algebra. *Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing*, pages 459–470, 2001.

[SDKZ02] Ilya Shmulevich, Edward R. Dougherty, Seungchan Kim, and Wei Zhang. Probabilistic Boolean Networks: a rule-based uncertainty model for gene regulatory networks. *Bioinformatics (Oxford, England)*, 18(2):261–274, February 2002.

[SH05] William Smith and Javad Hashemi. *Foundations of Materials Science and Engineering*. McGraw-Hill, 4th edition, 2005.

[SKK$^+$08] Svetlana Surkova, David Kosman, Konstantin Kozlov, Manu, Ekaterina Myasnikova, Anastasia A. Samsonova, Alexander Spirov, Carlos E. Vanario-Alonso, Maria Samsonova, and John Reinitz. Characterization of the Drosophila segment determination morphome. *Developmental Biology*, 313(2):844 – 862, 2008.

[SMC$^+$08] Antoine Spicher, Olivier Michel, Mikolaj Cieslak, Jean-Louis Giavitto, and Przemyslaw Prusinkiewicz. Stochastic P systems and the simulation of biochemical processes with dynamic compartments. *Biosystems*, 91(3):458 – 472, 2008.

[SMG04] Antoine Spicher, Olivier Michel, and Jean-Louis Giavitto. A topological framework for the specification and the simulation of discrete dynamical systems. In Peter M. A. Sloot, Bastien Chopard, and Alfons G. Hoekstra, editors, *Cellular Automata*, volume 3305 of *Lecture Notes in Computer Science*, pages 238–247. Springer Berlin / Heidelberg, 2004.

[SSP06] Zoltan Szallasi, Jörg Stelling, and Vipul Periwal, editors. *System Modeling in Cellular Biology: From Concepts to Nuts and Bolts*. The MIT Press, 1 edition, April 2006.

[Ste07] Malcolm S Steinberg. Differential adhesion in morphogenesis: a modern view. *Current Opinion in Genetics & Development*, 17(4):281 – 286, July 2007.

[TBDP08] Bryan C. Thorne, Alexander M. Bailey, Douglas W. Desimone, and Shayn M. Peirce. Agent-based modeling of multicell morphogenic processes during development. *Birth Defects Research (Part C): Embryo Today*, 81(4):344–353, 2008.

[Teaa] Repast Development Team. `http://repast.sourceforge.net/`. Repast home page.

[Teab] Swarm Development Team. `http://www.swarm.org/index.php/Main_Page`. Swarm home page.

[Tur52] Alan Turing. The chemical basis of morphogenesis. *Philosophical Transactions of the Royal Society of London. Series B, Biological Sciences*, 237(641):37–72, August 1952.

[UDZ05] Adelinde M. Uhrmacher, Daniela Degenring, and Bernard Zeigler. Discrete event multilevel models for systems biology. In Corrado Priami, editor, *Transactions on Computational Systems Biology I*, volume 3380 of *Lecture Notes in Computer Science*, pages 66–89. Springer, 2005.

[Uni] George Mason University. `http://www.cs.gmu.edu/~eclab/projects/mason/`. MASON home page.

[VB08] Cristian Versari and Nadia Busi. Efficient stochastic simulation of biological systems with multiple variable volumes. *Electronic Notes Theoretical Computer Science*, 194(3):165–180, 2008.

[VCMZ11] Mirko Viroli, Matteo Casadei, Sara Montagna, and Franco Zambonelli. Spatial coordination of pervasive services through chemical-inspired tuple spaces. *ACM Transactions on Autonomous and Adaptive Systems*, 2011. To Appear.

[VCO09] Mirko Viroli, Matteo Casadei, and Andrea Omicini. A framework for modelling and implementing self-organising coordination. In Sung Y. Shin, Sascha Ossowski, Ronaldo Menezes, and Mirko Viroli, editors, *24th Annual ACM Symposium on Applied Computing (SAC 2009)*, volume III, pages 1353–1360, Honolulu, Hawai'i, USA, 8–12 March 2009. ACM.

[VRVZ08] Cynthia Villalba, Alberto Rosi, Mirko Viroli, and Franco Zambonelli. Nature-inspired spatial metaphors for pervasive service ecosystems. In *Workshop on Spatial Computing*, Venice Italy, October 2008. Informal Proceedings.

[WBB$^+$98] Lewis Wolpert, Rose Beddington, J. Brockes, Thomas Jessell, P. Lawrence, and E. Meyerowitz. *Principles of Development*. Oxford University Press, 3 edition, 1998.

[WC] Uri Wilensky and CCL. `http://ccl.northwestern.edu/netlogo/index.shtml`. NetLogo home page.

[WOO07] Danny Weyns, Andrea Omicini, and James Odell. Environment as a first-class abstraction in multi-agent systems. *Autonomous Agents and Multi-Agent Systems*, 14(1):5–30, February 2007. Special Issue on Environments for Multi-agent Systems.

[YDZE02] Lingfa Yang, Milos Dolnik, Anatol M. Zhabotinsky, and Irving R. Epstein. Spatial resonances and superposition patterns in a reaction-diffusion model with interacting turing modes. *Phys. Rev. Lett.*, 88(20):208303, May 2002.

[YN08] Daniel Yamins and Radhika Nagpal. Automated global-to-local programming in 1-d spatial multi-agent systems. In *7th International Joint Conference on Agents and Multi-Agent Systems (AAMAS-08)*, pages 615–622, Estoril, Portugal, 12–16May 2008. IFAAMAS.

[ZV08] Franco Zambonelli and Mirko Viroli. Architecture and metaphors for eternally adaptive service ecosystems. In *IDC'08*, volume 162/2008 of *Studies in Computational Intelligence*, pages 23–32. Springer Berlin / Heidelberg, September 2008.