Computational modelling of mycobacterium infection and innate immune response in zebrafish

Viana de Carvalho, R.

1. Introduction

Based on:

Carvalho, Rafael V., Davids, Willem, Meijer, Annemarie H., Verbeek, Fons J.: Spatio-temporal Modelling and Simulation of Mycobacterium Pathogenesis Using Petri nets. In: BIONETICS2011. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, vol. 103, pp 236-241, Springer (2012).

And

Carvalho, Rafael V., Kleijn, Jetty, Meijer, Annemarie H., Verbeek, Fons J.: Modeling Innate Immune Response to Early Mycobacterium Infection. In: Computational and Mathematical Methods in Medicine, vol. 2012, 12 pages (2012).

Papel	Paper
E tudo que eu pensei e tudo que eu falei e tudo que me contaram era papel.	And everything I've though and everything I've said and everything they've told me was paper.
E tudo que descobri amei detestei: papel.	And everything I've discovered loved hated: paper.
Papel quanto havia em mim e nos outros, papel de jornal de parede de embrulho papel de papel papelão	Paper everything that was in me and in the others, paper news-paper wallpaper wrapping paper paper-paper papier-mâché
Carlos Drummond de Andrade, Brasil (1902 – 1987)	Carlos Drummond de Andrade, Brazil (1902 – 1987)

Abstract

Modeling of biological systems is evolving into the description, simulation and analysis of the behavior and interdependent relationships of biological phenomena. The process of formulating a model relies on the synergistic combination of experimentation, theory and formalization of the processes that happen in such systems. In this chapter we describe the characteristics of the modeling process and a broad view of the formal models used to model biological systems. We choose the Petri net formalism as modeling technique, describing its structure and characteristics applied in the models described in the following chapters. This thesis is about an application of Petri nets in biology; the examples will directly refer to or rely on biological context.

1.1 Modeling systems biology

Modeling is the art of representing, manipulating and communicating an object, process or concept. In the dictionary[1], the verb to 'model' is defined as "to plan or form after a pattern" or "to produce a representation or simulation of". As one can easily realize, for the modeling is necessary to observe, to gather information and generate hypotheses that are consistent with the data collected. Systems biology is an empirical science. It represents facts of life with a complex structure. Modeling biological systems is an endeavor to better understand the inner workings and emergent properties of such system.

The core of systems biology relies on the construction of models describing biological behavior. The aim is to study biological components, i.e. molecules, cells, organs or entire species, and their complex interactions, in order to understand the processes and emergent properties that happen within such systems. There is ample scientific use for models that represent biological behavior: education, using the biological information to describe system characteristics; analysis and prediction, testing which assumptions about the system best fit the data and eventually, to predict response to different stimuli; instrumentation or device, designing an "alternative system" to circumvent experiments that are too costly, time consuming or ethically undesirable.

In order to build a model that will represent biological behavior, it is necessary to collect the information, select the most relevant components and construct first a simplified version of the entity we want to model. This process is referred to as the modeling decisions; and basically it takes into account the properties of the information, the purpose of the model and the aspects of the structure and the process on which the model focuses. Therefore, model formalism is required to define, implement and simulate the model and it is part of the modeling decision process to choose the formalism that will be used to describe a biological problem. In this chapter, we focus on the modeling decisions and the formalisms that are commonly used in modeling biological behavior. First, we discuss about the model properties that are considered to make the modeling decision. Subsequently, we describe the formal methods used to model biological behavior. We emphasize the formalism chosen to model the case study presented in the following chapters. In this chapter we also briefly introduce how the formalism that we will use in this thesis, the Petri net, is successfully used in the domain of biology. We will explain that the current status of the PN formalism allows us to gradually build a model of increasing complexity. Modern biology research is capable of producing lots of data that together represent a biological system; the systems biology approach is to integrate all these different components and data in order to understand the system and its interacting components. We think that the concurrent aspects and different layers can be well

[1] Merriam-Webster online dictionary: www.merrian-webster.com/dictionary, access on March 2015

represented in a Petri net and therefore in this thesis, using a case study, we further investigate mapping of complex systems studied in the context of systems biology. The different modeling aspects are explored and probed with the different Petri net methods. A problem typical to systems biology is one in which multiple organisms are involved and interact on different levels – and in order to integrate these levels procedures need to be developed. Therefore, a system with an invader-host interaction common to disease models was chosen to further develop these procedures. More specifically, the invader host system of tuberculosis (*Mycobacterium*) is studied using the zebrafish model. It addresses infection and response mechanisms on various levels of detail. The *Mycobacterium* infection and the response is a very interesting system to study we aim to also use it to push the limits of modeling forward.

1.2 Modeling process and properties

Every modeling process starts by an analysis of the problem; this is essential for a clear definition of the questions one asks the model. What do we want to model? What is the aim? What should the model achieve? This step includes the definition of the **model objective** and it implies a certain level of understanding of the problem. In the attempt to understand the problem, it is necessary to state what the model will represent, and how to analyze its output. Once the objective of the modeling is defined, the next step is to **collect the information and data** about the problem consulting scientific literature and/or empirical data together with discussions with field experts. After stating the objective and gathering the data, the next step in the modeling process is to **define a hypothesis**. The hypothesis is the basis for the model and the results will depend on it. On basis of the hypothesis that we can formulate the **model structure**, correlating the acquired information and the model objective, considering only the data that are relevant for the specific purpose of the model. Therefore, in this step we define the level of detail that will be modeled and based on the model properties start to construct the model.

It is crucial to analyze the model properties to define the model structure. Therefore, interpretation of the requirements of the model and the aspects of the information/data are acquired to define the properties of the structure and process on which the model focuses.

A model is considered **descriptive** when it represents the observed data without considering directly the processes that produce these data. An example is a graphical representation of the steps of the cell cycle: G2 phase (cells with duplicated chromosome); Mitosis (chromosome separation and cell division); G1 phase (cells with one chromosome); Synthesis (chromosome duplication). It gives the information about the cell process but there is no data about the variables that trigger the process (molecules interaction, rates, time).

A **mechanistic model** assumes that a complex system can be understood by examining the working of its individual parts and the manner in which they are coupled. For instance, for modeling the genetic pathways underlying the cell division, one should consider the data about molecular interactions, rates and time consuming.

Other model properties are related to the dynamics of the process, which is studied. A model can be static or dynamic**.** A **static model** represents structures that do not provide time dependency. The model refers more to the structure than to the behavior of the system. These models are considered descriptive and not mechanistic.

A **dynamic model** describes relevant information that can be put in a chronological, time dependent order. It models systems that represent a process flow, what an object does essentially with many possibilities that may arise. Such models can be both descriptive and mechanistic. The models presented in Chapter 2 and 3 are examples of a dynamic descriptive model and a dynamic mechanistic model respectively.

Another two important model properties to be considered are directly related to the information and data acquired. A **qualitative model** describes processes without using exact experimental data. The model relies on the examination, analysis and interpretation of observations. The purpose is to describe underlying meanings and patterns of relationships, including classifications of types of phenomena and entities in the biological system. A qualitative model helps to obtain an in-depth understanding of biological behavior and the reasons that govern such behavior (why and how, not just what, where, when).

The **quantitative model** is based on systematic empirical investigation of biological phenomena from experimental data that are available. The model allows via statistical, mathematical and/or computational analysis, to make predictions about future states in its behavior. A quantitative model can contribute, for example, in the prediction of results from experiments and generation of further hypothesis about the biological behavior.

The construction of a model to describe a biological system does not always require exact quantitative data. A qualitative model is a starting point for modeling processes for which experimental data are still incomplete. When experimental data are available, a migration to a quantitative model can be realized by including variable and extending the model complexity. This thesis shows this process in its Chapters 2 to 5 where we start to model qualitatively and then migrate to a quantitative model. Each model represents an extension of the previous model adding a layer of complexity and more data information. The convenience in the process of transformation from qualitative to quantitative model is directly related to the formal method used to implement the model and will be discussed in the next sections.

1.3 Formal models

Further to the definition given in section 1.1, a model can be seen as description of a system designed to help an observer to understand how the system works and to be able to predict on the behavior of such system. Models are typically conceptual, existing as an idea, a computer program or a set of mathematical conceptualizations. Next, formal models are methods to represent the knowledge of a system using computational and/or mathematical structures. For systems biology, the theory of concurrency is at the basis of most approaches that have been applied. Many different formal methods, languages and modeling paradigms exist, that depend on the information, data and the model properties that guide the modeling process.

In systems biology, **Mathematical models** are, most often, equation based models that express mathematical relationships between quantities and how they change over time. The language in which they are specified is denoted by a set of equations that describes different parts, components or processes of a system and their behavior.

Next, **Computational models** are based on algorithmic process models that are executable and progress from state to state, not necessary time dependent. They use an algorithm/programing language to specify an abstract execution engine to mimic a design or natural phenomenon.

In an extended review, Fisher and Henzinger [35] distinguished between computational and mathematical models and how they are used to model biological phenomena. They survey the applicability and benefits of both approaches and how they can be interconnected. Essentially the distinctions are related to important information that drives the modeling decision, such as:

I. **Discrete or continuous:**

   A discrete model represents the behavior of the system in distinct spatial and/or temporal steps. The state variables change only at a countable number of points in time. In the time points an event or a change in state occurs. In a continuous model the behavior of the system is represented continuously. The state variables change in a continuous way, and not abruptly from one state to another. In fact, the number of states is infinite.

II. **Deterministic or nondeterministic:**

In deterministic models, series of events are determined at the onset and are precise. It is possible to predict what will happen in the system. In nondeterministic (stochastic) models, multiple outcomes of the processes can occur and the behavior of the system cannot be entirely predicted. In stochastic models, the rates and/or probabilities of events that can occur are included.

III. **Sequential or concurrent:**

As the name suggest, in a sequential model the events occur sequentially and a process cannot start before its predecessor has ended. It will always depend on information of the pre-process. In a concurrent model, different events can occur at the same time. Most process in biological system can be characterized by concurrency, and therefore should be modeled in such a manner

Although the description of mathematical models is based on equations, in systems biology computer power is used to describe and analyze such models. Therefore, algorithms and programs are used to implement such models. In the research presented in this thesis, we designate mathematical models as those models that are using process algebras, term rewriting systems or different mathematical structures; i.e. Stochastic Differential Equations, Ordinary Differential Equations, Partial Differential Equations and Delay Differential Equations. These mathematical models also include methods that are inspired by biological phenomena such as Brane Calculi [15], P-Systems [96], Biocham [14] and Calculus of Looping Sequence (CLS) [6]. Next, there are computational methods based on concurrent systems; i.e. Calculus of Communicating Systems (CCS) [27], π-Calculus[99], Bio-PEPA [2, 23], agent-based model (ABM) [46, 108], Petri nets [86, 98] as well as variants of this method [49, 61]. The modeling technique of Petri nets (PNs) is widely used to model biological behavior due its flexibility and strong emphasis on concurrency and local dependency. It comprises an abstract model of information flow, providing a graphical representation and a formal mathematical definition. The Petri Net model is the formalism that is used to model the biological behavior presented in this thesis. In the following section, we will provide an overview about Petri nets, its classes and definitions.

1.4 Petri nets

Based on a graphical representation and an underlying mathematical structure, the Petri net formalism is applicable to model the behavior of a concurrent distributed system that can be described in terms of system states and changes in these system states. Among the advantages to use Petri nets to model biological systems we can state that the model

- Has an intuitive graphical representation and a directly executable model

- Has a strong mathematical foundation providing a variety of analysis techniques
- Addresses structural and behavioral properties and also their relationship
- Integrates qualitative and quantitative methods as well as analysis techniques and simulation/animation
- Covers discrete and continuous, deterministic and stochastic, sequential and concurrent methods including hybrid techniques for quantitative and qualitative analysis
- Has a range of tools to support the implementation, simulation and analysis of the models

In the formal sense, a Petri net is a directed, finite, bipartite graph. Typically without isolated nodes, a PN is basically composed of three main elements [33]:

Places: are passive nodes that refer to conditions or local states of a system; they can be used, e.g. to represent resources; *Transitions*: are active nodes that describe local state changes in the system;

Directed arcs: specify relationships between local states and local actions by depicting the relations between places and transitions.

Tokens: are elements used to represent information in local states;

Places, transitions and arcs describe the static structure of the Petri net and how they are connected. A transition has *input places* to which it is connected by a number of places with a direct arc leading to the transition. Next, it has *output places* connected by a direct arc from the transition. Direct connections between two places or two transitions are not allowed. The tokens can be distributed in the places in order to define a state of the Petri net, referred to as a *marking*. The *state space* of a Petri net is the set of all possible markings.

The dynamic properties of the system are governed by a firing rule. It relates the transitions that can occur when enabled and then move tokens around the places in a Petri net. A transition can fire (occur) depending on the presence of tokens in its input places. The transitions fire by consuming the tokens from each of its input places and then producing (deposing) tokens on each of its output places. One of the elegant elements of the Petri Net formalism is the graphical notation; the notation helps to understand the flow of information through the net. A place is represented by a circle and a transition by a rectangle. Arcs connect a place to a transition. Tokens are represented by dots that pass through the net (cf. Figure 1.1).

The basic standard class of the formalism consists of place/transition Petri nets (PTNs). These nets are discrete and have no association with time or probability. Possible behaviors of the system are analyzed in terms of causalities and dependencies, without any quantification. A Petri net model can be enhanced with special read and inhibitor arcs as a means of modeling activation or inhibition of activities respectively. In addition, features can be added allow one to connect sub-models in a

hierarchical structure. In Chapter 3 we present a model that represents the bacterial-macrophage interaction in such a multi-scale structure.

There are several ways to add time to a net for quantitative modeling. In Continuous Petri nets (CPN) the discrete values of the net are replaced by continuous (real) values to represent concentrations over time [48]. In Stochastic Petri nets (SPN) an exponentially distributed firing rate (waiting time) - typically state dependent and specified by a rate function – is associated with each transition [85]. Hybrid Petri nets (HPN) [28] allow one to combine continuous and stochastic features of the process to be modeled.

Initially studied in [83, 88, 115, 116] and summarized by Marsan [78], Stochastic Petri nets can be considered as a timed Petri net in which the timings have stochastic values, where a firing delay (rate functions) is associated with each transition. It specifies the amount of time that must elapse before the transition can fire. This firing delay is a random variable following an exponential probability distribution. The semantics of a SPN with exponentially distributed firing delays for all transitions are described by a continuous time Markov chain (CTMC). Their firing transition follows the standard firings rule of QPNs, and do not consume time. The Stochastic Petri nets can also be enhanced with modifier arcs, which allow pre-places to modify the firing rate of a transition without influence on its enable state. There are also the special transitions: 1) deterministic which its firing delay is specified by an integer constant; 2) immediate which have zero delay and always high priority; and 3) scheduled which is a special case of deterministic transition which is specified at an absolute point in the simulation time at which it might occur (it will always depend if it is enable).

The Colored Petri nets, proposed by Jensen [60, 62], is an extension to the PN formalism in which information is added in the form of ‘colors’ (data types) assigned to tokens, allowing further operation and structure abstraction. The functional programming language Standard ML is used to manipulate and test data, providing a flexible way to create compact and parameterizable models. In a Colored Petri nets, to regulate the occurrence of transitions there are arc expressions that specify which tokens can flow over the arcs; and guards that are in fact Boolean expressions used to decide which transitions instances exist.

In the Colored Petri net, a transition is enabled (allowed to fire) if it has no input place, or if each of its input places is sufficiently marked by tokens: i.e. the arc expressions evaluate to a multi-set of token colors that should be available in the corresponding preceding place. In addition, the guard of the transition – if present – should evaluate to true for the given binding. When a transition fires a multi-set of colored tokens are consumed (taken) from each of the preceding places, according to the evaluation of the expression on the arc. A multi-set of colored tokens is produced (added), in correspondence with the arc expression, to each successor place. The overall state space of the Petri net is determined by the firing sequences consisting of iterated occurrences of transitions [33].

A typical example of a Colored Petri net and its components is depicted in **Figure 1.1**. Here both the graphical representation and the values that make of the QPN^C are given.. **Figure 1.1 (a)** shows the declaration of the data values assigned to the net defining two color-sets: *count* with integer values, and *individual* with a string *mm*; a constant *Max* with an integer value; the two variables: *k* and *x* used in arc expressions and the guard of the transition. The color-sets are assigned to places and the tokens on each place will have a color from a color-set assigned to the place. In **Figure 1.1 (b)** the net is defined with place *P1* (with color-set *individual*) containing 1 token with the color *mm*, place *P2* (with color-set *count*) containing one token with the integer value "1", and place *P3* (with color-set *individual*) without token. The transition *T1* is connected with the place *P1* by a read-arc that works as a test arc: if there is a token in *P1*, as described by the binding of *x* and that satisfies the firing rule, then the transition can occur, producing a token on the output places, but not consuming the token read in *P1*. Transition *T1* is connected to place *P2* by two arcs indicating that it will consume and produce tokens according to the arc expressions. **Figure 1.1 (c)** shows the new marking obtained when *T1* has read *P1* and consumed the token in *P2*, adding a new token in *P2* and P3 according to the firing rule. Repeating this process, the firing sequence will stop in the end state as shown in **Figure 1.1 (d)**. In this case, the guard condition turns false since the value of the token in *P2* does not fulfill the condition.

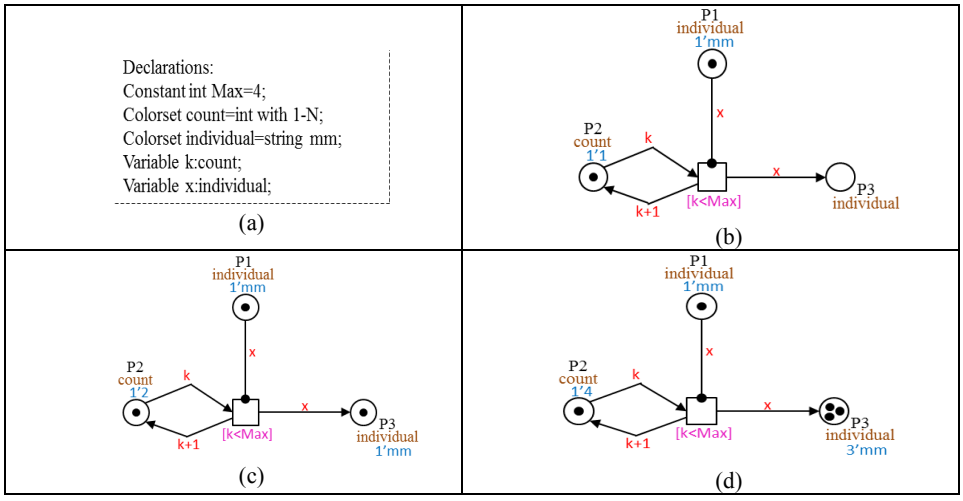

Figure 1.1. Colored Petri net example. **(a)** Declarations of the data types and variables. **(b)** Colored Petri net components. **(c)** State of the net after firing the transition. **(d)** End state of the net; the transition is not able to fire since the condition of the guard is not satisfied.

The Petri net and its colored extension permit to organize the formalism in a set of modules as a family of related Petri net classes, sharing structure, but being specialized by their kinetic information, dividing it in colored and uncolored: QPN – QPN^C (colored), SPN – SPN^C (colored),

CPN – CPN^C (colored), HPN – HPN^C (colored). The conversion of these classes can be realized by a folding process. This process groups similar model components in one colored model by defining color-set and the set of arc expression. The unfolding process dismembers a colored Petri net in one or more similar nets without colors. Moving between the colored and uncolored level changes the style of representation, but not necessary the net structure, though there may be loss of information in some direction. Heiner et al. [50] produced a classification of the different nets and **Figure 1.2** depicts their structure paradigm of the Petri nets formalism. . For practical reasons we adapt this classification as, in our case, this closely aligns with the tools that we use to compose the net. In Chapter 2 we present a qualitative colored Petri nets representing the dynamics of the infection process. In Chapter 4 and 5 we demonstrate the portability of the Petri nets formalism throughout its classes by presenting a QPN^C and a SPN^C respectively. A formal definition of the underlying Petri net classes used in this thesis can be found in **Appendix A, Appendix B and Appendix C.**

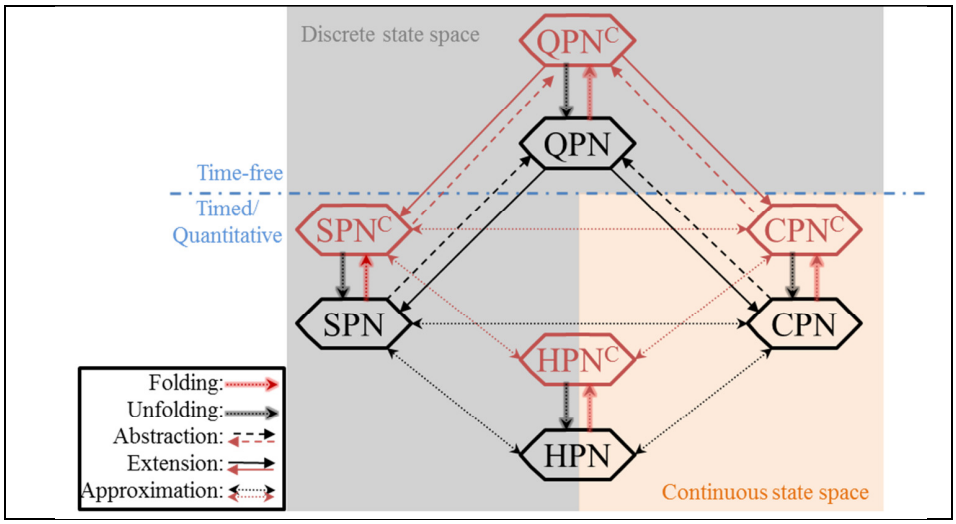

Figure 1.2. Paradigm structure of the Petri nets formalism. According to [50].

1.5 Model analysis

We have seen that for a model to represent the behavior of a biological system, it has to concisely summarize and replicate the information about the biological process as present in the system. Once a model has been constructed, it should serve as a clear description of the system, and can be unequivocally communicated. Therefore, an analysis of the model is required in order to verify and validate its structure and behavior. A reliable model is responsible to reproduce a biological process bringing evidences in reproduce a specific behavior and confidence on its predictive capability.

For the analysis of a model it is necessary to refer to the model objective and the information and data used in the modeling process. In the process of analysis we can test the model: does the model represent the hypothesis at hand?. Is the data concisely represented?. Does the model reproduce the problem? Moreover, the analysis of the model must consider the characteristics of the biological system that have been used to define the model properties (cf. Section 1.2). A **verification analysis** evaluates if the model meets its specification by checking the structure and behavior of the model. A **validation analysis** intends to ensure meeting the operational requirements by executing a model checking on the result.

One important (experimental) strategy to verify and validate the model is using simulation. The analysis of the results of a simulation can demonstrated the proper behavior of the model and also it can contribute to possibly further refine the model so the results approximate reality as close as possible. Although model simulations will never replace laboratory experiments, a model allows one to probe system behavior in ways that would not be possible in the lab. Simulations can be carried out quickly (often in seconds) and incur no real cost. Model behavior can be explored in conditions that are otherwise difficult to achieve in a laboratory settings. Every aspect of model behavior can be observed at all time-points. Furthermore, model analysis yields insights into why a system behaves the way it does, thus providing links between network structure modeled and observed in real experiments.

1.6 Petri nets in biology

In our descriptions until now it has been implicitly assumed that the Petri net formalism is applied on a case from the domain on biology, especially due to the concurrent nature that is overwhelmingly present in biological systems. The Petri net formalism has been successfully applied in systems biology in the representation of networks [67]; in particular, applications of Petri nets in regulatory and metabolic pathways [3, 45, 52, 59, 98, 128]. In addition to conventional graphical representations, the flow of information can be demonstrated and visualized. Different networks can be connected and what-if scenarios can be tried to understand the effect of presence and/or absence of a component. The common denominator of these domains is the concurrent behavior that needs to be captured in a model.

Systems biology requires integration and a next step in the application of Petri nets is to use it in the integration of the components in a systems biology analysis. Such components are described at different levels, the process level, the organismal level, the organ/tissue level, the molecular level and the gene level. A collection of PN's typically can integrate all these levels. We need to investigate how this can be done in the best possible manner and rather than constructing a theoretical framework, we investigate how the existing PN variants can be invoked in a multi-scale problem. The infection of *Mycobacterium* to a host is such a typical problem. It already includes two players each with the own agenda and still their strategies interfere. The *Mycobacterium tuberculosis* is the evil-doer for tuberculosis in human. In order to study the infection of mycobacterium a model system is used, the zebrafish. In zebrafish the *Mycobacterium marinum* is used to infect the host. The infection in the zebrafish model is used a case study to understand the process itself, how it can be represented in a Petri net and how it can be elevated in a Petri net representing a systems biology approach. In the next chapters the zebrafish infection is further explained in order to map it to a Petri net model. These models then gradually integrate more layers are able to represent this complex invader-host process.

1.7 Structure of this Thesis

This thesis is structured as follows: In Chapter 2, the steps related to the dynamics of the mycobacterial infection process and early immune response are used as basis for the first model. In this chapter, a qualitative model is presented in which one can visualize the process involved in the infection from the moment the bacteria enter in the host: Immune system detection, phagocytosis, migration, proliferation, granuloma formation and infection dissemination are the steps that drive the dynamics of the infection process. Chapter 3 presents a model that represents how the bacteria explore regulatory pathways, to evade host immune responses and enhance the infection inside the immune cell. In this chapter the introduction of hierarchical modeling process is introduced. The hierarchical model provides an overview of the important host-cell signaling pathways that occur at multiple (molecular, intracellular and intercellular) scales. Chapter 4 demonstrates the power of the computational method by coupling the two models from the previous chapters in one hierarchical structure. It presents a qualitative model that represents the infection process from the molecular interaction to granuloma dissemination. It is possible to correlate the information of the model, its structure and its results to behavior observed *in vivo*. Moreover, a 3D visualization is added to support the interpretation of the biological process. In Chapter 5, the qualitative model presented previously is extended to a quantitative model that, through support of empirical data, enables to do quantitative analysis as well as perform simulations and predictions. Data from zebrafish infection

studies were used as a basis for such quantitative model and different simulation scenarios were performed in order to verify and validate the model.

Following the content of Chapters 1 to 5, it gradually optimizes the knowledge and functionality of the modeling decisions by increasing the complexity of the model so as to better reproduce the biological behavior. As a result, new methodologies of modeling a biological system using computational methods were developed. An overview and discussion of this thesis is presented in Chapter 6 with the insights obtained from the research explored in previous chapters as well as pointing to new directions in the research of modeling biological systems.