

7
Towards Understanding the Role and Function of Regulatory Networks in Microorganisms

Krist V. Gernaey, Morten Lind, and Sten Bay Jørgensen

7.1
Introduction

Microbial function is carefully controlled through an intricate network of proteins and other signaling molecules, which enables microorganisms to react to changes in their environment. Thus microorganisms constitute examples of entire autonomous chemical plants, which are able to produce and reproduce despite a shortage of raw materials and energy supplies. Understanding the intracellular regulatory networks of microorganisms is important to process systems engineering for several reasons. One reason is that the microbial systems still constitute relatively simple biological systems, the study and understanding of which may provide a better understanding of higher biological systems such as human beings. Furthermore microbial systems are used, often following genetic manipulation, to produce relatively complex organic molecules in an energy-efficient manner. Understanding the regulatory networks in microorganisms, and especially understanding how to couple the microbial regulatory functions and higher level process and production control functions, is a prerequisite for process engineering.

The focus of this chapter is discussing basic modeling problems when describing regulatory networks in microorganisms. In this introduction, we first present arguments to explain why researchers from so many different disciplines, but especially from the systems engineering field, are interested in gaining an increased understanding of the functioning and design principles of these regulatory networks. Second, fundamental modeling problems are highlighted. The introduction finishes with a statement of the purpose of this chapter and an overview of the remainder of the chapter.

Computer Aided Process and Product Engineering. Edited by Luis Puigjaner and Georges Heyen

ISBN: 3-527-30804-0

7.1.1 Why Gain an Understanding of Regulatory Network Function?

From an industrial point of view, a microorganism can be considered an autonomous plant suited for the production of complex biomolecules. Industrial production of chemicals such as food and cosmetics ingredients is, for example, increasingly based on biotransformation processes (Cheetham 2004), where the conversions of raw materials to useful products are catalyzed either by microorganisms or by enzymes obtained from microorganisms.

On a macroscale, for example, in a bioreactor where millions of microorganisms reside, the conversion of raw materials to valuable products by the microorganisms has traditionally been monitored using probes for pH, dissolved oxygen, gas phase composition, and biomass concentration measurements. In recent years, however, interest in system-level understanding of regulatory networks in biological systems, including microorganisms such as *Escherichia coli* (a prokaryotic organism) and *Saccharomyces cerevisiae* (a eukaryotic organism), has been an important research theme. This increasing interest in the microscale is, to a large extent, boosted by the fact that biology has evolved from being a data-poor science to a data-rich science, an evolution driven by progress in molecular biology, particularly in genome sequencing and high-throughput measurements (Kitano 2002; Vukmirovic and Tilghman 2000). Indeed, contrary to earlier efforts in developing system-level understanding of biological systems, it is now possible to collect informative system-wide data sets on protein-DNA interactions, protein-protein interactions, and increasingly small-molecule interactions as well (Ideker and Lauffenburger 2003). An ever-increasing number of advanced analytical methodologies allow detailed monitoring of the dynamics of intracellular processes (e.g., Chassagnole et al. 2002). In the post-genomic era, the availability of genome sequence data of several organisms, including *E. coli* and *S. cerevisiae*, has already led to a focus shift from molecular characterization and sequence analysis to developing an understanding of functional activity and the interaction of genes and proteins in pathways (Salgado et al. 2004; Vukmirovic and Tilghman 2000; Wolkenhauer et al. 2003), a research area called functional genomics. In fact, microorganisms are networks of genes, which make networks of proteins, which regulate genes, and so on ad infinitum (Vukmirovic and Tilghman 2000). Gene expression and regulation, i.e., to understand the organization and dynamics of genetic, signaling, and metabolic pathways, is considered to be one of the main research challenges of the next 50 years (Wolkenhauer et al. 2003).

Developing a system-level understanding of biological systems can be derived from insight into four key properties (Kitano 2002): (1) System structure, for example, the network of gene interactions; (2) System dynamics, for example, the dynamic response of a biological system to a change in the substrate concentration; (3) The control method, that is, understanding of the mechanisms that control the state of the cell; (4) The design principles of the cell, for example, simulations can support strategies to construct biological systems. Reaching a system-level understanding of biological systems necessitates multidisciplinary research efforts to unravel the complexity of biological systems.

One could, of course, wonder why not only biologists, but people coming from very different research fields, are involved and interested in developing an increased understanding of biological function. First of all, involving other research disciplines can be considered a necessity. Secondly, the versatility of microorganisms to produce industrially relevant chemicals, by expression of the appropriate gene(s), is an important factor promoting research aimed at gaining an increased understanding of biological function. Thirdly, the similarities between microorganisms and chemical plants, combined with increased data availability, almost naturally lead to an interest of systems engineering in understanding biological function. Each of these points will be presented in a little more detail below.

Biology has grown to a scientific area that generates far more data than biologists are used to handle. The amount of complex data that are and will be generated with the technologies now available, and the need for modeling to understand the way networks function, requires – for efficiency reasons – that disciplines outside of traditional biology collaborate on the problem of understanding biological function (Vukmirovic and Tilghman 2000). The most obvious collaborators for this endeavor are systems theoreticians and engineers.

The industrial interests in the understanding of biological function is illustrated by the tremendous and steadily growing list of products resulting from biotransformation processes mentioned in the review paper of Cheetham (2004). Clearly, improved understanding of the regulatory mechanisms responsible for the expression of the gene encoding a product of interest might lead to higher production rates (more product can be produced within an existing industrial facility), increased production yields (raw materials can be utilized more efficiently), and shorter time to market. Thus, for an industrial biotransformation process, the results of improved understanding of biological function are directly related to increased profit. The bacterium *E. coli*, to name one popular example, was called a "workhorse microorganism" for recombinant protein production and a fundamental understanding of intracellular processes, such as transcription, translation, and protein folding, make this microorganism even more valuable for the expression of recombinant proteins (Baneyx 1999). Knowledge of the mechanisms of complex regulatory networks involved in the transformation of extracellular signals into intracellular responses is important to improve the productivity of microorganisms. The *E. coli* lactose utilization *(lac)* operon, which will be used later in this chapter to illustrate the complexity of regulatory networks, has served as one of the paradigms of prokaryotic regulation, and therefore a considerable number of the promoters used to drive the transcription of heterologous genes (genes carrying the genetic code for a product of interest) have been constructed from *lac*-derived regulatory elements (Baneyx 1999; Makrides 1996).

The interest of systems engineering groups in contributing to an improved understanding of microbial function becomes clear by considering the number of review and position papers that were published in recent years (e.g., de Jong 2002; Doyle 2004; Ferber 2004; Hasty et al. 2001; Ideker and Lauffenburger 2003; Kitano 2002; Smolen et al. 2000; Wolkenhauer et al. 2003). Microbial function is carefully controlled through an intricate network of proteins and other signaling molecules. Free-

living bacteria have to maintain a constant monitoring of extracellular physicochemical conditions in order to respond and modify their gene expression patterns accordingly (Lengeler et al. 1999; Salgado et al. 2004). Microorganisms by themselves thus constitute examples of entire autonomous chemical plants, which are able to produce and reproduce despite a shortage of raw materials and energy supplies. Microorganisms can sense changes in the surrounding environment, and subsequently control the expression of genes in reaction to these changes. Such adaptation of the cell to changes in the environment is crucial for the survival of the cell, since it allows economical use of cellular resources (Lengeler et al. 1999), as a result of regulating the expression of all genes to produce the optimal amount of gene product at any given point in time. The energy consumption for protein synthesis and the relatively short half-life of the mRNA molecules are reasons for a cell to control both the types and amounts of each protein (Wolkenhauer et al. 2003). Making a link to chemical production plants, cell behavior can be compared with adjusting the production capacity and the operation strategy of a chemical plant to the availability of limiting amounts of raw materials, aiming at minimizing plant operating costs. In view of the similarities between the functioning of a microorganism and a chemical plant, it is not overly surprising that systems-engineering thinking is increasingly applied to these biological systems.

Systems engineering has different applications. Reverse engineering, aimed at unraveling the functionality of regulatory networks, is one of the major goals in systems biology. However, more and more effort is also directed into forward engineering, aiming at the design of regulatory networks with a desired functionality (Elowitz and Leibler 2000; Ferber 2004; Hasty et al. 2001). This research area is also called synthetic biology, to distinguish it more clearly from the reverse engineering efforts in systems biology. Contrary to systems biologists, who analyze data on the activity of thousands of genes and proteins, synthetic biologists simplify and build. They create models of genetic circuits, build the circuits, see if they work, and adjust them if they don't (Ferber 2004). In the synthetic biology field, one of the future visions is the construction of cells as small factories for complex chemical compounds such as pharmaceuticals.

7.1.2 Levels of Abstraction, Function, and Behavior

It is important to realize that models of biological systems play a central role in both reverse and forward engineering. However, a model of biological systems represents different types of knowledge and assumptions about the system depending on the problem to be solved.

Thus, the aim of reverse engineering is to *interpret* the biological system in order to explain how its structure and behavior originate from interactions of its subsystems. The interpretation is based on a model of the expected structure and behavior. The model can be based on either previous experience or represent a purpose or design intention. In both cases, the aim is to test whether the model (the hypothesis)

is an adequate representation of empirical data. In contrast, the aim in forward engineering is to *predict* structure and behavior of a biological system from knowledge of the structure and behavior of its parts, and to test if the predictions match subsequent empirical observations or design intentions. In prediction, the model is assumed to be adequate and used to produce hypotheses about unobservable structure or behavior. Models have accordingly different roles in reverse and forward engineering of biological systems.

A general problem in the modeling of dynamic systems is to determine a proper level of abstraction. Most natural and artificial systems can be modeled on a variety of levels but the choice of level is of particular importance for biological systems due to their extreme complexity. Unfortunately, levels can be defined relative to several dimensions in the modeling problem. For example, we can describe the spatial structure (the anatomy) on many part-whole levels, and we can also describe the behavior (dynamics) at several part-whole levels of temporal resolution.

Another way to define levels in biological systems is to consider their functional organization. The idea here is to describe the biological system as a goal-directed system and to decompose the system into subsystems so that each subsystem serves the needs or provides the means for its superordinate system. The analysis that brings about this type of system information is usually called means-end analysis or functional modeling, and has been developed within cognitive science and artificial intelligence research.

The use of means-end analysis to define levels of abstraction is a very powerful approach to handle the modeling of complex dynamic systems (Lind 1994). It is of particular importance for modeling systems with embedded control systems, such as biological systems. Control systems play a direct role in the constitution of functional levels (Lind 2004b) and their function can therefore not be described properly without means-end concepts.

Note that when using concepts of means-end analysis we must distinguish carefully between the concepts of behavior and function. The two notions are often confused such that function is thought to be more or less synonymous with behavior. We stress the teleological meaning of function; it represents the role the system has in the fulfillment of a purpose or goal. Behavior refers to what happens when a system reacts to an intervention or a disturbance. Descriptions of behavior have accordingly no connotations to purposes or goals and are therefore distinct from functional descriptions.

We will later return to a discussion of means-end analysis and functional concepts in modeling complex dynamic systems.

7.1.3 Overview of the Chapter

The main purpose of this chapter is to discuss basic modeling problems that arise when attempting to describe regulatory networks and their function in microorganisms. The focus is on the representation of the regulatory mechanisms in micro-

organisms applied for production purposes. First, the central dogma of biology will be introduced briefly. The *E. coli lac* operon is subsequently used as an example of a regulatory network structure in microorganisms, illustrating the complexity of such networks. The *lac* operon example is followed by a discussion of the essential steps in the central dogma, identifying possible sites for control actions. Formalisms to model the regulatory networks are then introduced briefly, and strategies developed to deal with the complexity of regulatory networks in microorganisms are highlighted. Finally, means-end analysis and functional modeling are presented as suitable methods to represent the complex interactions in regulatory networks, and their use is illustrated by means of the *lac* operon example. The chapter ends with a discussion and conclusions.

7.2
Central Dogma of Biology

According to the central dogma of biology, a term coined by Sir Francis Crick, three processes, illustrated in Fig. 7.1, are responsible for the conversion of genetic information. (1) DNA replication: a process involving several enzymes and duplicating a double stranded nucleic acid to give identical copies; (2) Transcription: a DNA segment constituting a gene or an operon is read and transcribed into a single stranded sequence of RNA, the messenger RNA (mRNA), by the RNA polymerase enzyme; (3) Translation: the mRNA sequence is translated into a sequence of amino acids, where the ribosome reads three bases (a codon) at one time from the mRNA, translates them into one amino acid, and subsequently joins the amino acids together in an amino acid chain (protein formation). The resulting proteins, depending on their structure, may function as transcription factors (or regulatory proteins) binding to regulatory sites of other genes, as enzymes catalyzing metabolic reactions, or as components of signal transduction pathways.

In prokaryotic cells, transcription and translation take place simultaneously. In eukaryotic cells, the mRNA is formed in the cell nucleus, which is separated from the rest of the cell. The mRNA undergoes further processing and modifications before it is transported out of the nucleus, where the ribosomes take care of the translation.

New research results appearing in the early 1970s meant that the basic principle of the central dogma, that information flows uniquely from DNA to RNA to protein,

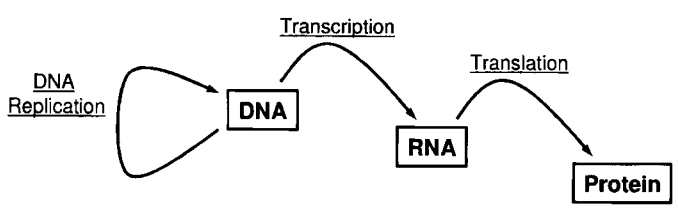

Figure 7.1 Schematic illustration of the central dogma of molecular biology.

needed adjustment. Indeed, with the discovery of reverse transcriptase in retroviruses, the central dogma (Fig. 7.1) was extended to include the ability to convert RNA into DNA. Prions also form an exception to the original formulation of the central dogma, since these proteins can induce misfolding of other proteins.

7.3 Complexity of Regulatory Networks

Of course, Fig. 7.1 is a simplified representation of the complex processes taking place in, for example, a prokaryotic microorganism. In reality, the central dogma of molecular biology includes quite a number of possibilities for regulation of DNA replication and protein production processes, which in this chapter will be first illustrated with an example of the production of enzymes (proteins) in a prokaryotic organism.

7.3.1 An Example of Transcriptional Regulation: the *lac* Operon

The example focuses on transcriptional regulation, the intracellular control mechanisms that influence the rate of the process responsible for converting the genetic information contained in the DNA into mRNA. In prokaryotes, genes are grouped into operons (Fig. 7.2). An operon can thus consist of several structural genes, where each structural gene encodes a protein. The genetic information contained in the genes in one operon together provides the cell with the capability to perform a coordinated function, for example, the execution of one metabolic pathway to produce a specific amino acid, or the (partial) conversion or degradation of one specific substrate to a metabolic intermediate. All genes in an operon are transcribed at once, resulting in polycistronic messenger RNA (mRNA), that is, mRNA encoding for several proteins.

An example of an operon, in this case the well-known *E. coli lac* operon, is provided in Fig. 7.2A. The regulation of the transcription of the *lac* operon is the result of a combination of different mechanisms.

7.3.1.1 Absence of Extracellular Glucose: Induction of the *lac* Operon

Mechanism

The first mechanism that will play a role in the transcription of the *lac* operon is induction, which is schematically represented in Fig. 7.2. The foundation for the current level of understanding of this regulatory mechanism is the operon model formulated by Jacob and Monod (1961), where the clear distinction between structural and regulatory genes was introduced.

The *lac* operon consists of three structural genes (Fig. 7.2), containing the genetic code for enzymes that will be responsible for the uptake and conversion of the sub-

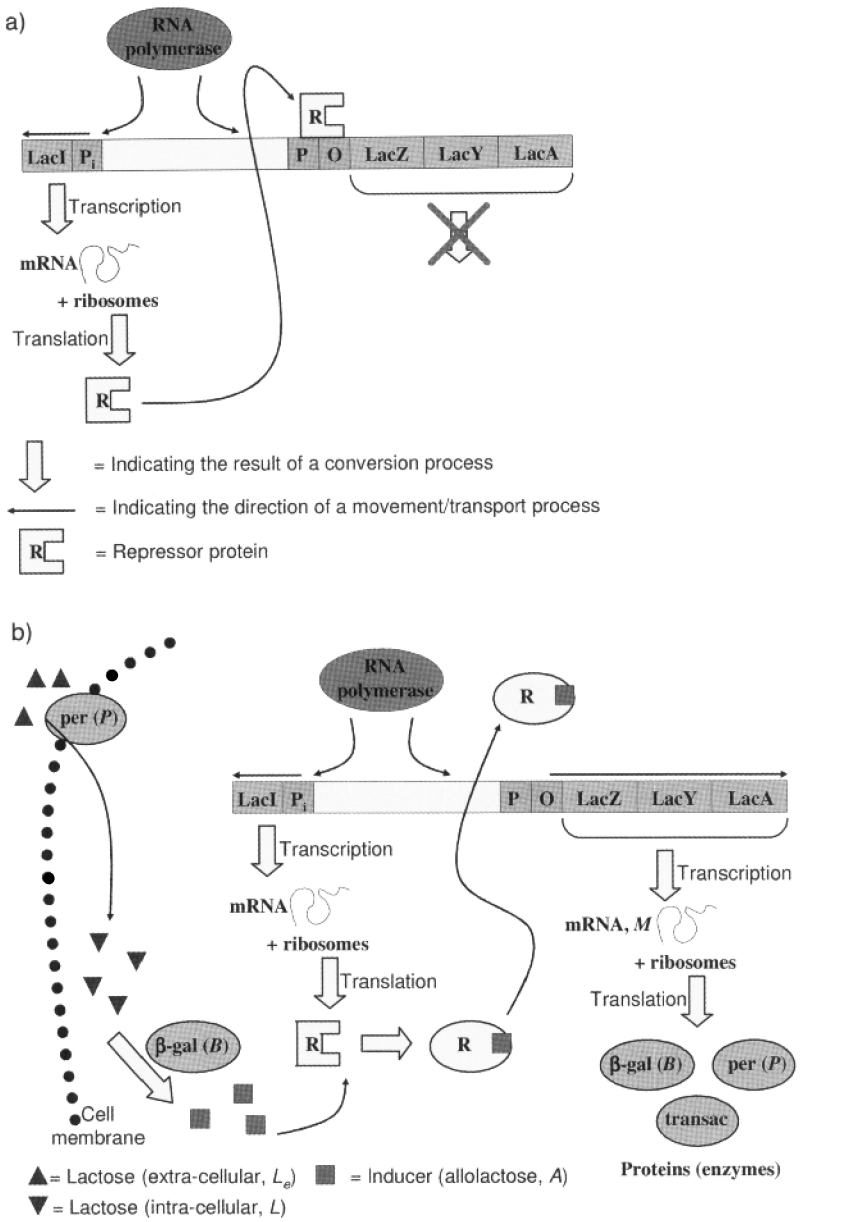

Figure 7.2. Induction of the *lac* operon in the absence of glucose in the growth medium (based on the model of Yildirim and Mackey (2003)). **A** Repressed *lac* operon; **B** Induced *lac* operon (A, B, P, L, L_e and M, in italics in the figure, indicate the variables considered in the model of Yildirim and Mackey 2003). LacI = gene encoding for repressor protein; P_i = promoter region for repressor protein; P = promoter region for structural genes; O = operator region for structural genes; LacZ = β-galactosidase gene; LacY = β-galactoside permease gene; LacA = β-galactoside transacetylase gene; β-gal = β-galactosidase; per = β-galactoside permease; transac = β-galactoside transacetylase.

strate lactose into its building blocks glucose and galactose. In the simplified representation in Fig. 7.2, the structural genes are preceded by one operator and one promoter.

In the absence of extracellular glucose and lactose, the *lac* operon is repressed. The repression of the *lac* operon originates from the presence of a fourth gene, containing the genetic code for a repressor protein. This *lac* repressor gene, or regulatory gene, provides one of the keys for understanding the regulatory mechanism that allows *E. coli* bacteria to grow on lactose in the absence of glucose. The *lac* repressor gene has its own promoter (P_i in Fig. 7.2) allowing RNA polymerase to bind to P_i and to transcribe the *lac* repressor gene. The ribosomes translate the *lac* repressor mRNA, to form the *lac* repressor protein. In the absence of lactose, the *lac* operon is repressed, meaning that the *lac* repressor protein is bound to the operator region of the *lac* operon, preventing the RNA polymerase to bind to the promoter of the structural genes, and thus repressing the transcription of the structural genes (see Fig. 7.2A).

Allolactose is the inducer of the *lac* operon and results from the intracellular conversion of lactose following uptake trough the cell membrane (Lengeler et al. 1999; Wong et al. 1997; Yildirim and Mackey 2003). Indeed, in the absence of extracellular glucose, and when lactose is present in the growth medium, lactose is transported into the cell by the β-galactoside permease (Fig. 7.2B). Intracellular lactose is subsequently converted into glucose, galactose, and allolactose. The *lac* repressor protein undergoes a conformational change after binding the inducer allolactose, and is then no longer capable of binding to the operator region of the structural genes (see Fig. 7.2B). RNA polymerase can now bind to the promoter of the structural genes and produce mRNA, which is subsequently converted into proteins (β-galactosidase, β-galactoside permease, and β-galactoside transacetylase) by the ribosomes. This induction mechanism of the *lac* operon is a positive feedback loop: increasing intracellular lactose concentrations will lead to an increase in the expression of the *lac* operon, and thus result in an increased production of, for example, permease enzyme molecules, which will again lead to increased intracellular lactose concentrations, until the maximum protein production rate is reached. Depletion of extracellular lactose will result in repression of the *lac* operon.

A First Principles Model Example: Model of the *lac* Operon Induction

Modeling plays an important role in unraveling regulatory mechanisms. A model for the induction of the *lac* operon was proposed by Yildirim and Mackey (2003) and will be used here as an example. Since this model only considers the induction mechanism, the model is only valid in the absence of extracellular glucose. The model consists of five states (see Fig. 7.2B): intracellular lactose (L), allolactose (A), mRNA resulting from the transcription of the structural genes (M), β-galactosidase (B), and β-galactoside permease (P). The system is modeled with five nonlinear delay differential equations (DDEs) provided in Eqs. (1–5), and has two external inputs, the extracellular lactose concentration (L_e), which is assumed to be constant, and the growth rate (μ). Note also that spontaneous mRNA generation has been omitted in Eq. (1), since its contribution could be neglected.

$$\frac{dM}{dt} = \alpha_M \cdot \frac{1 + K_1 \cdot \left(e^{(-\mu \cdot \tau_M)} \cdot A(t - \tau_M)\right)^2}{K + K_1 \cdot \left(e^{(-\mu \cdot \tau_M)} \cdot A(t - \tau_M)\right)^2} - \gamma_M \cdot M - \mu \cdot M \quad (1)$$

$$\frac{dB}{dt} = \alpha_B \cdot e^{(-\mu \cdot \tau_B)} \cdot M(t - \tau_B) - \gamma_B \cdot B - \mu \cdot B \quad (2)$$

$$\frac{dA}{dt} = \alpha_A \cdot B \cdot \frac{L}{K_L + L} - \beta_A \cdot B \cdot \frac{A}{K_A + A} - \gamma_A \cdot A - \mu \cdot A \quad (3)$$

$$\frac{dL}{dt} = \alpha_L \cdot P \cdot \frac{L_e}{K_{Le} + L_e} - \beta_{L1} \cdot P \cdot \frac{L}{K_{L1} + L} - \alpha_A \cdot B \cdot \frac{L}{K_L + L} - \gamma_L \cdot L - \mu \cdot L \quad (4)$$

$$\frac{dP}{dt} = \alpha_P \cdot e^{(-\mu \cdot (\tau_B + \tau_P))} \cdot M\left(t - (\tau_B + \tau_P)\right) - \gamma_P \cdot P - \mu \cdot P \quad (5)$$

Such a model is, of course, based on a number of model assumptions. In the case of this model, the delay times in the DDEs are assumed to be related to different biological phenomena. The delay in Eq. (1) represents the fact that there is a delay τ_M between the start of transcription and the production of a complete mRNA. The delay τ_B in Eq. (2) represents the delay between the start of mRNA translation and the appearance of β-galactosidase, and τ_B thus corresponds to the time needed for translation. The delay $\tau_B + \tau_P$ in Eq. (5) includes the assumption that β-galactosidase production needs to be finished before the production of the β-galactoside permease can start (delay τ_B), whereas τ_P represents the time needed to produce β-galactoside permease. The selection of DDEs with constant delays to model this regulatory mechanism actually includes the assumption that translational regulation does not influence the protein production rate. Indeed, translational regulation would lead to variations in the delays τ_B and τ_P. Furthermore, transcriptional control is only modeled as influencing transcription initiation. The constant delay τ_M in Eq. (1) includes the assumption that no regulatory mechanism influences transcriptional elongation and transcription termination.

Besides the assumptions underlying the choice of the delay times in the model example, it is of utmost importance to have a proper understanding of the assumptions that were made when describing the transcriptional regulation mechanism of the *lac* operon. Actually, the model example in Eqs. (1–5) does not provide a detailed description of this regulatory mechanism (Santillán and Mackey 2004). Instead, the model example lumps the regulatory mechanism into one Hill-type equation, describing the production of mRNA as a function of the inducer, the allolactose concentration (Eq. (1)). The dynamics of the *lac* repressor protein, the *lac* repressor protein-allolactose complex and the RNA polymerase enzyme are not considered explicitly. The original paper by Yildirim and Mackey (2003) can be consulted for further detail on the kinetic expressions.

A set of parameters, suitable initial conditions, and steady-state values obtained with these parameters can be found in Yildirim and Mackey (2003), as well as a demonstration of the capabilities of the model to describe the dynamics observed experi-

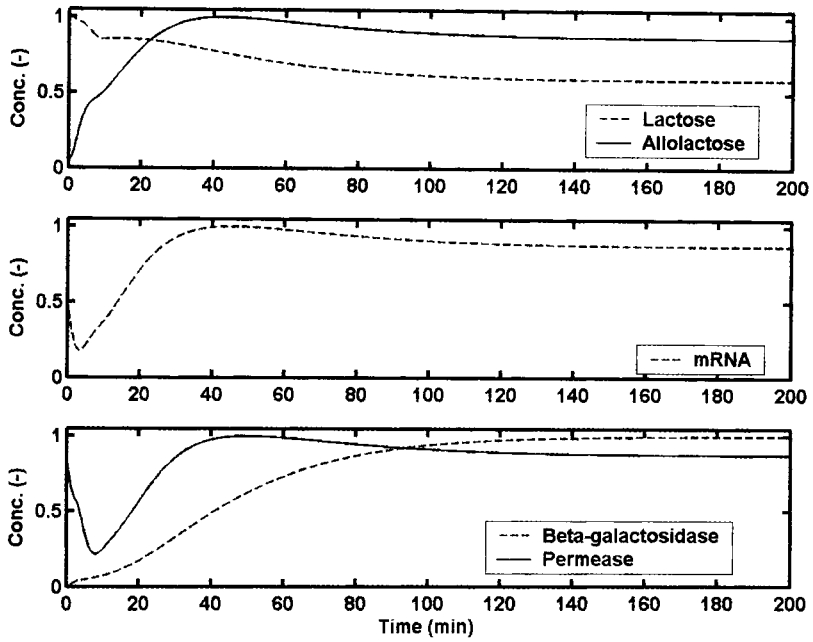

Figure 7.3 Relative concentration dynamics of intracellular lactose (L), allolactose (A), mRNA (M), β-galactosidase (B), and β-galactoside permease (P) predicted by the model of Yildirim and Mackey (2003) for a step-change of glucose to lactose feeding at $t = 0$ ($L_e = 0.08$ mM, $\mu = 0.0226$ min^{-1}). The initial values provided by Yildirim and Mackey (2003) were used.

mentally. Figure 7.3 provides the relative concentration dynamics predicted by the model for a step change in the feed from glucose to lactose at $t = 0$. For the figure, the data were scaled by dividing each concentration time series by its maximum value. After the appearance of lactose, the model predicts an increase of the allolactose concentration, resulting in induction of the *lac* operon and subsequent production of β-galactosidase.

7.3.1.2

Presence of Extracellular Glucose: Inducer Exclusion and Carbon Catabolite Repression

Glucose, not lactose, is the preferred carbon source of *E. coli* bacteria. In the presence of both glucose and lactose, *E. coli* bacteria will first grow on glucose, and the enzymes encoded in the *lac* operon will not be produced. However, when all glucose is consumed, the presence of lactose will induce the production of the enzymes encoded in the *lac* operon, and thus provide *E. coli* bacteria with the capability of growing on lactose as an alternative substrate. When all lactose is consumed, the production of the enzymes encoded in the *lac* operon will be turned off again, thereby economizing on the cellular resources.

The *lac* operon inducer mechanism (Fig. 7.2) alone cannot explain why the *lac* operon is repressed when both extracellular glucose and lactose are present. Clearly, there must be additional regulatory mechanisms beside the *lac* operon inducer mechanism. Figure 7.4 provides a diagram of the main regulatory mechanisms of the *lac* operon included in the model of Wong et al. (1997), and indicates that the *lac* operon is indeed controlled by glucose at two levels (Lengeler et al. 1999; Wong et al. 1997): (1) Inducer exclusion; and (2) Catabolite repression. Extracellular glucose, while being transported into the cell by the phosphoenolpyruvate:sugar phosphotransferase system (PTS), an important transport system of *E. coli* bacteria, is converted to glucose 6-phosphate (G6P). The cell growth rate is assumed to depend on the G6P concentration. At this point, the model of Wong et al. (1997) assumes that the glucose uptake rate via the PTS is related to the external glucose concentration via Monod kinetics. It is important to realize, though, that the PTS itself is a protein complex consisting of several enzymes that will transfer a phosphate group from phosphoenolpyruvate (PEP) to glucose during glucose uptake, resulting in G6P (Postma et al. 1993). In the absence of extracellular glucose, cyclic adenosine monophosphate (cAMP) is synthesized by the adenylate cyclase (AC) enzyme, accumulates in the cell, and binds to the cAMP receptor protein (CRP). cAMP is considered an alarmone for carbon starvation of *E. coli* (Lengeler et al. 1999). Alarmones are molecules that signal stress conditions. The phosphorylated form of one of the enzymes of the PTS (IIA^{Glc}) is an activator of AC (Postma et al. 1993). The cAMP:CRP complex binds to the CRP binding region, which is located near the *lac* promoter, and will enhance transcription initiation and also transcription of the structural genes by the RNA polymerase:σ factor complex (RNAP:σ in Fig. 7.4). In the presence of extracellular glucose, no cAMP will be generated since IIA^{Glc} is in its non-phosphorylated form, and no cAMP:CRP complex will be formed, resulting in catabolite repression, or the repression (inactivation) of certain sugar-metabolizing operons (such as the *lac* operon in this example) in favor of the utilization of an energetically more favorable carbon source (glucose in this example). In Fig. 7.4, the presence of extracellular glucose results in inhibition of the transport of lactose by the *lac* permease, a phenomenon known as inducer exclusion. It has been demonstrated that it is not extracellular glucose itself, but the non-phosphorylated form of the PTS enzyme IIA^{Glc} that inhibits the uptake of lactose by the *lac* permease (Postma et al. 1993).

Several first-principles models have been formulated to describe the combined effects of inducer exclusion, carbon catabolite repression, and induction on the *lac* operon (e.g., Kremling et al. 2001; Santillán and Mackey 2004; Wong et al. 1997). An accurate description of the phenomena observed when *E. coli* bacteria grow on a mixture of glucose and lactose necessitates inclusion of the glucose effects in mathematical models of the *lac* operon, which results in rather complex models. In the model of Santillán and Mackey (2004), an additional layer of complexity in the regulation of the *lac* operon is considered explicitly by taking into account in the model that the *lac* operon has three different operators, two different cAMP:CRP binding sites and two different promoters. Also, the model takes into account that DNA can fold in such a way that a single repressor molecule can bind to two different operators. Considering all possible interactions between the *lac* operon, the repressor, the cAMP:CRP com-

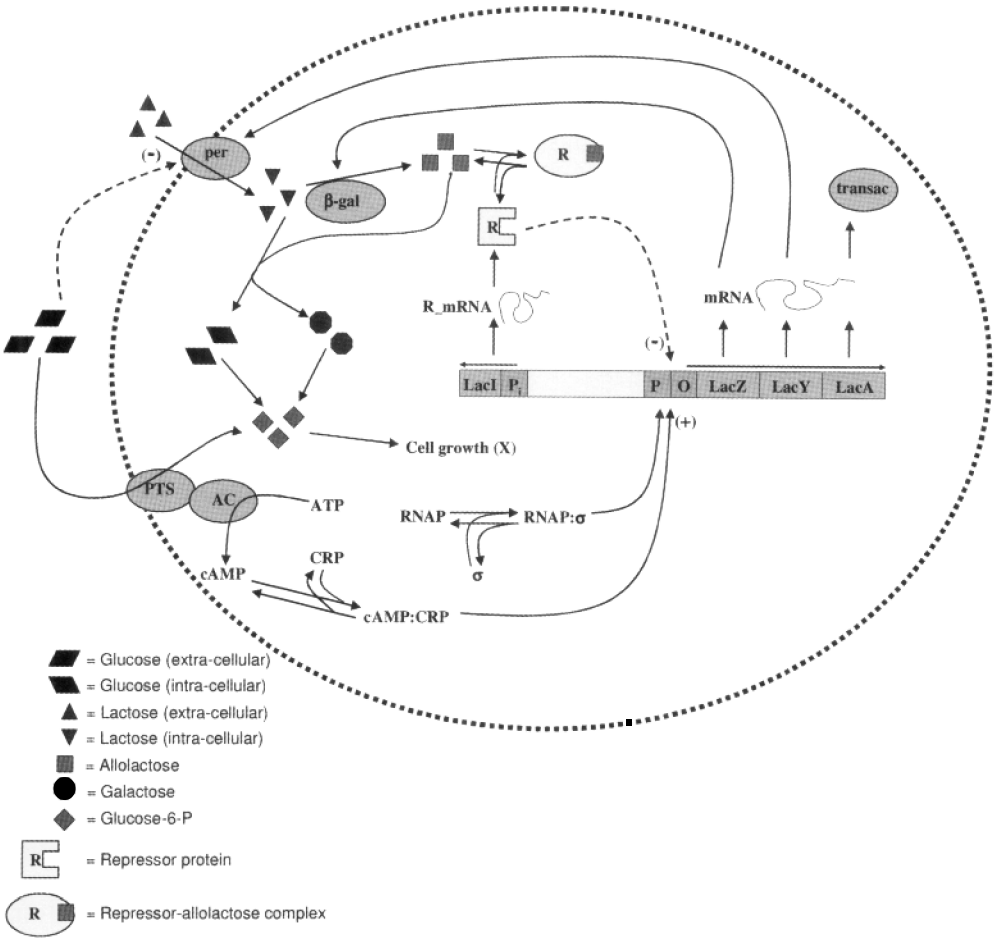

Figure 7.4 A diagram of the *lac* operon, schematically representing mechanisms for inducer exclusion, catabolite repression, and induction of the *lac* operon (Wong et al. 1997). See the main text for an explanation of the symbols.

plex, and the RNAP:σ complex results in 50 different binding states for the *lac* operon.

7.3.2 Potential Sites for Control Actions

The *lac* operon example is entirely focused on transcriptional regulation, more specifically transcription initiation, meaning that only part of the mechanisms that control the transcription of the structural genes, and thus the production of mRNA, are

considered. In fact, the emphasis is on transcriptional regulation in the majority of the studies on regulatory networks in microorganisms. In a review on modeling transcriptional regulation, Smolen et al. (2000) explained this by the fact that two key approximations have historically been used to model genetic regulatory systems: (1) control is exercised at the transcriptional level, (2) the production of a protein product is a continuous process, with the rate determined by the balance between gene activation and gene repression. As a consequence, there are few or no studies that model both translational and transcriptional control in any specific genetic system (Smolen et al. 2000).

However, prokaryotic cells are capable of rapidly adjusting to a wide range of environmental conditions (Lengeler et al. 1999), and this adjustment is achieved in two ways: (1) Instant responses involving a change in the activity of critical metabolic enzymes; and (2) Delayed, more long-term responses, involving positive or negative regulation of gene activity in a coordinated fashion. Transcriptional regulation is, of course, very important. However, Lengeler et al. (1999) provided examples that illustrated that regulation of protein synthesis not only takes place at the level of transcription initiation, that is, regulation of the binding of the RNA polymerase to the promoter, but also occurs at the levels of transcriptional elongation (i.e., during the formation of the mRNA chains) and termination (i.e., during the final stages of mRNA formation). Moreover, the mRNA is not a stable intermediate, and mRNA degradation provides a major control point of gene expression in virtually all organisms (Makrides 1996). Furthermore, Lengeler et al. (1999) indicated the importance of regulation during translation and mention protein stability as an additional factor that can be influenced by regulatory mechanisms. Finally, posttranslational modification of proteins is considered as a fine-tuning mechanism to adjust the activity of enzymes.

Summarizing, transcriptional regulation alone provides only part of a more complicated picture. Information on mRNA-levels in the cell provides an indication of gene expression and transcriptional regulation, but should also be combined with protein measurements to track the final gene expression product.

7.4
Methods for Mapping the Complexity of Regulatory Networks

Models are ideally suited for the representation of complex regulatory networks. The *lac* operon example is first compared to the size of the genome to further illustrate the complexity. As mentioned above, most systems can be modeled on a variety of levels of abstraction. The role of modeling is discussed and illustrated with a design example. Current developments in the construction of high-level models will be illustrated with the search for network motifs in regulatory networks, which is a high-level modeling example. A signal-oriented detailed first principles modeling methodology will subsequently be introduced as an attractive example of a low-level modeling approach. Finally, high- and low-level modeling approaches will be contrasted, and the link between high- and low-level models will be explained.

7.4.1 Complexity of Regulatory Networks

Regulatory networks are complex, which was illustrated using the *lac* operon example. There are two distinct characterizations of complexity that both apply to regulatory networks (Doyle 2004): (1) The classical notion of behavior associated with the mathematical properties of chaos and bifurcations (behavioral complexity); and (2) The descriptive or topological notion of a large number of constitutive elements with nontrivial connectivity (organizational complexity).

Chaos, bifurcations, and the occurrence of multiple static or dynamic states in biological systems are beyond the scope of this chapter. Instead, we are more interested in the organizational complexity of regulatory networks, more specifically in methodologies that allow representation of the complex regulatory networks and its many elements in a systematic way.

The *lac* operon example illustrates the degree of organizational complexity involved in the transcriptional regulation of a single prokaryotic operon, and gives an indication that proteins are the main catalysts, structural elements, signaling messengers, and molecular machines of living cells. The classical view of protein function focused on the local action of a single protein molecule, for example, the catalysis of one specific reaction in the metabolism of an organism. However, today there is a more expanded view of protein function, where a protein is defined as an element in the network of its interactions (Eisenberg et al. 2000). Each gene in the genome of an organism encodes for a protein. Thus, a first indicator of the overall organizational complexity of the regulatory networks is the number of genes in the genome. The genome of the well-studied prokaryote *E. coli* consists of 4408 genes with 179 transcriptional regulators (Salgado et al. 2004), whereas the genome of a typical eukaryote, *S. cerevisae*, consists of 6270 genes (Lee et al. 2002).

The absolute numbers of genes might already provide an indication that the organizational complexity of eukaryotic organisms is higher compared to prokaryotic organisms. Most proteins interact with several other proteins, resulting in complicated protein-protein interaction networks. It is exactly these multiple simultaneous interactions of many proteins in the network that need to be understood and represented to understand the functioning of a living cell. As a reaction on sensing a change in the extracellular environment, the gene expression pattern will be modified. Contrary to prokaryotic cells, eukaryotic cells have a nucleus. For Eukaryotes provoking a change in the gene-expression usually requires the movement of a protein from the body of the cell to the nucleus in response to the changes in the extracellular environment (Downward 2001). Thus, the cell compartmentalization will also necessitate the mapping and representation of transport processes between different cell compartments for eukaryotic cells, whereas such intracellular transport processes usually don't need to be considered for prokaryotic cells.

7.4.2 The Essential Role of Modeling

A conceptual problem arises of how to understand the operation of these complex systems. Positive and negative feedback within signaling pathways, crosstalk between pathways, time delays that may result from mRNA or protein transport, and nonlinear interactions all need to be considered to understand the operation of genetic regulatory systems (Smolen et al. 2000). Mathematical modeling of the dynamics of regulatory networks in microorganisms is therefore assumed to take on an essential role for a number of reasons (Mackey et al. 2004; Smolen et al. 2000): (1) Mathematical models can integrate biological facts and insights, that is, process knowledge on regulatory networks can be represented and summarized in a mathematical model; (2) Models can be helpful in identifying design principles for the regulatory networks; (3) Modeling can contribute to developing an understanding of the responses of both normal and mutant organisms to stimuli; (4) Model analysis can reveal potentially new dynamical behaviors that can then be searched for experimentally; and (5) Models can be used to verify the consistency and completeness of reaction sets hypothesized to describe specific systems. Failure of realistic mathematical models to explain experimentally observed behavior often points to the existence of unknown biological details, and can thereby also act as a guide for experimentalists.

Many modeling formalisms have been applied to the description of regulatory networks and were reviewed in detail by de Jong (2002), including directed graphs, Bayesian networks, Boolean networks, nonlinear ordinary differential equations (ODEs), piecewise-linear differential equations, qualitative differential equations (QDEs), partial differential equations (PDEs), and stochastic equations and rule-based formalisms. Discussing the advantages and drawbacks of each modeling formalism is beyond the scope of this chapter. Instead, we will limit ourselves to highlighting positive and negative aspects related to applying the most widespread modeling formalism for the detailed representation of regulatory networks, nonlinear ODEs.

Representing regulatory network dynamics with differential equations has certain advantages (Smolen et al. 2000; Hasty et al. 2001): (1) the model yields a continuous description allowing, in principle, for a more accurate physical representation of the system; (2) the models are supported by dynamical systems theory or, in other words, a large body of theory and methodology is available to characterize the dynamics produced by these models; (3) despite being computationally expensive, simulations with detailed models are still rapid compared to *in vivo* experimental work, allowing researchers to examine many hypotheses and concentrate experimental effort on the most promising of them.

Using differential equations also has disadvantages (Alur et al. 2002; de Jong et al. 2002; Smolen et al. 2000; Stelling et al. 2002): (1) The approach is computationally more intensive than, for example, the Boolean approach, where discrete updating of model states is applied. (2) Differential equation models require the assumption of a specific kinetic scheme, whereas the necessary mechanistic detail is in many cases not (yet) available. (3) There is often a lack of *in vivo* or *in vitro* measurements of

kinetic parameters in the models. Parameter values are indeed only available for a limited number of well-studied systems such as the *E. coli lac* and *trp* operon. Application of system identification methods combined with the increasing availability of data might alleviate this problem. (4) Cell compartments modeled with differential equations are assumed to be spatially homogeneous. In some situations this assumption is not appropriate. (5) Differential equations do not yield a good description of systems where only a limited number of molecules are involved. For identical initial conditions, two regulatory systems may reach different steady states as a consequence of stochastic processes resulting from the low number of molecules involved.

7.4.2.1

A Differential Equation Modeling Example: the Repressilator

Simulations with detailed mathematical models are important tools to analyze or to predict the behavior of regulatory networks and to subsequently draw conclusions regarding their design principles (Hasty et al. 2001). The repressilator (Elowitz and Leibler 2000) is an example of a rather simple synthetic network consisting of three transcriptional repressor systems, each consisting of a repressor gene encoding for a repressor protein. The names of the specific proteins are not important in the frame of this paper, and will therefore be omitted. When the genes (e.g., gA) are transcribed to mRNA, which is subsequently translated, the result is the production of a repressor protein (e.g., pA). The repressilator is a synthetic network, and was designed such that a negative feedback loop was obtained: The first repressor protein (pA) inhibits the transcription of the second repressor gene (gB). The second repressor protein (pB) inhibits the transcription of the third repressor gene (gC). And finally, the third repressor protein (pC) inhibits the transcription of the first repressor gene (gA). This is schematically presented in Fig. 7.5.

The repressilator example can be represented by a system of six coupled ODEs (Elowitz and Leibler 2000), where m_A, m_B, and m_C represent the mRNA concentrations, and p_A, p_B, and p_C represent the protein concentrations.

$$\frac{dm_A}{dt} = -m_A + \frac{\alpha}{(1+p_C^n)} + \alpha_0 \tag{6}$$

$$\frac{dp_A}{dt} = -\beta \cdot (p_A - m_A) \tag{7}$$

$$\frac{dm_B}{dt} = -m_B + \frac{\alpha}{(1+p_A^n)} + \alpha_0 \tag{8}$$

$$\frac{dp_B}{dt} = -\beta \cdot (p_B - m_B) \tag{9}$$

$$\frac{dm_C}{dt} = -m_C + \frac{\alpha}{(1+p_B^n)} + \alpha_0 \tag{10}$$

$$\frac{dp_C}{dt} = -\beta \cdot (p_C - m_C) \tag{11}$$

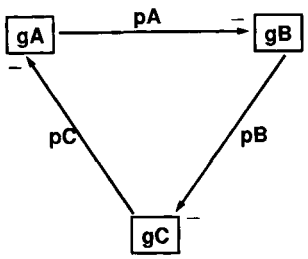

Figure 7.5 Scheme of the repressilator (Elowitz and Leibler 2000).

The parameters $\alpha + \alpha_0$, α_0, n, and β in Eqs. (6–11) represent the mRNA production rate of the derepressed promoters, the mRNA production rate of the repressed promoters (due to the "leakiness" of the promoter), a Hill coefficient, and the ratio of the protein decay rate to the mRNA decay rate, respectively.

Elowitz and Leibler (2000) demonstrated that depending on the selection of the model parameters, the system has a stable or unstable steady state. Both cases are illustrated in Fig. 7.6, and were obtained by simply varying the parameter α in the model. Note that both the protein concentrations and the time axis were normalized

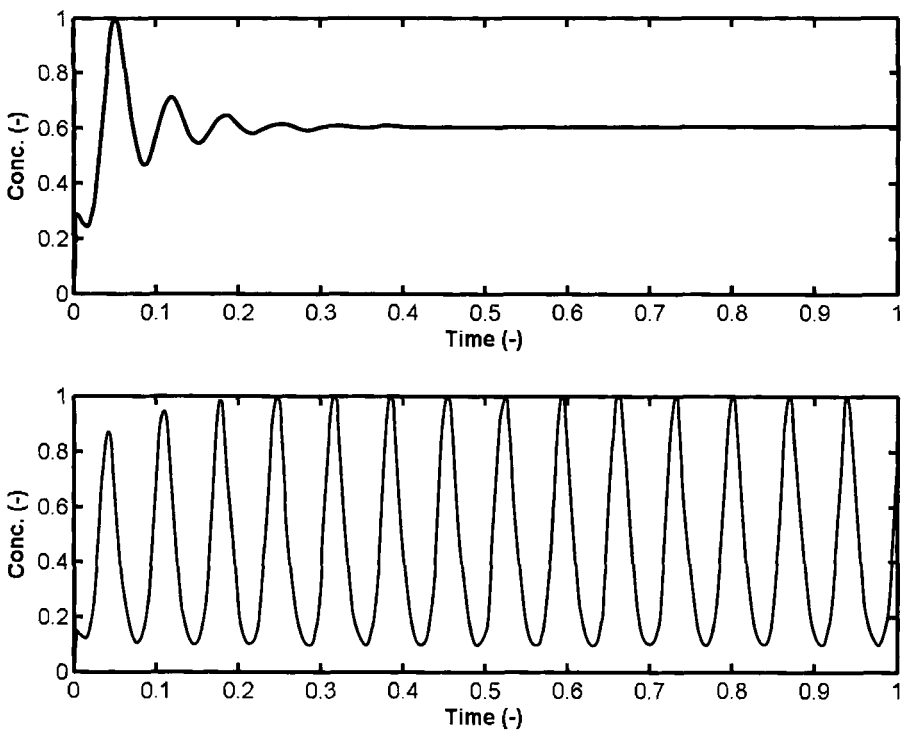

Figure 7.6 Evolution of the repressor protein concentration pA vs. time (both in relative units) for a stable and unstable steady state of the repressilator (Elowitz and Leibler 2000). The only parameter that was varied between both simulations is α.

in Fig. 7.6 by dividing through the maximum protein concentration and the end time of the simulation, respectively. Based on the modeling work, it was concluded that oscillations are favored by the following cellular design principles: strong promoters coupled to efficient ribosome-binding sites, tight transcriptional regulation (low α_0), cooperative repression characteristics, and comparable protein and mRNA decay rates. These model-based design principles were subsequently used to construct an *E. coli* mutant showing oscillatory behavior *in vitro* (Elowitz and Leibler 2000).

7.4.3
Modularizing Complex Regulatory Networks

In the following, existing methods to decompose the complex network interactions into smaller elementary units will be highlighted. However, it should be mentioned explicitly that we did not attempt to make a complete overview of all available methodologies. Rather, we have chosen to present two approaches so that we could illustrate the difference between high-level and low-level modeling.

There seems to be general agreement that suitable methodologies to represent the organizational complexity of regulatory networks should rely on hierarchical structures consisting of multiple modular elementary blocks. A module can generally be considered as a component or a subsystem of a larger system, and generally has some or all of the following properties (Csete and Doyle 2002): (1) identifiable interfaces; (2) can be modified and evolved somewhat independently; (3) facilitates simplified or abstract modeling; (4) maintains some identity when isolated or rearranged; and (5) derives additional identity from the rest of the system. However, what kind of modular structure should be selected for this purpose remains an open question and recent research has provided a number of attractive suggestions. The development of methodologies that allow a modular representation and simulation of large-scale dynamic systems is considered as one of the most important research topics in systems biology (Wolkenhauer et al. 2003). However, Csete and Doyle (2002) point out that the protocols (the rules that prescribe allowed interfaces between modules, permitting system functions that could not otherwise be achieved by isolated modules) are far more important to biological complexity than the modules.

7.4.3.1
Network Motifs

One way to deal with the organizational complexity of regulatory networks in microorganisms is the recognition of elementary modules, called network motifs. Such network motifs seem to be present in all kinds of complex networks (Milo et al. 2002) and can serve as elementary building blocks to reconstruct the connectivity in a regulatory network. For the prokaryote *E. coli*, Shen-Orr et al. (2002) extracted data from a database (Salgado et al. 2004) on direct transcriptional interactions between transcription factors and the operons they regulate, and augmented these data with a literature search, resulting in 141 transcription factors. A transcription factor, or a tran-

scriptional regulator, is a protein that binds to regulatory regions of the DNA and helps control gene expression. The LacI gene encoding for the *lac* repressor protein is an example of a transcription factor (Fig. 7.2).

Shen-Orr et al. (2002) found that a considerable part of the regulatory network of *E. coli* was composed of repeated appearances of only three elementary network motifs (Fig. 7.7). In the feedforward loop network motif, a general transcription factor X regulates the expression of a second specific transcription factor Y, whereas both transcription factors jointly regulate the expression of a structural gene Z. Coherent and incoherent feedforward loops are distinguished. In a coherent feedforward loop the direct effect of the general transcription factor on the expression of the structural gene has the same sign as the net indirect effect through the specific transcription factor. In an incoherent feedforward loop the direct and indirect effect have opposite signs. The coherent feedforward loop, the most frequently occurring feedforward loop motif in *E. coli* (Mangan and Alon 2003), was originally thought to be designed to be sensitive to persistent, rather than short and fast, transient inputs (Shen-Orr et al. 2002), that is, as a circuit that can reject transient activation signals from the general transcription factor (X in Fig. 7.7). A more detailed mathematical analysis of the feedforward loop motif (Mangan and Alon 2003) indicated that coherent feedforward loops act as a sign-sensitive delay element, meaning that the coherent feedforward loop responds rapidly to step changes in the general regulator concentration X in one direction (e.g., OFF to ON), and with a considerable delay to step changes in the general regulator concentration X in the other direction (e.g., ON to OFF). The practical functioning of this coherent feedforward loop regulatory mechanism was demonstrated with the L-arabinose (*ara*) utilization system in *E. coli* (Mangan et al. 2003). The influence of step changes in the global regulator cAMP on the expression of the L-arabinose system was investigated, and it was demonstrated that the ON response following a step increase of the cAMP concentration was indeed much slower compared to the OFF response (provoked with the addition of glucose in the growth medium). It was concluded that *E. coli* might have an advantage in a rapidly varying environment with this type of asymmetric response. When glucose is suddenly present (corresponding to a cAMP OFF step) it is utilized immediately. However, when glucose is depleted from the growth medium (corresponding to a cAMP ON step), the cell can save on the energy spent for protein production by only responding to persistent cAMP ON stimuli.

In a single input module network motif, a number of structural genes Z_1, Z_2, ..., Z_N are controlled by a single transcription factor X. The single input module can be compared to a single-input multiple-output (SIMO) block architecture in control (Doyle 2004), and is typically found in systems of genes that encode for a complete metabolic pathway. Shen-Orr et al. (2002) further indicate, based on mathematical analysis, that single input modules can show a detailed execution sequence of expression of the structural genes, resulting from differences in the activation thresholds of the different structural genes. In the dense overlapping regulons network motif there is a layer of overlapping interactions between a group of transcription factors X_1, X_2, ..., X_N and a group of structural genes Z_1, Z_2, ..., Z_N. In control terminology, dense overlapping regulons can be compared to a multiple-input multi-

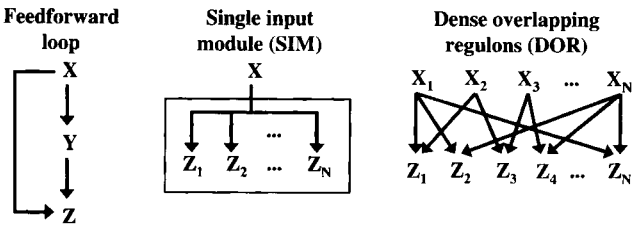

Figure 7.7 Elementary network motifs found in the *E. coli* transcriptional regulation network (Shen-Orr et al. 2002).

ple-output (MIMO) block architecture (Doyle 2004). The dense overlapping regulons seem to group operons that share a common biological function (Shen-Orr et al. 2002).

Shen-Orr et al. (2002) indeed illustrate that the motifs allow a representation of the *E. coli* transcriptional network in a compact, modular form. However, reality is more complex: the transcriptional network can be thought of as the "slow" part of the cellular regulation network (with a time scale of minutes). An additional layer of faster interactions, including interactions between proteins (often on a subsecond timescale) contributes to the full regulatory behavior and will probably introduce additional network motifs. This was confirmed by Yeger-Lotem et al. (2004), who extended the search algorithms for network motifs from genome-wide transcriptional regulatory network data to also include protein-protein interaction data (Yeger-Lotem and Margalit 2003) and applied the methodology to *S. cerevisiae*. For a more complete review of recently developed methods to search for network motifs in high-throughput data see Wei et al. (2004).

Again, eukaryotic organisms are more complex than prokaryotic organisms. In a study with *S. cerevisiae* on regulator-gene interactions, Lee et al. (2002) identified six frequently occurring network motifs, compared to only three for *E. coli*. Besides the

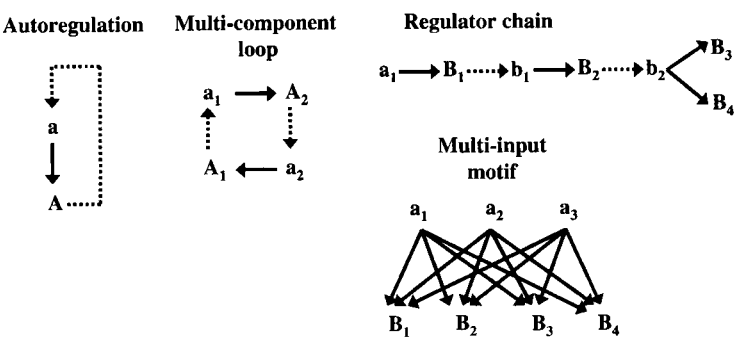

Figure 7.8 Network motifs in the regulatory network of *S. cerevisiae* (Lee et al. 2002). *Solid arrows* indicate binding of a regulator to a promoter. *Dotted arrows* indicate links between regulators and genes encoding for a regulator. *Capitals* indicate genes (e.g., A), whereas *normal font* (e.g., a) indicates proteins.

feedforward loop and the single input module (single-input motif in Lee et al. 2002) found for *E. coli* (Shen-Orr et al. 2002), autoregulation, multicomponent loop, regulator chain, and multiple-input motif network motifs were identified (Fig. 7.8). In an autoregulation motif, the regulator binds to the promoter region of its own gene. This mechanism was found for about 10% of the yeast genes encoding transcription factors.

7.4.3.2

A Signal-Oriented Modeling Approach to Modeling Regulatory Networks

Models are ideally suited to represent the knowledge about complex systems. Hierarchical modular modeling approaches are needed, since they lead to high model transparency at different levels of abstraction. Such model transparency is beneficial for engineers, but certainly also for biologists. Moreover, a modular structure contributes to allowing easy modification of the model by the model user. One can just modify one model module and subsequently plug the updated module into the overall model. A two-level hierarchical approach for modeling cell signaling mechanisms was proposed (Asthagiri and Lauffenburger 2000), where signaling modules would be defined as units whose underlying mechanisms can be studied first in isolation, and then integrated into a larger flow diagram of networked modules. Modules may be networked in a manner similar to the assembly of unit operations. Signaling outputs would be directed between different modules providing the interconnectivity and optimization; network performance can be assessed from a process systems perspective.

According to Lengeler et al. (1999), cellular control is hierarchical, meaning that there are global control networks that are superimposed on the specific control systems, and that can overrule the specific control systems. In prokaryotes, operons and regulons are at the lowest level of the control hierarchy as specific control systems. A regulon is a group of operons that are regulated by a common, but specific, regulator. The global control networks are coupled to complex signal transduction systems, which sense changes in the extracellular environment that require more drastic cellular adaptations than simply the expression or the repression of a few operons. Groups of operons and/or regulons controlled by such a global regulator are called a modulon. Finally, a stimulon represents groups of genes that will respond to the same stimulus. In the example of the *lac* operon, the repressor protein can be considered a local regulator, whereas cAMP can be considered a global regulator. In this cellular control hierarchy, functional units naturally appear by applying a set of three biological criteria (Kremling et al. 2000; Lengeler et al. 1999): (1) the presence of an enzymatic network with a common physiological task; (2) its control at the genetic level by a common regulatory network, corresponding to operons, regulons, and modulons; and (3) the coupling of this regulatory network to the environment through a signal transduction network.

The prokaryotic cellular control hierarchy of Lengeler et al. (1999) is applied in the signal-oriented first principles modeling approach of Kremling et al. (2000), where complex metabolic and regulatory networks are decomposed into physiologically meaningful smaller functional units. Each functional unit is built up by combining

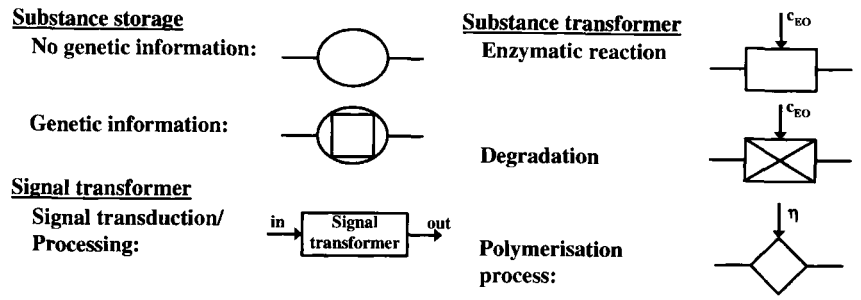

Figure 7.9 Elementary modeling objects for the signal-oriented modeling approach (Kremling et al. 2000).

a number of elementary modeling objects (Fig. 7.9). When building a mathematical model of a unit, each elementary model block in the representation of a regulatory network structure (see Fig. 7.10 for an example) gets a mathematical equation assigned to it. As a result, functional units in complex networks are represented as mathematical modeling objects. The method was first applied to the modeling of the *lac* operon (Kremling et al. 2001).

The development of the systems biology markup language (SBML), an XML-based language for representing models of systems of biochemical reactions, and for

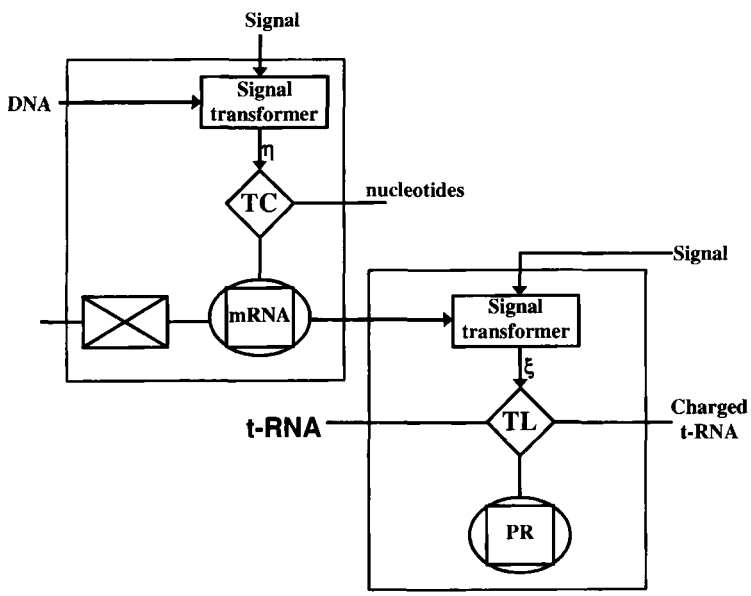

Figure 7.10 Representation of transcription (TC) and translation (TL) processes using the elementary modeling symbols of Kremling et al. (2000). η and ξ are the transcription and translation efficiency, respectively.

exchanging these models between simulation and model analysis tools (Hucka et al. 2003), is an important joint effort of a number of research teams. The use of SBML should facilitate the exchange of models between users of different software platforms. Indeed, instead of writing and validating model code for each software platform, a validated model in one software platform can be exported as an SBML model, which can subsequently be loaded by another software platform. The mere existence of the current SBML definition already contributes to modular representation of models, since a model for part of the processes in a cell can now be exchanged easily between researchers interested in cell modeling, and incorporated into other models and simulation software packages. Moreover, in the long term, it is envisaged that SBML will include the possibility of building large models by reusing a number of previously defined submodels (Finney and Hucka 2003). Clearly, this future SBML development is ideally suited for building large cell models with a modular structure.

7.4.3.3 Bridging the Gap between Network Motifs and the Signal-Oriented Modeling Approach

The network motifs (see Section 7.4.3.1) can be linked to the low-level modeling of regulatory networks (Doyle 2004), where the motifs represent modular components that recur across and within given organisms. One hierarchical modeling classification is proposed (Doyle 2004), where the top level corresponds to a network, which is comprised of interacting regulatory motifs. A module is at the lowest level in the hierarchy and describes transcriptional regulation.

It is important to realize here that the network motifs are extracted from system-wide (genome-wide) molecular interaction datasets by applying statistical methods. They provide a general indication of the connectivity and the structure of the regulatory network, however, without any indication on the exact kinetics of each interaction. Network motifs might point in the direction of a model structure that can be applied to describe the connectivity in part of the network, but there are many model candidates that can correspond to each motif (Mangan and Alon 2003). However, development of a detailed (low-level) simulation model necessitates experimental data that can be used to discriminate between model candidates and to estimate kinetic model parameters (Mangan et al. 2003). The signal-oriented modeling approach (see Section 7.4.3.2) on the other hand, is based on detailed experimental work aimed at generating dynamic data for the key metabolites participating in the interactions related to a very small part of the genome. The signal-oriented modeling approach includes detailed mechanistic information on the kinetics of each interaction between model states, resulting in a detailed nonlinear ODE-based model. Both approaches consider the regulatory network at a different abstraction level (Ideker and Lauffenburger 2003). The network motifs can be considered as high-level pathway models, whereas the signal-oriented modeling approach belongs to a class of extremely detailed low-level models.

High-level and low-level models are of course connected. In fact, there are relatively few well-documented systems where detailed low-level modeling can be applied (de Jong 2002; Ideker and Lauffenburger 2003), whereas high-level informa-

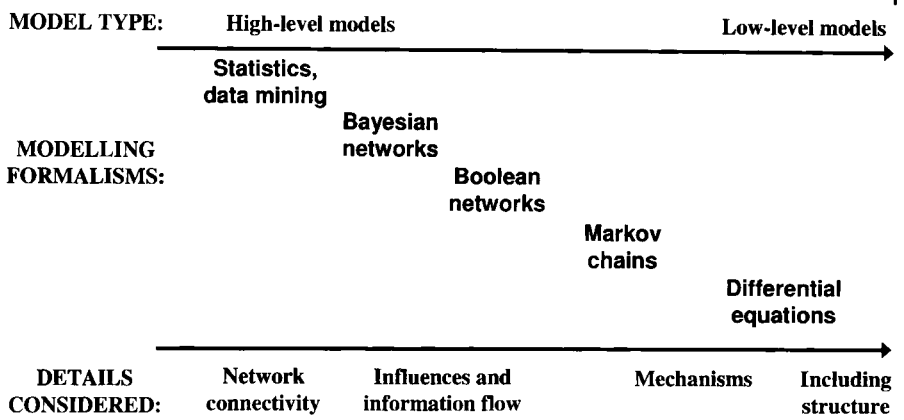

Figure 7.11 Illustration of the use of different modeling formalisms to move from abstracted high-level models to specific low-level models (Ideker and Lauffenburger 2003).

tion on protein-DNA interactions and protein-protein interactions is available for an increasing number of microorganisms. Bridging the gap between the high-level and the low-level models or, in other words, increasing the throughput with which interesting and important biological problems can be brought from the high-level to the low-level modeling state is a major challenge for systems biology (Ideker and Lauffenburger 2003). Bridging the gap between high-level and low-level modeling might necessitate the sequential use of a hierarchy of modeling formalisms (see Fig. 7.11), where each formalism corresponds to an adequate description of a certain level of abstraction of the regulatory network (de Jong 2002; Ideker and Lauffenburger 2003). An example of a procedure to evolve from high-level to low-level models is provided in Ronen et al. (2002).

7.5 Towards Understanding the Complexity of Microbial Systems

The models presented above represent selected but important aspects of microbial regulatory function as they can be expressed using dynamic systems concepts and theories. These theories have proven to be very powerful in dealing with analysis and design problems in control engineering and it is therefore natural to expect that similar successes can be obtained when they are applied to microbial systems. However, this expectation is based on the assumption that the complexity of engineering systems and microbial systems are comparable and measurable on the same scale. In that respect, Csete and Doyle (2002) indeed concluded that the complexity of engineering systems, taking a Boeing 777 with its more than 150,000 subsystem modules as an example, is almost comparable to the complexity of biological systems. The modeling of microbial systems should therefore not represent fundamental new challenges, except maybe from the problem that the number of ODEs required to

describe their behavior will be significantly higher than for engineering systems, and that more nonlinear phenomena might be involved. This assumption however ignores basic interpretation problems in model building. These interpretation problems are tractable and their importance therefore remains unrecognized in most systems engineering problems, which becomes a major problem in modeling microbial systems.

7.5.1 The Interpretation Problem

The interpretation problem originates in the multifunctional nature of microbial systems. Where a subsystem in most engineering systems only serves one or a few functions, it may serve many interdependent functions in microbial systems. A function is not an inherent property of a subsystem, but is defined relative to other subsystems and by the purpose of the system of which it is a component. A protein may thus serve at least three different functions. It can serve as a substance (material or product, e.g., in protein degradation reactions) in a metabolic process, it can serve as an enzyme promoting another reaction, and it can act on the DNA for promoting or blocking the expression of genetic information (transcription factor). The complexity of microbial systems originates in this unique ability of proteins to enter into a multitude of functional relations.

The identification of functions requires knowledge of how a subsystem contributes to the whole. This knowledge about the functional organization of the system is a prerequisite for the formulation of a set of ODEs describing the system, because it determines the level of abstraction adopted and the system features to be included in the equations (Lind 2004b). As mentioned before, a distinction must be made between organizational (functional) complexity and behavioral complexity (Doyle 2004). Behavioral complexity can be expressed by ODEs, but we need other concepts to model the organizational complexity. The purpose of a model of the organizational complexity is to define, in formalized language, the functional relations between subsystems and the biological system as a whole. Such a model comprises an abstract qualitative representation which can be used to communicate the understanding obtained for the biological system. Often, informal sketches or graphics are used to communicate functional knowledge. However, more formal concepts are required in order to ensure clear semantics and consistency of the models. A formalized model of the functional organization is therefore a complement to, and not merely a mediocre or less accurate version of, an ODE model.

In the following we will discuss the interpretation problem in more detail in order to further motivate the application of functional concepts in the modeling of microbial systems, and to introduce and explain the basic concepts. We will subsequently present formalized generic concepts to model control (regulatory) functions. A key advantage of generic concepts is that they can be applied on an arbitrary level of abstraction and thus facilitate the modeling of complex control functions in microbial systems. Another advantage of the formalized concepts is their completeness.

7.5.1.1

Frameworks of Interpretation

In order to develop a deeper understanding of the problems in modeling complex systems, it is important to realize that modeling activity, in addition to mathematical aspects, involves a process of interpretation where the modeler makes sense of the events and phenomena in the problem under investigation. The interpretation problem is fundamental to humanities and social sciences but has thus far not been considered particularly relevant for the natural or the engineering sciences because interpretation is often considered in conflict with objectivity. However, when considering complex systems we must apply perspectives or make abstractions in order to handle the modeling problem at hand, and thus interpretations are unavoidable. However, interpretation of a phenomenon is always relative to a conceptual framework. According to Goffman (1974) we can distinguish between two so-called primary frameworks of interpretation. A framework of interpretation serves as a frame of reference and is seen as rendering what would otherwise be a meaningless and chaotic situation into something that is meaningful and with structure. The two primary frameworks are called the natural and the social frameworks, respectively, and are defined as follows:

- ...Natural frameworks identify occurrences seen as undirected, unoriented, unanimated, unguided, "purely physical." Such unguided events are ones understood to be due totally, from start to finish, to "natural" determinants. It is seen that no willful agency causally and intentionally interferes and that no actor continuously guides the outcome. Success or failure with regard to these events is not imaginable...
- ...Social frameworks, on the other hand, provide background understanding for events that incorporate the will, aim, and controlling effort of an intelligence, a live agency, the chief one being the human being. What the agent... does can be described as "guided doings." These doings subject the doer to "standards," to social appraisal of his action based on its... efficiency, economy, safety, etc.

Events and occurrences in engineering systems can clearly be interpreted within a natural framework. Engineering systems are, however, designed to exploit physical phenomena such that human purposes and aims can be fulfilled and therefore be understood within a social framework of interpretation. The natural and social frameworks are both broad categories. The natural frameworks include, for example, physics and chemistry, and similarly the social frameworks include several subcategories. Note that concepts of function and purpose belong to the social framework of interpretation.

Habermas (1989) compared different approaches to functionalism within social science. His analysis identifies three approaches. Two of these are in Goffman's sense of social frameworks for understanding the plan or *intention* of a system or an activity:

- We can understand the plan teleologically, in which case it is based on the artisan model of *instrumental action* through which an end is reached through appropriate means.

- We can also conceive the plan dialectically, in which case it is based on the dramaturgic model of *communicative action*, in which an author makes an experience transparent through the role playing of actors.

Habermas' analysis also indicates that we could define an additional framework of interpretation that could be called biological and was not considered by Goffman. This framework is characterized as follows:

- We can also use a model borrowed from biology. According to this model, systems can be understood as organized unities that under changing circumstances maintain themselves in a specific state through self-regulation and reproduction.

Accordingly, four different frameworks may be applied in the interpretation of an event or a phenomenon:

1. the natural framework,
2. the framework of instrumental action,
3. the framework of communicative action,
4. the biological framework.

Note that the four frameworks of interpretation should be seen as different ways to assign meaning to an observed event or phenomenon. Each framework defines a context for understanding the system according to a particular point of view. As mentioned below, the frameworks are often combined in the interpretation of complex systems.

7.5.1.2

Interpretation of Complex Biotechnological Systems

Interpreting complex biological systems will often require the application of more than one of the frameworks. For example, the behavior of a "dancing" bee can be described within a communicative framework by its communicative function. But it may also be described within a biological framework (at a higher abstraction level) by its function for the survival of the species.

In order to understand the organization of bio(techno)logical systems it is necessary to apply the instrumental action, the biological, and possibly also the communicative framework, for example, when considering quorum sensing where a population of microorganisms are informed about a certain event. Since there is no blueprint (i.e., no designer of the cell), its regulatory function must be explained in evolutionary terms, where it must be seen as emerging from a selection process, leading to a competitive advantage for the cell. When the behavior of a cell population in a bioreactor is controlled from the outside it must be seen as an object of instrumental action.

A major challenge in the interpretation of complex microbial systems is therefore to understand how to combine different interpretations of the same subsystem or how to combine the interpretations of subsystems that belong to different frameworks. As an example, the central dogma, which includes the transcription, translation, and expression of information in the DNA and RNA (communicative action), should be combined with the metabolic reactions (biological framework) and the control of cell population in a reactor (instrumental action).

7.5.2 Functional Analysis

The instrumental, communicative and biological frameworks support functional explanations, i.e., answers to a "why" question having the general form "in order to" (Achinstein 1983). The explanations are different, however. Within the framework of instrumental action the explanation of an event or object relates it to the intention of the actor. Within the communicative framework an occurrence is ascribed a communicative function (e.g., a message), and the occurrence is explained by its effect or role in an act of communication. Within the biological framework, observed events are seen as contributing to survival and adaptation of the system to its environment, e.g., an organ is ascribed a function in view of its contribution to the survival of the organism of which it is a part.

Functional explanations express the reasons (not the causes) for the occurrence of an event and are therefore an integral part of means-end analysis. Means-end analysis is an old topic in philosophical logic with ancient roots in works of Aristotle but has more recently been developed within artificial intelligence (Simon 1981) and cognitive science research (Bratman 1987). Means-end analysis is the basis for multilevel flow modeling (MFM), which is a methodology for modeling complex industrial systems (Lind 1994) by integrating different frameworks. MFM is not intended to generate detailed dynamic models. Instead, it allows one to represent systems at different levels of abstraction and as such supports the building of detailed dynamic models in the conceptual phase (Gofuku and Lind 1994). MFM has an inherent logic that allows formal analysis of the organizational complexity, and is therefore also attractive for application to regulatory networks in microorganisms.

7.5.2.1 Formalization of the Concept of Function

One of the key research problems in means-end analysis is the formalization of the concept of function. Formalization is necessary in order to be able to build models of means-end relations in systems that are logically consistent and in order to be able to use the models for computational purposes. The formalization involves the solution of two problems. The first problem is to define a logic that can be used to make inferences about means-end relations. The other problem is to identify a basic set of so-called elementary functions, which can be used as generic modeling concepts. The question of means-end logic was addressed by Larsson (1996) for applications of MFM in fault diagnosis and by Larsen (1993) for problems of start-up planning. We will not consider these logics here, but instead focus on the problem of elementary functions, which is of particular interest in the present context of modeling regulatory networks in microbial systems.

7.5.2.2 Elementary Functions

Before we address the problem of elementary functions it should be mentioned that the concept of function actually has two core meanings. One meaning is related to

the concept of action and the other is related to the concept of role. The first meaning is used when we define a function by what a system or actor is doing, and the second when we refer to the entities involved in the action. Consider the following example: the function of the pump is to "move the water." Here the function of the pump in the first meaning is the intended result of the pump's action, whereas the function in the second meaning is the role played by the pump in its interaction with the water, i.e., that it is the agent of the action. The function (role) of the water is similarly the object of action (the patient). By distinguishing between actions and roles we are accordingly able to define functions of systems more precisely. This clarification of the concept of function relies on the linguistic analyses of verb semantics (see, e.g., Lyons 1977). The solution to the problem of elementary functions depends therefore on whether we mean function in the sense of action or role.

Elementary Roles and Embedding Relations

Elementary roles (such as agent, patient, instrument, etc.) have been defined by linguists, but some disagreements of minor importance for the discussion in this chapter still remain. Role relations are important for understanding system complexity. Thus, the same object or system could have several roles at the same time or different roles at different times. In this way system processes can be embedded into each other. With the roles mentioned above we have the following possibilities for role shifting:

- An item may be the patient (product) of an action (transformation) and then become the agent (e.g., catalyst) of another action.
- An item may be the patient (product) of an action (transformation) and then become the patient (material) of another transformation.
- An item may be the agent of an action (transformation) and then become the patient (material) of another action (transformation).

An item may participate in this way in several processes at the same time provided it can play the roles simultaneously.

Elementary Actions

The possibility of defining a set of elementary action types has been addressed by Von Wright (1963), and has been explored further for application in means-end analysis of complex dynamic systems by Lind (1994, 2002, 2004a,b).

The elementary action types are actually derived from a set of corresponding elementary change types. The idea is that an action results in a change of state. Conceptually, the change caused by the action would not appear if the action was not done. The definition of an action therefore contains a reference to a hypothetical situation that is not realized because the action was done. Now, by defining a change as a transition between two states, we can define four so-called elementary changes shown in Table 7.1. Each change in the table is defined by both a linguistic description and a logic formula, which is composed of a proposition p representing the world state, a temporal operator T (Then) and one of the four change verbs "happens," "remains," "disappears," and "remains absent." In this way the formula ~pTp (~p Then p) is a logic representation of the change described by "p happens."

We shall not go into details about the logic definitions here. However, it is notable that the list of elementary changes is a logically complete list so that all changes in the world can be defined provided we define the state in question by a proposition p. We will actually also need to combine elementary changes. Each elementary change has a corresponding elementary action type as indicated in Table 7.1. The action formula contains the temporal operator T (Then) and an additional operator I (Instead) used to indicate the hypothetical state. The logical formula ~pTpI~p represents the action "produce p." It is seen that if the action was not done the state of the world would be ~p instead of p. The list of elementary actions can actually be expanded with four additional action types not shown in the table. These actions would correspond to actions where the agent refrains from intervening with the world. The total number of elementary actions is accordingly eight.

The four (eight) elementary action types define a generic set of actions that have the great advantage of being defined on a logical basis. This means that the completeness of the action types is ensured. The elementary action types (Table 7.1) therefore form a very attractive basis for the definition concepts for modeling system functions. Note that the action types are generic because they are defined without specifying the proposition p. The action types can therefore be specialized to specific problem domains by proper specification of p.

Another remarkable aspect of the action types is that they have a direct correspondence with the types of control functions used in control engineering. The correspondence is shown in Table 7.2. The completeness of the action types implies

Table 7.1 The elementary action types of Von Wright (1963). p denotes a state, T denotes "Then," and I denotes "Instead."

Types of elementary change		Types of elementary action	
Description	**Formula**	**Description**	**Formula**
p Happens	~pTp	Produce p	~pTpI~p
p Remains	pTp	Maintain p	pTpI~p
p Disappears	pT~p	Destroy p	pT~pIp
p Remains absent	~pT~p	Suppress p	~pT~pIp

Table 7.2 Correspondence between the elementary action types and control actions.

Elementary action	Control action
Produce	Steer
Maintain	Regulate
Destroy	Trip
Suppress	Interlock

accordingly that any control function can be described by proper combinations of these four functions. Note that the descriptions of the controls do not represent the implementation of the controls. The descriptions only define the control purpose.

7.5.3
A Language for Modeling Functions of Microbial Systems

The elementary action types can be used as a systematic basis for the derivation of modeling concepts for a particular problem domain. As an example, MFM (Lind 1994) can be mentioned. A basic set of modeling concepts adapted to the domain of microbial systems is proposed in Fig. 7.12. Each of the actions shown can be defined formally as specific interpretations of the elementary action types or as compositions of two elementary actions (Lind 2004a). We will not provide all details here. Instead, we prefer to demonstrate with examples (see Section 7.5.4) how the modeling concepts can be used to represent the functional organization of microbial systems.

The MFM modeling language (Fig. 7.12) comprises three types of concepts. It contains a set of concepts for representing action (functions), concepts to represent goal states, and a set of concepts for representing means-end relations between actions, sets of actions, and goals. It should be stressed that MFM represents the actions or transformations done to material, energy, or information flows (fluxes) in a complex system. However, it does not represent the flows or fluxes themselves. This may seem disturbing, but the abstractions provided by MFM describe how the systems of

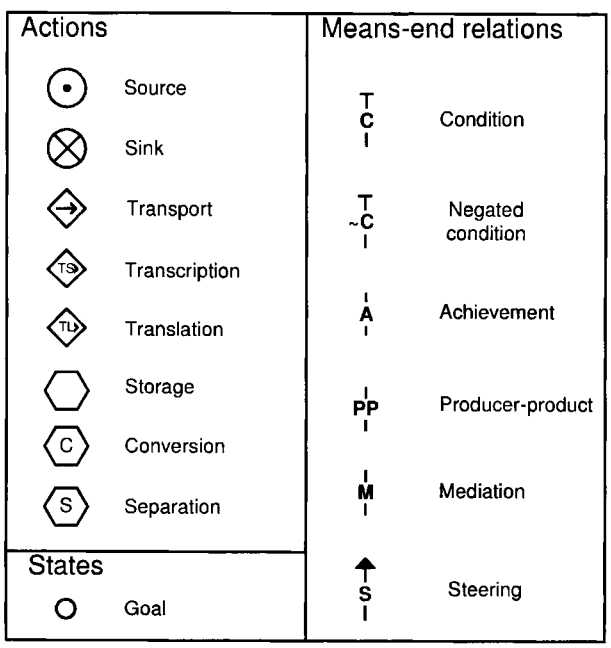

Figure 7.12 MFM modeling concepts adapted to the microbial domain.

transformation of the various substances (energy, material or information) are organized into means-end networks. The levels of abstraction can therefore not be defined without implicitly thinking in concepts of flows or fluxes.

It should be noted that MFM also includes concepts to model part-whole relations, as well as concepts to model relations between functions and physical structures, but these relations are not used here (see Lind (1999) for more detail on these relations). The concepts in Fig. 7.12 will be explained briefly in the following. A deeper understanding can be obtained by studying the application examples presented in Section 7.5.4.

7.5.3.1

The Means-End Relations

Goals and functions can be connected by conditions, achievements, producer-product, mediation, and steering relations. Each of the relations will be discussed separately.

- The condition: a goal can define a condition that is necessary for the enablement of a function. This conditioning is expressed by a relation (C) between the objective and the function.
- The negated condition: a goal can define a condition that is necessary for the disablement of a function. This negated conditioning is expressed by a relation (~C) between the goal and the function.
- The achieve relation: goals are achieved by system functions. This relation is defined by the achieve relation (A). The (A) relation is a means-end relation where the goal is the end and the function or systems of functions are the means.
- The producer-product relation: functions can be related through a means-end relation called a producer-product (PP) relation. This relation is used when the interactions between a set of functions (an activity or process) result in a product that again serves another function in the system.
- The mediation relation: functions can also be related through a mediate (M) relation. This relation is used when a system has the role of being an intermediate between a system and another system that serves as an object of action. An example of such mediation could be the transportation of energy by the pumping of water. Here, there is a mediate relation between the pumping function and the transportation of energy.
- The steering relation: functions can also be related through a steering relation (S). This relation is used when the interactions between a set of functions (an activity or process modeled by a so-called flow structure) determine the state of another function.
- The connection relation: MFM also includes a so-called connection relation, which is not really a means-end relation, and is also not shown in Fig. 7.12. A connection is used to relate the functions (actions) into functional (flow) structures. A connection is symbolically represented as a line linking two functions and represents a contextual linkage of two functions. This means that they relate to the same goal perspective or that they share substances (change properties that belong to the

same substance). The connection relation can be further specialized by taking into account causal directions in the interaction between functions.

7.5.3.2 The Flow Functions

MFM also defines a set of so-called flow functions (the actions), which are used in building models together with the relations described above. The symbols used for functions are shown in Fig. 7.12. Each of the functions represents an action on a substance that may be mass, energy, or momentum. The different substances are indicated by symbology. In action terms a source provides a substance, i.e., makes it available. Similarly a sink removes substances. A transport changes the spatial location of a substance, a conversion changes the composition of a material flow, and a separation separates a flow of material into its components. It is clear that some of the functions both apply to modeling material and energy flows. However, there are also functions that are dedicated to modeling transformation of, e.g., information flows. Two such functions are defined here for modeling the processing of genetic information in microbial networks, namely the transcription, and the translation functions.

7.5.4 Modeling Examples

In the following, the application of MFM to regulatory networks will be illustrated with examples that illustrate the capabilities of the methodology in decomposing the regulatory network into its elementary modules. Again, the *lac* operon will be used as an example since its regulation has already been described in detail (see Section 7.3.1).

7.5.4.1 Induction of the *lac* Operon

Figure 7.13 represents the induction mechanism of the *lac* operon (see also Fig. 7.2) using the symbols introduced in Fig. 7.12. The logic of the model can be explained by starting with the bottom part of Fig. 7.13, which represents the uptake of lactose and the conversion of lactose to allolactose, the inducer of the *lac* operon. The model shows in box I that the transport of lactose over the cell wall is carried out ("mediated") by the β-galactoside permease, which is produced as the result of the translation of the mRNA (top part of the model). The subsequent conversion of lactose to allolactose is catalyzed ("enabled") by the β-galactosidase. The allolactose is afterwards assumed to be distributed in the cellular cytoplasm by diffusion ("transport function"). The functions described comprise the means for achieving (A in Fig. 7.13) the conditions for allolactose to be present in the cell.

By changing perspective and thus moving upwards in the model we now consider the set of functions in boxes II and III in Fig. 7.13, describing how the state of the repressor (R) is influenced by the various functions of the microbial system. The

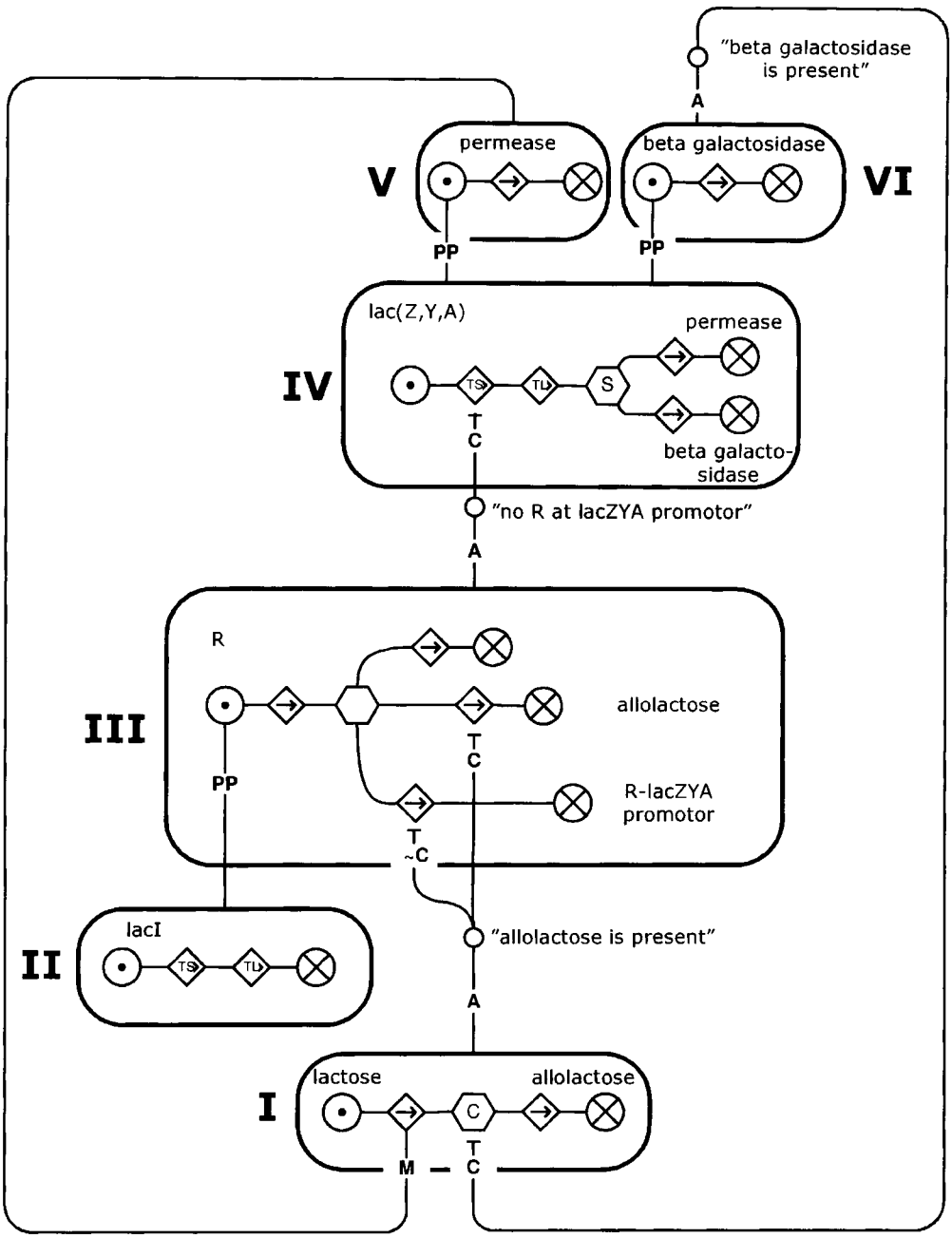

Figure 7.13. MFM representation of the *lac* operon induction mechanism (see also Fig. 7.2).

principles to describe the production of repressor protein in detail (box II) are similar to the flow model in box IV, representing the transcription and translation of the structural genes of the *lac* operon. The lacI gene is transcribed and the resulting mRNA is translated into the repressor protein. Box II thus depicts the flow model for the transcription and translation part of the central dogma (Fig. 7.1). The function in box II provides the presence of R. In other words, it provides the means for producing R. Box II represents the producer, whereas the fact that R is becoming present in box III is the product.

The repressor protein can follow three possible paths, represented in box III: (1) it can follow the bottom path where it binds with the operator of the *lac* operon, and will thus block transcription; (2) it can follow the middle path, where it binds with the inducer allolactose and undergoes a conformational change; and (3) it can follow the upper path, where the repressor protein is degraded (modeled as a "Sink"). The presence of allolactose conditions two of the transports in box III. When allolactose is present the repressor protein binds to the allolactose, whereas in the absence of allolactose the repressor protein binds to the *lac* operon operator. In the first case, the functions described in box III comprise the means for achieving a de-repressed *lac* operon and thus transcription of the *lac* operon structural genes occurs. Again, we can now move upwards in the model. The transcription and translation of the *lac* operon is represented in box IV. The *lac* operon structural genes are transcribed into a polycistronic mRNA, and during the subsequent translation process the mRNA results in the different proteins (β-galactoside permease and β-galactosidase). Conversion of the polycistronic mRNA to several proteins is modeled by combining the translation process with the subsequent separation function in Fig. 7.13. It is important to mention that the production of the third protein encoded in the *lac* operon, β-galactoside transacetylase, is not shown in Fig. 7.13, since that enzyme is assumed not to play any significant role in the induction mechanism. The diffusion of both proteins into the cytoplasm is finally illustrated in box V and VI. Note that β-galactoside permease is a mediator (transport enzyme), whereas the role of β-galactosidase is an enabler (a catalyst). Now the loops to box I are closed.

7.5.5
Inducer Exclusion and Carbon Catabolite Repression

Figure 7.14 is an extension of Fig. 7.13, including the regulatory effects of glucose on the *lac* operon, catabolite repression, and inducer exclusion (see also Fig. 7.4). Again, we start at the bottom of the figure to explain the modeled flows. The PTS system (box VII and VIII) is responsible for the uptake and the phosphorylation of glucose, resulting in G6P (for an explanation of the abbreviations, see Section 7.3.1.2). Note that both the energy level (box VIII: transfer of a phosphate group from PEP to G6P) and the component level (box VII) are represented, indicating the strength of MFM to represent a system at different abstraction levels. In the simplified schematic representation of the PTS for this purpose, it is assumed that a phosphate group is transferred from the phosphorylated PTS enzyme P-IIA^{Glc} to glucose during uptake

of glucose. The resulting PTS enzyme IIA^{Glc} again receives a phosphate group from PEP, i.e., it is involved in a loop. In fact, phosphorylation of IIA^{Glc} takes place through a series of conversion steps, involving several of these loops. These conversions are not shown in detail since they do not contribute to the regulatory mechanisms, and are instead represented by one extra catalysis step between PEP and the catalysis of IIA^{Glc} to $PIIA^{Glc}$ conversion. For the detailed mechanism of the PTS system, Lengeler et al. (1999) and Postma et al. (1993) should be consulted.

With respect to the glucose effects on the regulation of the *lac* operon, the formation of IIA^{Glc} and $PIIA^{Glc}$ has been described in box VII in all the detail needed. Again, we can change perspective and consider the functions where the presence of IIA^{Glc} or $PIIA^{Glc}$ will have an influence on the state of the system. The $P\text{-}IIA^{Glc}$, the species that will be abundant in the absence of extracellular glucose, activates the conversion of ATP to cAMP by the adenylate cyclase (AC) in box X. Similarly to β-galactosidase, AC is modeled as an enabler (catalyst), but the transcription and translation processes leading to AC formation are not presented since these mechanisms were considered as not contributing substantially to the regulation of the *lac* operon. We have thus assumed that the enzyme AC is present ("Source" in box IX) and undergoes diffusion in the cytoplasm, and subsequently catalyzes the ATP to cAMP conversion in box X. The cAMP forms a complex with CRP, and this complex subsequently boosts the transcription of the structural genes of the *lac* operon (box IV), thereby releasing the catabolite repression of the *lac* operon. Thus, the catabolite repression mechanism has been described.

Finally, in the presence of glucose IIA^{Glc} will be abundant and will inhibit the uptake of lactose by the permease. Inducer exclusion is thus modeled as a negated condition, i.e., the absence of IIA^{Glc} will be the condition to reach full activity of the permease enzyme.

7.6 Discussion and Conclusions

To reveal and understand regulatory network mechanisms constitutes one of the most significant scientific challenges in the post-genomic era. Many researchers are devoted to uncovering these networks and utilize many different (novel) techniques to enable gene annotation, transcription factor identification, as well as characterization and representation of protein-DNA and protein-protein interactions. Behind these attempts remains a fundamental question of how to combine data from these many different sources for revealing the function of regulatory networks in microorganisms. The intuitive approach for a systems engineer is to generate a model of the system under study. However it is not always clear which modeling methods and what abstractions to apply. Therefore, this chapter first highlighted fundamental modeling problems in describing regulatory networks in microorganisms, and subsequently illustrated the potential of means-end analysis (also called functional modeling) to represent the functionality of complex regulatory networks at different levels of abstraction.

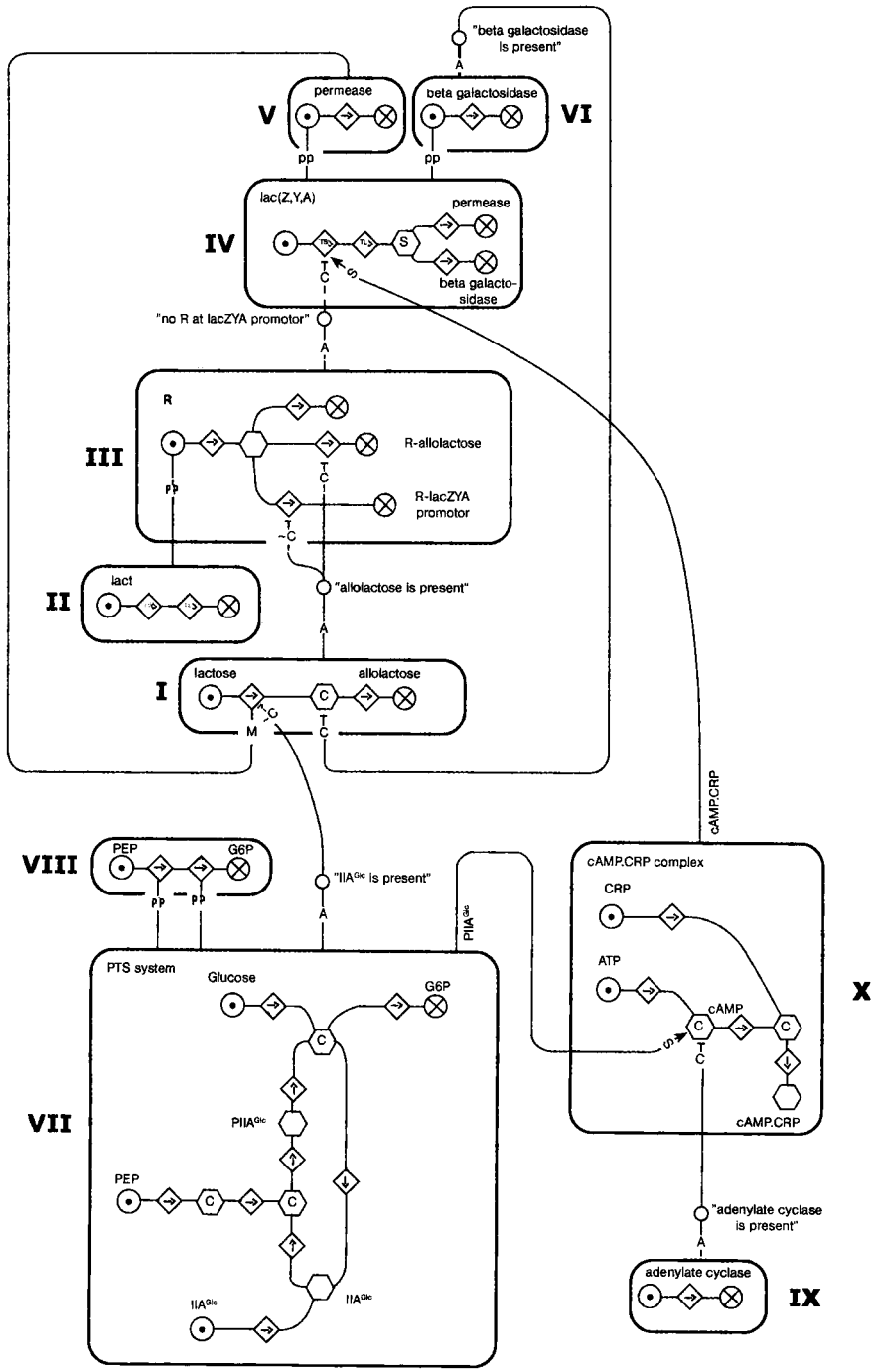

Figure 7.14 MFM representation of the glucose effects on the *lac* operon (see also Fig. 7.4).

As mentioned above, microbial function is carefully controlled through an intricate network of proteins and other signaling molecules, which was demonstrated in a couple of examples mainly drawn form the *lac* operon. From a production process perspective, it is definitely an interesting question how systems engineers should couple the detailed description and understanding of the functioning of microorganisms (the microscale) to the higher process level descriptions (the macroscale). The proposed MFM-modeling-based methodology is especially suitable to support this coupling of the microbial regulatory functions and the higher level process and production control functions, since the same set of symbols might be used to represent the flows at the process as well as at the detailed (micro) level. This ability clearly distinguishes the proposed methodology from existing methods to represent regulatory network mechanisms: The network motifs (Lee et al. 2002; Shen-Orr et al. 2002) only represent the connectivity between system states, and do not allow a representation of the connectivity with higher process levels. The elementary modeling objects developed by Kremling et al. (2000) allow the representation of the regulatory networks at a very detailed level, and would probably also be suited to connect the regulatory network with higher process level functions. However, the representation with elementary modeling objects does not contain the degree of information available in the MFM models of regulatory networks, where the actions and means-end relation symbols (see Fig. 7.12) provide a high degree of transparency on the way system states interact with each other.

Thus, a first conclusion of this chapter is that the proposed representation of regulatory network systems, based on MFM, is ideally suited for supporting systems engineers in detailed model building in bioreactor systems. The applied modeling concept has been demonstrated to enable modeling the changes in qualitative behavior of microorganisms, and is as such able to summarize available process knowledge. If quantitative dynamic models were desired, then these could be developed within each region of qualitative behavior using the logic in the MFM model as a support in the generation of detailed mathematical descriptions.

By providing a methodology to represent the regulatory networks at several abstraction levels, this chapter is of relevance to process systems engineering for several reasons. One reason is that microorganisms constitute relatively simple biological systems, and the study and understanding of these relatively simple biological systems may, with suitable extensions, enable better understanding of multicellular biological systems. Furthermore, microbial systems are increasingly used, often following genetic manipulation, to produce relatively complex organic molecules in an energy-efficient manner. Understanding the details of intracellular regulatory networks is a prerequisite for efficient coupling of microbial regulatory functions with higher-level process and production control functions. In other words, the final result of applying process engineering might be improved considerably when process-relevant parts of the intracellular regulatory networks are better understood, and the methodology proposed in this chapter can significantly contribute to represent and subsequently develop that understanding. Finally, and maybe most importantly, applying the MFM modeling method to regulatory networks in microorganisms almost naturally leads to modularizing the network into elemental

building blocks that are understandable for systems engineers as well as biologists. Thus, the proposed modeling method could contribute substantially by providing a formalism that allows biologists and systems engineers to communicate efficiently about regulatory network functions.

To reach a basic level of understanding of the function of these autonomous plants, which is what microorganisms are from an industrial point of view, a systematic description of fundamental regulatory and metabolic functions is proposed in this chapter: The proposed description, which is based on MFM (Lind 1994), might eventually lead to combining the basic understanding of microbial behavior with the semiotics of control. This combination leads to simple schematics for describing fundamental roles of molecules in cells, and their reactions for control and coordination of microbial behavior. In this respect, the flexibility of the MFM modeling formalism is especially noteworthy. In fact, in the *lac* operon example the lacI gene is expressed constitutively. This means that transcription and translation of the gene to the resulting repressor protein does not necessarily have to be modeled in detail in box II of Fig. 7.13. Indeed, since we assume that no regulatory mechanisms are involved in this process, the presence of the repressor protein could have been modeled by only including a "source" for the repressor protein in box III, thereby omitting box II from the model. Thus, MFM models are flexible and can be extended easily. This is, for example, also illustrated by the straightforward extension of the *lac* operon induction mechanism (Fig. 7.13) to also include the glucose effects (Fig. 7.14). Clearly, when further building on existing MFM models, the "source" symbols are obvious candidates for extending these models, aimed at including more detail. An MFM representation of the DNA replication process could for example be coupled to the presence of the lacI and lac(Z,Y,A) genes in Figs. 7.13 and 7.14.

One could also argue that this chapter has mainly addressed prokaryotic organisms, and that this will limit the applicability of the MFM modeling methodology severely. We claim that application of the proposed methodology to other organisms, for example, eukaryotic unicellular organisms, should be no problem except for obtaining the necessary fundamental knowledge. Again, an example will illustrate this. In eukaryotes, the mRNA might undergo several processing steps before it is transported out of the nucleus, where the ribosomes will finally take care of the translation of the mRNA to a protein. Applying the proposed MFM methodology to such eukaryotic systems, box IV in Fig. 7.13 (describing transcription and translation of the prokaryotic *lac* operon) would definitely need several extensions to allow the detailed representation of similar eukaryotic mechanisms. This extension could be obtained by splitting up the prokaryotic version of box IV (Fig. 7.13) into several boxes for the eukaryotic case, where each box represents a separate perspective. One box for transcription and its regulation, one box for the subsequent eukaryotic mRNA processing steps, one box for the transport of the processed mRNA out of the nucleus, and finally one box for the translation process. However, it is also evident from the examples in Figs. 7.13 and 7.14 that these extensions can be made in a straightforward way by using the symbols and conventions provided in Fig. 7.12. Thus, a second conclusion of this chapter is that MFM modeling is highly flexible, allowing systems engineers to easily extend existing models (e.g., by adding flow

models related to the means for production of some "sources" in an existing model), and to transfer MFM modeling concepts to other (more complex) biological systems.

Finally, it should also be pointed out that the proposed modeling methodology is not only useful in reverse engineering, where it could be applied to represent hypotheses on the operation of complex regulatory network systems. In our opinion MFM models could also be used in forward engineering to design regulatory network building blocks such as the repressilator (Elowitz and Leibler 2000) before developing a detailed mathematical description.

References

1 *Achinstein P.* (1983) The nature of explanation. *Oxford University Press*, Oxford, UK.

2 *Alur R. et al.* (2002) Modeling and analyzing biomolecular networks. *Comput. Sci. Eng.* 4 (Jan/Feb), 20–31.

3 *Asthagiri A. R., Lauffenburger D. A.* (2000) Bioengineering models of cell signaling. *Ann. Rev. Biomed. Eng.* 2, 31–53.

4 *Baneyx F.* (1999) Recombinant protein expression in *Escherichia coli*. *Curr. Opin. Biotechnol.* 10, 411–421.

5 *Bratman M. E.* (1987) Intention, plans and practical reason. *Harvard University Press*, Cambridge, MA.

6 *Chassagnole C., Noisommit-Rizzi N., Schmid J. W., Mauch K., Reuss M.* (2002) Dynamic modeling of the central carbon metabolism of *Escherichia coli*. *Biotechnol. Bioeng.* 79, 53–73.

7 *Cheetham P. S. J.* (2004) Bioprocesses for the manufacture of ingredients for foods and cosmetics. *Adv. Biochem. Engin. Biotechnol.* 86, 83–158.

8 *Csete M. E., Doyle J. C.* (2002) Reverse engineering of biological complexity. *Science* 295, 1664–1669.

9 *de Jong H.* (2002) Modeling and simulation of genetic regulatory systems: a literature review. *J. Comp. Biol.* 9, 67–103.

10 *Doyle F. J. III* (2004) A systems approach to modeling and analyzing biological regulation. Proceedings of the International Symposium on Advanced Control of Chemical Processes (ADCHEM2003), 11–14 January 2004, Hong Kong.

11 *Downward J.* (2001) The ins and outs of signalling. *Nature* 411, 759–762.

12 *Eisenberg D., Marcotte E. M., Xenarios I., Yeates T. O.* (2000) Protein function in the post-genomic era. *Nature* 405, 823–826.

13 *Elowitz M. B., Leibler S.* (2000) A synthetic oscillatory network of transcriptional regulators. *Nature* 403, 335–338.

14 *Ferber D.* (2004) Microbes made to order. *Science* 303, 158–161.

15 *Finney A., Hucka M.* (2003) Systems biology markup language (SBML) level 2: structures and facilities for model definitions. http: www. sbml. org/.

16 *Goffman E.* (1974) Frame analysis. *Penguin Books*, London.

17 *Gofuku A., Lind M.* (1994) Combining multilevel flow modeling and hybrid phenomena theory for efficient design of engineering systems. Proc. 2nd IFAC Workshop on Computer Software Structures Integrating AI/KBS Systems in Process Control.

18 *Habermas J.* (1989) On the logic of the social sciences. *MIT Press*, Cambridge, MA.

19 *Hasty J., McMillen D., Isaacs F., Collins J. J.* (2001) Computational studies of gene regulatory networks: In numero molecular biology. *Nature Rev. Gen.* 2, 268–279.

20 *Hucka M., Finney A., Sauro H. M., Bolouri H. et al.* (2003) The systems biology markup language (SBML): a medium for representation and exchange of biochemical network models. *Bioinformatics* 19, 524–531.

21 *Ideker T., Lauffenburger D.* (2003) Building with a scaffold: emerging strategies for high- to low-level cellular modelling. *Trends Biotechnol.* 21, 255–262.

22 *Jacob F., Monod J.* (1961) Genetic regulatory mechanisms in the synthesis of proteins. *J. Molec. Biol.* 3, 318–356.

23 *Kitano H.* (2002) Systems biology: a brief overview. *Science* 295, 1662–1664.

24 *Kremling A., Bettenbrock K., Laube B., Jahreis K., Lengeler J. W., Gilles E. D.* (2001) The organization of metabolic reaction networks. III. Application for diauxic growth on glucose and lactose. *Metabol. Eng.* 3, 362–379.

25 *Kremling A., Jahreis K., Lengeler J. W., Gilles E. D.* (2000) The organization of metabolic reaction networks: a signal-oriented appro-

ach to cellular models. *Metabol. Eng.* 2, 190–200.

26 *Larsen M. N.* (1993) Deriving action sequences for start-up using multilevel flow models. PhD Thesis, Department of Automation, Technical University of Denmark.

27 *Larsson J. E.* (1996) Diagnosis based on explicit means-end models. *Artif. Intell.* 80, 29–93.

28 *Lee T. I. et al.* (2002) Transcriptional regulatory networks in Saccharomyces cerevisiae. *Science* 298, 799–804.

29 *Lengeler J. W., Drews G., Schlegel H. G.* (1999) Biology of the prokaryotes. Thieme Verlag, Stuttgart, Germany/Blackwell Science, Oxford, UK.

30 *Lind M.* (1994) Modeling goals and functions of complex industrial plant. *Appl. Artif. Intell.* 8, 259–283.

31 *Lind M.* (1999) Plant modeling for human supervisory control. *Trans. Inst. Measure. Control* 21, 171–180.

32 *Lind M.* (2002) Promoting and opposing. NKS-R-07 Project Report, Ørsted DTU, Technical University of Denmark, Kongens Lyngby, Denmark.

33 *Lind M.* (2004a) Description of composite actions-towards a formalization of safety functions. NKS-R-07 Project Report, Ørsted DTU, Technical University of Denmark, Kongens Lyngby, Denmark.

34 *Lind M.* (2004b) Means and ends of control. Proc. IEEE Conf. Systems Man and Cybernetics, 10–13 October 2004, The Hague, Holland.

35 *Lyons J.* (1977) Semantics 2. *Cambridge University Press*, Cambridge, MA.

36 *Mackey M. C., Santillán M., Yildirim N.* (2004) Modeling operon dynamics: the tryptophan and lactose operons as paradigms. *C. R. Biologies* 327, 211–224.

37 *Makrides S. C.* (1996) Strategies for achieving high-level expression of genes in *Escherichia coli*. *Microbiol. Rev.* 60, 512–538.

38 *Mangan S., Alon U.* (2003) Structure and function of the feed-forward loop network motif. *Proc. Natl. Acad. Sci.* 100, 11980–11985.

39 *Mangan S., Zaslaver A., Alon U.* (2003) The coherent feedforward loop serves as a sign-sensitive delay element in transcription networks. *J. Mol. Biol.* 334, 197–204.

40 *Milo R., Shen-Orr S., Itzkovitz S., Kashtan N., Shklovskii D., Alon U.* (2002) Network motifs: simple building blocks of complex networks. *Science* 298, 824–827.

41 *Postma P. W., Lengeler J. W., Jacobson G. R.* (1993) Phosphoenolpyruvate:carbohydrate phosphotransferase systems of bacteria. *Microbiol. Rev.* 53, 543–594.

42 *Ronen M., Rosenberg R., Shraiman B. I., Alon U.* (2002) Assigning numbers to the arrows: parameterizing a gene regulation network by using accurate expression kinetics. *Proc. Natl. Acad. Sci.* 99, 10555–10560.

43 *Salgado H. et al.* (2004) Regulon DB (version 4. 0): transcriptional regulation, operon organization and growth conditions in *Escherichia coli* K-12. *Nucleic Acids Res.* 32, D303–D306.

44 *Santillán M., Mackey M. C.* (2004) Influence of catabolite repression and inducer exclusion on the bistable behavior of the *lac* operon. *Biophys. J.* 86, 1282–1292.

45 *Shen-Orr S. S., Milo R., Mangan S., Alon U.* (2002) Network motifs in the transcriptional regulation network of *Escherichia coli*. *Nature Genetics* 31, 64–68.

46 *Simon H. A.* (1981) The sciences of the artificial. *MIT Press*, Cambridge, MA.

47 *Smolen P., Baxter D. A., Byrne J. H.* (2000) Modeling transcriptional control in gene networks-methods, recent results, and future directions. *Bull. Math. Biol.* 62, 247–292.

48 *Stelling H., Klamt S., Bettenbrock K., Schuster S., Gilles E. D.* (2002) Metabolic network structure determines key aspects of functionality and regulation. *Nature* 420, 190–193.

49 *Von Wright G. H.* (1963) Norm and action-a logical enquiry. Routledge and Kegan Paul, London.

50 *Vukmirovic O. G., Tilghman S. M.* (2000) Exploring genome space. *Nature* 405, 820–822.

51 *Wei G.-H., Liu D.-P., Liang C.-C.* (2004) Charting gene regulatory networks: strategies, challenges and perspectives. *Biochem. J.* 381, 1–12.

52 *Wolkenhauer O., Kitano H., Cho K. H.* (2003) Systems biology. *IEEE Control Syst. Mag.* 23, 38–48.

53 *Wong P., Gladney S., Keasling J. D.* (1997) Mathematical model of the *lac* operon: inducer exclusion, catabolite repression, and diauxic growth on glucose and lactose. *Biotechnol. Prog.* 13, 132–143.

54 *Yeger-Lotem E. et al.* (2004) Network motifs in integrated cellular networks of transcription-regulation and protein-protein interaction. *Proc. Natl. Acad. Sci.* 101, 5934–5939.

55 *Yeger-Lotem E., Margalit H.* (2003) Detection of regulatory circuits by integrating the cellular networks of protein-protein interactions and transcription regulation. *Nucleic Acids Res.* 31, 6053–6061.

56 *Yildirim N., Mackey M. C.* (2003) Feedback regulation in the lactose operon: a mathematical modeling study and comparison with experimental data. *Biophys. J.* 84, 2841–2851.