Universitat Autònoma de Barcelona

PML – A MODELING LANGUAGE FOR PHYSICAL KNOWLEDGE REPRESENTATION

By
Juan José Ramos González

SUBMITTED IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY
AT
UNIVERSITAT AUTÒNOMA DE BARCELONA
BELLATERRA, BARCELONA
JULY 2003

UNIVERSITAT AUTÒNOMA DE BARCELONA
DEPARTMENT OF
TELECOMMUNICATIONS AND SYSTEM ENGINEERING

The undersigned hereby certify that they have read and recommend to the Faculty of PhD Studies for acceptance a thesis entitled "**PML – A modeling Language for Physical Knowledge Representation**" by **Juan José Ramos González** in partial fulfillment of the requirements for the degree of **Doctor of Philosophy**.

Dated: July 2003

Research Supervisor: ______________________________
Miquel Àngel Piera i Eroles

A María José, Julia y mis padres.

Contents

List of Tables

List of Figures

Abstract

The topic of this thesis is the automated modeling of physical systems. Modeling automation has been a common objective in many of the present modeling tools. Reuse of predefined models is probably the main approach adopted by many of them in order to reduce the modeling burden. However, to facilitate reuse is difficult to achieve and, as it is discussed thoroughly in the thesis, reusability of models can not be assured when they are predefined to represent the system dynamics in a particular physical context. In order to avoid the reuse constraints due to the system dynamics formulation, a modeling language should be defined with a clear separation between the physical behaviour representation aspects (declarative physical knowledge) and the computational aspects concerning to model simulation (procedural computational knowledge). The physical knowledge will represent the system behaviour and it will support the analysis of the model reusing context in order to set the system dynamics formulation.

The aim of this work is the design of a modeling language, PML, able to automate the modeling process by assuring the reusability of ready-made models independently of the physical context where they have been defined. The reuse of a predefined model contemplates both the construction of new models (structured modeling) and the model usage for different experimentation purposes. New models are constructed by coupling predefined models according to the physical system topology. Such structured models are manipulated in order to obtain the representation of the system dynamics which are of interest for the experimentation purposes.

PML is an object oriented modeling language designed to represent system behaviour by means of modular structures (modeling classes). The PML modeling classes describe physical concepts well-known by the modeller. The physical knowledge declared by the modeling classes is used to analyze structured models in order to generate automatically the mathematical representation of the system dynamics. The simulation model is obtained by means of an equation-based object oriented modeling language.

Acknowledgements

I would like to thank Dr. Miquel Àngel Piera, my supervisor, for his many suggestions and constant support during this research. He shared with me his knowledge and provided many useful references and friendly encouragement. I am also thankful to Professor Ignasi Serra for his confidence in me when he gave me the opportunity to begin my research and educational activities. I wish to thank to Dr. Romualdo Moreno and Dr. Ramon Vilanova, also members of the department staff. Without all of them, this work and other endeavors would never reach a successful destination.

I wish to thank also to the people whose friendship and support has helped me during these years. Special mention to Dr. Mattson, Dr. Nilsson and Dr. Andersson for their guidance during my first steps in the system modeling field, and to my good friend Quim for many reasons. Their contribution has been also invaluable to the completion of this project.

Of course, I am grateful to my family. Agradecido a mis padres por darme la oportunidad y los medios para haber llegado hasta aquí. Agradecido a María José por su paciencia, apoyo y compresión. Agradecido a Julia, por saber entender que *papá* tenía que trabajar. Agradecido a Elena y Carlos, mis hermanos. Sin todos ellos, este trabajo nunca hubiera existido.

Tarragona, Juanjo Ramos
August, 2003

1

Introduction

This thesis is about the representation of the physical knowledge required to build the model of a physical process system. The thesis presents a modeling environment designed to give support in the modeling process in scientific and engineering activities related to simulation.

Nowadays there is an increasing industrial interest in the use of modeling and simulation tools. Many sectors of the process industries are under intense and growing pressures from the world economies. As a result, industries are currently faced with a number of challenges such as increasing responsiveness to market forces, benefit margin improvement and intensifying competition or safety enhancements according to a growing body of legislation and standards. All of these key issues are closely related to the behaviour of the plant: a plant which is not efficient will not only consume more raw materials and energy but also will tend to have many problems in other areas, such as flexible operations to cover market demands, safety incidents, excess waste production, inconsistent quality and so on. In order to avoid these problems and improve competitiveness, production and manufacturing should be highly efficient, flexible and responsive to market dynamics, safe, clean and reliable. To achieve these goals the plant must be operated properly and the operational characteristics of the plant must reflect not only the process itself but also the control or safety systems and the supervisory and management procedures. It is becoming essential to develop the process and its “operating systems” as an integrated whole, taking full account of both the steady-state and dynamic behaviour of the production environment from the very beginning. This is, for instance, the essence of CAPE (Lien and Perris 1996): *"the application of a system's modeling and simulation approach to a process and its control, safety, utility and environmental protection systems as an integrated whole, from the viewpoints of development, design and operations"*.

Despite most of these objectives could be covered by means of simulation, there are many industrial branches where the modeling and simulation techniques are scarcely used. One of the main reasons is that the development of mathematical models for complex systems is difficult and time consuming. Here, and through this thesis, complexity should be understood in the sense of systems composed by many subsystems and components, whose behaviour can usually be described from the very basic principles, but where the amount of components and the interactions among them is the source of the modeling difficulties. Note that setting up the model of those non-trivial physical systems from the very basic principles is a hard task subject to human and technical errors such as inexperience with the physical system, ignorance of specific physical laws, corrupted measured data, etc. These risks, combined with the close dependency relationship between modeling and simulation, lead to a lack of confidence in simulation results.

The topic of this thesis is to develop a modeling environment with the aim of contributing to the improvement of the modeling techniques' reliability and availability. A significant effort has been made by the research community to develop software modeling tools able to cope with the growing demands for simulation models of ever increasing complex industrial processing units, with component models stemming from different application fields. Since early seventies, the effort has mainly been focused in creating different software tools that simplify the modeling task. A brief draw on the continuous-time modeling and simulation historical evolution can be found in (Åström *et al.* 1998). These previous and current research efforts have influenced the work reported in this thesis.

1.1. Basic modeling concepts

The modeling process can be defined in a wide sense as the activity of developing models. A model is a collection of information that describes the characteristics of a system in a specific situation which is of interest for certain purpose. Some relevant aspects of the modeling activity have to be explored before giving a more concise definition of model suitable with the context where this thesis is developed. A good starting point, widely used in the literature, is the model definition given by Minsky in (Minsky 1965):

> *" To an observer $\mathcal{O}$, an object $\mathcal{M}$ is a model of a system $\mathcal{S}$ to the extent that $\mathcal{O}$ can use $\mathcal{M}$ to answer questions that interest him about $\mathcal{S}$"*

This definition of model as an *object* covers almost everything from some understanding of how a particular system works to an explicit mathematical representation of the system dynamics. This thesis is interested on those models which can be constructed and manipulated by computer programs since the observer $\mathcal{O}$ will use $\mathcal{M}$ by means of a software tool.

The definition establishes a ternary relation among the observer, the system and the model. At the core of this relation is the observer's experimentation interest, which somehow will address the model construction.

The experimentation interest will guide the way in that the modeller observes the system. The modeller focus his modeling interest in those particular issues of the system he wants experiment with. The experimentation interest is frequently stated in terms of the experimentation objectives (see for instance (Zeigler *et al.* 2000)):

DEFINITION 1.1 – *Experimentation objectives*

The experimentation objectives can be defined as the set of system issues to which the observer wants to find answers. ❑

The experimentation objectives are essential to focus the model construction since they define the role of the model in the application domain (system design, control, management, etc). The formulation of the experimentation objectives establishes the experimental framework (Zeigler *et al.* 2000).

DEFINITION 1.2 – *Experimental framework*

The experimental framework is a specification of the conditions under which the system is observed or experimented with. ❑

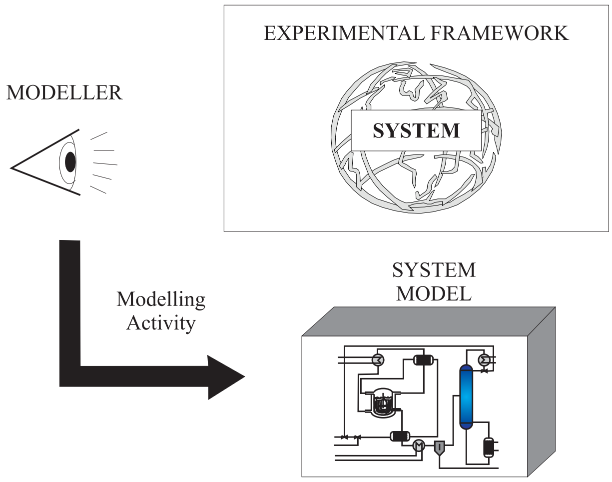

Figure 1.1. The ternary relation among the modeller, the system and the system model: the observer puts the system into the experimental framework to build the model.

At a first stage of a modeling and simulation project the experimental framework should be defined according to the experimentation objectives. The experimental framework specifies the set of system outcomes measures according to the experimentation objectives. For example, given a certain industrial process, one observer may be interested in the scheduling of the production process and another observer may be interested the control system design of the involved process units. The experimentation objectives, and in consequence the experimental framework, will be completely different in both cases: the first observer will demand information about measures such as the material processing time, busy resources or fabrication bottlenecks; the second observer will demand information about the physicochemical phenomena occurring in the process units in order to design the process control system.

As the Figure 1.1 illustrates, the observer puts the system into the experimental framework defined by its experimentation interest. The modeller will answer to these specifications with a model able to satisfy the experimentation objectives.

The modeling activity is difficult and usually requires of expertise in the model application domain with a deep knowledge of the particular modeling and simulation tool used to develop and validate models. Many different experimental frameworks can be defined for a system, so different models can be built for the same system. Therefore, the modeling cost becomes of more relevance since the same system will require different models in order to answer the questions made by each observer according to his experimentation objectives.

Different modeling and simulation environments have been developed during the last decades in order to reduce the modeling burden and its related cost. Two main aspects can be distinguished in every modeling environment: the *representation formalism* and the *modeling methodology*. The representation formalism is defined by the language used to describe the system behaviour. The modeling methodology will define a set of rules specifying the way in that the knowledge about the system physical behaviour must be organized (or structured) in order to build the model.

A third important aspect to be considered is the relationship between the system model (as it is specified according to the language and the methodology defined by the modeling environment) and the model formulation suitable for the simulation environment (simulator). We will name this relationship the *simulator relation*. This relationship states an agreement between the model and its solution engine which must be fulfilled in order to guarantee a correct generation of the experimentation results.

The simulator becomes the fourth entity involved in a modeling and simulation project, together with the system, the experimental framework and the model. The simulator will impose certain constraints to the model formulation in order to apply the numerical solvers. In many modeling approaches, the model as it is defined by the modeller does not match with the model formulation required by the solvers. In these cases, the model built according to the rules defined by the modeling tool has to be manipulated in order to derive a model formulation suitable for the simulator.

To sum up these factors, we may say that the model construction process is guided by the experimental framework, the representation language, the modeling methodology and the simulator relation. A short overview of these essential aspects is presented bellow within the context of this thesis.

System behaviour representation formalism

A representation formalism can be defined by the set of linguistic symbols and formulas which configure the structure of a language. The philosophical concept of *form* refers to the structure of anything as opposed to the contents (meaning). So, a formalization does not go into the semantics aspect of the language but focus on the syntactical rules which precisely state how well-formed sentences have to be defined. However, a language definition should embrace not only the formal aspects, but also the semantic and pragmatic aspects. The semantics of the language describes the meaning of the constructions made with the basic language concepts. The language pragmatics relates the language concepts to the concepts outside the language.

The mathematical formalism has been the most wide spread formalism used to represent the physical system dynamics in most of the present commercial modeling and simulation tools. Different types of mathematical models can be set up according to three main aspects: the dynamics with respect to *time* (continuous or discrete evolution), the spatial dependencies of the behaviour to be represented (lumped or distributed parameters) and the amount and type of knowledge we have about the system's behaviour. These characteristics lead to four type of models which practically range the whole spectrum of system dynamics modeling and simulation problems (Cellier 1991):

i) Continuous time system models (e.g. chemical or electrical systems) where the interest is focused on the evolution of certain dynamic over time and therefore can be expressed by means of *ordinary differential equations* (ODE) or *differential-algebraic equations* (DAE) such as Equation 1.1,

$$f(\dot{x}, x, u, t) = 0 \tag{1.1}$$

where u represents the input signals, x represents the state and t represents the time as the independent variable. *Partial derivative equations* (PDE) can be used in order to represent both the time and the spatial dependencies, but we do not consider them in this thesis.

ii) Discrete time system models (e.g. digital controllers or sampled continuous systems) where the dynamics are represented at discrete time instants. They are described by means of *difference equation* (DE) sets such as Equation 1.2.

$$x_{k+1} = f(x_k, u_k, t_k) \tag{1.2}$$

where u_k represents the input signals and x_k represents the state at the t_k time instant.

iii) Discrete-event models are used to represent a system whose state changes a finite number of times within a finite time interval. Examples of the application domain of this model type are the representation of the batch operation mode of a chemical process, the check-in queues in an airport or the representation of a flexible manufacturing cell. Different formalisms have been developed to represent such type of systems as , for instance, Finite State Automata (FSA) (Hopcroft and Ullman 1979), Petri nets and Colored Petri nets (Jensen 1997) or the Discrete Event System Specification (DEVS) (Zeigler 1990) formalism amongst others.

iv) Incomplete knowledge about the system behaviour can make quantitative models (i.e., models such as Equations 1.1 or 1.2) difficult to apply. A *Qualitative* model can express incomplete knowledge of physical values and relationships (Kuipers 1989). Qualitative simulation (Kuipers 1986) allows useful quantitative predictions to be inferred.

The simulation tools considered in this thesis can deal with the ODE, DAE and DE mathematical formalisms (see Equations 1.1 and 1.2). Many of the current modeling environments also use these formalisms to represent the physical behaviour. Some of them will be discussed in Chapter 2.

Model development methodology

According to the notion of model given by Minsky, the engineering activity somehow leads the way in that models are obtained. Two different approaches, each showing different methodologies, are usually distinguished in the modeling process. They are frequently related to the model application.

The first modeling approach is usually referred as *black box modeling* and the modeling process is called *system identification* (Johansson 1993). For instance, in control engineering a very usual modeling problem consists in determining a model which relates some input and output signals. In the *identification problem* it is attempted to find the structure and the parameters of the model. This is also the type of modeling in *system design problems* where, by means of an optimization process, the parameters of a model are found to fit with an input-output signal relationship.

The second modeling approach to develop models, where this thesis is concerned, consists in formulating the existing physical knowledge about the system behaviour. The models represent the laws derived from the fundamental principles of physics, i.e., conservation of mass, energy and momentum. Such modeling procedure usually begins from identifying the set of system components and gives rise to a set of (ideal) model components with some network structure of connections. The two following examples will illustrate the modeling approach.

Example 1.1 – *Modeling an electrical circuit system*

The Figure 1.2 shows a RLC circuit. In order to obtain a model of this system, first all the components are identified and their behaviour is described in terms of the proper physics laws such as, for instance, Ohms law; thereafter, the network structure which describes the interactions (in this case, the connection topology of the basic electrical components) should be represented with Kirchoff's voltage law. The physical knowledge applied to the analysis of the circuit or, at least, of one particular view of the circuit can be expressed with the equation:

$$U_i(t) \;=\; R * i(t) \;+\; \frac{1}{C}\int i(t)dt \;+\; L\frac{di(t)}{dt} \tag{1.3}$$

This mathematical expression can be solved analytically or by simulation. ❑

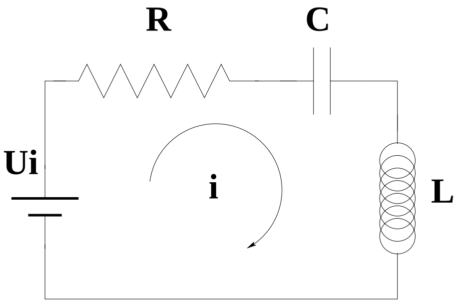

Figure 1.2. Electrical RLC circuit. R, L and C are system parameters, U_i is the voltage input signal and i is the electrical current.

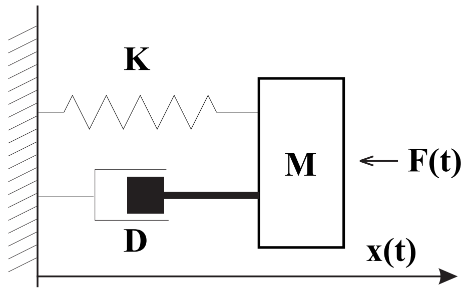

Figure 1.3. Mechanical system with a mass, a spring and a damper. M, D and K are the system parameters, $F(t)$ is an external force applied to the system and $x(t)$ represents the mass position.

In some ideal applications, the model components can easily be identified with the system components. In Example 1.1 the model components have been identified with the system components. These components are simple enough to be represented applying to the fundamental principles of physics and the network structure can represent the interactions between them. The model of a mechanical system can also be structured in a network of masses, springs, dumpers, etc.

Example 1.2 – *Modeling a mechanical system*

A simple mechanical system with a mass, a spring and a damper is shown in Figure 1.3. The essential mechanics involved in the motion of the body with (point) mass M is contained in Newton's second law. By identifying the forces acting on the body, the motion can be represented with the mathematical expression:

$$M\frac{d\dot{x}(t)}{dt} = F(t) - D\frac{dx(t)}{dt} - Kx(t) \tag{1.4}$$

Equation 1.4 includes the representation of the forces acting over the mass, i.e., the spring, the damper and the external forces. ❑

The way in that models are constructed will depend on the development methodology provided by the modeling tool. Many of the modeling methodologies within the engineering domain are based on the idea of decomposing (or structuring) a complex system (in the sense of systems composed by many subsystems and components) into subsystems until these subsystems are simple enough. These subsystems can be more easily represented, usually from the basic physical principles. Hence, the model of the complex system can be built from its subsystems models (model components). Each modeling methodology will define the set of rules to define such model components (some of the most relevant methodologies will be visited in Chapter 2).

A model component can be seen as a building block that can be reused as a part of a more complex model. The modeling component must declare certain physical behaviour and also must define how it can interact with other modeling components.

Definition 1.3 – *Modeling Component*

A *modeling component* is a basic model building block used to represent some physical behaviour. A modeling component should also define, in an explicit or implicit manner, the interaction with other modeling components in order to be able to link them with some network structure to declare additional physical behaviour. ❑

A modeling component may be almost everything from a simple variable representing a physical property to a complex software structure representing a process unit such as a chemical reactor. The set of definable modeling components configures the model development methodology since the way in that models are constructed is dependent on the nature of the modeling component.

Let us consider the Example 1.1 discussed above. At a very low level of abstraction, variables may be considered as modeling components representing physical quantities and parameters (e.g., $i(t)$ represents the electrical current in the RLC circuit). Variables are combined in the equations to represent the physical behaviour. Hence the equation can be viewed as the connection network of modeling components. The component connection network (i.e. the equation) is derived by the modeller applying to his physical knowledge. Equations obtained in this way are combined to represent the system behaviour. In the example, the Equation 1.3 can be derived from the physical laws ruling the behaviour of each electrical component (e.g., Ohm's law for the resistor) together with laws ruling their interactions in the circuit (e.g.,Kirchoff's voltage law). We can roughly speak about structured modeling when the methodology is based on variables as the basic building components and equations as the connectivity mechanism.

Let us assume that the modeling methodology supports a higher level of abstraction providing with representation structures where expressions such as $R * i(t)$ can be defined as a modeling component to represent the voltage drop in a electrical resistor. In this case, it can be established an analogy between the modeling component and the electrical component and becomes easier to identify the system component in the modeling component. Furthermore, the connection network of modeling components is much more simpler and can be derived just from Kirchoff's voltage law. The problem here is that expressions such as $R * i(t)$ can not be used always to represent the resistor behaviour, since in a different physical context the resistor behaviour must be represented by the expression $v(t)/R$ (e.g., when the current through the resistor has to be calculated from the voltage drop). The reuse of such structures is bounded to certain physical context implicit in the representation mechanism.

As a basic building block, a modeling component can be reused to define new models. This is the basis for a structured modeling process where the user can select modeling components from a repository and connect them to define a model. In order to support a structured modeling methodology, three basic features should be provided:

- The user should be able to identify easily the system physical behaviour he needs to represent in the modeling component. This could be achieved if there exists an analogy between the modeling components and the represented system components which can be explained from a physical perspective.

- There should be a unique association between a system component and the modeling component which represents it. The reuse of such modeling component should be context independent.

- The connection network of the modeling components should preserve the analogy with the topology of the subsystem connections in order to make easy the definition of new models.

Different interpretations of these characteristics are performed by each modeling methodology. For example, the *flowsheeting* modeling methodology (Westerberg and Benjamin 1983, Marquardt 1991), very extended in the process engineering domain, is based on the idea of decomposing a system into a set of subsystems which typically represent process units (e.g., pumps, valves or heat exchangers). The models of these subsystems are the modeling components. The network structure of subsystem connections is represented in terms of mass or energy transfer. The modeling components and their connections preserve the analogy with the system components and their connection topology. The *Bond Graph* modeling (Thoma 1990) is a methodology focused in the description of energy flows among system components. A bond graph model is basically a network of modeling components where the edges represent the energy flow paths and the nodes (modeling components) represent basic physical phenomena such as energy dissipation (resistance element), flow storage (capacitor element) or effort storage (inductance element).

The identification and definition of the modeling components that can be reused to build models of complex systems is not always an easy task. For example, in flowsheeting modeling there is a straightforward analogy between the modeling components and the system components. Nevertheless, these modeling components are usually complex models (e.g. a distillation column) built within an experimental framework. This is the main reason why they can not be reused in a context different to the one that was considered when the modeling component was built. Bond graph modeling components are much more simpler since they represent basic physic principles. However, bond graph modeling components do not always match with the system components so it is difficult to define their connection topology.

Usually, when the modeling components become more complex, their connection network becomes simpler. However, when the modeling component becomes more and more complex there is a risk to constrain its reusing context. The advantages and limitations of the most relevant current modeling methodologies will be analyzed in Chapter 2.

The modeling process

The modeling process was roughly defined at the beginning of this section as the activity of building models. A unified procedure to perform the model building process has not been described in the literature yet (Lohmann and Marquardt 1996). Of course, there are many works devoted to the systematization of the modeling process according to a particular methodology. See, for instance, object oriented modeling (Rumbaugh *et al.* 1991), object oriented modeling in chemical process (Nilsson 1993) or bond-graph modeling (Karnopp and Rosenberg 1968) amongst others.

Even it is not in our aim to define such unifying procedure, we will give now a set of main objective factors which somehow characterize the way a model is built, leaving aside more subjective considerations such as the modeller background. These factors are the experimental framework, the means provided by the modeling tool in order to represent the physical behaviour and the adequacy of the model.

As it has been discussed at the very beginning of this section, the model development is performed within certain experimental framework and the built model should satisfy the experimentation purpose. Since different experimentation objectives will demand different system outcomes to be observed through the model, it can be expected that different models will be required, provided that an universal model is not a realistic option. Therefore, one essential factor driving the modeling process is the experimental framework (see definitions 1.1 and 1.2).

Intuitively, the model construction process, performed to contemplate the experimentation objectives, has an inherent notion of different representations of the system: we begin from a system representation which is successively manipulated and adapted according to the experimentation objectives. These representations of the system can be seen as different specifications of the modeled system describing the system knowledge at different levels.

The layered structure of system knowledge given by (Klir 1985) in the General System Theory framework will be used to describe this iterative process. This structure establishes four levels of system knowledge:

1. *Source*: what variables to measure and how to observe them. This level identifies a portion of the real world that we wish to model and the means to observe it.

2. *Data*: data collected from the modeled system. That is, measures and observations made for the system.

3. *Generative*: means to generate data in a data system. At this level we have the ability of recreate the collected data at the previous level by means of some type of representation, typically formula. Within the simulation context, we may think about this representation as a computer program to generate data.

4. *Structure*: components (at lower levels) coupled together to form a generative system. This is specific type of generative knowledge where the system is viewed as set of coupled subsystems.

This approach, sometimes referenced in the modeling community (Marquardt 1991, Zeigler *et al.* 2000), is interesting for our purposes since it defines a starting point to describe the modeling process as a progressive procedure where the model evolves towards a specification able to answer the questions about the system that motivated its construction.

Each level provides relevant information about the system. We may relate the source and data levels to the experimentation objectives and the consequent experimental framework since the lowest levels identify the portion of the real world that we wish to observe. The generative and structure levels can be related to the modeling process since, in the traditional context of modeling and simulation community, the system specification is done at these levels by defining a program able to generate the required system data.

So in certain way, we may think in the modeling and simulation activity as a procedure where the modeller moves from the higher levels of system knowledge to the lower levels. The modeling process will be considered as the task of obtaining a specification able to represent the system at generative or structure levels, taking into account that we are concerned with the system dynamics description.

Level	Name	System specification
0	Causal explanation	Causal formulation of the expected behaviour
1	Non causal explanation	Formulation of the expected behaviour
2	Phenomenological structure	Physical phenomena and interactions
3	Topological structure	System components and connection structure

Table 1.1
Levels of system specification

In order to characterize the modeling process, we will distinguish four layers to define the system specification (see Table 1.1):

- **Topological structure:** This specification of a system describes its physical structure, i.e., which are its subsystems and how are they interconnected. This is a usual way to think about a system.

- **Phenomenological structure:** This specification is more concerned with the phenomena occurring in the system and the physical interactions derived from the matter and energy transfer among its subsystems. The analysis of this structure is essential in the modeling process since at this layer the modeller should contemplate the experimental framework in order to define a system specification where the expected behaviour is represented.

- **Non causal explanation:** at this layer the phenomenological structure is formulated by means of mathematical equations. Is named non causal since the specification does not express any assumption on the physical causality, i.e., the formulation does not bother about which are the causes and which are the effects. For example, the formulation of a electrical resistor behaviour with the equation $V = IR$ does no assume the current is the cause and the voltage drop is the effect, it is simply a equality relation among variables and parameters.

- **Causal explanation:** this system specification must resolve the physical causality, which was no relevant at the previous layer. While the non causal explanation is the formulation of the expected behaviour (what the system does in relation with the experimental frame), the causal explanation is the formulation able to explain how the system does what it is expected to do. Therefore, the causal mathematical formulation should have the computational structure suitable to the numerical solver. This means that the system specification at this layer should fulfill the *simulator relation*.

According to the Klir's levels of system knowledge, the topological and phenomenological specifications are at the structure level, and the non causal and causal specifications are at the generative level.

The modeling tool should give to the modeller the means to start the model construction at one of these layers. Hence, a very important factor to be considered in the modeling process is the development methodology defined by the modeling tool and the system behaviour representation mechanism it provides since they will establish the layer at which the system specification has to be built. Depending on the modeling methodology, the system can be specified at a level which does not fulfill the *simulator relation*, i.e., the model can not be used to generate the experimentation results. In those cases, the modeling tool should provide with the means to move from the level were the model has been built into the level where the system data can be recreated. The main current modeling approaches and methodologies will be discussed in the Chapter 2.

The third factor to be considered in the modeling process is the adequacy of the model. The adequacy of a model can be considered as a measure established with respect the ability of a model to give a causal explanation for the phenomena of interest (Nayak 1995), but also it should be considered as a combination of validity and simplicity. We will understand by validity, or correctness, the capability of the model to generate, at least, the data collected from the system (data level at the Klir structure) within the experimental framework, even there are stronger forms of validity (see for instance (Zeigler *et al.* 2000)). Since we have assumed the non viability of universal models, we should consider simplicity to be always linked to the experimental framework. The simplicity property involves two aspects: the set of represented phenomena, which should be the minimum able to give a causal explanation for the behaviour of interest, and the accuracy or degree of detail used to formulate the phenomena. In other words, a simple model should not include irrelevant phenomena and should not make needlessly complex formulations of the relevant phenomena. All these factors can be gathered in the following definition:

Definition 1.4 – *Adequate models*

A model can be considered to be an adequate model if:

1. is able to give a causal explanation to the phenomena of interest,
2. is accurate enough within certain level of tolerance,
3. is the simplest model able to accomplish with the two previous conditions.

❑

Now we will try to explain the modeling process according to these three factors: the experimental framework, the means provided by the modeling tool in order to define the system specification and the adequacy of the built model. A good visualization of the modeling process is given in (Lohmann and Marquardt 1996). As Figure 1.4 shows, the modeling process can be seen as a trajectory in a three dimensional space spanned by the coordinates of specification, objectives and experimental framework and adequacy.

From this point of view, modeling can be seen as an incremental process where successive steps are taken towards the desired system specification achievement. By starting form scratch, a system specification is usually defined at the topological structure level looking at system as a set of components (subsystems) connected to perform certain task. According to the experimental framework, the phenomena of interest should be selected and the physical interactions have to be identified from the system topology. This specification has been defined as the phenomenological structure. In the next step, the phenomena of interest and the physical interactions can be formulated by means of the mathematical formalism. Finally, the mathematical specification should be manipulated in order to set the proper computational causality according to the

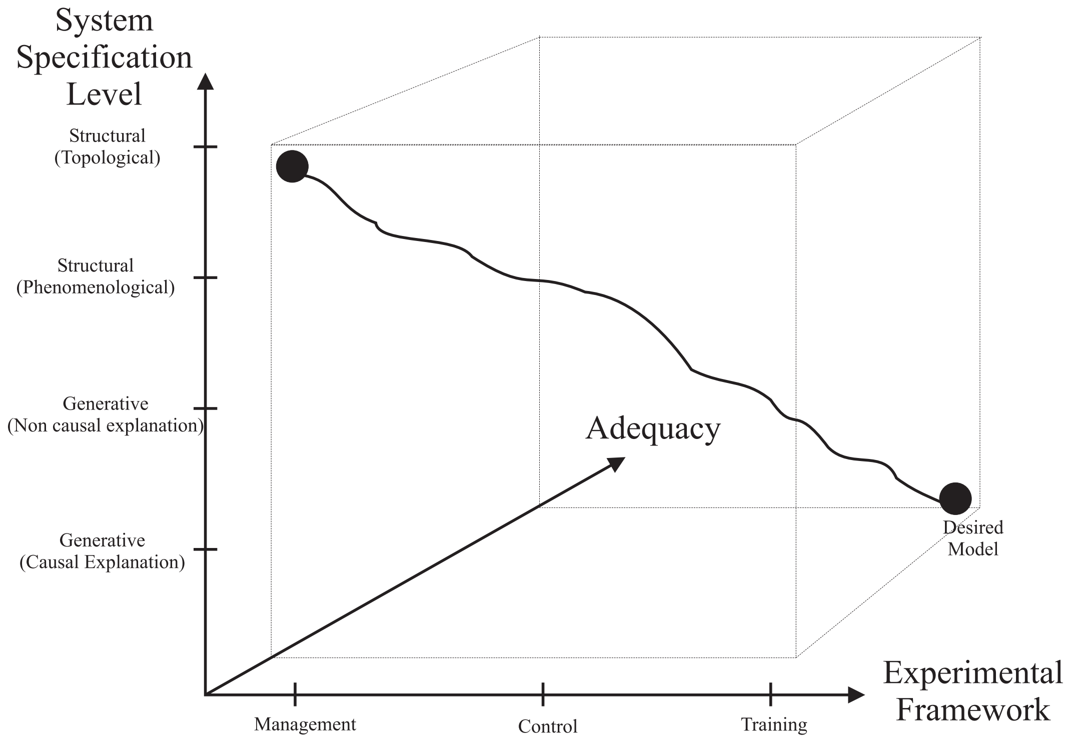

Figure 1.4. The modeling process trajectory.

simulator relation. Model adequacy should be present at every step, trying to find an agreement between the model validity and simplicity.

To follow the modeling process trajectory will require of two types of analysis. At the beginning of the trajectory, the modeller is more concerned with the physical aspects of the system he wants to model in order to consider all the related modeling factors. When the modeling problem gets closer to the causal explanation level the modeller has to be more concerned with the computational aspects of the model. Two important definitions are introduced to characterize the automation degree achieved by a tool to perform the modeling trajectory.

Definition 1.5 – *Physical analysis procedure*

By physical analysis procedure should be understood any activity performed applying to physical knowledge in order to move from the higher levels of system specification (topological or phenomenological) into the non causal specification level. ❑

Definition 1.6 – *Computational analysis procedure*

The computational analysis procedure is the symbolic mathematical manipulation activity required to move from the non causal explanation level into the causal level. ❑

The following example illustrates the modeling process trajectory by means of a simple system such as an electrical circuit.

Example 1.3 – *Modeling process with an electrical circuit*

Let us go back to the RLC circuit shown at Figure 1.2. At the topological structure level, we identify the electrical components (the voltage source U_i, the resistor R, the capacitor C and the inductance L) and their connection topology (a mesh where all the components are serial connected).

We can now deduce from this topology the phenomenological structure of the system: the physical behaviour at each component and the physical interactions among them (defined by the Kirchoff's mesh law.)

The phenomenological structure can be now formulated at the non causal explanation level without bothering about the causality:

$$\begin{aligned}
v_R(t) &= R * i_R(t) \\
C\frac{dv_C(t)}{dt} &= i_C(t) \\
L\frac{di_L(t)}{dt} &= v_L(t) \\
i_R(t) &= i_C(t) = i_L(t)\,[= i(t)] \\
U_i(t) &= v_R(t) + v_C(t) + v_L(t)
\end{aligned} \tag{1.5}$$

where v_R, v_C and v_L are respectively the voltages drops across the resistor, capacitor and inductance, and i_R, i_C and i_L the current through them.

Finally, we should resolve causality in order to set the causal explanation matching the simulator relation:

$$
\begin{aligned}
\frac{dv_C(t)}{dt} &= i(t)/C \\
\frac{di(t)}{dt} &= v_L(t)/L \\
v_R(t) &= R * i(t) \\
v_L(t) &= U_i(t) - (v_R(t) + v_C(t))
\end{aligned}
\tag{1.6}
$$

❑

Previous example tried to illustrate the modeling trajectory through the four levels of system specification. The experimental framework has been the analysis of the circuit dynamics and the adequacy factor made us to decide that, for instance, it was not necessary to consider the thermal effect over the resistance, capacity and inductance parameters.

The starting point of the modeling trajectory will be strongly conditioned by the modeling tool. Both the methodology and the representation formalism supported by the modeling tool will introduce certain rules to define the system specification at one of the described levels. This is because of the modeling tool defines the modeling components (see Definition 1.3) which can be used to build a model. If such modeling components are close to the topological view of the system, the tool supports the modeling process at the topological level, whereas if the modeling components are close to the generative levels (variables and equations), most of the modeling process will not be supported by the modeling tool and, therefore, the task will be performed by the modeller.

Obviously, the most interesting for a modeling tool is to support the system specification at the topological level, where predefined models of the system components could be reused, and perform the rest of the modeling trajectory automatically in order to reduce the modeling burden.

1.2. Modeling automation by means of reusability

Automating the modeling process has been a common objective in many of the present modeling tools. A usual trend in many of them is to facilitate the modeling process supporting the reuse of predefined modeling components (see Definition 1.3). Guiding users to build models with appropriate modeling rules, which provide information on the selection of the modeling components (component and interaction description mechanisms), may lead to a significant reduction of the model construction cost, while enhancing the quality of the models in terms of adequacy (validity and satisfactory degree of detail), consistency and robustness.

We should consider that the reuse of a modeling component has two different facets: first, a technical facet which should guarantee to a great extent that, when two valid modeling components are properly reused to define a new modeling component, the resulting modeling component is also valid and is able to explain the aggregated physical knowledge; second, a pragmatics facet which should make easy to the modeller the selection of the appropriate modeling component. This second facet will depend on the capability of the modeling component to represent physical elements and concepts familiar to the modeller.

As it has been discussed in the previous section, essential aspects in the modeling process are both the representation formalism (the modeling language) and the modeling methodology. They should be considered to be separated things. Two modeling environments may share one of these aspects and differ in the other, leading to completely different trajectories in the modeling process. For example, the equation oriented flowsheeting (Westerberg and Benjamin 1983) and the bond graph modeling share the same equation based language, but they define two completely different modeling methodologies since, at is has been discussed in Section 1.1, the nature of the modeling components differs in each approach.

The language used to represent the physical knowledge, as well as the modeling methodology, establish the type of definable modeling components (such as variables and equations or physical quantities and process units). According to the pragmatics facet of reusability, the modeling components should have a semantic interpretation close to the way engineers look into the

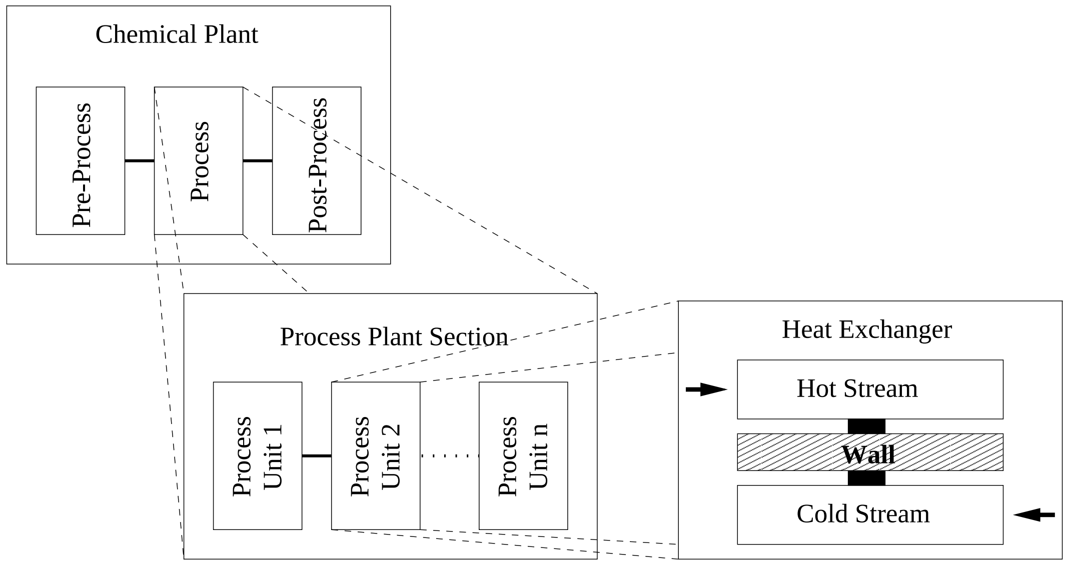

Figure 1.5. Structured modeling of chemical process. The models of the process units are successively connected according to the system topology.

system. Thus, it would be much more easy to specify the system model at the topological level (see Table 1.1) reusing complex modeling components such as process unit models (see Figure 1.5), than to make the specification at the causal explanation level by means of variables and equations.

From the model user's point of view, a complex modeling component provides more readable information about the represented physical behaviour than, for instance, a set of equations does. For example, a balance equation such as Equation 1.4 represents the physical knowledge (Newtons second law) in an implicit manner, i.e., the represented physical behaviour is an interpretation of the equation. Explicitly, it is just a second order differential equation. The mathematical (equation-based) language used in Examples 1.1 and 1.2 to express at the causal explanation level the physical knowledge about the system has explicit information of the procedural aspects of a model (the simulator relation: how the model can be solved within a computer), but it is not a powerful way to offer explicit information about the represented knowledge.

In a structured modeling method, as is illustrated at Figure 1.5, the model of the system is defined by connecting the models of the subsystems. Hence, such a method relies on the capability to reuse the modeling components which can be predefined in a modeling repository. The capability of providing support to build models at the topological structure level by means of reusability is closely related to the modeling component ability to represent the physical knowledge in an explicit manner.

A short discussion on the requirements imposed to the modeling environment in order to support this modeling method follows. First, we discuss which should be the scope of reusability. In a second term, we will analyze the implications which may affect the modeling language in order to support the reuse of predefined modeling components and also to help the modeller in the modeling process.

Scope of reusability

Developing a modeling environment able to support the system specification at the topological structure level and the automation of the process needed to follow the modeling trajectory towards the causal explanation level, requires first to establish which is the scope of the reusability of a modeling component. We will consider two main aspects with respect to the reusability:

- **Aggregation of modeling components:** this is the basis for a structured modeling method. It should be possible to aggregate, or couple, modeling components in order to declare new modeling components. A clear example is the aggregation of subsystem models to build a new model such as, for instance, Figure 1.5 illustrates.

- **Free reusing physical context:** the reuse of a modeling component should not be confined to the physical context where it was defined. The physical context is determined by the rest of aggregated modeling components and by the experimental framework since it drives the way in that the modeller looks at the system when he defines the modeling component.

Aggregation of modeling components does not necessarily mean that the system specification is made at one of the structure levels (topological or phenomenological) shown at Table 1.1. Actually, many of the classical modeling approaches such as the CSSL and block-oriented tools allow reuse of predefined modeling components. However, the system specification is made at the causal explanation level (this matter is treated with detail in Chapter 2). In this case, the reusing context of a modeling component is certainly constrained by the causal formulation stated at this level. Furthermore, the experimental framework and the adequacy must be completely defined before the model is formulated.

Our interest is to set the requirements in order to support model reuse at the topological level. Let us assume a modeling tool where the system specification can be performed at the topological structure level, so the model of a system can be set up by aggregating predefined modeling components representing its subsystems. According to this methodology, the modeling tool should be able to analyze the specification at the topological level in order to obtain the aggregated behaviour and formulate it at the non causal explanation level. Therefore, the modeling path through the phenomenological and non causal levels should be covered by the modeling tool, preferably without involving the model user.

The same model component may take part in the construction of different structured models. Generally, the reusing context of a modeling component can differ from one structured model to other. The main question to be solved by a modeling tool supporting this methodology is which behaviour can and must be predefined in each modeling component without constraining its reusing context.

First to answer this question, we should consider that the physical reusing context of a modeling component may affect to the aggregated behaviour in three forms:

1. In its simplest way, the reusing context may affect to the physical causality of the aggregated behaviour without variation of the phenomenological structure. The causal formulation of the behaviour represented at each modeling component must properly be set by the modeling tool according to the resulting physical causality.

2. A more complex situation is found when the reuse of a modeling component in different contexts leads to different phenomenological structures since the physical causality in the aggregated behaviour is dependent on the phenomenological structure. This is the situation occurring when the reusing context sets conditions over the phenomena which should have been predefined at the modeling component. The modeling tool should be prepared to anticipate such effect on the phenomenological structure and process the topological specification to derive it.

3. The third effect on the aggregated behaviour may occur when the experimentation objectives and the desired adequacy required by the model user are different from the ones stated at the modeling component definition. We are proposing that a modeling component can be reused within an experimentation framework which is different from the framework where it was built. In this case, first to establish the phenomenological structure, the modeling tool should provide the user with the mechanisms to select which of the represent behaviour is of interest for his particular experimental framework and the desired adequacy level.

These side effects due to the physical reusing context of a modeling component are discussed through the following examples . Consider that the electrical circuit shown at Figure 1.2 is specified at the topological structure level and the modeling tool is able to move from this level into the non causal explanation level achieving the formulation in Equation 1.5. Assume that a similar circuit is now specified at the topological level with the same modeling components except one of them, in such a way that the physical causality varies but the phenomenological structure remains unaltered. For example, the voltage source $U_i(t)$ at the circuit in Example 1.3 is replaced by a current source $i(t) = A\sin(wt)$. In this case, the non causal explanation shown in Equation 1.5 remains unaltered. However, the causal explanation will be different from Equation 1.6 since the physical causality has changed. The new physical causality leads to the following new causal explanation:

$$\begin{array}{ll} \frac{di(t)}{dt} = Aw\cos(wt) & i(t) = A\sin(wt) \\ \frac{dv_C(t)}{dt} = i(t)/C & v_R(t) = R * i(t) \\ \frac{di(t)}{dt} = v_L(t)/L & U_i(t) = v_L(t) + v_R(t) + v_C(t) \end{array} \tag{1.7}$$

The reusing context of every modeling component may lead to different causal formulations of the behaviour due to the variation of the physical causality. A typical example is an electrical drive that works as a DC motor if we feed the armature and field circuits, and it works as a generator if we apply a torque to its shaft (this system will be analyzed with some depth in the next chapter).

It has been assumed that the modeling tool is able to move automatically from the topological into the phenomenological specification level. The non causal explanation level was previously defined as the non causal mathematical formulation of the phenomenological structure (see Table 1.1). In many reusing contexts of a modeling component, even the same physical phenomena occur (i.e. the phenomenological structure remains unaltered), the causality of the equations used to formulate them is different since the physical cause-effect relationships differ. To complete the modeling path towards the causal explanation level, the modeling tool should deal with the variation of the physical causality at the non causal explanation level.

Moving from the non causal into the causal explanation level is basically a problem of equation manipulation (Cellier and Elmqvist 1993), which has been defined as a computational analysis procedure (see Definition 1.6). This manipulation procedure has two main steps: first, to determine which equations have to be used to solve each unknown variable; second, the proper equation symbolic manipulation in order to set a computable code sequence. This problem is treated by some equation based modeling approaches, such as the equation-based object oriented modeling languages (e.g, (Elmquist and Al. 1999)), which are able to manipulate the equations to assign the proper computational causality in order to set the causal explanation.

A more difficult to solve problem is found when the contribution of a reused modeling component to the phenomenological structure of the aggregated behaviour depends on its reusing

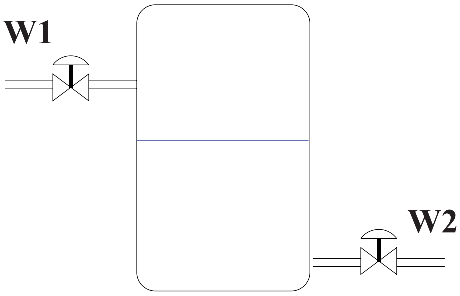

Figure 1.6. Reservoir system to store matter.

context. The phenomenological structure specification represents the phenomena occurring in the system and the physical interactions derived from the matter and/or energy transfer among its subsystems (see Table 1.1). Therefore, it is not hazardous to anticipate that if the same modeling component may lead to different phenomenological structures, the phenomena and physical interactions which should be predefined in the modeling component depend on its reusing context.

The following example will be used to illustrate which are the main difficulties (for a modeling environment) to contemplate the variation of the phenomenological structure depending on the reusing context of the modeling components.

Example 1.4 – *Mathematical model of a reservoir system*

Consider the system shown in Figure 1.6. We should build a model representing its behaviour. According to the structured method at the topological level, we will develop several modeling components representing the behaviour of the physical components, i.e., a tank and two valves.

We should also consider that these modeling components may act as submodels in future structured models. Therefore, the modeling components should represent also the interactions with other modeling components (e.g. valves, pipes or pumps). However, the modeling component interaction mechanism is not relevant at this moment. We simply assume that it makes possible the derivation of the aggregated behaviour.

Let us consider that modeling language is based on a mathematical formalism. In this case, some decisions have to be taken from the very beginning to obtain a valid model of the tank system. These decisions are taken according to the experimental framework and and adequacy coordinates and could be, for instance, the dynamic of interest (e.g. matter balance), what type of matter is involved (e.g. single component or multi-component material), which of its properties are represented (e.g. the matter mass) and which material inflows or outflows are present in the system. These considerations would lead to a mathematical modeling component representing the tank such as:

$$\begin{aligned}
\frac{dm}{dt} &= w_1 + w_2 \\
m &= \rho S h \\
P_t &= P_a \\
P_b &= P_a + \rho g h
\end{aligned} \tag{1.8}$$

where m is the accumulated mass and w_1 and w_2 are variables representing the input/output mass flows, P_t and P_b are the pressures at top and bottom tank outlets respectively, S is the tank section, h is the liquid level, P_a is the atmospheric pressure and g the gravitational constant. We will consider that w_1 and w_2 define the tank model interaction by sharing them with other modeling components (e.g. a valve model).

Hence, the valve could be represented by the following modeling component:

$$\begin{aligned}
w * |w| &= C_v A_p \Delta P \\
\Delta P &= p_1 - p_2
\end{aligned} \tag{1.9}$$

where w is the mass flow, p_1 and p_2 are the pressures at the valve ends, C_v is the valve constant and A_p is the opening factor.

By connecting these modeling components according to the system topology, the modeling tool derives the following aggregated behaviour:

$$\begin{aligned}
\frac{dm}{dt} &= w_1 + w_2 \\
m &= \rho S h \\
w1 * |w1| &= C_{v1} A_{p1} (p_1 - P_a) \\
w2 * |w2| &= C_{v2} A_{p2} (P_a + \rho g h - p_2)
\end{aligned} \tag{1.10}$$

where p_1 and p_2 can be considered as the boundary conditions for model experimentation.

With this mathematical formulation all the physical objects and knowledge (the tank, the matter, the occurring phenomena) do not have their counterparts modeling components in the specification. They are represented by means of variables and equations, establishing an implicit representation of the system in a particular context. It can be considered that the system shown in Figure 1.6 has been specified at the non causal explanation level (it is not predefined if w_2 is input or output flows) according to certain phenomenological structure. ❑

When a system is structured in subsystems in order to build separately each subsystem model, the resulting coupled behaviour can not be generally represented just by adding the equations of each component model and manipulating them to assign the proper computational causality. Let us consider for instance the model of the valve in Equation 1.9. This model has been defined to represent a matter transport according to a physical law. It can be reused at any context requiring a matter transport representation provided that the represented phenomenon can be formulated according to Equation 1.9 (same adequacy and experimental framework). For instance, the mass transfer between two tanks whose models are defined in a similar way as in Example 1.4 by the model in Equation 1.8. Let us assume that, in a different reusing context, the same valve is used to regulate the mass flow between a heat exchanger and a chemical reactor. The physical interactions derived from the mass transfer through the valve are completely different in this case where the matter properties carried out by the stream (such as temperature, density or specific heat) must be represented. Hence, the transfer of these properties through the valve would require new equations to be added in its model. Even the same physical valve may be used without distinction in each case, the need of adding more equations in the second case leads to a different valve model just because the valve reusing context differs introducing side effects on the phenomenological structure to be represented.

Regardless of the adequacy and the experimental framework coordinates, if the modeling of the same system component (e.g. the valve) leads to different phenomenological structures in different contexts means that the resulting aggregated behaviour does not only depend on the modeling component but also on the other modeling components to whom it is coupled. By considering that:

1. the same modeling component may require of different behaviour representations depending on its reusing context, and

2. a structured modeling method should lay on the assumption that the reusing context of a modeling component can not be predefined,

the reuse of predefined modeling components seems to lead towards a contradictory condition: a modeling component should predefine the representation of a behaviour which is dependent on its reusing context, which should not be predefined.

In order to avoid such a contradiction, it must be taken into account that the phenomenological structure is not a property local to each modeling component, on the contrary, its a property global to the whole topological structure. Therefore, if the system has to be specified at the topological level, it should be possible for the modeling tool to derive the adequate phenomenological structure according to the reusing context of every aggregated modeling component. Which means that the phenomenological structure can not be implicitly described by the topological specification.

Finally, it should be also considered how the aggregated behaviour could be affected if the experimentation objectives and the desired adequacy of the model user change from the ones stated at the modeling component definition. The variation of the experimental framework affects to the phenomena of interest, which must be clearly represented at the phenomenological level. The adequacy level determines how the phenomena of interest has to be formulated at the non causal explanation level. The variation of the adequacy level means that different mathematical formulations can be set according to the modeller interest. This implies that the modeling tool should offer the possibility to select the proper formulation (according to Definition 1.4) in the movement from the phenomenological structure level into the non-causal explanation level. It can be found in the literature many works devoted to the generation of adequate models (e.g. (Nayak 1995, Acebes 1996, Piera 1993)).

The experimental framework can be modified by the model user in two ways: either neglecting represented behaviour which is not of interest for his experimentation objectives (we may call this *behaviour pruning*), either *aggregating* new behaviour to an already defined modeling component. For example, we could be interested in neglecting the thermal phenomena in a tank with heat capacity model. To cover this facility means that different phenomenological structures can be set according to the experimentation objectives, each of them representing the proper set of phenomena. Supporting this facility implies that the modeling tool allows the selection of the phenomena of interest and, afterwards, is able to move automatically from the topological level into the phenomenological level, giving rise to different structures according to the experimental framework (this modeling tool feature has been already demanded in relation to the physical reusing context analysis). An opposite experimental framework variation is the aggregation of new phenomena to an already defined model. For instance, we could be interested in aggregating to the tank model in Example 1.4 the thermal phenomena representation. This way to modify the experimental framework may have strong implications over the rest of modeling components since aggregating new behaviour affects not only to the modified model, but also to any model where the modified model participates as a component.

Considering these facilities (experimental framework and adequacy variation) would extend the reusing context of a predefined model, both in the methodological sense of structured modeling, and in the sense of the adaptability of a ready-made model to different experimental frameworks and adequacy levels.

Implications over the modeling tool

Previous considerations about reusability as the mean to automate the model construction will condition the design of the modeling tool. It can be easily explained from the specification levels point of view: to the modeller concern models should be built at the topological level; to the simulator concern, models should be specified at the causal explanation level. So, which is the role of a modeling tool?. The non trivial solution to this question is: the modeling tool provides the user with a representation formalism (defined by the modeling language) and a methodology to specify models at the topological level and afterwards translates this specification into a formulation a the causal explanation level.

At the core of the modeling methodology should be the reuse mechanism as the mean to support such a structured approach. The modeling methodology is configured by the set of definable modeling components since they establish the starting point at the modeling process trajectory (see Figure 1.4). The type of modeling components, as behaviour representation structures, is defined by the modeling language. Therefore, we need to consider how the modeling language should be designed in order to support a methodology where the modeling trajectory starts at the topological specification and finishes at the causal explanation.

According to its pragmatics facet, the reuse of a modeling component at the topological level could become significantly easier if it represents a physical concept or entity familiar to the modeller. Hence, to the model developer concern the modeling language formalism should be related to the physical concepts (phenomena, laws, process units, etc.) as closely as possible. To the computer simulation concern the modeling formalism should be tightly related to the numerical aspects of the represented behaviour (the simulator relation must be fulfilled at the end of the modeling process). Therefore, the modeling language should also embrace the computational aspects in addition to the pragmatic and semantic aspects.

The following discussion intends to state which are the set of required properties to be fulfilled by the modeling language in order to accomplish with the proposed modeling automation by means of reusability. The modeling language will be characterized according to the following aspects:

- Its adaptation capabilities in the face of the physical reusing context variation (context adaptability).
- The language facilities as the vehicle to represent meaningfully the physical concepts familiar to the model user (the language expressiveness) while preserving the simulator relation.

We should recall here that the reusing physical context can vary in two forms:

1. Affecting to the physical causality of the modeling component as a part of a larger model. We may find two situations:
   (a) The phenomenological structure remains unaltered.
   (b) The phenomenological structure is altered.
2. Introducing modifications on the experimentation objectives and the desired adequacy. This will likely affect to the phenomenological structure.

Generally, it is not possible to anticipate the context where a modeling component is going to be reused. When the physical reusing context of a modeling component does not affect to the phenomenological structure (case 1.a), the behaviour represented by the modeling component can be predefined (it will not change with the reusing context). However, as it is illustrated through Example 1.4, this assumption can not become widespread stated since the reusing context of a modeling component may condition the physical behaviour which should have been predefined (cases 1.b and 2). By considering the effects on the phenomenological structure, we will characterize a modeling language in terms of its capabilities to give support to case 1.a and/or to cases 1.b and 2.

The terms *dynamic* or *static* are usually associated to a property which varies, or does not vary, with respect some independent coordinate (e.g. time). In this case, we are considering the reusing context as the independent coordinate since it can vary from one reusing case to other.

We may characterize the representation formalism defined by the modeling language in terms of its capabilities to support the effects of the reusing context on the phenomenological structure specification.

Definition 1.7 – *Static Modeling Formalism*

A formalism is said to be *static* when the reusing physical context must be predefined in order to represent the physical behaviour of the system in a modeling component. Once the reusing context is set, the represent behaviour can not be reused in a different one. ❑

According to the given definition of static formalism, the mathematical formalism used to define the modeling components in Example 1.4 should be considered static. First to made the specification at the non causal explanation level (Equations 1.8 for the tank modeling component and 1.9 valve modeling component), most of the factors in the modeling process have been stated (phenomena of interest, adequacy and experimentation objectives) giving rise to the phenomenological structure. Hence, the modeling components in Example 1.4 have completely determined which system phenomena are represented and how they are formulated. There are two main consequences:

1. Attending to the modeling component definition: decisions about the physical context have been made to define the modeling components represented by Equations 1.8 and 1.9. Therefore, its reuse is constrained to this context.

2. Attending to reusability: the mathematical formulation of the model should be carefully interpreted in order to restore the represented physical behaviour before the decision on the model validity in a particular reusing context could be taken.

In Example 1.4, the specification has been made at the phenomenological structure level before its formulation at the non causal level. We have decided which phenomena are represented in each modeling component and we have derived the aggregated behaviour from the defined phenomenological structure (even we have not specified how the interaction mechanism should work for simplicity reasons). Despite of the apparent analogy between the modeling components and the physical system components, the reusing context at the topological specification level is constrained by the phenomena predefined at the modeling component, i.e., by the represented phenomenological structure. Which means that the specification structure respond to the phenomenological structure defined by the physical context where the modeling components are defined. Therefore, it can be affirmed that the system specification is not made at the topological level despite of the coincidence between the specification topology and the component connection topology of the physical system.

The main consequences of a static formalism are the constraints imposed to the model reusing context, which limit the capabilities of a structured modeling methodology to support the system specification at the topological level. By considering that:

- The phenomenological structure is not a property local to one single component (the phenomena which should have been predefined in a modeling component may depend on its reusing context).
- The same topological specification may lead to different phenomenological structures depending on the experimental framework.

the modeling tool should be able to analyze the structure of the topological specification and derive the proper phenomenological structure according to the reusing context of each modeling component and to the experimental framework. Therefore, the modeling tool should be based on a representation formalism suitable for applying to such analysis. This type analysis has been defined as *physical analysis procedure* (see Definition 1.5). The physical analysis requires the use of physical knowledge, so the modeling language should support its representation in order to automate the modeling trajectory from the topological into the causal explanation level. We can give now a definition of a language providing with this characteristic.

Definition 1.8 – *Dynamic Modeling Formalism*

The formalism defined by the modeling language is said to be *dynamic* if it supports the representation of the physical knowledge required to perform the analysis procedure necessary to move from the topological structure specification level into the non causal explanation level. Two basic features are demanded to such a formalism:

- The reusing physical context does not have to be predefined in order to build a modeling component.
- It should permit the representation of the physical knowledge required to formulate the system behaviour at the non causal level according to different experimental framework and adequacy targets.

❑

Such a modeling language has to be concerned with the physical aspects of the system behaviour. The mathematical formulation of the behavior can be automatically derived from the physical knowledge represented by the model once the physical analysis is performed.

Finally, we should consider the pragmatic facet of the modeling language. Almost as important as the context adaptability provided by the language, is its expressiveness as a measure of its ability to represent the physical knowledge and behaviour in terms of concepts easily recognizable by the modeller (the language pragmatics). This is applicable to both the construction of new modeling components and to the reusability of predefined ones. The potential user of a modeling component must retrieve the represented behaviour before deciding whether is possible to reuse the modeling component for his particular problem. Thus, the facilities provided by the modeling language to interpret the represented behaviour becomes a relevant aspect.

For instance, an equation expresses how the system dynamics can be formulated and computed once the physical context where the system is modeled has been set (e.g. Equations 1.8 and 1.9). However, it has not any explicit relation to the physical knowledge used to formulate it. We may say that the equations are a *implicit* formulation of the physical behaviour and of the physical knowledge used to postulate it. A formalism representing the physical knowledge in an implicit manner can be considered as a static formalism, since it would not permit the analysis of the represented behaviour from a physical point of view.

A dynamic formalism should explicitly represent the required physical knowledge because of two main reasons: to support the physical analysis procedure and also to give to the modeller a clear feedback about the behaviour represented in the modeling components.

In order to summarize previous discussion, here follows a list of the main characteristics demanded to the modeling tool:

- Able to support a structured approach.
- Able to specify the system model at the topological level by means of reusability.
- A modeling language defining a dynamic formalism. The defined formalism should support the explicit representation of the required physical knowledge.

1.3. Problem statement

The main objective of the previous sections has been to set the basis for a methodological approach leading to the automation of the modeling process. We have defined this process as a trajectory through different levels of system specification (see Figure 1.4). At the highest level (topological structure) the system is specified according to the usual way in that engineers look at the system. At the lowest level (causal explanation) the system is specified according to the experimental framework and adequacy targets. At this level the simulator relation must be fulfilled.

The proposed modeling approach is based on the reuse of predefined modeling components as the mean to reduce the modeling burden. In order to extend the reusability of a modeling component, we have proposed the need of including the treatment of possible variations on the experimental framework and the model adequacy.

The modeling trajectory requires of different manipulation procedures in order to move through the system specification levels. As it has been discussed, these manipulations involve both a physical analysis of the specification at the structure levels and a mathematical analysis at the generative levels.

The mathematical analysis basically consists in a computational causality assignment procedure which translates the non causal mathematical formulation of the behaviour into a caused set of equations by means of symbolic manipulation. This problem is solved by well established approaches such as the equation-based object oriented modeling tools and the bond-graph modeling (these approaches will be visited in Chapter 2).

As it has been discussed, the physical analysis procedure should be based on the physical knowledge represented by the modeling components. There are two main questions raising: which is the required physical knowledge?; how do we organize this knowledge?, i.e., what sort of representation structures should be defined in order to embody the required knowledge.

The present work tries to answer these questions by proposing a modeling environment where all the related aspects in the modeling process are taken into account.

Objectives and overview of the proposed solution

This thesis presents a modeling environment able to automate the described modeling process. By automated modeling we understand the capability to generate the adequate causal explanation to the system behaviour within certain experimental framework by means of the analysis of the system specification made at the topological level. The modeling environment is based on the language PML (*Physical Modeling Language*) designed for this purpose.

The modeling trajectory with PML is shown at Figure 1.7. The trajectory starting point is the topological system specification defined by the modeller. The trajectory ending point is a causal explanation of the system behaviour according to the experimental framework and adequacy defined by the modeller. All the steps through the trajectory are performed by the modeling tool named PMT (*Physical Modeling Tool*).

The system specification at the topological level focuses on the behavioral concerns of physical systems. The behavioral concerns refer to physical aspects of the system such as, for instance, its structure and the phenomena occurring at each system component. We introduce the following definition applicable through the rest of the thesis.

Definition 1.9 – *Topological Model*

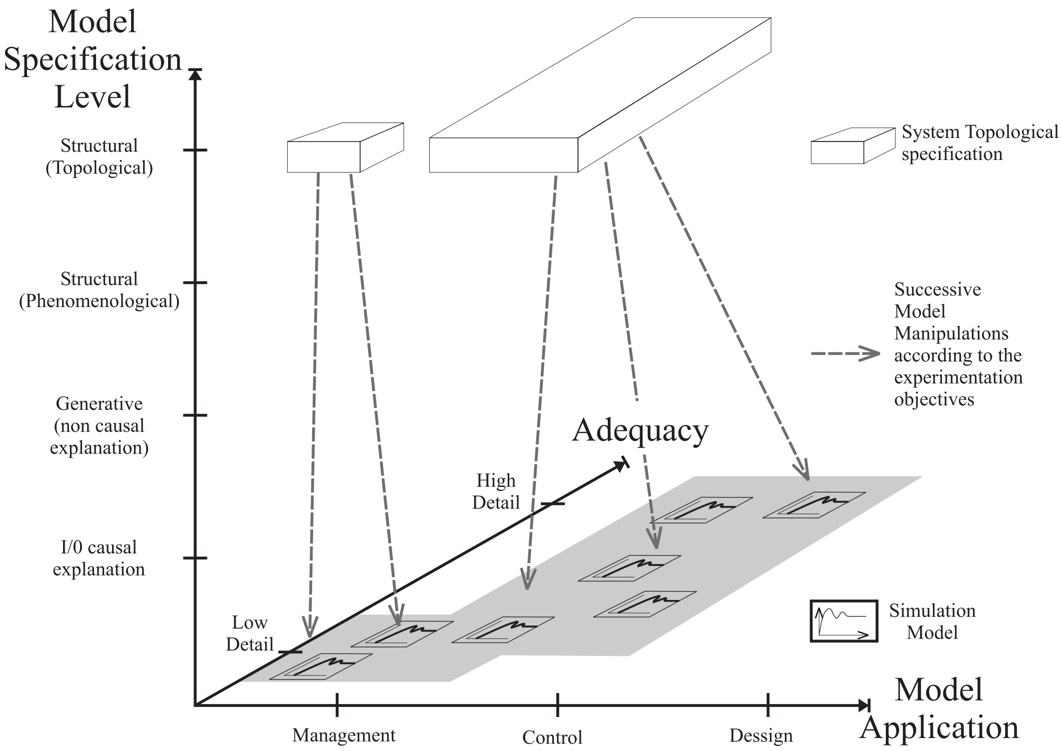

Figure 1.7. The automated modeling process trajectory.

The topological model $\mathcal{M}$ for a system $\mathcal{S}$ is the representation of the system at the topological structure specification level. Hence, $\mathcal{M}$ is a structured representation where the modeling components are coupled according to the connection topology of the system components. This representation should be manageable by a computer in order to obtain the aggregated behaviour. ❑

The topological model construction should be based on the reusability of predefined modeling components according to the system structure. A main factor considered in the design of the modeling environment has been to extend reusability to the features described in 1.2.

The model topology is analyzed from a physical point of view (see Definition 1.5) in order to derive the aggregated behaviour. The experimental framework is taken into account through this step in order to set the expected behaviour. The obtained aggregated behaviour is represented at the phenomenological structure level. This structure represents the set of phenomena described at the topological model together with the physical interactions derived from the matter and energy transfers defined by the topological model. The specification at this level in PML is named as Functional Model.

Definition 1.10 – *Functional Model*

The functional model $\mathcal{M}$ for a system $\mathcal{S}$ is the representation of the system physical behaviour at the phenomenological structure specification level. Hence, $\mathcal{M}$ is a structured representation where the components represent the occurring phenomena (according to the experimental framework) and the connection structure represent the physical interactions. This representation should be manageable by a computer in order to obtain the non causal formulation of the behaviour. ❏

This modular structure is not predefined in the modeling components defined with PML. It is the result of analyzing the physical causality of a model once their reusing context is determined at the topological model.

The next step of the automated modeling process is the formulation of the phenomenological structure by means of non causal equations, i.e., is the mathematical formulation of the aggregated behaviour. It is important to remark that the adequacy of the model can be adapted to the desired level of accuracy by selecting the proper mathematical formulation.

Definition 1.11 – *Mathematical Model*

The mathematical model $\mathcal{M}$ for a system $\mathcal{S}$ is the representation at the non causal specification level of the system physical behaviour which is of interest for some engineering activity. The formulation of $\mathcal{M}$ should be based on a formalism suitable to recreate the system expected behaviour. This representation should be manageable by a computer in order to obtain the causal explanation of the expected behaviour. ❑

The mathematical model is considered as a non causal explanation since , even the behaviour has been formulated, it does not assume which are the causes and which the effects at each physical interaction. This question is solved at the next step where the causal explanation is generated by means of mathematical manipulations (see Definition 1.6).

Definition 1.12 – *Simulation Model*

The simulation model $\mathcal{M}$ for a system $\mathcal{S}$ is the representation of the system physical behaviour which is of interest for some engineering activity. The model $\mathcal{M}$ is the specification of the system physical behaviour in some experimental framework with the stated adequacy. This representation should fulfill the simulator relation in order to answer questions about $\mathcal{S}$. ❑

This definition reflects the model definition coined by Minsky (see Section 1.1): the possible variations of the experimental framework are contemplated in the translation from the topological into the functional model (*... $\mathcal{O}$ can use $\mathcal{M}$ to answer questions that* ***interest*** *him about $\mathcal{S}$*); the model adequacy is contemplated int the translation from the functional into the mathematical model (*... $\mathcal{O}$* ***can use $\mathcal{M}$ to answer questions*** *that interest him about $\mathcal{S}$*).

The PML modeling environment has been designed around the following aspects in order to support the described modeling approach:

The representational framework defined by the PML language has two main purposes. In its pragmatics facet, the language should provide the modeller and the model user with representation structures where the system behaviour is described in terms of concepts familiar to engineers. In its semantics facet, the language embodies the physical knowledge required to make applicable the physical analysis procedure (see Definition 1.5). The formalism defined by the PML language is uncoupled from the mathematical foundation required for the physical behavior computation and responds to the concept of dynamic formalism (see Definition 1.8).

The modeling methodology responds to the object-oriented paradigm (see Chapters 2 and 3), supporting the basic characteristics of **structured** development by providing modular modeling components, **encapsulation** of component behaviour, reliable model construction by means of **aggregation** and knowledge **classification** by means of inheritance mechanism.

The modeling tool PMT which facilitates the interaction with the PML language and facilitates the access to the benefits derived from the object oriented method.

The main idea underlying the architecture of the modeling environment is the definition of different borders between the tasks which can be related to the model development, manipulation and operation. The thesis work has been focused on the model development and manipulation tasks. However, we have taken into account the model operation in the design of both the language PML and the modeling tool PMT. The architecture of the modeling environment is shown in Figure 1.8 and its explanation follows.

The physical behaviour representation concerns and the computational concerns have been separated as a consequence of the distinction between the structure and generative specification levels. The representation of the system behaviour has been treated as a problem of knowledge

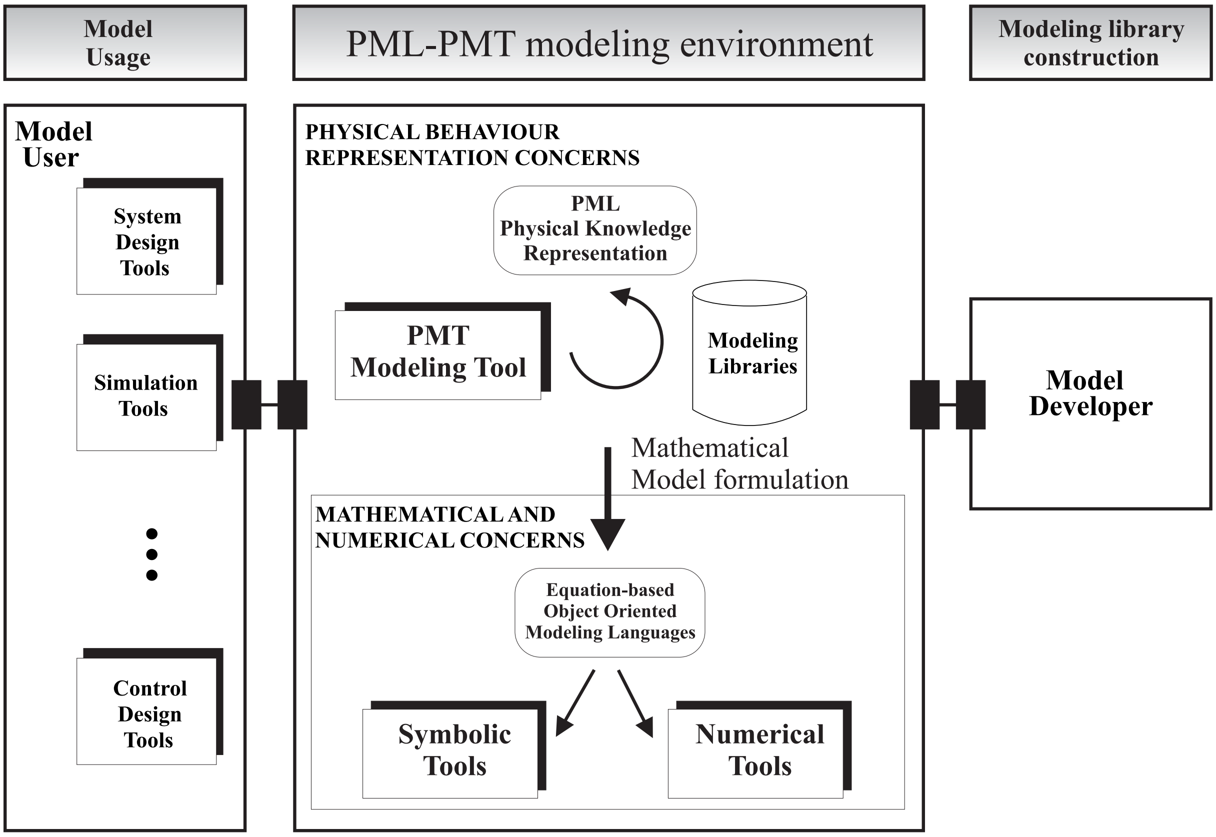

Figure 1.8. Schematic representation of the PML modeling environment architecture.

representation bearing in mind the modeling approach described through this chapter. According to the physical aspects of the modeling activity, the language design has been focused on determining the expressiveness required to declare the physical behaviour and the knowledge needed to interpret it. Two principal benefits are expected to be obtained: on the one hand, the model definition can be done in terms of physical concepts expressed with a natural language familiar to the engineers' way of thinking in systems by means of representation structures which the modeller can easily define and reuse; on the other hand, the modeling tool will be able to analyze the model from a physical point of view when the model computational concerns must be satisfied. The mathematical aspects of the model remains hidden to the user since the translation from the structure level into the generative level is automated by PMT.

A second separation is made between the model development concerns and its operation or usage concerns. Two types of users are distinguished: the user who merely builds models by reusing the predefined ones (by means of aggregation or simply parameterization of existing models), named model user; and the user who will build the model libraries to satisfy the requirements of the model user, named model developer or modeller. Obviously, the interpretation of automated modeling will be different in each case. This aspect will be described in detail in Chapter 2. Model operation or usage is tightly related to the different model based engineering activities. Even the simulation is at the core of most of the engineering activities, the computational model formulation usually has specific characteristics for each purpose. The adequacy of the simulation model to the experimentation purpose will be possible because of the model physical analysis capability.

We have contemplated the possibility to define manipulation rules in order to guide the modeling trajectory according to the engineering activity without intervention of the user. These manipulation rules are closely related to the model usage purposes. The model manipulation responds to the main requirement of automating the generation of the simulation model taking into account the model user experimentation purpose by contemplating the possible hypothesis formulated by the user according to the following main aspects:

- Simplifications to be performed on the model definition in order to formulate computational efficient models. These simplifications will be usually determined by the experimental framework. For instance, all the chemical and thermodynamic phenomena described in a reaction process unit model can be likely neglected if the experimentation goal is to simulate the matter balance in order to design a level controller.

- The representation of certain behaviour according to the detail required for the experimentation. For instance, the generation of a linearized version of a non linear behaviour represented in a model to be used in control design.

Except for certain capabilities given to PMT to prune the system behaviour represented at the topological model (see Chapter 5), the definition and processing of the manipulation rules is left for future research.

The PML-PMT modeling environment has been designed to accomplish with this architecture. PML defines the representational framework for physical modeling, and the modeling tool PMT is an application for manipulating models described by PML.

1.4. Main contributions and scope of the thesis

The topic of this thesis is the design of a modeling environment to represent the physical behaviour and knowledge in an explicit manner, improving the mechanisms which can be used to support a guided and automated modeling process. A main motivation has been the improvement of modeling techniques' reliability and availability, by means of making easy and minimizing the cost of the model development task.

To accomplish with this objective, this thesis introduces a modeling framework where the model development can be automated to a great extent, making this task less expensive and less error prone. The modeling framework is based on the language PML (*Physical modeling Language*) designed to reach a big capability to reuse the physical behaviour and knowledge predefined in modeling components. The modeling components can be reused to build new models, but also to use already defined models for different experimentation purposes, being able to obtain adequate simulation models within different experimental frameworks. The thesis will show how this extension to reusability in PML improves the capabilities to reuse models with respect to other modeling approaches.

The modeling framework reported in this thesis has the following main characteristics:

- With respect the representational framework:
  - The modeling language PML defines a dynamic formalism (see Definition 1.8). In difference with the equation based formalisms, PML introduces a new formalism to extend reusability in order to contemplate the variation of the experimental framework and the adequacy factor.
  - The defined formalism supports the explicit representation of the physical knowledge required to analyze models from a physical point of view.

- With respect the modeling methodology:
  - A structured approach is supported. PML has been designed to fit with the Object Oriented paradigm.
  - The structured models are specified at the topological level by means of reusing predefined modeling components.

Despite of the PML language has been conceived as general modeling purpose language, many of its contributions and features make full sense within the domain of process system modeling. This is because of the main model reuse limitations in a structured approach are due to the exchange of materials and energy among the aggregated modeling components. More concisely, the propagation of the material properties are difficult to predefine at the topological level since they are defined at the phenomenological level, i.e., once the reusing context of every modeling component has been stated and analyzed.

The main contributions of this thesis are summarized in the following points:

- With respect to the PML language pragmatic, the semantics is very closed to the user's physical understanding of the system, which leads to the improvement of the assistance capabilities provided to both the model developer and the model user. The expressiveness of a language can be significantly improved when it is designed for some specific purpose. Such a language, designed for use in a specific field such as mathematics, logic or programming, is said to be a formal language. The PML language set of linguistic symbols allows an unambiguous identification of the physical concepts involved in the modeling process. Two main achievements can be reported:

  i) the representational framework permits models to be expressed in terms of the knowledge which the user has about the system, and

  ii) relevant physical knowledge can be coded to analyze the model from a physical point of view. Hence, the modeling tool can use this knowledge to automate the modeling task and derive the appropriate simulation model.

- The PML topological model, as system specification, is loosely coupled to the experimentation activity. The model-experiment relationship appears in the modeling process because the topological model manipulation is guided by the experimentation goals.

- The model reusability has been extended by supporting a computer-aided modeling process to cope with different simulation or experimentation purposes.

- The capability to adapt a model according to different experimental frameworks is a consequence of the possibility to use the physical knowledge represented at the modeling components in the topological model in order to set the proper functional model. This feature has been defined as *dynamic modularity* (see Chapter 3).

- The separation between the phenomenological specification (the functional model) and its mathematical formulation (the mathematical model) permits the variation of the simulation model adequacy at model user demand. This feature has been defined as *dynamic binding* (see Chapters 4 and 5).

1.5. Outline of the thesis

This thesis has been organized in six chapters. The first one has given a brief overview of the concerns of the work. Chapter 2 is also an introductory chapter where the main modeling methodologies are visited, focusing the discussion in the object oriented modeling method. The main characteristics of the PML language are outlined. Chapter 3 analyzes with more detail the limitations of the declaration of mathematical models as class structures from the reusability point of view. The need of real modular software structures capable to support the system behaviour description independently from its reusing context is introduced. The final part of the chapter presents the basic modeling objects in PML and introduces the procedure used to translate the model defined by the user into the computational model required for simulation engines. Chapter 4 presents *Physic modeling Language* (PML), the object oriented modeling language designed to support modular structures where the problem of representing the physical behaviour and the problem of finding a suitable formalism for the simulation procedure are separated. Chapter 5 describes how the modeling trajectory is performed. The physical analysis procedure performed to translate the topological into the functional model is also presented. The main contributions of the PML language are explored by means of several examples. Finally, in Chapter 6, the conclusions of this thesis are discussed. The present state of the PML modeling environment is summarized and the future work is outlined.