A unified representation for networked dynamical system modelling

Thierry Bastogne

A unified representation for networked dynamical system modelling

submitted to Simulation Modelling - Practice and Theory

T. Bastogne

Centre de Recherche en Automatique de Nancy (CRAN),

Nancy-Université, CNRS,

BP 239, F-54506 Vandœuvre-lès-Nancy Cedex, France,

Phone: (33) 3 83 68 44 73 - Fax: (33) 3 83 68 44 62

thierry.bastogne@cran.uhp-nancy.fr

May 29, 2007

Abstract

A unified description of networked dynamical systems, entitled *multiport diagram*, is proposed in this paper. The multiport diagram is an object-oriented model structure described into a general mathematical framework, independently from any computer language. It is shown that usual diagrammatic representations of networks, *e.g.* block diagrams, signal flow graphs, compartmental networks and bond graphs, are subclasses of the multiport diagram. By bridging the gap between these four network representations, this unified diagram should contribute to facilitate education of networked system modelling in systems and control theories. [1]

Keywords

System modelling, networked dynamical systems, object-modelling techniques, mathematical models, block diagrams, signal flow graphs, compartmental networks, bond graphs.

1 Introduction

In recent years the analysis and modelling of networked dynamical systems have been the subject of considerable interdisciplinary interest in physics, mathematics, computer science, systems engineering, biology, economics, and sociology. Since Kirchhoff's [19] and Firestone's [13] works, several formalisms, located at the intersection of graph theory on one hand, and systems theory and control on the other, have been developed for the analysis and modelling of networked dynamical systems. Three main classes of network representations may be considered:

[1] Simulation Modeling Practices and Theory, doi: 10.1016/j.simpat.2007.04.001

- the signal-based formalisms, *e.g.* block diagrams and signal flow graphs;
- the mass-based formalisms, *e.g.* compartmental networks;
- the energy-based formalisms, *e.g.* bond graphs.

Block diagrams associated with the notion of transfer function have been developed in the 40s for the control design of composite systems [6]. A block within a block diagram defines a dynamical system in itself. The relationships between each elementary dynamical system in a block diagram are illustrated by the use of signals connecting the blocks. Collectively the blocks and lines in a block diagram describe an overall dynamical system [15, 11]. This formalism is still used by computer aided modelling softwares such as Simulink© or Scicos/Scilab©.

Signal flow graphs are close in spirit to block diagrams. In a signal flow graph, signals are denoted by nodes and the transfer functions by branches between nodes. Originally devised by Mason for linear networks [21], they are a mainstay of network theory and are commonly applied to areas as diverse as automatic control and data communications.

Frequently used in biology and mathematics for modelling transportation and diffusion phenomena, compartmental networks are directed graphs, in which the nodes are conceptual storage tanks called compartments [17, 5, 1]. A compartment is an amount of a certain matter that acts kinetically like a distinct, homogenous, well-mixed amount of that matter. Each oriented arc represents a mass transfer which may hold for various transport, transformation or interaction phenomena between the species inside the system. Each arc is labeled with the fractional transfer coefficient, which is the fraction of the compartment from which the arrow originates, transferred per unit time. Compartmental networks are mass conservative and positive systems.

Bond graphs were introduced in 1959 by Paynter [23, 24] and developed by Karnopp [18] and Rosenberg [26]. Bond graphs are labelled and directed graphs, in which the nodes (elements) represent submodels and the arcs (bonds) represent an ideal energy point-to-point connection between the elements. The arcs or edges are idealised descriptions of physical phenomena with respect to energy, *e.g.* storage, irreversible and reversible transformations, distribution, supply and demand. Bond graphs rely on basic principles of physics, *i.e.* energy conservation, positive entropy production and power continuity.

In parallel, in computer science, the concept of objects and instances in computing had its first major breakthrough in the 60s with Simula 67, the first object-oriented language [16]. This led to the design of object-oriented languages developed for complex systems modelling [12, 10, 22, 20, 7, 27, 9, 2]. Despite these efforts, it seems that main aspects of the object-orientation developed initially in computer science are not fully appreciated in the systems science community [8]. Such modelling languages are often proprietary languages. Their frequent evolutions can lead to the impossibility to perpetuate and capitalize on the modelling effort. This reason has motivated the specification of the object-modelling paradigm into a mathematical framework [4] by using the behavioral approach of the systems theory proposed by

Willems [29, 30].

A few papers have proposed to relate these different formalisms, particularly the bond graphs and some object-oriented languages like Modelica in [8] or VHDL [25]. Unfortunately, up to now, there is no unified framework in which differences and relations between these diagrammatic representations could be globally stressed instead of comparing them two by two. Moreover, these comparative studies require a minimum knowledge about object-oriented modelling languages. The contributions of this paper are twofold:

- a unified object-oriented diagram, entitled *multiport diagram*, is proposed. The objective is to show that block diagrams, signal flow graph, compartmental networks and bond graphs are particular cases of the multiport diagram;
- the object-oriented paradigm is described into a mathematical framework which makes it independent from any computer language.

Section 2 introduces notations, definitions and conjectures about object-oriented models which will be used in the sequel. Clarifications and illustrations of the object-oriented framework for dynamical systems modelling are developed in [4]. The multiport diagram is defined in Section 3. In Sections 4, 5, 6 and 7, block diagrams, signal flow graphs, compartmental networks and bond graphs are described as subclasses of the multiport diagram. A synthesis is provided in Section 8.

2 Behavioral representations and relationships of the object-orientation

This section introduces two mathematical representations of an object. These representations, entitled *complete* and *partial behavioral representations*, are used in the sequel to mathematically define object-oriented model structures, and so independently from any computer language. They are inspired from the behavioral approach of the systems theory, introduced by J.C. Willems in 1986 [29, 30]. Basic notations used in this paper are presented in Table 1.

2.1 Complete and partial behavioral representations of an object

An object is generally defined in computer science as a structure encapsulating data (state) and data evolution (behavior). In case of dynamic systems, these two attributes can be described by two sets: a set of variables and a set of time trajectories. This perception is closed to the concepts of universal and behavioral sets introduced by Willems in [29] and leads to a first behavioral definition of an object.

Definition 2.1 *The* ***complete behavioral representation*** *of an object class* $\mathbf{O}$ *is defined as follows*

$$\mathbf{O} = (\mathcal{U}^*_{\mathbf{O}}, \mathcal{B}^*_{\mathbf{O}}), \tag{1}$$

where

Table 1: Basic notation of sets

Ref.	Description
$\mathbf{D}$	Object class
D	Instance (object)
$\mathcal{B}$	Behavioral set
$\mathcal{C}$	Composition set
$\mathcal{H}$	Generalization set
$\mathcal{L}$	Internal variable set
$\mathcal{P}$	Parameter set
$\mathcal{U}$	Universal set
$\mathcal{W}$	External variable set
$\mathbb{C}$	Set of complexes
$\mathbb{N}$	Set of integers
$\mathbb{R}$	Set of reals

- $\mathcal{U}^*_{\mathbf{O}}$ *is the universal set of* $\mathbf{O}$, i.e. *the set which contains all the elements involved in* $\mathbf{O}$. $\mathcal{U}^*_{\mathbf{O}}$ *is defined by*

$$\begin{aligned}\mathcal{U}^*_{\mathbf{O}} &= \{t \in \mathcal{T}_{\mathbf{O}}, p^* \in \mathcal{P}^*_{\mathbf{O}}, l^*(t) \in \mathcal{L}^*_{\mathbf{O}}, w^*(t) \in \mathcal{W}^*_{\mathbf{O}}\} \\ &= \mathcal{T}_{\mathbf{O}} \times \mathcal{P}^*_{\mathbf{O}} \times \mathcal{L}^*_{\mathbf{O}} \times \mathcal{W}^*_{\mathbf{O}}, \end{aligned} \tag{2}$$

  where $\times$ *is the Cartesian product,* $\mathcal{T}_{\mathbf{O}}$ *the time axis,* $\mathcal{P}^*_{\mathbf{O}}$ *the parameter set,* $\mathcal{W}^*_{\mathbf{O}}$ *the external variable set and* $\mathcal{L}^*_{\mathbf{O}}$ *the internal variable set.* $t \in \mathcal{T}_{\mathbf{O}}$ *denotes the time variable,* $p^* \in \mathcal{P}^*_{\mathbf{O}}$ *: the vector of parameters,* $w^*(t) \in \mathcal{W}^*_{\mathbf{O}}$*: the vector of external variables and* $l^*(t) \in \mathcal{L}^*_{\mathbf{O}}$*: the vector of internal variables. A parameter is defined as a particular variable which is kept constant during the time range* $\mathcal{T}_{\mathbf{O}}$.

- $\mathcal{B}^*_{\mathbf{O}}$ *is the behavioral model,* i.e. *a non empty subset of* $\mathcal{U}^*_{\mathbf{O}}$ *which contains the possible values of* $t, p^*, l^*(t), w^*(t)$ *satisfying a set of constraint equations. The complete behavioral model of an object is defined by*

$$\mathcal{B}^*_{\mathbf{O}} = \left\{ t, p^*, l^*(t), w^*(t) \,\middle|\, f(t, p^*, l^*, w^*) = 0 \right\}, \tag{3}$$

  where $f(\cdot)$ *is an implicit system of behavioral equations. When output variables* $y^*(t)$ *are explicit, with* $y^*(t) \subset w^*(t)$*, then* $f(\cdot)$ *can be replaced by an explicit system* $g(\cdot)$ *of behavioral equations,* e.g. $y^*(t) = g(t, p^*, l^*, w^*)$.

Two object classes, $\mathbf{A}$ and $\mathbf{B}$, are equal if their complete behavioral models, $\mathcal{B}^*_{\mathbf{A}}$ and $\mathcal{B}^*_{\mathbf{B}}$ are identifical. A major difference between objects used at the origin in computer science and those used for physical systems modelling is that contrary to conventional objects, physical phenomena are associated with a

temporal semantics, physical systems are dynamical systems. Consequently, the universal set $\mathcal{U}^*_{\mathbf{O}}$ includes a time axis.

In object-orientation, the principal utility of the encapsulation process remains the privatization of the access to the data. Indeed, the concept of object also makes it possible to legalise and limit the access of a limited number of variables entitled external variables. They enable the object to communicate with its environment. Internal variables correspond either to state variables or to algebraic variables.

The object-orientation is based on the notion of class. A class is a paradigm defining the elements and behaviour for a particular type of object. Any object designed from this paradigm is an *instance* of this class. Instances are the representatives of the object classes in the model. Classes are arranged into specialization-generalization hierarchies, subclasses provide specialised behavior, whereas super-classes are more generic. Any object class can be reused to compose new objects. The taking into account of those instantiation, composition and generalization relationships leads to a second behavioral representation.

Definition 2.2 *The **partial behavioral representation** of an object class* $\mathbf{O}$ *is defined by*

$$\mathbf{O} = (\mathcal{C}_{\mathbf{O}}, \mathcal{H}_{\mathbf{O}}, \mathcal{U}_{\mathbf{O}}, \mathcal{B}_{\mathbf{O}}), \tag{4}$$

where

- $\mathcal{C}_{\mathbf{O}}$ *denotes the composition set.* $\mathcal{C}_{\mathbf{O}} = \{\mathbf{A}\}$ *means that* $\mathbf{O}$ *is composed of* $\mathbf{A}$*.*
- $\mathcal{H}_{\mathbf{O}}$ *is generalization (inheritance) set.* $\mathcal{H}_{\mathbf{O}} = \{\mathbf{A}\}$ *implies that* $\mathbf{A}$ *is a super-classes of* $\mathbf{O}$*, or* $\mathbf{O}$ *is a subclass of* $\mathbf{A}$*.*
- $\mathcal{U}_{\mathbf{O}}$ *is the partial universal set containing elements exclusively belonging to* $\mathbf{O}$*. In other words, variables of elements defined in* $\mathcal{C}_{\mathbf{O}}$ *and* $\mathcal{H}_{\mathbf{O}}$ *are not specified in* $\mathcal{U}_{\mathbf{O}}$*. Consequently* $\mathcal{U}_{\mathbf{O}} \subset \mathcal{U}^*_{\mathbf{O}}$*.* $\mathcal{U}_{\mathbf{O}}$ *is defined by*

$$\begin{aligned} \mathcal{U}_{\mathbf{O}} &= \{t \in \mathcal{T}_{\mathbf{O}}, p \in \mathcal{P}_{\mathbf{O}}, l(t) \in \mathcal{L}_{\mathbf{O}}, w(t) \in \mathcal{W}_{\mathbf{O}}\} \\ &= \mathcal{T}_{\mathbf{O}} \times \mathcal{P}_{\mathbf{O}} \times \mathcal{L}_{\mathbf{O}} \times \mathcal{W}_{\mathbf{O}}. \end{aligned} \tag{5}$$

  $\mathcal{P}_{\mathbf{O}} \subset \mathcal{P}^*_{\mathbf{O}}$ *is the partial set of parameters of* $\mathbf{O}$*,* $\mathcal{W}_{\mathbf{O}} \subset \mathcal{W}^*_{\mathbf{O}}$ *is the partial set of external variables and* $\mathcal{L}_{\mathbf{O}} \subset \mathcal{L}^*_{\mathbf{O}}$ *is the partial set of internal variables. Vectors* $p, w(t), l(t)$ *are sub-vectors of* $p^*, w^*(t), l^*(t)$*.*

- $\mathcal{B}_{\mathbf{O}}$ *is the partial behavioral model of* $\mathbf{O}$ *defined by*

$$\mathcal{B}_{\mathbf{O}} = \left\{ t, p, l(t), w(t) \,\middle|\, \exists (p', l', w') \text{ with } f(t, p, p', l, l', w, w') = 0 \right\}, \tag{6}$$

  where $p^* = (p, p')$*,* $l^* = (l, l')$ *and* $w^* = (w, w')$*.* p'*,* l' *and* w' *are parameters and variables defined in components and super-classes of* $\mathbf{O}$*.*

The partial representation relies on the definition of four sets instead of two for the complete representation. In other words, a complete representation of an object is a partial representation for which

the sets $\mathcal{C}_{\mathbf{O}}$ and $\mathcal{H}_{\mathbf{O}}$ are empty. A complete representation has for advantage to completely define an object independently of all other objects. On the other hand, the interest of a partial representation is to simplify the definition of an object by specifying only its own characteristics and by not repeating the common points that it shares with existing objects. The time axis of a networked system and the ones of its components are assumed to be the same. Accordingly, for simplicity the time axis $\mathcal{T}_{\mathbf{O}}$ is removed in the sequel.

Assumption 2.1 *In the sequel, it is assumed that any object class* $\mathbf{O}$ *is globally and structurally identifiable. Structural identifiability [28] deals with the possibility to give a unique value to each parameter of a mathematical model structure. The uniqueness of this solution is assessed in an idealized or theoretical framework where the system and the model have identical model structures, the data are noise-free, and where the input signals and the measurement times can be chosen at will. In these conditions and given an initial state vector* x_0*, a parameter* p_i^* *of* p^* *is globally identifiable for almost any* $\theta^* \in \mathcal{P}$ *if*

$$\mathcal{B}^*_{\mathbf{O}}(p^*, x_0) \equiv \mathcal{B}^*_{\mathbf{O}}(\theta^*, x_0) \Leftrightarrow p_i^* = \theta_i^* \quad \forall i \in \{1, \cdots, n_p\}, \tag{7}$$

where n_p *denotes the number of parameters. An object class* $\mathbf{O}$ *is globally and structurally identifiable if and only if all its parameters are globally identifiable.*

In [14], a differential algebra approach is developed to examine identifiability of object-oriented models.

2.2 Instantiation relationship

Any object designed from the paradigm of a given object class is an *instance* of this class. The instantiation process usually involves setting parameters and sometimes initial values of state variables of the object. The instantiation relationships is represented by the colon symbol (':').

Definition 2.3 *An* ***instance*** O *of an object class* $\mathbf{O}$ *is defined as follows*

$$O : \mathbf{O}(p^* = p_0, x(0) = x_0), \tag{8}$$

where $x(t) \subset l(t)$ *is the vector of state variables, a subvector of the internal variables.*

Conjecture 2.1 *The instantiation relationship between an object class* $\mathbf{O}$ *and its instance* A *is noted* $A : \mathbf{O}$ *and satisfies*

$$\mathcal{U}_A = \mathcal{U}_{\mathbf{O}} \tag{9}$$

$$\mathcal{B}_A = \mathcal{B}_{\mathbf{O}} \tag{10}$$

$$\mathcal{C}_A = \mathcal{C}_{\mathbf{O}} \tag{11}$$

$$\mathcal{H}_A = \mathcal{H}_{\mathbf{O}}. \tag{12}$$

In other terms, a class and its instances are identical by their form and their behavior. However their parameters and their variables generally contain different values.

2.3 Composition and generalization relationships

Let $\mathbf{O}$ be an object class and $\mathbf{A}$ a super-class ($\mathcal{H}_{\mathbf{O}} = \{\mathbf{A}\}$ with $\mathbf{A} \neq \mathbf{O}$) or a component of $\mathbf{O}$ ($\mathcal{C}_{\mathbf{O}} = \{\mathbf{A}\}$ with $\mathcal{C}_{\mathbf{O}} = \{\mathbf{O}\} \overset{def}{\Longrightarrow} \mathcal{C}_{\mathbf{O}} = \{\emptyset\}$). Let $(\mathcal{U}^*_{\mathbf{O}}, \mathcal{B}^*_{\mathbf{O}})$ be the complete representation of $\mathbf{O}$ with $\mathcal{U}^*_{\mathbf{O}} = \mathcal{P}^*_{\mathbf{O}} \times \mathcal{W}^*_{\mathbf{O}} \times \mathcal{L}^*_{\mathbf{O}}$.

Conjecture 2.2 *Although different by nature, the composition and generalization relationships imply that parameters and variables of* $\mathbf{A}$ *are added to those of* $\mathbf{O}$ *to build up augmented spaces:*

$$\mathcal{P}^*_{\mathbf{O}} = \mathcal{P}_{\mathbf{O}} \times \mathcal{P}^*_{\mathbf{A}} \tag{13}$$

$$\mathcal{W}^*_{\mathbf{O}} = \mathcal{W}_{\mathbf{O}} \times \mathcal{W}^*_{\mathbf{A}} \tag{14}$$

$$\mathcal{L}^*_{\mathbf{O}} = \mathcal{L}_{\mathbf{O}} \times \mathcal{L}^*_{\mathbf{A}}. \tag{15}$$

The complete behavioral model is then given by

$$\mathbb{B}^*_{\mathbf{O}} = \mathbb{B}_{\mathbf{O}} \cap \mathbb{B}^*_{\mathbf{A}}. \tag{16}$$

In other terms, the complete behavior of $\mathbf{O}$ can be regarded as the intersection $\mathbb{B}_{\mathbf{O}}$ and $\mathbb{B}^*_{\mathbf{A}}$ in the universal set $\mathcal{U}^*_{\mathbf{O}}$. From an algebraic point of view, the equation (16) implies that the behavioral equation system of $\mathbf{O}$ is augmented by the behavioral equations of $\mathbf{A}$.

Conjecture 2.3 *Given definitions 2.1 and 2.2, and conjectures 2.1 and 2.2, any partial behavioral representation of an object class* $\mathbf{O}$ *may be transformed into a complete representation if the behavioral representations of its components and super-classes are known.*

3 Multiport diagram

This section introduces an object-oriented model structure entitled *Multiport Diagram*. Figure 1 describes the structure of the Multiport diagram by a UML class diagram. It is composed of modules connected by links through ports. Modules and links have both behavioral models which can be either black-box models or first-principle models, *e.g.* balance equations for conservative system modelling.

Definition 3.1 *A* ***Multiport diagram*** Δ *is a model structure composed of* modules *connected by* links. *A multiport diagram is itself an object class and its partial behavioral representation is defined as follows*

$$\Delta = (\mathcal{C}_\Delta, \mathcal{H}_\Delta, \mathcal{U}_\Delta, \mathcal{B}_\Delta) \tag{17}$$

with

$$\mathcal{C}_\Delta = \{\mathcal{M}, \mathcal{X}\} \tag{18}$$

$$\mathcal{H}_\Delta = \mathcal{U}_\Delta = \mathcal{B}_\Delta = \{\emptyset\} \tag{19}$$

$\mathcal{M} = \{M_1, \cdots, M_m\}$ *is a set of module instances and* $\mathcal{X} = \{L_1, \cdots, L_n\}$ *a set of link instances,* i.e. *a set of connections between the modules.* m *and* n *are the number of modules and links in* Δ. *Note that modules are equipped with ports to be connected by links.*

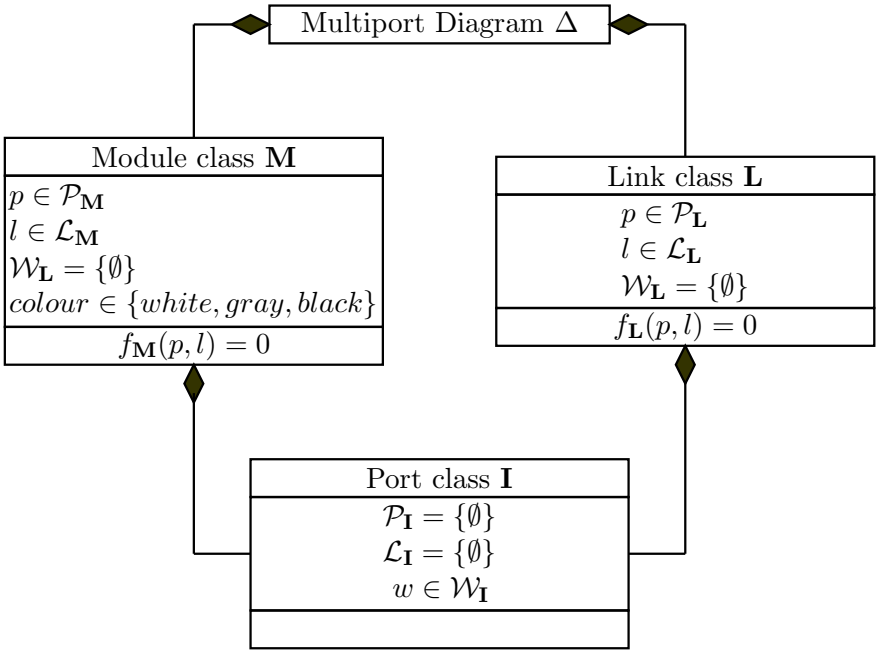

Figure 1: UML class diagram of the Multiport diagram

3.1 Module

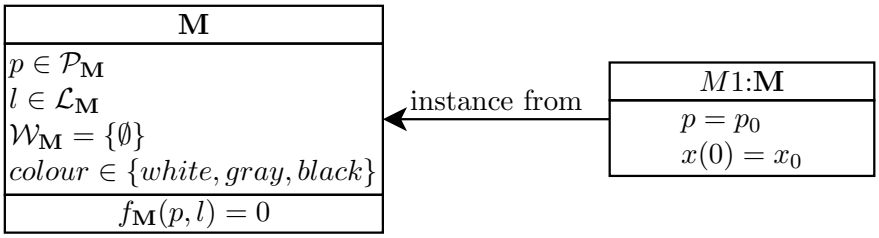

Figure 2: Module class and module instance

Definition 3.2 *A **module** $\mathbf{M}$ is an object-class whose instances are associated with components or entities of the networked system. Its partial behavioral representation is given by*

$$\mathbf{M} = (\mathcal{C}_\mathbf{M}, \mathcal{H}_\mathbf{M}, \mathcal{U}_\mathbf{M}, \mathcal{B}_\mathbf{M}) \tag{20}$$

$$\mathcal{C}_\mathbf{M} = \{list\,of\,sub-modules\,which\,compose\,\mathbf{M}\} \tag{21}$$

$$\mathcal{H}_\mathbf{M} = \{list\,of\,superclasses\,of\,\mathbf{M}\} \tag{22}$$

$$\mathcal{U}_\mathbf{M} = \{p \in \mathcal{P}_\mathbf{M}, l(t) \in \mathcal{L}_\mathbf{M}\} \tag{23}$$

$$\mathcal{B}_\mathbf{M} = \left\{ p, l(t) \,\middle|\, \exists (p', l') \;\; with \;\; f_\mathbf{M}(t, p, p', l, l') = 0 \right\} \tag{24}$$

where p and $l(t)$ denote the parameters and internal variables of the module. $x(0)$ is the initial state vector with $x(t) \subset l(t)$. The dynamical behavior of the module is described by an implicit system $f_\mathbf{M}(\cdot)$. (p', l') are parameters and internal variables defined in sub-modules and superclasses of $\mathbf{M}$.

The graphic notation of a module is given in Figure 2. As indicated in equation (24), a module has no external variable $\mathcal{W}_\mathbf{M} = \{\emptyset\}$. It means that a module needs ports to communicate with its environment.

3.2 Ports

A *port* $\mathbf{I}$ is a communication interface associated with a module, *i.e.* an ordered set of external variables. There is no standard or limited description of interfaces in the object-modelling paradigm. Personal interfaces can be freely developed by the user. However, as suggested in [10], '*modelers are advised not to define the interfaces of their models arbitrarily, but to restrain themselves, and only use proven connection mechanisms*'. Three port classes are proposed herein: the power interfaces ($\mathbf{I}_P$), the material interfaces ($\mathbf{I}_M$) and the signal interfaces ($\mathbf{I}_S$). A port has no parameter ($\mathcal{P}^*_{\mathbf{I}} = \{\emptyset\}$), no internal variable ($\mathcal{L}^*_{\mathbf{I}} = \{\emptyset\}$) and no behavioral equation ($\mathcal{B}^*_{\mathbf{I}} = \mathcal{U}^*_{\mathbf{I}}$). It only contains external variables which makes it possible for a module to exchange energy, material and information with other modules of the network.

Definition 3.3 *A* power port*, noted* $\mathbf{I}_P$*, is an ordered pair of* across/through *variables* (α, φ) *based on the Firestone's analogy [13]. Its complete behavioral representation is limited to the definition of the external variable set*

$$\mathcal{W}^*_{\mathbf{I}_P} = \{(\alpha, \varphi)\} = \mathbb{R}^2. \tag{25}$$

where α *and* φ *denote the* across *and* through *variables. As illustrated in Figure 3, the icon of a power port is a white disc. The power flow* $P(t)$ *associated with such a port is given by* $P(t) = \alpha(t) \cdot \varphi(t)$*. By convention, the positive flow of* through *variables is oriented into the module. This convention is used to establish power balance equations in each module.*

In Figure 3, $M1.I1 : \mathbf{I}_P$ and $M2.I1 : \mathbf{I}_P$ are two power ports associated with the modules $M1$ and $M2$ respectively.

Definition 3.4 ***Material ports****, noted* $\mathbf{I}_M$*, define material transfer rates. As previously, its complete behavioral representation is limited to the definition of the external variable set*

$$\mathcal{W}^*_{\mathbf{I}_M} = \{(x, \varphi)\} = \mathbb{R}^2 \tag{26}$$

where φ *denotes a* through *variable and* x *is the amount of material transiting through the port. By convention, the positive flow of* through *variables is oriented into the module. This convention is used to establish mass balance equations in each module. The icon of a material port is a black square*

In Figure 3, $M2.I2 : \mathbf{I}_M$ and $M3.I2 : \mathbf{I}_M$ are two material ports associated with the modules $M2$ and $M3$ respectively.

Definition 3.5 ***Input*** *and* ***output signal ports****, noted* $\mathbf{I}_{S+}$ *and* $\mathbf{I}_{S-}$*, send/receive input/output signals. Their external variable sets are given by*

$$\mathcal{W}^*_{\mathbf{I}_{S+}} = \{u\} = \mathbb{R}^r \tag{27}$$

$$\mathcal{W}^*_{\mathbf{I}_{S-}} = \{y\} = \mathbb{R}^q \tag{28}$$

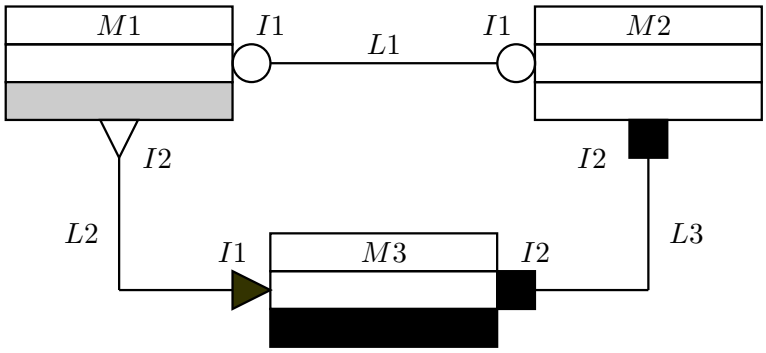

Figure 3: Interconnection of modules

where u and y denote input and output vectors of size r and q respectively. The icons of input and output signal ports are a black and a white arrow.

Figure 3 shows an input signal port $M3.I1 : \mathbf{I}_{S+}$ and an output signal port $M1.I2 : \mathbf{I}_{S-}$.

3.3 Links

Definition 3.6 *A **link** $\mathbf{L}$ is an object class which describes the connexion mode,* i.e. *the interconnection equations, between two modules. A link has to be connected to a module by a port. Similarly to the ports, three classes of links have been defined, entitled* power, material *and* signal link, *noted $\mathbf{L}_P$, $\mathbf{L}_M$ and $\mathbf{L}_S$*

respectively. Their relative behavioral representations are given by

$$\mathbf{L}_P = (\mathcal{C}_{\mathbf{L}_P}, \mathcal{H}_{\mathbf{L}_P}, \mathcal{U}_{\mathbf{L}_P}, \mathcal{B}_{\mathbf{L}_P}) \tag{29}$$

$$\mathcal{C}_{\mathbf{L}_P} = \{I1, I2 : \mathbf{I}_P\}$$

$$\mathcal{H}_{\mathbf{L}_P} = \{\emptyset\}$$

$$\mathcal{U}_{\mathbf{L}_P} = \{\emptyset\}$$

$$\mathcal{B}_{\mathbf{L}_P} = \left\{ \exists \begin{pmatrix} I1.\alpha(t) \\ I1.\varphi(t) \\ I2.\alpha(t) \\ I2.\varphi(t) \end{pmatrix} \middle| \begin{bmatrix} I1.\alpha(t) = I2.\alpha(t) \\ I1.\varphi(t) + I2.\varphi(t) = 0 \end{bmatrix} \right\} \tag{30}$$

$$\mathbf{L}_M = (\mathcal{C}_{\mathbf{L}_M}, \mathcal{H}_{\mathbf{L}_M}, \mathcal{U}_{\mathbf{L}_M}, \mathcal{B}_{\mathbf{L}_M}) \tag{31}$$

$$\mathcal{C}_{\mathbf{L}_M} = \{I1, I2 : \mathbf{I}_M\}$$

$$\mathcal{H}_{\mathbf{L}_M} = \{\emptyset\}$$

$$\mathcal{U}_{\mathbf{L}_M} = \{\emptyset\}$$

$$\mathcal{B}_{\mathbf{L}_M} = \left\{ \exists \begin{pmatrix} I1.\varphi(t) \\ I2.\varphi(t) \end{pmatrix} \middle| \begin{bmatrix} I1.\varphi(t) + I2.\varphi(t) = 0 \\ I1.\varphi(t) = f(I1.x(t), I2.x(t)) \end{bmatrix} \right\} \tag{32}$$

$$\mathbf{L}_S = (\mathcal{C}_{\mathbf{L}_S}, \mathcal{H}_{\mathbf{L}_S}, \mathcal{U}_{\mathbf{L}_S}, \mathcal{B}_{\mathbf{L}_S}) \tag{33}$$

$$\mathcal{C}_{\mathbf{L}_S} = \{I1 : \mathbf{I}_{S+}, I2 : \mathbf{I}_{S-}\}$$

$$\mathcal{H}_{\mathbf{L}_S} = \{\emptyset\}$$

$$\mathcal{U}_{\mathbf{L}_S} = \{\emptyset\}$$

$$\mathcal{B}_{\mathbf{L}_S} = \left\{ \exists \begin{pmatrix} I1.u(t) \\ I2.y(t) \end{pmatrix} \middle| I1.u(t) = I2.y(t) \right\} \tag{34}$$

Each class of link is composed of two ports of the same class, *e.g.* a power link is composed of two power ports. The behavioral model of the power links, described in equation (30), corresponds to the generalized Kirchhoff laws applied to the across/through variables of the power ports. The equation of the material links is a generalization of the Kirchhoff's current law, expressing mass continuity. The first equation $P1.\varphi + P2.\varphi = 0$ describes the mass continuity in the material link. The second one $P1.\varphi = p \cdot f(P1.x, P2.x)$ defines the relationship between the matter flow φ and the amount of matter in the two connected modules. The behavioral model of the signal ports, defined in equation (34), is a causal assignment between the output and input variables. Figure 3 shows three link instances $L1 : \mathbf{L}_P$, $L2 : \mathbf{L}_S$ and $L3 : \mathbf{L}_M$.

3.4 Behavioral representation of the Multiport diagram

According to the definition of the multiport diagram, the latter is only composed of modules, ports and links. Consequently, if every module, port and link of the diagram is defined, *i.e.* if its behavioral representation is known, then the complete behavioral representation of the Multiport diagram can be obtained by applying the conjectures 2.1 and 2.2. In practice, that generally leads to create a large system of differential algebraic equations to solve in the simulation step.

4 Block diagrams

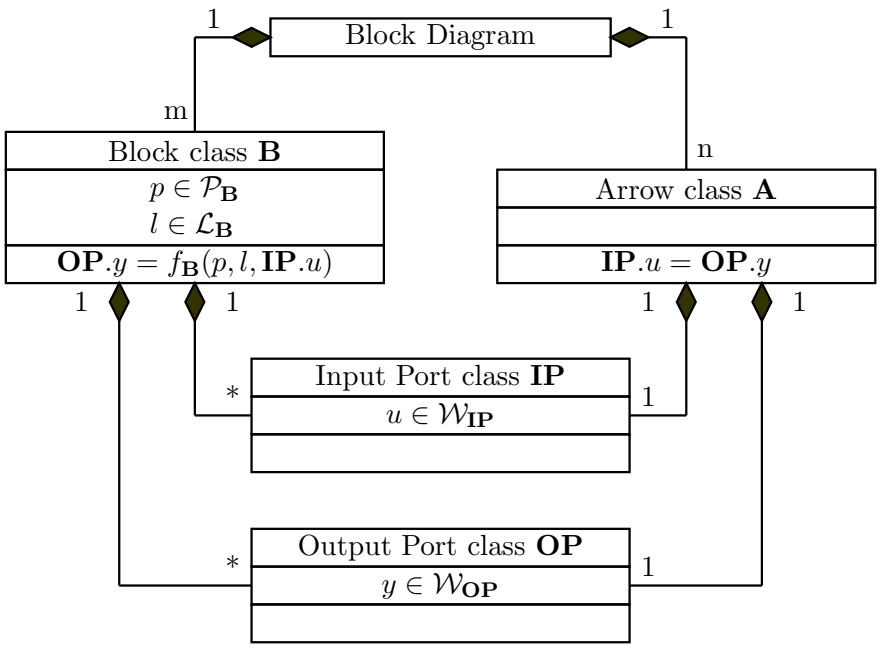

Figure 4: Meta-model of a block diagram

Figure 4 shows the UML class diagram of a block diagram. The latter is decomposed into m blocks **B** and n arrows **A**. Blocks are equipped with input/output ports **IP**/**OP** by which they can exchange information with other blocks. The number of ports is variable according to the blocks while an arrow has exactly one input port and one output port. An arrow is an oriented link between an output variable and an input variable defined by a variable assignment $\mathbf{IP}.u = \mathbf{OP}.y$. Each block encapsulates parameters p, internal variables l and a behavioral model. This model is mainly composed of a set of explicit equations gathered in $f_{\mathbf{B}}(\cdot)$, *e.g.* transfer function or state-space representations, which completely describe the input-output relationships of the block. By comparing figures 1 and 4, it appears that the block diagram is a particular case of multiport diagram, *i.e.* an incomplete version which is not provided with material and power ports. Blocks, input/output ports and arrows are subclasses of modules, input/output signal ports and signal links. Note also that the behavioral equations of a block are explicit while the ones of a module may be implicit.

Example 4.1 *Figure 5 presents the block diagram of a closed-loop system. The dynamical behavior of the*

system S is represented by a second-order transfer function where s denotes the differentiation operator, i.e. $s \cdot u(t) \stackrel{def}{=} du/dt$. *The control system is composed of a comparator C and a proportional controller PC. The comparator forms the error signal e between the reference signal r and the output signal y. The control signal u is produced by the controller. Its gain k is a parameter.*

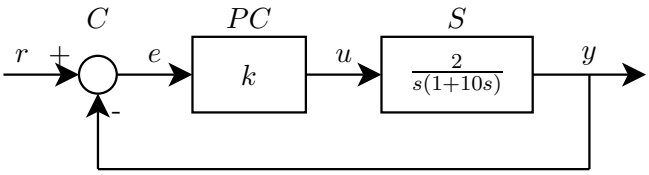

Figure 5: Example of block diagram

In Figure 6, the closed-loop system is represented by a multiport diagram. C, PC and S are three instances of the block class **B** *defined in Figure 4. $L1$, $L2$ and $L3$ are three instances of the arrow class* **A**. *C has two input ports $C.P1$, $C.P3$ and one output port $C.P2$; PC has one input port $PC.P1$ and one output port $PC.P2$; S has one input port $S.P1$ and one output port $S.P2$. The complete behavioral model of the multiport diagram is obtained by gathering equations of its components. That leads to*

$$\begin{aligned} C: &\quad C.P2.y = C.P1.u - C.P3.u \\ L1: &\quad PC.P1.u = C.P2.y \\ PC: &\quad PC.P2.y = PC.k \cdot PC.P1.u \\ L2: &\quad S.P1.u = PC.P2.y \\ S: &\quad S.P2.y = \frac{2}{s(1+10s)} S.P1.u \\ L3: &\quad C.P3.u = S.P2.y \end{aligned} \tag{35}$$

By introducing $C.P1.u = r$, $PC.P1.u = C.P2.y = e$, $S.P1.u = PC.P2.y = u$ and $C.P3.u = S.P2.y = y$, and after a few substitutions, we then obtain equations of the original block diagram presented in Figure 5. Accordingly, these two representations are mathematically equivalent.

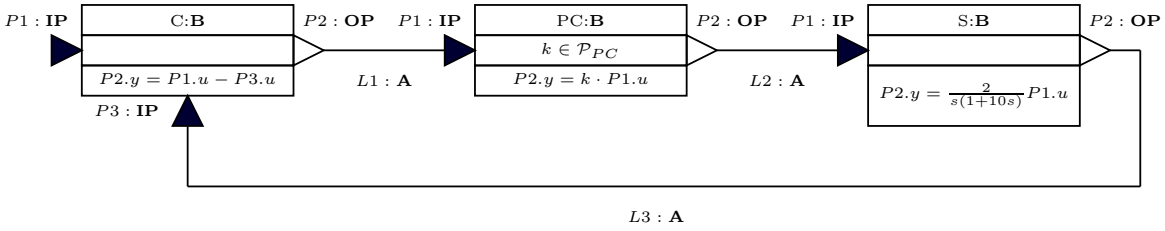

Figure 6: Equivalent multiport diagram of the block diagram

5 Signal Flow graphs

Figure 7 shows the UML class diagram of a signal flow graph. The latter is decomposed into m nodes **N** and n branches **B**. Nodes are equipped with input/output ports **IP**/**OP** to be interconnected by branches. The number of ports is variable according to the nodes while a branch has exactly one input port and one output port. A branch is an oriented link between an output variable and an input variable defined by an

explicit relationship, *i.e. its transmission function* $\mathbf{OP}.y = g_{\mathbf{B}}(p, l, \mathbf{IP}.u)$ where p and l are parameters and internal variables. The behavioral model of a node is an addition rule, *i.e. the output value of a node equal to the sum of all signals entering the node* . Like for the block diagrams, a signal flow graph may be regarded as a particular multiport diagram. Nodes, input/output ports and branches are subclasses of modules, ports and links respectively.

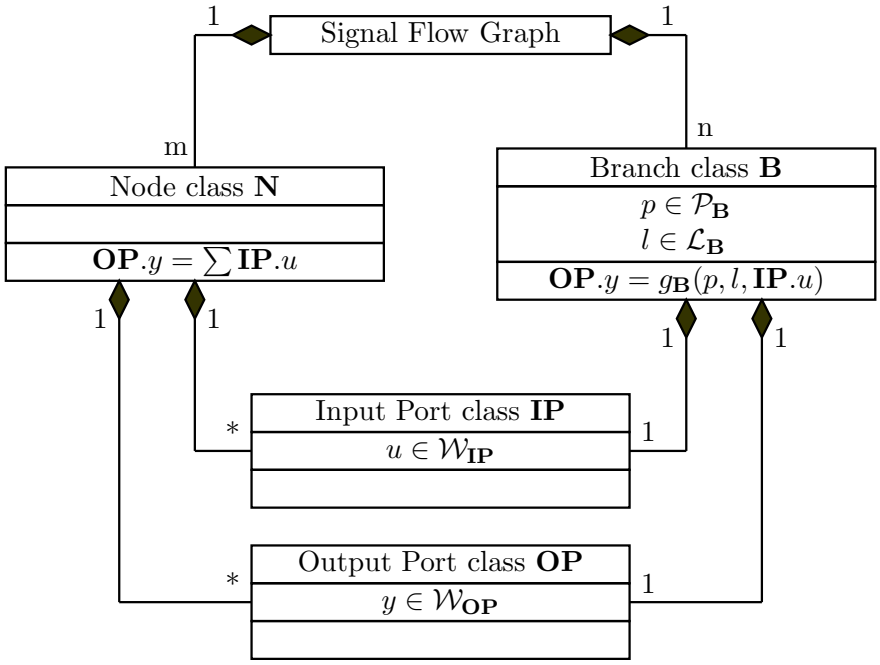

Figure 7: Meta-model of a signal flow graph

Example 5.1 *Figure 8 shows the signal flow graph of the closed-loop system presented in the previous section. The equivalent multiport diagram of this signal flow graph is described in Figure 9.* r*,* e*,* u *and* y *are four instances of the node class* **N** *defined in Figure 7.* $L1$*,* $L2$*,* $L3$ *and* $L4$ *are four instances of the branch class* **B***. Each node is equipped with input/output ports* P_i*. The complete behavioral model of the object diagram is obtained by gathering equations of its components. Equations of the nodes are derived from the generic behavioral model of the node class* **N** *given in Figure 7. That leads to*

$$\begin{aligned}
r: &\quad r.P1.y = r \\
L1: &\quad e.P1.u = r.P1.y \\
e: &\quad e.P2.y = e.P1.u + e.P3.u \\
L2: &\quad u.P1.u = L2.k \cdot e.P2.y \\
u: &\quad u.P2.y = u.P1.u \\
L3: &\quad y.P1.u = \tfrac{2}{s(1+10s)} u.P2.y \\
y: &\quad y.P2.y = y.P1.u \\
L4: &\quad e.P3.u = -1 \cdot y.P2.y
\end{aligned} \tag{36}$$

As previously, by introducing $e.P1.u = r.P1.y = r$*,* $e.P2.y = e$*,* $Su.P2.y = u.P1.u = u$ *and* $y.P2.y = y.P1.u = y$ *and after a few substitutions, equations of the original block diagram presented in Figure 5*

are recovered. Accordingly, the signal flow graph and its object-oriented representation are mathematically equivalent.

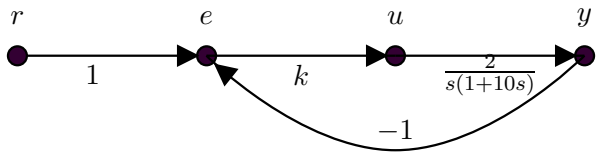

Figure 8: Example of signal flow graph

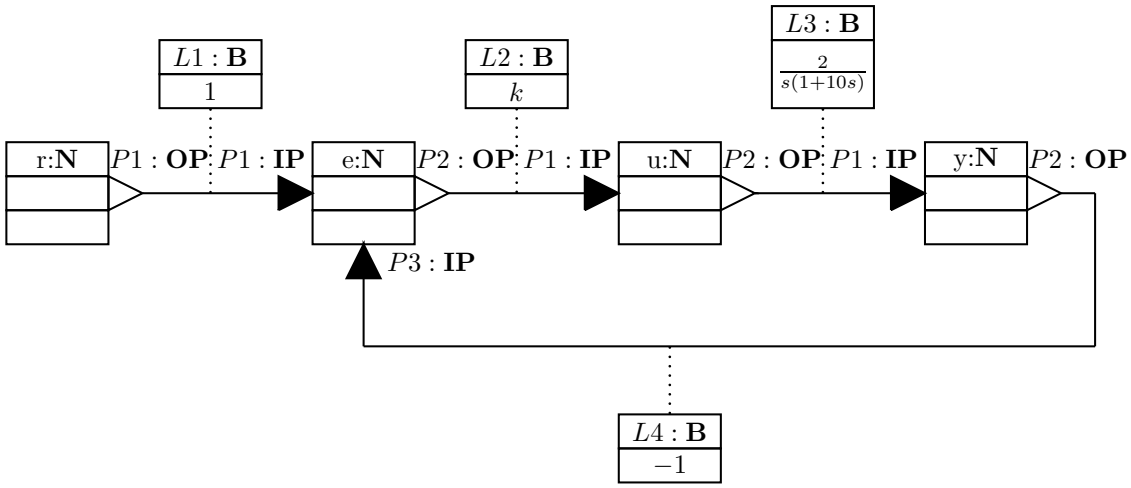

Figure 9: Equivalent multiport diagram of the signal flow graph

6 Compartmental networks

Figure 10 shows the UML class diagram of a compartmental network. The latter is decomposed into m compartments **C** and n oriented arcs **A**. Compartments are equipped with material ports to represent transfer of material into and/or out of compartments through arcs. The number of material ports is variable according to the compartments while a branch has exactly two material ports. An arc is an oriented link between two material ports $P1, P2 : \mathbf{I}_M$. Its behaviour is defined by two equations and one parameter p which corresponds to the fractional transfer coefficient. The first equation $P1.\varphi = p \cdot f(P1.x, P2.x)$ defines the relationship between the matter flow φ and the amount of matter $(P1.x, P2.x)$ in the two compartments. The second one $P1.\varphi + P2.\varphi = 0$ describes the mass continuity in the arc. The dynamical behavior of a compartment is defined by a mass balance equation $\frac{d\mathbf{I}_M.x}{dt} = \sum \mathbf{I}_M.\varphi$. Like for the previous diagrammatic representation, the comparison between UML diagrams in Figure 1 and 10 shows that a compartmental network may be regarded as a particular multiport diagram in which compartments, material ports and arcs are subclasses of modules, ports and links respectively.

Example 6.1 *Figure 11 presents the compartmental network for many infectious diseases including measles, mumps and rubella. This model is composed of three compartments S (for susceptible), I (for infectious) and R (for recovered). It is dynamic in that the numbers of people in each compartment s, i, r may fluctuate over time. Arcs are labelled with the transition rates between compartments. Between S and I, the*

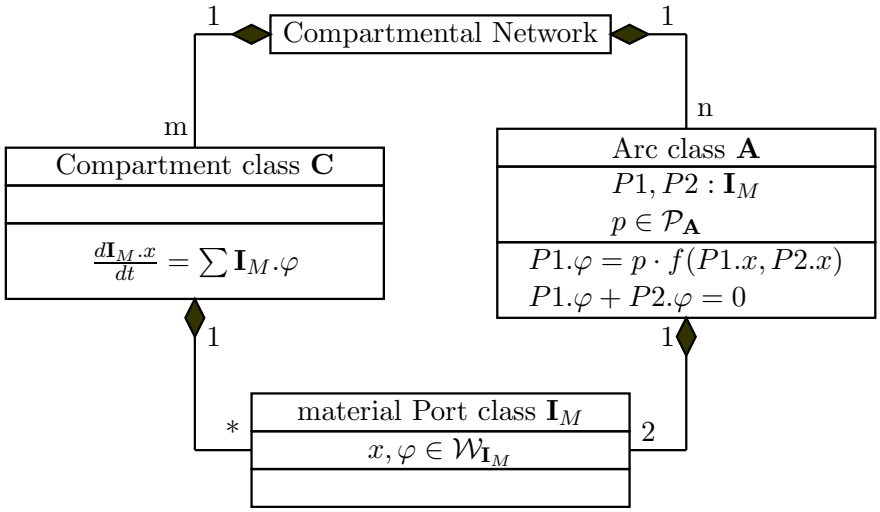

Figure 10: Meta-model of a compartmental network

transition rate is λ, the force of infection, which is simply the rate at which susceptible individuals become infected by an infectious disease. Between I and R, the transition rate is δ (simply the rate of recovery). The SIR system described above can be expressed by the following set of differential equations

$$\frac{dS}{dt} = -\lambda SI \tag{37}$$

$$\frac{dI}{dt} = \lambda SI - \delta I \tag{38}$$

$$\frac{dR}{dt} = \delta I. \tag{39}$$

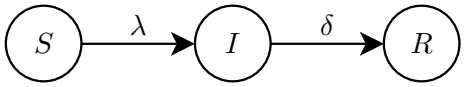

Figure 11: Example of compartmental network

The object-oriented representation of this compartmental network is described in Figure 12. S, I and R are three instances of the compartment class **C** *defined in Figure 10. $L1$ and $L2$ are two instances of the arc class* **A**. *Each compartment is equipped with material ports P_i. Here is the complete list of behavioral*

equations included in this diagram

$$
\begin{aligned}
S: \quad & S.\tfrac{ds}{dt} = S.P1.\varphi \\
& S.P1.x = s \\
L1: \quad & S.P1.\varphi = L1.\lambda \cdot S.P1.x \cdot I.P1.x \\
& S.P1.\varphi + I.P1.\varphi = 0 \\
I: \quad & I.\tfrac{di}{dt} = I.P1.\varphi + I.P2.\varphi \\
& I.P1.x = i \\
& I.P2.x = i \\
L2: \quad & I.P2.\varphi = L2.\lambda \cdot I.P2.x \\
& I.P2.\varphi + R.P1.\varphi = 0 \\
R: \quad & R.\tfrac{dr}{dt} = R.P1.\varphi \\
& R.P1.x = r
\end{aligned}
\tag{40}
$$

This equation system is totally equivalent to the one produced by the compartmental network in Figure 11. The compartmental network and its object-oriented representation are therefore mathematically equivalent.

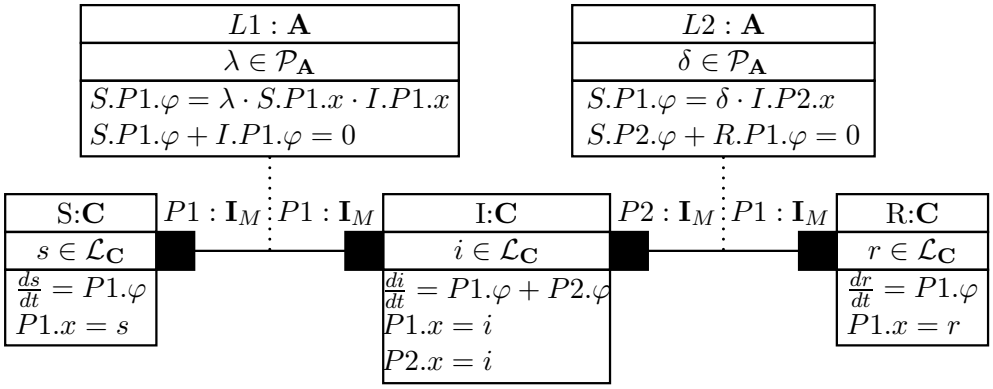

Figure 12: Equivalent multiport diagram of the compartmental network

7 Bond graphs

Figure 13 shows the UML class diagram of an acausal bond graph. The latter is decomposed into m elements $\mathbf{E}$ and n bonds $\mathbf{B}$. Elements may be storage elements, dissipative elements, transformers, gyrators, sources or junctions. Each element is equipped with power ports ($\mathbf{I}_P$) to descirbe transfer of power into and/or out of elements through bonds. The number of power ports is variable according to the elements while a bond has exactly two power ports. By convention, for each power port, the positive power flow is oriented into the element. Each power port is defined by a couple of effort/flow variables noted e and f respectively. A bond is a non-oriented link between two power ports $P1, P2$. Its behaviour is defined by two equations corresponding to the generalized Kirchhoff laws. The dynamical behavior of an element is defined by an implicit equation $f_{\mathbf{E}}(\cdot)$. As previously, the comparison between UML diagrams in Figure 1 and 13 shows that an acausal bond graph may be considered as a particular multiport diagram in which

elements, power ports and bonds are subclasses of modules, ports and links respectively.

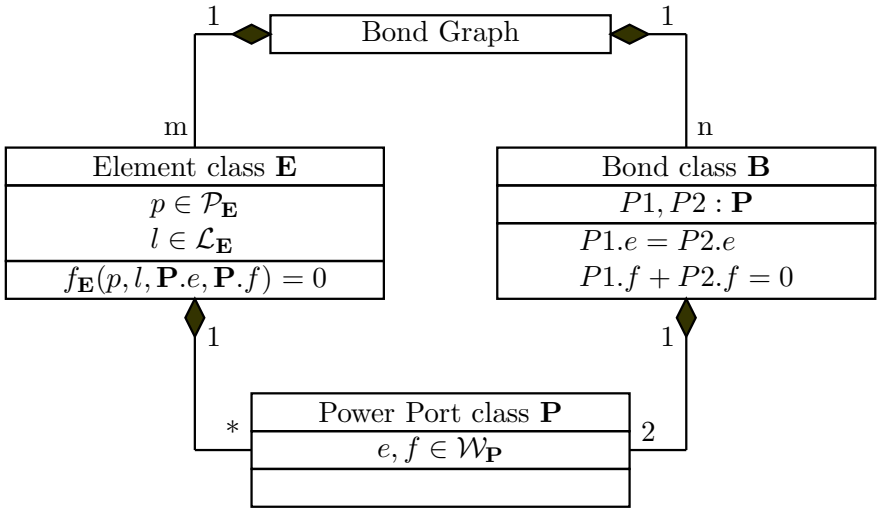

Figure 13: Meta-model of an acausal bond graph

Example 7.1 *Figure 14 presents an electrical RLC circuit composed of a voltage source u_0, a resistor R, a capacitor C and an inductor L. Bonds are labelled with the effort/flow variables.*

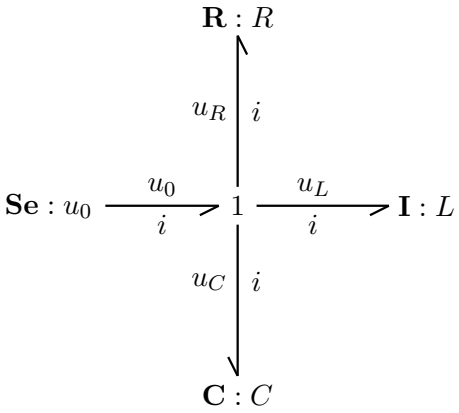

Figure 14: Example of acausal bond graph

The object-oriented representation of this bond graph is described in Figure 15. $u0$, R, L, C and $J1$ are five instances of the element class **E** *defined in Figure 13. $L1, L2, L3$ and $L4$ are four instances of the bond class* **B**. *Each element is equipped with power ports P_i. Here is the complete list of behavioral*

equations included in this diagram

$$
\begin{aligned}
u_0: \quad & u0.P1.e = u_0 \\
& u0.P1.f = i_0 \\
R: \quad & R.P1.e = u_R \\
& R.P1.f = i_R \\
& u_R - R.R' \cdot i_R = 0 \\
C: \quad & C.P1.e = u_C \\
& C.P1.f = i_C \\
& i_C - C.C' \cdot du_C/dt = 0 \\
L: \quad & L.P1.e = u_L \\
& L.P1.f = i_L \\
& u_L - L.L' \cdot di_L/dt = 0 \\
J1: \quad & J1.P1.e - J1.P2.e - J1.P3.e - J1.P4.e = 0 \\
& J1.P1.f = J1.P2.f = J1.P3.f = J1.P4.f \\
L1: \quad & u0.P1.e = J1.P1.e \\
& u0.P1.f + J1.P1.f = 0 \\
L2: \quad & u0.P1.e = J1.P1.e \\
& u0.P1.f + J1.P1.f = 0 \\
L3: \quad & u0.P1.e = J1.P1.e \\
& u0.P1.f + J1.P1.f = 0 \\
L4: \quad & u0.P1.e = J1.P1.e \\
& u0.P1.f + J1.P1.f = 0
\end{aligned}
\tag{41}
$$

This equation system is totally equivalent to the one produced by the bond graph in Figure 14. Accordingly, the bond graph and its object-oriented representation are mathematically equivalent.

8 Synthesis

A synthesis of the previous diagrammatic representations is presented in Figure 16. Block diagrams, signal flow graphs, compartmental networks and bond graphs share the same architecture composed of three object classes: modules, ports and links. As previously shown, bond graph elements, blocks, compartments and branches are subclasses of the module class. Nodes, arrows, arcs and bonds are subclasses of signal, material and power links. Accordingly, the multiport diagram can be regarded as a superclass of block diagrams, signal flow graphs, compartmental networks and bond graphs. Although it is defined independently from any computer language, it has been shown in [3] that the multiport diagram could be easily implemented into an object-oriented simulation language like Modelica. As a result, this multiport diagram can be used as a pattern to implement a classical diagram into any object-oriented modelling language.

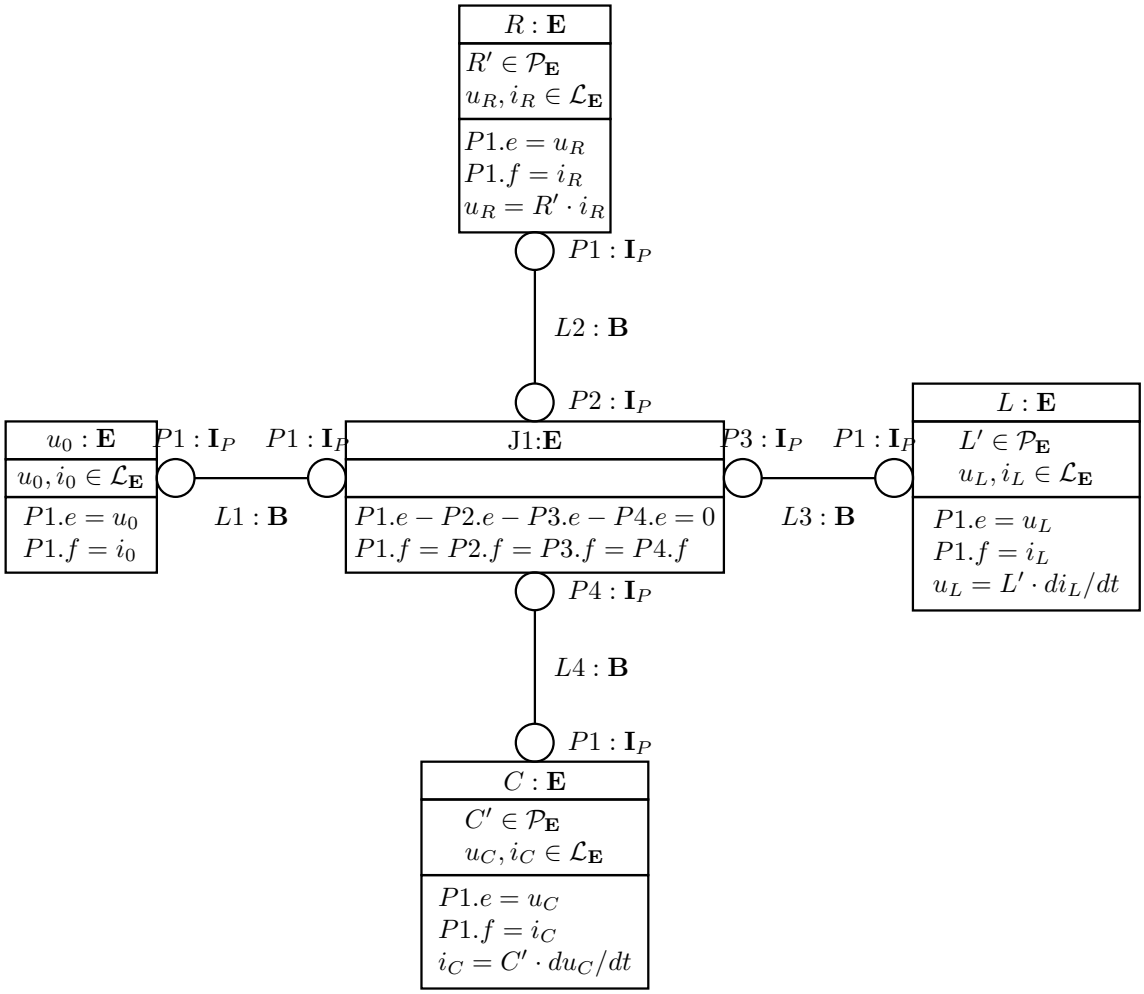

Figure 15: Object-oriented diagram of the bond graph

Moreover, this unified formalism shows that a mix of block diagrams, signal flow graphs, compartmental networks and bond graphs can easily be implemented into an object-oriented modelling environment.

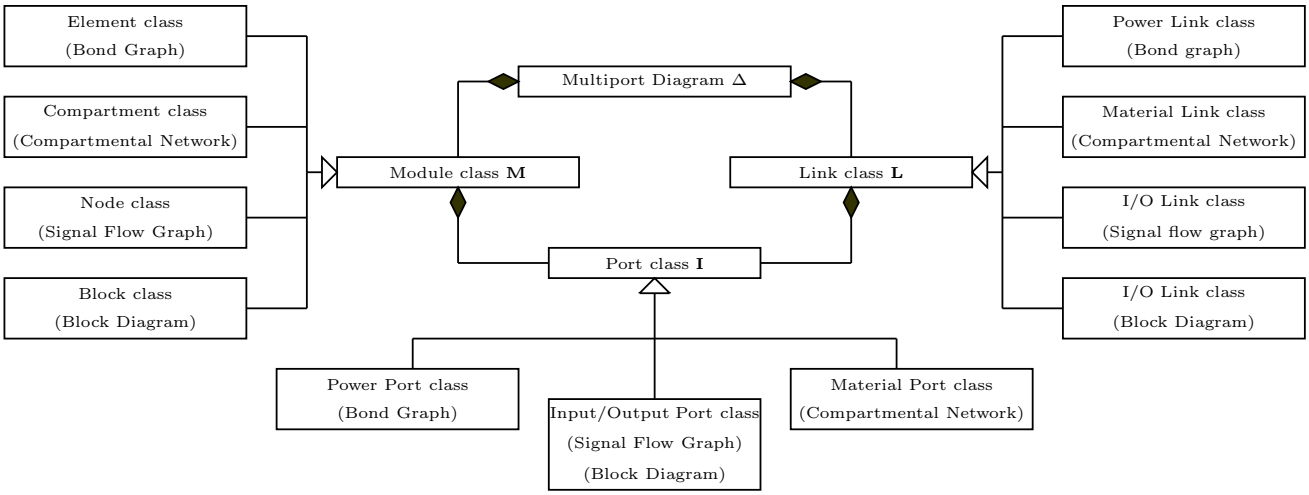

Figure 16: Multiport diagram: a synthesis of diagrammatic representations

9 Conclusion

This paper deals with the meta-modelling of diagrammatic representations devoted to networked systems modelling. In a first part, the object-oriented modelling paradigm is described in a mathematical framework (behavioral representations) independently from any computer language in order to provide a more general character to this analysis. The object-oriented paradigm is not a model structure but a modelling framework, based on some properties like instantiation and inheritance, in which any diagrammatic models such as block diagrams can be developed. In a second part, this paper presents a global and structural analysis of four diagrammatic representations used in systems theory: block diagrams, signal flow graphs, compartmental networks and bond graphs. It is shown that all these component-based models have the same structure based on only three object classes, entitled modules, ports and links. Modules are often composed of differential algebraic equations describing the dynamic behavior of each component. Differences between these diagrams mainly come from ports and links devoted to the description of signal, mass or power transfers between components. The multiport diagram is presented as a generalized model structure which can bring several benefits to the modelling of networked dynamical systems. (1) This meta-model bridges the gap between the main diagrammatic representations of systems theory and should facilitate the education of networked systems modelling. (2) It provides the basic elements (modules, ports and links) for the implementation of diagrammatic representations into any object-oriented modelling environment. (3) This analysis implies that constitutive elements of block diagrams, signal flow graphs, compartmental networks and bond graphs can be mixed together in the same multiport diagram. (4) The multiport diagram is not only restricted to these four graphs. It can also be extended by personal ports developed by the user or by dynamic links described by differential equations. A java software application, *jPhyDia*[2], is currently developed to interface the multiport diagram to Modelica© for a simulation purpose.

References

[1] G. Bastin and V. Guffens. Congestion control in compartmental network systems. *Systems and Control Letters*, 55(8):689 – 696, 2006.

[2] T. Bastogne. A multiport object-oriented diagram for batch system modelling. Methodology and implementation. *Simulation Practice and Theory*, 12(6):425–449, October 2004.

[3] T. Bastogne. Behavioral descriptions of the object-oriented paradigm for physical system modeling. In *16th World IFAC Congress on Automatic Control*, Pragues, Czech Republic, July 2005. Invited session 'Modeling and Interconnection of Open Dynamical Systems' organized by J.C. Willems.

[2] A beta version of jPhyDia is available at:

`http://www.iris.cran.uhp-nancy.fr/francais/idsys/personnes/Perso_TB/HomeTB.html`

[4] T. Bastogne. Behavioral interpretation of the object-oriented paradigm for interconnected dynamic system modeling. *Control Engineering Practice, (2007), doi:10.1016/j.conengprac.2006.12.006*, 2007.

[5] L. Benvenuti and L. Farina. Positive and compartmental systems. *IEEE Transactions on Automatic Control*, 47(2):370–373, 2002.

[6] H. W. Bode. *Network Analysis and Feedback Amplifier Design.* Van Norstrand, Princeton, New Jersey, 1945.

[7] W. Borutzky. Bond graph modeling from an object oriented modeling point of view. *Simulation Practice and Theory*, 7:439–461, 1999.

[8] W. Borutzky. Relations between bond graph based and object-oriented physical systems modeling. In *International Conference on Bond Graph Modeling and Simulation, ICBGM'99*, pages 11–17, San Francisco, CA, January 17-20 1999.

[9] P.C. Breedveld. Port-based modeling of mechatronic systems. *Mathematics and Computers in Simulation*, 66(2-3):99–127, June 2004.

[10] F. E. Cellier. *Continuous System Modeling.* Springer-Verlag, New York, 1991.

[11] J. J. DiStephano, A. R. Stubberud, and I. J. Williams. *Theory and Problems of Feedback and Control Systems.* Schaum's Outline Series, 1967.

[12] H. Elmqvist. *A structured model language for large continuous systems.* PhD thesis, Dept. of Automatic Control, Lund Institute of Technology, Sweden, 1978. Report CODEN: LUTFD2(/TFRT-1015).

[13] B. F. Firestone. A new analogy between mechanical and electrical systems. *J. Acoust. Soc. Am.*, 4:249–267, 1933.

[14] M. Gerdin and T. Glad. On identifiability of object-oriented models1. In *14th IFAC Symposium on System Identification, SYSID 2006*, Newcastle, Australia, March 2006.

[15] T.D. Graybeal. Block diagram network transformation. *Elec. Eng.*, 70:985–990, 1951.

[16] J. R. Holmevik. Compiling SIMULA: a historical study of technological genesis. *IEEE Annals of the History of Computing*, 16(4):25–37, 1994.

[17] J. Jacquez and C. Simon. Qualitative theory of compartmental systems. *SIAM Review*, 30(1):43–79, March 1993.

[18] D. C. Karnopp and R. C. Rosenberg. *System Dynamics; A Unified Approach.* John Wiley, New York, 1974.

[19] G. Kirchhoff. Uber den durchgang eines elektrischen stromes durch eine ebene, insbesondere durch eine kreisformige. *Poggendorfs Annalen der Physik und Chemie*, 64:497–514, 1845.

[20] H. Mann. Physical modeling with multipoles. In *Proc. of the 1999 IEEE Symposium on Computer-Aided Control System Design*, Kona, Hawaii, August 1999.

[21] S. J. Mason. Feedback theory - Some properties of signal flowgraphs. *Proceedings of the Institute of Radio Engineers (IRE)*, 41:1144–1156, September 1953.

[22] O. Otter and H. Elmqvist. Energy flows modeling of mechatronic systems via object diagrams. In *2nd International Conference on Mathematical Modelling (MATHMOD)*, pages 705–710, Vienna, Austria, 1997.

[23] H. M. Paynter. Hydraulics by analog - An electronic model of a pumping plant. *J. Boston Society of Civil Engineering*, pages 197–219, July 1959.

[24] H. M. Paynter. *Analysis and Design of Engineering Systems*. MIT Press, Cambridge, 1961.

[25] F. Pêcheux, B. Allard, C. Lallement, A. Vachoux, and H. Morel. Modeling and simulation of multi-discipline systems using bond graphs and VHDL-AMS. In *International Conference on Bond Graph Modeling and Simulation (ICBGM)*, pages 149–155, 2005.

[26] R. C. Rosenberg and D. C. Karnopp. *Introduction to Physical Systems Dynamics*. Mc-Graw Hill, New York, 1973.

[27] M. Tiller. *Introduction to Physical Modeling With Modelica*. Kluwer International Series in Engineering and Computer Science, Norwell, Massachusetts, 2001.

[28] E. Walter and L. Pronzato. *Identification of Parametric Models from experimental data*. Springer-Verlag, Masson, 1997.

[29] J. C. Willems. From time series to linear systems. *Automatica*, 1986. Part I: Vol. 22, No. 5, pp. 561-580, 1986, Part II: Vol. 22, No. 6, pp. 675-694, 1986, Part III: Vol. 23, No. 1, pp. 87-115, 1987.

[30] J. C. Willems. Paradigms and puzzles in the theory of dynamical systems. *IEEE Transactions on Automatic Control*, 36(3):259–294, March 1991.