

# BEHAVIORAL DESCRIPTIONS OF THE OBJECT-ORIENTED PARADIGM FOR PHYSICAL SYSTEM MODELING

T. Bastogne

*Université Henri Poincaré, Nancy 1,  
Centre de Recherche en Automatique de Nancy (CRAN),  
CNRS-INPL-UHP UMR 7039,  
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*

**Abstract:** This paper deals with the mathematical description of the object-oriented paradigm for physical system modeling through the behavioral approach of systems theory introduced by J.C. Willems in the eighties. Two behavioral representations of an object are defined and three relationships of the object-orientation : instantiation, composition and generalization are examined. Finally, the behavioral description of a object-oriented model structure, entitled *physical diagram*, is developed. *Copyright © 2005 IFAC*

**Keywords:** Object-modeling techniques, mathematical models, system models

## 1. INTRODUCTION

This paper deals with modeling and interconnection of open dynamical systems. The aim consists in determining model structures able to both fit the dynamic behavior of a system and describe with a high-fidelity its internal structure, *i.e.* its components and their interconnexion relationships. Such models are essentially developed for analysis, simulation or understanding of complex and unknown phenomena. Since the seventies, several object-oriented techniques have been developed for complex systems modeling (Elmqvist, 1978; Cellier, 1991; Otter and Elmqvist, 1997; Borutzky, 1999; Mann, 1999; Breedveld, 2004; Bastogne, 2004). These efforts have made real the development of a lot of object formalisms and in 1996, a unification attempt of existing languages was initiated and led to a new specification language: Modelica (Tiller, 2001). Despite these efforts, it seems that due to differences of terminology, *e.g.* object *vs* system, main aspects of the object-orientation developed initially in computer science are not fully ap-

preciated in the systems science community. In fact, the notions of *object* in computer science and *system* in automatic control, systemics or cybernetics are similar. Nevertheless some basic differences can be highlighted.

- The concept of object developed in computer science was not initially associated with a temporal semantic as this is the case in automatic control with the notion of *dynamic system*.
- In control applications, controllers are causal<sup>1</sup> systems, *i.e.* based on the definition of input and output variables, whereas physical systems are non causal by nature (Cellier, 1991). These differences of causality have to be taken into account in the modeling procedure. However

---

<sup>1</sup> In systems theory, the causality notion involves physical realizability. A system is non causal if its response occurs prior to the input stimulus and causal otherwise. Herein, the question is which variable is caused by the other ? If there is an invariant solution the system is causal but if there is no solution, *e.g.* the chicken-egg problem, the system is regarded as non causal.

contrary to the non-causal object-modeling techniques, the systems theory is still based on a causal perception of the process.

- The idea of object is based on concepts like instantiation or inheritance which clearly increase the modularity of the model, *i.e.* updating parts of the model according to the process evolution or easily removing an object without changing the remainder.
- Finally, the representations based on the systems theory, *e.g.* block-diagrams or bond graphs, can be completely described by a mathematical expression, *e.g.* a state-space description. However the object-oriented models of physical systems are algorithmic by nature and have no particular mathematical description.

This latter point is essential since it prevents all objective analysis of the model independently from its implementation languages. That is a particularly crucial issue for the model identification step which requires to assess the identifiability of the model structure. Moreover, the diversity of the existing languages and their frequent evolutions can lead to the impossibility to perpetuate and capitalize on the modeling effort.

Specifying the object-modeling paradigm in a mathematical framework independently from any computer language is therefore the problem addressed in this paper. This problem is analog to the question of meta-modeling in computer science. A lot of specification languages have been proposed, but lately, an effort has been made to standardise the notation, resulting in the Unified Modeling Language (UML) (Rumbaugh *et al.*, 1999). Nevertheless, UML is based on discrete dynamic models and therefore is not very well suited to describe continuous-time behaviors. On the other hand, the behavioral approach of the systems theory, introduced by J.C. Willems in 1986 (Willems, 1986; Willems, 1991) provides a mathematical formalism based on set theory that is well suited to the interconnected systems modeling and may be a solution to the addressed problem. We firstly propose to use the Willems' behavioral formalism as a specification language of the object-oriented models devoted to interconnected systems modeling. Secondly, the concept of object, three object-relationships and an object-oriented model structure called 'physical diagram' are specified. A few examples based on a mechatronic system are also developed to illustrate each theoretical point of the paper.

## 2. BEHAVIORAL REPRESENTATIONS OF AN OBJECT

An object class is usually defined as a structure encapsulating data (state or attributes) and data operations (behavior). Instances of a class are the objects representing the class in the model, they have the same attributes and operations but may have different

attribute values. Parameters  $p$  and variables  $w$  of an object  $O$  will be noted  $O.p$  and  $O.w$  in the sequel. This section aims at defining a general mathematical representation of an object class by describing it as a special dynamic system through the behavioral formalism of system theory introduced by Willems in (Willems, 1986). Two behavioral representations of an object class are proposed.

**Definition 1.** The **absolute behavioral representation** of an object class  $\mathbf{O}$  is defined as follows:

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

where:

- $\mathbb{U}_{\mathbf{O}}^*$  is the universal set, *i.e.* the set of elements under consideration in  $\mathbf{O}$ . A major difference between the objects used at the origin in computer science and those used for physical systems modeling is that contrary to conventional objects, physical phenomena are associated to a temporal semantics, physical systems are dynamic systems. Consequently, the universal set includes a time axis. Parameters and variables are the two other types of attributes used in  $\mathbf{O}$ . 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 legalize and limit the access of a limited number of variables. The latter are generally entitled external variables. The external variables of an object will be described by *manifest* variables and the internal or local variables by *latent* variables. Hence, a mathematical expression of the universal set is:

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

with:  $t \in \mathbb{T}_{\mathbf{O}}$ ,  $p^* \in \mathbb{P}_{\mathbf{O}}^*$ ,  $l^*(t) \in \mathbb{L}_{\mathbf{O}}^*$ ,  $w^*(t) \in \mathbb{W}_{\mathbf{O}}^*$ .  $\times$  denotes the cartesian product,  $\mathbb{T}_{\mathbf{O}}$  the time axis,  $\mathbb{P}_{\mathbf{O}}^*$  the parameter set,  $\mathbb{W}_{\mathbf{O}}^*$  the manifest signal set and  $\mathbb{L}_{\mathbf{O}}^*$  the latent variable set.  $t \in \mathbb{T}$  denotes the time variable,  $p^*$ : the vector of parameters,  $w^*(t)$ : the vector of manifest variables and  $l^*(t)$ : the vector of latent variables.

- $\mathbb{B}_{\mathbf{O}}^*$  denotes the behavioral model. The latter corresponds to the set of the admissible data that the object can generate:  $\mathbb{B}_{\mathbf{O}}^* \subset \mathbb{U}_{\mathbf{O}}^*$ . The complete behavioral model of an object can be expressed as follows:

$$\mathbb{B}_{\mathbf{O}}^* = \left\{ t, p^*, l^*, w^* \left| \begin{array}{l} f(t, p^*, l^*, w^*) = 0 \\ y = g(t, p^*, l^*, z^*, u^*) \end{array} \right. \right\}, \quad (3)$$

where  $w^* = (u^*, y^*, z^*)$ ,  $u^*(t)$  and  $y^*(t)$  denote input and output variables while  $z^*(t)$  is a sub-vector of manifest variables which cannot be *a priori* qualified as inputs or outputs. The behavioral equations of  $\mathbf{O}$  are split up into two parts. The implicit equations are gathered into  $f(\cdot)$  and

the explicit equations into  $g(\cdot)$ .  $\mathbb{B}_O^*$  may be regarded as a set of admissible time trajectories, *i.e.* the possible time behaviors of the variables.

Object classes can be combined with each other by composition and generalization relationships in order to create new object classes. These composition and generalization links lead to consider a second behavioral representation of an object.

**Definition 2.** The **relative behavioral representation** of an object class  $O$  is defined as follows:

$$O = (\mathbb{C}_O, \mathbb{H}_O, \mathbb{U}_O, \mathbb{B}_O), \quad (4)$$

where:

- $\mathbb{C}_O$  denotes the composition set. It contains the object classes which compose  $O$ , *e.g.*  $\mathbb{C}_O = \{A, B\}$  means that  $O$  is composed of two object classes  $A$  et  $B$ .
- $\mathbb{H}_O$  denotes the generalization set. Classes are arranged into specialization-generalization hierarchies, subclasses provide specialised behavior, whereas superclasses are more general. Data and behavior are inherited down a hierarchy. The generalization set of  $O$  is noted  $\mathbb{H}_O$ , *e.g.*  $\mathbb{H}_O = \{A, B\}$  means that  $A$  and  $B$  are two superclasses of  $O$ .

The relative representation relies on the definition of four sets instead of two for the absolute representation. In other words, an absolute representation of an object is a relative representation for which the sets  $\mathbb{C}_O$  and  $\mathbb{H}_O$  are empty. An absolute representation has for advantage to completely define an object independently of all other objects. On the other hand, the interest of a relative 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 its components or superclasses.

### 3. DESCRIPTION OF OBJECT RELATIONSHIPS

#### 3.1 Instantiation relationship

A class is a paradigm defining the behavior and the variables 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. The class-instance relationship is symbolized by  $\Rightarrow$ , *e.g.*  $O \Rightarrow A$  means that  $A$  is an instance of the class  $O$ .

**Proposition 3.** A class and its instances are identical by their form and their behavior :

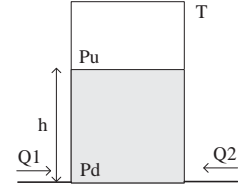


Figure 1. Tank:  $T$ -class

$$\mathbb{U}_A = \mathbb{U}_O \quad (5)$$

$$\mathbb{B}_A = \mathbb{B}_O \quad (6)$$

$$\mathbb{C}_A = \mathbb{C}_O \quad (7)$$

$$\mathbb{H}_A = \mathbb{H}_O. \quad (8)$$

However their parameters and their variables may contain different values.

**Example 4.** Figure 1 depicts a generic tank with two inlet and outlet openings at the bottom. This type of tank can be described by the object-class  $T$  defined by its absolute behavioral representation :

$$T = (\mathbb{U}_T^*, \mathbb{B}_T^*), \quad (9)$$

with :

$$\mathbb{U}_T^* = \mathcal{R}^+ \times \mathcal{R}^{+4} \times \{\emptyset\} \times (\mathcal{R}^{3+} \times \mathcal{R}^2) \quad (10)$$

$$\mathbb{B}_T^* = \left\{ t, \begin{pmatrix} \rho \\ g \\ A \\ h \end{pmatrix}, \begin{pmatrix} L(t) \\ P_d(t) \\ P_u(t) \\ Q_1(t) \\ Q_2(t) \end{pmatrix} \middle| Eq.(12) \right\} \quad (11)$$

$$\begin{cases} P_d(t) - P_u(t) = \rho \cdot g \cdot L(t) \\ Q_1(t) + Q_2(t) = A \cdot dL(t)/dt \\ P_u(t) = 10^5 \\ 0 \leq L(t) \leq h \end{cases} \quad (12)$$

$\rho$ ,  $g$ ,  $A$  and  $h$  are the parameters of the tank, *i.e.* the density of the fluid contained in  $T$ , the gravitation constant, the section area and the height of the tank.  $P_u(t)$  and  $P_d(t)$  are the pressures of the fluid at the top and at the bottom of the tank respectively.  $Q_1(t)$  and  $Q_2(t)$  are the input-output flow rates and  $L(t)$  is the level of water in the tank. The empty set in the equation (10) denotes the absence of latent variable.

Let us consider a particular case of tank with the same characteristics as  $T$  but with :  $A = 1m^2$  and  $h = 3m$ . A model of this tank, noted  $T_1$ , is thus regarded as an instance of  $T$  and is defined by :

$$T_1 \Leftarrow T(A = 1, h = 3). \quad (13)$$

#### 3.2 Composition and generalization relationships

Let  $O$  be an object class defined by its relative behavioral description with either  $\mathbb{H}_O = \{\emptyset\}$  and  $\mathbb{C}_O = \{A\}$ , or  $\mathbb{C}_O = \{\emptyset\}$  and  $\mathbb{H}_O = \{A\}$  where  $A \neq O$ . In the first case, the class  $O$  is composed of a class  $A$  and in the second case the class  $A$  is a superclass of  $O$ . Let  $(\mathbb{U}_O^*, \mathbb{B}_O^*)$  be the absolute representation of  $O$  with:  $\mathbb{U}_O^* = \mathbb{T}_O \times \mathbb{P}_O^* \times \mathbb{W}_O^* \times \mathbb{L}_O^*$ . The following

proposition defines the consequences of the composition and generalization processes on the absolute representation of  $\mathbf{O}$ .

*Proposition 5.* The composition and the generalization relationships imply that the parameters of  $A$  are simply added to those of  $\mathbf{O}$  such that:

$$\mathbb{P}_{\mathbf{O}}^* = \mathbb{P}_{\mathbf{O}} \times \mathbb{P}_{\mathbf{A}}^*. \quad (14)$$

The manifest and latent variables of  $\mathbf{A}$  are added to the manifest and latent variables of  $\mathbf{O}$  respectively:

$$\mathbb{W}_{\mathbf{O}}^* = \mathbb{W}_{\mathbf{O}} \times \mathbb{W}_{\mathbf{A}}^* \quad (15)$$

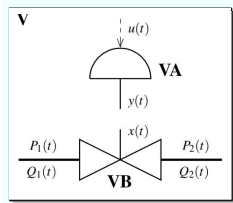
$$\mathbb{L}_{\mathbf{O}}^* = \mathbb{L}_{\mathbf{O}} \times \mathbb{L}_{\mathbf{A}}^*. \quad (16)$$

The absolute behavioral model is then given by:

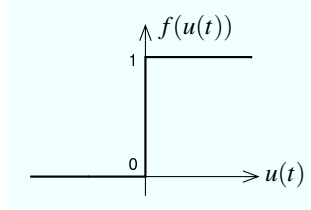
$$\mathbb{B}_{\mathbf{O}}^* = \begin{cases} \mathbb{B}_{\mathbf{O}} \cap \mathbb{B}_{\mathbf{A}}^* & \text{if } \mathbb{U}_{\mathbf{O}} \cap \mathbb{U}_{\mathbf{A}}^* \neq \{\emptyset\} \\ \mathbb{B}_{\mathbf{O}} \times \mathbb{B}_{\mathbf{A}}^* & \text{if } \mathbb{U}_{\mathbf{O}} \cap \mathbb{U}_{\mathbf{A}}^* = \{\emptyset\}. \end{cases} \quad (17)$$

In other terms, if  $\mathbf{O}$  and  $\mathbf{A}$  have no common variable, the absolute behavioral model of  $\mathbf{O}$  is regarded as the cartesian product of  $\mathbb{B}_{\mathbf{O}}$  and  $\mathbb{B}_{\mathbf{A}}^*$ . However, if the composition and generalization processes are correctly established, the behavioral models of  $\mathbf{O}$  and  $\mathbf{A}$  are non disjoint, *i.e.* at least one common variable, and the complete behavior of  $\mathbf{O}$  is the intersection  $\mathbb{B}_{\mathbf{O}}$  and  $\mathbb{B}_{\mathbf{A}}^*$ . From an algebraic point of view, the equation (17) implies that the behavioral equations system of  $\mathbf{O}$  is augmented by the behavioral equations of  $\mathbf{A}$ . For the composition process, the special case:  $\mathbf{A} = \mathbf{O}$  is handled by setting:

$$\mathbb{C}_{\mathbf{O}} = \{\mathbf{O}\} \xrightarrow{\text{def}} \mathbb{C}_{\mathbf{O}} = \{\emptyset\}. \quad (18)$$



(a) Valve



(b) Actuator law

Figure 2. Valve:  $\mathbf{V}$ -class

*Example 6.* Figure 2 depicts a control valve composed of a body and an actuator. The object-class  $\mathbf{V}$  associated with the valve is described by its relative behavioral representation:

$$\mathbf{V} = (\mathbb{C}_{\mathbf{V}}, \mathbb{H}_{\mathbf{V}}, \mathbb{U}_{\mathbf{V}}, \mathbb{B}_{\mathbf{V}}) \quad (19)$$

$$\mathbb{H}_{\mathbf{V}} = \{\emptyset\}$$

$$\mathbb{C}_{\mathbf{V}} = \{\mathbf{VA}, \mathbf{VB}\}$$

$$\mathbb{U}_{\mathbf{V}} = \mathcal{R}^+ \times \{\emptyset\} \times \{\emptyset\} \times \{\emptyset\}$$

$$\mathbb{B}_{\mathbf{V}} = \left\{ t, \left( \begin{array}{l} \mathbf{VA}.y(t) \\ \mathbf{VB}.x(t) \end{array} \right) \middle| \mathbf{VA}.y(t) = \mathbf{VB}.x(t) \right\}$$

$\mathbf{VA}$  denotes the object-class *valve actuator* and  $\mathbf{VB}$  the object-class *valve body*.  $\mathbf{VA}.y(t)$  and  $\mathbf{VB}.x(t)$  denote the position of the actuator stem and the relative position of the plug/seat respectively. This specific

notation implies that  $y(t)$  and  $x(t)$  are two variables defined in  $\mathbf{VA}$  and  $\mathbf{VB}$  respectively but not in  $\mathbf{V}$ . The behavioral equation of  $\mathbb{B}_{\mathbf{V}}$  corresponds to the mechanical coupling between the plug of the valve and the actuator stem. The object-class  $\mathbf{VB}$  associated with the body of the valve is defined by:

$$\mathbf{VB} = (\mathbb{U}_{\mathbf{VB}}^*, \mathbb{B}_{\mathbf{VB}}^*) \quad (20)$$

$$\mathbb{U}_{\mathbf{VB}}^* = \mathcal{R}^+ \times \mathcal{R} \times \mathcal{R} \times (\mathcal{R}^2 \times \mathcal{R}^{2+})$$

$$\mathbb{B}_{\mathbf{VB}}^* = \left\{ t, K_v, x(t), \left( \begin{array}{l} Q_1(t) \\ Q_2(t) \\ P_1(t) \\ P_2(t) \end{array} \right) \middle| \text{Eq.(22)} \right\} \quad (21)$$

$$\left( \begin{array}{l} Q_1(t) = x(t) \cdot K_v \cdot (P_1(t) - P_2(t)) \\ Q_2(t) = -Q_1(t) \end{array} \right) \quad (22)$$

$K_v$  is a coefficient and  $P_1(t)/P_2(t)$ ,  $Q_1(t)/Q_2(t)$  denote the upstream/downstream pressures and flow rates of the valve. The object-class  $\mathbf{VA}$  of the actuator is defined by :

$$\mathbf{VA} = (\mathbb{U}_{\mathbf{VA}}^*, \mathbb{B}_{\mathbf{VA}}^*) \quad (23)$$

$$\mathbb{U}_{\mathbf{VA}}^* = \mathcal{R}^+ \times \{\emptyset\} \times \mathcal{R} \times \mathcal{R}$$

$$\mathbb{B}_{\mathbf{VA}}^* = \{t, y(t), u(t) \mid y(t) = f(u(t))\}$$

$f(\cdot)$  is defined in figure 2(b) and  $u(t)$  denotes the input signal of the actuator. The behavioral model of the actuator is given by the characteristic law  $f(\cdot)$ . According to the definitions of  $\mathbf{VB}$  and  $\mathbf{VA}$ , and the proposition 5, the absolute behavioral representation of  $\mathbf{V}$  is given by:

$$\mathbb{U}_{\mathbf{V}}^* = \mathcal{R}^+ \times \mathcal{R} \times \mathcal{R}^3 \times (\mathcal{R}^3 \times \mathcal{R}^{2+}).$$

$$\mathbb{B}_{\mathbf{V}}^* = \left\{ t, \mathbf{VB}.K_v, \left( \begin{array}{l} \mathbf{VA}.y(t) \\ \mathbf{VB}.x(t) \end{array} \right), \left( \begin{array}{l} \mathbf{VA}.u(t) \\ \mathbf{VB}.Q_1(t) \\ \mathbf{VB}.Q_2(t) \\ \mathbf{VB}.P_1(t) \\ \mathbf{VB}.P_2(t) \end{array} \right) \middle| \dots \right\} \quad (24)$$

$$\left. \begin{array}{l} \mathbf{VA}.y(t) = \mathbf{VB}.x(t) \\ \mathbf{VB}.Q_1(t) = \mathbf{VB}.x(t) \cdot \mathbf{VB}.K_v \cdot \mathbf{VB}.\Delta P(t) \\ \mathbf{VB}.\Delta P(t) = (\mathbf{VB}.P_1(t) - \mathbf{VB}.P_2(t)) \\ \mathbf{VB}.Q_2(t) = -\mathbf{VB}.Q_1(t) \\ \mathbf{VA}.y(t) = f(\mathbf{VA}.u(t)) \end{array} \right\}$$

*Example 7.* Let  $\mathbf{C}$  be the object-class of a controller shown in figure 3(a) and described by:

$$\mathbf{C} = (\mathbb{U}_{\mathbf{C}}^*, \mathbb{B}_{\mathbf{C}}^*) \quad (25)$$

$$\mathbb{U}_{\mathbf{C}}^* = \mathcal{R}^+ \times \{\emptyset\} \times \{\emptyset\} \times \mathcal{R}^3$$

$$\mathbb{B}_{\mathbf{C}}^* = \left\{ t, \left( \begin{array}{l} U(t) \\ W(t) \\ M(t) \end{array} \right) \middle| U(t) = \mathcal{C}(W(t), M(t)) \right\}$$

The control law  $\mathcal{C}(\cdot)$  of the controller is not specified. Hence, the object-class  $\mathbf{C}$  is incomplete and cannot be used alone.  $U(t)$ ,  $W(t)$  and  $M(t)$  are the control, reference and measurement signals. Let us now consider a level controller :  $\mathbf{LC}$  based on a two states control law. Therefore,  $\mathbf{LC}$  is regarded as a subclass of  $\mathbf{C}$  and is defined by:

$$\begin{aligned}
\mathbf{LC} &= (\mathbb{C}_{\mathbf{LC}}, \mathbb{H}_{\mathbf{LC}}, \mathbb{U}_{\mathbf{LC}}, \mathbb{B}_{\mathbf{LC}}) & (26) \\
\mathbb{C}_{\mathbf{LC}} &= \{\emptyset\} \\
\mathbb{H}_{\mathbf{LC}} &= \{\mathbf{C}\} \\
\mathbb{U}_{\mathbf{LC}} &= \mathcal{R}^+ \times \mathcal{R}^{3+} \times \{\emptyset\} \times \{\emptyset\} \\
\mathbb{B}_{\mathbf{LC}} &= \left\{ t, h, \epsilon(t) \mid \begin{array}{l} \mathcal{C}(\epsilon) \text{ is given in figure (3(b))} \\ \epsilon(t) = \mathbf{C}.W(t) - \mathbf{C}.M(t) \end{array} \right\}
\end{aligned}$$

In the definition of  $\mathbf{LC}$ , only the specific details of

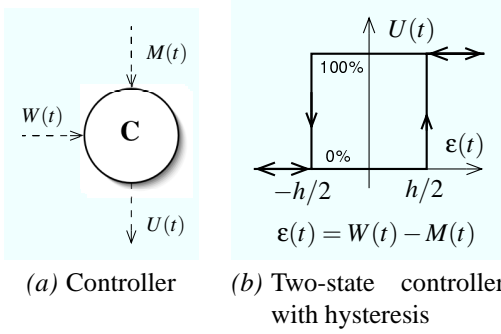


Figure 3.  $\mathbf{LC}$ -class

the level controller are given, *i.e.* one parameter:  $h$ , and the control law  $\mathcal{C}$ . Based on the relative definitions of  $\mathbf{C}$  and  $\mathbf{LC}$ , the absolute description of  $\mathbf{LC}$  is thus given by:

$$\begin{aligned}
\mathbb{U}_{\mathbf{LC}}^* &= \mathcal{R}^+ \times \mathcal{R}^+ \times \mathcal{R} \times \mathcal{R}^3 & (27) \\
\mathbb{B}_{\mathbf{LC}}^* &= \left\{ t, h, \epsilon(t), \begin{pmatrix} U(t) \\ W(t) \\ M(t) \end{pmatrix} \mid \begin{array}{l} U(t) = \mathcal{C}(W(t), M(t)) \\ \mathcal{C}(\epsilon) \text{ is given in figure (3(b))} \\ \epsilon(t) = W(t) - M(t) \end{array} \right\}
\end{aligned}$$

### 3.3 Compositions of object-relationships

Propositions 3 and 5 allow the modeler to systematically transform a relative behavioral representation into an absolute representation. But the composition and generalization relationships can also be combined with each other. Firstly, let us specify each application:

- Let  $\mathcal{C}(\cdot)$  be a composition application such that if  $\mathbf{A} \in \mathbb{C}_{\mathbf{O}}$  then  $\mathbf{O} = \mathcal{C}(\mathbf{A})$  means that  $\mathbf{O}$  is composed of  $\mathbf{A}$  or  $\mathbf{A}$  is a component of  $\mathbf{O}$ . It is straightforward to show that  $\mathcal{C}$  is reflexive, non symmetric and transitive.
- Let  $\mathcal{H}(\cdot)$  be a generalization application such that if  $\mathbf{A} \in \mathbb{H}_{\mathbf{O}}$  then  $\mathbf{O} = \mathcal{H}(\mathbf{A})$  means that  $\mathbf{O}$  is a subclass object of  $\mathbf{A}$ . It is straightforward to show that  $\mathcal{H}$  is non reflexive, non symmetric but transitive.

The two possible compositions of those object applications are thus:

- $\mathbf{C} = \mathcal{C}(\mathbf{B})$  and  $\mathbf{B} = \mathcal{H}(\mathbf{A})$ , *i.e.*  $\mathbf{C} = \mathcal{C} \circ \mathcal{H}(\mathbf{A})$ :  $\mathbf{B}$  is an object class composed of another object class that inherits from an object-class  $\mathbf{A}$
- $\mathbf{C} = \mathcal{H}(\mathbf{B})$  and  $\mathbf{B} = \mathcal{C}(\mathbf{A})$ , *i.e.*  $\mathbf{C} = \mathcal{H} \circ \mathcal{C}(\mathbf{A})$ :  $\mathbf{B}$  is an object class inheriting from another object class composed of the object class  $\mathbf{A}$ .

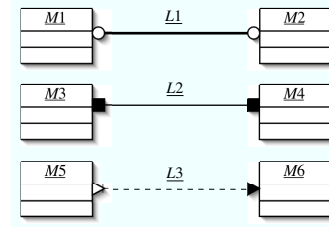


Figure 4. Interconnections of modules

The complete behavioral representation of  $\mathbf{C}$  is then:

$$\mathbb{P}_{\mathbf{C}}^* = \mathbb{P}_{\mathbf{C}} \times \mathbb{P}_{\mathbf{A}}^* \times \mathbb{P}_{\mathbf{B}}^* \quad (28)$$

$$\mathbb{W}_{\mathbf{C}}^* = \mathbb{W}_{\mathbf{C}} \times \mathbb{W}_{\mathbf{A}}^* \times \mathbb{W}_{\mathbf{B}}^* \quad (29)$$

$$\mathbb{L}_{\mathbf{C}}^* = \mathbb{L}_{\mathbf{C}} \times \mathbb{L}_{\mathbf{A}}^* \times \mathbb{L}_{\mathbf{B}}^* \quad (30)$$

$$\mathbb{B}_{\mathbf{C}}^* = \begin{cases} \mathbb{B}_{\mathbf{C}} \cap \mathbb{B}_{\mathbf{A}}^* \cap \mathbb{B}_{\mathbf{B}}^* & \text{if } \mathbb{U}_{\mathbf{C}} \cap \mathbb{U}_{\mathbf{A}}^* \cap \mathbb{U}_{\mathbf{B}}^* \neq \{\emptyset\} \\ \mathbb{B}_{\mathbf{C}} \times \mathbb{B}_{\mathbf{A}}^* \times \mathbb{B}_{\mathbf{B}}^* & \text{if } \mathbb{U}_{\mathbf{C}} \cap \mathbb{U}_{\mathbf{A}}^* \cap \mathbb{U}_{\mathbf{B}}^* = \{\emptyset\}. \end{cases}$$

## 4. PHYSICAL DIAGRAM

This section intends to find out the behavioral descriptions of an object-oriented model structure, entitled *physical diagram*, developed in (Bastogne, 2004).

**Definition 8.** A **physical diagram**  $\Delta$  is a model structure composed of *modules* connected by *links*. Its relative behavioral representation is defined as follows:

$$\Delta = (\mathbb{C}_{\Delta}, \mathbb{H}_{\Delta}, \mathbb{U}_{\Delta}, \mathbb{B}_{\Delta}) \quad (31)$$

Its composition set  $\mathbb{C}_{\Delta}$  is given by:

$$\mathbb{C}_{\Delta} = \{\mathbb{M}, \mathbb{X}\} \quad (32)$$

where  $\mathbb{M} = \{M_1, \dots, M_m\}$  is a set of modules and  $\mathbb{X} = \{L_1, \dots, L_n\}$  a set of links connecting the modules.  $m$  and  $n$  are the number of modules and links in  $\Delta$ .  $\mathbb{H}_{\Delta}$ ,  $\mathbb{U}_{\Delta}$  and  $\mathbb{B}_{\Delta}$  are generally empty. However, additional data and equations that are not included in the modules can be stated in  $\mathbb{U}_{\Delta}$  and  $\mathbb{B}_{\Delta}$ .

**Definition 9.** A **module**  $\mathbf{M}$  is an object-class associated with a real component or entity of the physical system. Its relative behavioral representation is given by  $\mathbf{M} = (\mathbb{C}_{\mathbf{M}}, \mathbb{H}_{\mathbf{M}}, \mathbb{U}_{\mathbf{M}}, \mathbb{B}_{\mathbf{M}})$  with:  $\mathbb{U}_{\mathbf{M}} = \mathbb{T}_{\mathbf{M}} \times \mathbb{P}_{\mathbf{M}} \times \mathbb{L}_{\mathbf{M}} \times \mathbb{W}_{\mathbf{M}}$ .

**Definition 10.** A **link**  $\mathbf{L}$  is an object class which describes the connexion mode, *i.e.* the interconnection equations, between two modules. Three classes of links have been defined, entitled *power link*, *material link* and *signal link*. Their relative behavioral representations are given by:

$$\mathbf{L}_P = (\mathbb{C}_{\mathbf{L}_P}, \mathbb{H}_{\mathbf{L}_P}, \mathbb{U}_{\mathbf{L}_P}, \mathbb{B}_{\mathbf{L}_P}) \quad (33)$$

$$\mathbb{C}_{\mathbf{L}_P} = \mathbb{H}_{\mathbf{L}_P} = \{\emptyset\}$$

$$\mathbb{U}_{\mathbf{L}_P} = \mathcal{R}^+ \times \{\emptyset\} \times \{\emptyset\} \times \mathcal{R}^4$$

$$\mathbb{B}_{\mathbf{L}_P} = \left\{ t, \begin{pmatrix} \alpha_1(t) \\ \varphi_1(t) \\ \alpha_2(t) \\ \varphi_2(t) \end{pmatrix} \mid \begin{array}{l} \alpha_1(t) = \alpha_2(t) \\ \varphi_1(t) + \varphi_2(t) = 0 \end{array} \right\}$$

## 5. CONCLUSION

$$\begin{aligned} \mathbf{L}_M &= (\mathbf{C}_{\mathbf{L}_M}, \mathbf{H}_{\mathbf{L}_M}, \mathbf{U}_{\mathbf{L}_M}, \mathbf{B}_{\mathbf{L}_M}) & (34) \\ \mathbf{C}_{\mathbf{L}_M} &= \mathbf{H}_{\mathbf{L}_M} = \{\emptyset\} \\ \mathbf{U}_{\mathbf{L}_M} &= \mathcal{R}^+ \times \{\emptyset\} \times \{\emptyset\} \times \mathcal{R}^2 \\ \mathbf{B}_{\mathbf{L}_M} &= \left\{ t, \begin{pmatrix} \varphi_1(t) \\ \varphi_2(t) \end{pmatrix} \middle| \varphi_1(t) + \varphi_2(t) = 0 \right\} & (35) \end{aligned}$$

$$\begin{aligned} \mathbf{L}_S &= (\mathbf{C}_{\mathbf{L}_S}, \mathbf{H}_{\mathbf{L}_S}, \mathbf{U}_{\mathbf{L}_S}, \mathbf{B}_{\mathbf{L}_S}) & (36) \\ \mathbf{C}_{\mathbf{L}_S} &= \mathbf{H}_{\mathbf{L}_S} = \{\emptyset\} \\ \mathbf{U}_{\mathbf{L}_S} &= \mathcal{R}^+ \times \{\emptyset\} \times \{\emptyset\} \times \mathcal{R}^2 \\ \mathbf{B}_{\mathbf{L}_S} &= \left\{ t, \begin{pmatrix} u(t) \\ y(t) \end{pmatrix} \middle| u(t) := y(t) \right\} & (37) \end{aligned}$$

The physical links have four manifest variables composed of two across variables  $(\alpha_1(t), \alpha_2(t))$  and two through variables  $(\varphi_1(t), \varphi_2(t))$  in accordance with the Firestone's terminology. Their behavioral model, *i.e.* their connection laws shown in equation (34), corresponds to the generalized Kirchhoff laws applied to their across/through variables. The material links have two manifest variables composed of two through variables  $(\varphi_1(t), \varphi_2(t))$ . Their behavioral model, *i.e.* their connection law shown in equation (35), corresponds to a mass balance equation. The signal links have two manifest variables composed of two input/output variables  $(u(t), y(t))$  respectively. Their behavioral model, *i.e.* their connection law shown in equation (37), is a causal assignment between the output variable and the input variable.

Behavioral descriptions of object-classes and object relationships allow the complete description of the physical diagram. The absolute behavioral representation of the physical diagram is finally given by:

$$\Delta = (\mathbf{U}_\Delta^*, \mathbf{B}_\Delta^*), \quad (38)$$

with:

$$\mathbf{U}_\Delta^* = \mathbf{T}_\Delta \times \mathbf{P}_\Delta^* \times \mathbf{L}_\Delta^* \times \mathbf{W}_\Delta^* \quad (39)$$

$$\mathbf{B}_\Delta^* = \mathbf{B}_\Delta \cap \mathbf{B}_{\mathbf{C}_\Delta} \cap \mathbf{B}_{\mathbf{H}_\Delta} \quad (40)$$

$$\mathbf{P}_\Delta^* = \mathbf{P}_\Delta \times \mathbf{P}_{\mathbf{C}_\Delta} \times \mathbf{P}_{\mathbf{H}_\Delta} \quad (41)$$

$$\mathbf{L}_\Delta^* = \mathbf{L}_\Delta \times \mathbf{L}_{\mathbf{C}_\Delta} \times \mathbf{L}_{\mathbf{H}_\Delta} \quad (42)$$

$$\mathbf{W}_\Delta^* = \mathbf{W}_\Delta \times \mathbf{W}_{\mathbf{C}_\Delta} \times \mathbf{W}_{\mathbf{H}_\Delta} \quad (43)$$

and:

$$\mathbf{B}_{\mathbf{C}_\Delta} = \mathbf{B}_X \cap \mathbf{B}_M \quad (44)$$

$$= (\mathbf{B}_{L_1}^* \times \cdots \times \mathbf{B}_{L_n}^*) \cap (\mathbf{B}_{M_1}^* \times \cdots \times \mathbf{B}_{M_m}^*)$$

$$\mathbf{Z}_{\mathbf{C}_\Delta} = \mathbf{Z}_X \times \mathbf{Z}_M \quad (45)$$

$$= \mathbf{Z}_{L_1}^* \times \cdots \times \mathbf{Z}_{L_n}^* \times \mathbf{Z}_{M_1}^* \times \cdots \times \mathbf{Z}_{M_m}^*.$$

where  $\mathbf{Z} = \{\mathbb{P}, \mathbb{W}, \mathbb{L}\}$ . Accordingly, the behavioral expression of a physical diagram may be completely deduced from the ones of its modules. Equations (40) and (45) finally show that analyzing the behavior of a physical diagram leads to handle a differential algebraic equation system.

The issue addressed in this paper is the mathematical representation of the object paradigm for physical system modeling. The behavioral formalism of the systems theory, introduced by J.C. Willems in the eighties, is advocated for this purpose. Two behavioral representations of an object are expressed. These two representations are equivalent, they just constitute two different ways to define a same object. Three relationships of the object-orientation : instantiation, composition and generalization are examined. The behavioral description of an object-oriented model structure, entitled *physical diagram*, is also proposed. It is finally pointed out that making-up object models in the behavioral framework generally leads to manipulate differential-algebraic equations systems.

## REFERENCES

- Bastogne, T. (2004). A multiport object-oriented diagram for batch system modelling. Methodology and implementation. *Simulation Practice and Theory* **12**(6), 425–449.
- Borutzky, W. (1999). Bond graph modeling from an object oriented modeling point of view. *Simulation Practice and Theory* **7**, 439–461.
- Breedveld, P.C. (2004). Port-based modeling of mechatronic systems. *Mathematics and Computers in Simulation* (66), 99–127.
- Cellier, F. E. (1991). *Continuous System Modeling*. Springer-Verlag.
- Elmqvist, H. (1978). A structured model language for large continuous systems. PhD thesis. Dept. of Automatic Control, Lund Institute of Technology. Sweden. Report CODEN: LUTFD2/(TFRT-1015).
- Mann, H. (1999). Physical modeling with multipoles. In: *Proc. of the 1999 IEEE Symposium on Computer-Aided Control System Design*. Kona.
- Otter, O. and H. Elmqvist (1997). Energy flows modeling of mechatronic systems via object diagrams. In: *Proc. of the 2nd MATHMOD*. Vienna. pp. 705–710.
- Rumbaugh, J., I. Jacobson and G. Booch (1999). *The Unified Modeling Language Reference Manual*. Addison-Wesley.
- Tiller, M. (2001). *Introduction to Physical Modeling With Modelica*. Kluwer International Series in Engineering and Computer Science.
- Willems, J. C. (1986). From time series to linear systems. *Automatica*. 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.
- Willems, J. C. (1991). Paradigms and puzzles in the theory of dynamical systems. *IEEE Transactions on Automatic Control* **36**(3), 259–294.