Behavioral interpretation of the object-oriented paradigm for interconnected dynamic system modeling

Thierry Bastogne

Behavioral interpretation of the object-oriented paradigm for interconnected dynamic system modeling

T. Bastogne
Centre de Recherche en Automatique de Nancy, CRAN
Nancy-Université, CNRS UMR 7039
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

This paper presents a mathematical interpretation of the object-oriented modeling paradigm inspired from the Willems' behavioral approach of systems theory. The object modeling of interconnected dynamic systems is introduced independently from any computer language and expressed as a set computation problem. Two behavioral representations (complete and partial) of an object are defined. Three object relationships, *i.e.* instantiation, composition and generalization are examined in the behavioral framework. Each definition is illustrated by basic examples, *e.g.* a storage tank, a resistor, a control valve and an electrical circuit. The implementation of the behavioral representations into the object-oriented language Modelica is finally presented. [1]

Keywords Object-modeling techniques, mathematical models, dynamic systems, complex systems.

1 Introduction

Since the seventies, several object-oriented techniques have been developed for interconnected dynamic system modeling [Elmqvist, 1978, Cellier, 1991, Otter and Elmqvist, 1997, Borutzky, 1999a, Mann, 1999, Breedveld, 2004, Bastogne, 2004]. These efforts have made real the development of a lot of object formalisms such as Allan, Dymola, NMF, ObjectMath, Omola, SIDOPS+ and Smile. In 1996, a unification attempt of existing languages was initiated and led to a new specification

[1] To be published in International Journal of Systems Science

language: Modelica [Tiller, 2001]. Despite these efforts, it still seems that main aspects of the object-orientation developed initially in computer science are not fully appreciated in the systems science community [Borutzky, 1999b]. 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[2] 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 process. However contrary to 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 rest of the model.

Currently, the most widespread object-oriented modeling techniques devoted to interconnected dynamic systems, *e.g.* the language Modelica, lead to handle differential algebraic equation systems. In [Gerdin and Glad, 2006], a differential algebra approach is developed to examine identifiability of object-oriented models.

The purpose of this paper is to show that object-oriented modeling of interconnected dynamic system may also be described as a problem of set computation. Set computation has a lot of practical applications in system control, particularly in robust control and in parameter estimation [Jaulin et al., 2001]. To this end, a new mathematical interpretation of the object-oriented paradigm for interconnected dynamic system modeling is developed herein. This new approach of object orientation relies on a behavioral formalism of systems theory proposed by Willems in [Willems, 1986, Willems, 1991, Polderman and Willems, 1998].

The Willems' behavioral approach of systems theory is used in section 2 to define two behavioral representations of an object class. In the next section, three object relationships, *i.e.* instantiation, composition and generalisation are examined in the behavioral framework. Section 4 is devoted to the behavioral representation of interconnected dynamic systems. At last, the links between behavioral representations and the object-oriented language Modelica are pointed out in section 5.

[2] 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.

2 Behavioral representations of an object

Table 1 sums up some notations used in the sequel. An object is generally defined 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 [Willems, 1986] and leads to a first behavioral definition of an object.

Table 1: Basic notation of sets

Ref.	Description
$\mathbf{D}$	Object class
D	Instance (object)
$\mathcal{B}$	Behavioral set
$\mathcal{C}$	Composition set
$\mathcal{E}$	Equation 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

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:

- $\mathcal{U}^*_{\mathbf{O}}$ *is the universal set of* $\mathbf{O}$*,* i.e. *the set that 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 × *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 corresponds to 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* $\mathcal{U}^*_{\mathbf{O}}$ *that 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.*

A major difference between 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 $\mathcal{U}^*_{\mathbf{O}}$ 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 behavioral model $\mathcal{B}^*_{\mathbf{O}}$ is analog to a set of admissible time trajectories [Polderman and Willems, 1998].

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. Moreover, 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 of an object is defined.

Definition 2.2 *The* ***partial behavioral representation*** *of an object class* **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.* $\mathcal{C}_{\mathbf{O}} = \{A\}$ *implies that* **O** *is composed of* A.

- $\mathcal{H}_{\mathbf{O}}$ *is generalization (inheritance) set.* $\mathcal{H}_{\mathbf{O}} = \{\mathbf{A}\}$ *implies that* **A** *is a super-classes of* **O***, or* **O** *is a subclass of* **A**.

- $\mathcal{U}_{\mathbf{O}}$ *is the partial universal set containing elements exclusively belonging to* **O***. In other words, variables of elements of* $\mathcal{C}_{\mathbf{O}}$ *or* $\mathcal{H}_{\mathbf{O}}$ *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:*

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

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, an 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.

3 Description of object relationships

3.1 Instantiation relationship

Conjecture 3.1 *The instantiation relationship between an object class $\mathbf{O}$ and its instance A is noted $A : \mathbf{O}$ and satisfies :*

$$\mathcal{U}_A = \mathcal{U}_{\mathbf{O}} \tag{7}$$

$$\mathcal{B}_A = \mathcal{B}_{\mathbf{O}} \tag{8}$$

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

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

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.

Example 3.1 ***Storage tank**. Figure 1 depicts a generic tank with two openings at the bottom. This type of tank can be described by the object-class $\mathbf{T}$ defined by its complete behavioral representation :*

$$\mathbf{T} = (\mathcal{U}^*_T, \mathcal{B}^*_T), \tag{11}$$

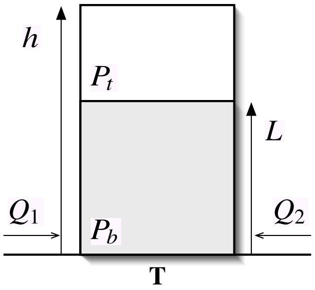


Figure 1: Tank: **T**-class

with :

$$
\begin{aligned}
\mathcal{U}^*_{\mathbf{T}} &= \mathbb{R} \times \mathbb{R}^{+4} \times \mathbb{R}^{+} \times (\mathbb{R}^{2+} \times \mathbb{R}^{2}) & (12)\\
\mathcal{B}^*_{T} &= \left\{ t, \begin{pmatrix} \rho \\ g \\ A \\ h \end{pmatrix}, P_t(t), \begin{pmatrix} L(t) \\ P_b(t) \\ Q_1(t) \\ Q_2(t) \end{pmatrix} \middle| \ \mathcal{E}_T \right\} \\
\mathcal{E}_T &= \begin{pmatrix} P_b(t) - P_t(t) = \rho \cdot g \cdot L(t) \\ Q_1(t) + Q_2(t) = A \cdot dL(t)/dt \\ P_t(t) = 10^5 \\ 0 \leq L(t) \leq h \end{pmatrix}
\end{aligned}
$$

ρ, 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. $P_t(t)$ and $P_b(t)$ 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. $L(t), P_b(t), Q_1(t)$ and $Q_2(t)$ are regarded by the modeler as external variables,* i.e. *variables susceptible of exchange with other objects.*

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 :*

$$T_1 : \mathbf{T}(A = 1, h = 3). \qquad (13)$$

3.2 Composition and generalization relationships

Let **O** be an object class whose the composition or the generalization sets are not empty, *e.g.* either $\mathcal{H}_{\mathbf{O}} = \{\emptyset\}$ and $\mathcal{C}_{\mathbf{O}} = \{\mathbf{A}\}$, or $\mathcal{C}_{\mathbf{O}} = \{\emptyset\}$ and $\mathcal{H}_{\mathbf{O}} = \{\mathbf{A}\}$ with $\mathbf{A} \neq \mathbf{O}$. In the first case **A** is a component of **O** while in the second case **A** is a super-class of **O**. Let $(\mathcal{U}^*_{\mathbf{O}}, \mathcal{B}^*_{\mathbf{O}})$ be the complete representation of **O** with: $\mathcal{U}^*_{\mathbf{O}} = \mathcal{T}_{\mathbf{O}} \times \mathcal{P}^*_{\mathbf{O}} \times \mathcal{W}^*_{\mathbf{O}} \times \mathcal{L}^*_{\mathbf{O}}$.

Conjecture 3.2 *The composition and generalization relationships imply that parameters and variables of A are added to those of* $\mathbf{O}$ *such that:*

$$\begin{aligned}\mathcal{P}^*_{\mathbf{O}} &= \mathcal{P}_{\mathbf{O}} \times \mathcal{P}^*_{\mathbf{A}} &(14)\\ \mathcal{W}^*_{\mathbf{O}} &= \mathcal{W}_{\mathbf{O}} \times \mathcal{W}^*_{\mathbf{A}} &(15)\\ \mathcal{L}^*_{\mathbf{O}} &= \mathcal{L}_{\mathbf{O}} \times \mathcal{L}^*_{\mathbf{A}}. &(16)\end{aligned}$$

The complete behavioral model is then given by:

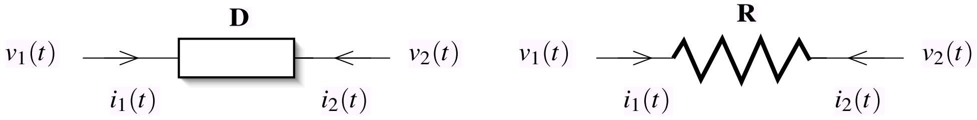
$$\mathcal{B}^*_{\mathbf{O}} = \mathcal{B}_{\mathbf{O}} \cap \mathcal{B}^*_{\mathbf{A}} \tag{17}$$

The special case: $\mathbf{A} = \mathbf{O}$ is handled by setting:

$$\mathcal{C}_{\mathbf{O}} = \{\mathbf{O}\} \overset{def}{\Longrightarrow} \mathcal{C}_{\mathbf{O}} = \{\emptyset\}. \tag{18}$$

From an algebraic point of view, composition and generalisation relationships imply that the behavioral equation system of $\mathbf{O}$ is augmented by the behavioral equations of A. But equation (17) also shows that the complete behaviour of $\mathbf{O}$ corresponds to the intersection of two behavioral sets $\mathcal{B}_{\mathbf{O}}$ and $\mathcal{B}^*_{\mathbf{A}}$.

Example 3.2 ***Electrical resistor***.



(a) **D**: electrical dipole class

(b) **R**: resistor class

Figure 2: Object models of electrical dipoles

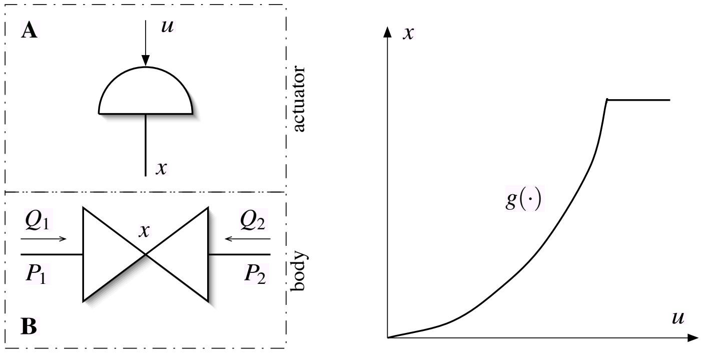
Figure 2 shows two classes of electrical components. ***D*** *is the generic class of an electrical dipole and* ***R*** *corresponds to the resistor class. The complete behavioral description of* ***D*** *is given by :*

$$\mathbf{D} = (\mathcal{U}^*_{\mathbf{D}}, \mathcal{B}^*_{\mathbf{D}}) \tag{19}$$

$$\mathcal{U}^*_{\mathbf{D}} = \mathbb{R} \times \{\emptyset\} \times \{\emptyset\} \times \mathbb{R}^4 \tag{20}$$

$$\mathcal{B}^*_{\mathbf{D}} = \left\{ t, \begin{pmatrix} i1(t) \\ i_2(t) \\ v_1(t) \\ v_2(t) \end{pmatrix} \middle| \; i_1 + i_2 = 0 \quad \forall v_1(t), v_2(t) \right\} \tag{21}$$

The class ***R*** *can be viewed as a subclass of* ***D****. A partial description of* ***R*** *is*



(a) Valve

(b) Actuator law

Figure 3: Valve: **V**-class

then defined by :

$$\mathbf{R} = (\mathcal{C}_{\mathbf{R}}, \mathcal{H}_{\mathbf{R}}, \mathcal{U}_{\mathbf{R}}, \mathcal{B}_{\mathbf{R}}) \tag{22}$$

$$\mathcal{H}_{\mathbf{R}} = \{\mathbf{D}\} \tag{23}$$

$$\mathcal{C}_{\mathbf{T}} = \{\emptyset\} \tag{24}$$

$$\mathcal{U}_{\mathbf{T}} = \mathbb{R} \times \mathbb{R}^{+} \times \{\emptyset\} \times \mathbb{R}^{2} \tag{25}$$

$$\mathcal{B}_{\mathbf{T}} = \left\{ t, R, \begin{pmatrix} i(t) \\ u(t) \end{pmatrix} \middle| \; \exists \begin{pmatrix} \mathbf{D}.i1 \\ \mathbf{D}.i2 \\ \mathbf{D}.v1 \\ \mathbf{D}.v2 \end{pmatrix} \; \textit{with:} \; \begin{bmatrix} u(t) = \mathbf{D}.v_1(t) - \mathbf{D}.v_2(t) \\ i(t) = \mathbf{D}.i_1(t) \\ u(t) = R \cdot i(t) \end{bmatrix} \right\}, \tag{26}$$

where $\mathbf{O}.x$ *denotes a variable* x *of the object class* $\mathbf{O}$*.*

Example 3.3 ***Control valve****. Figure 3 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:*

$$\begin{aligned} \mathbf{V} &= (\mathcal{C}_{\mathbf{V}}, \mathcal{H}_{\mathbf{V}}, \mathcal{U}_{\mathbf{V}}, \mathcal{B}_{\mathbf{V}}) \\ \mathcal{H}_{V} &= \{\emptyset\} \\ \mathcal{C}_{\mathbf{V}} &= \{A1 : \mathbf{A}, B1 : \mathbf{B}\} \\ \mathcal{U}_{\mathbf{V}} &= \mathbb{R} \times \{\emptyset\} \times \{\emptyset\} \times \{\emptyset\} \\ \mathcal{B}_{\mathbf{V}} &= \left\{ t \; \middle| \; \exists \begin{pmatrix} A1.x(t) \\ B1.x(t) \end{pmatrix} \; \textit{with: } B1.x(t) = A1.x(t) \right\} \end{aligned} \tag{27}$$

$A1$ *is an instance of the object-class* actuator $\mathbf{A}$ *and* $B1$ *an instance of the object-class* body $\mathbf{B}$*.* $A1.x(t)$ *and* $B1.x(t)$ *denote two different variables: the position of the actuator stem and the relative position of the plug/seat respectively. The behavioral equation of* $\mathcal{B}_{\mathbf{V}}$ *corresponds to the mechanical coupling* *between the plug of the valve and the actuator stem. The complete behavioral representation of* **B** *is defined by:*

$$\begin{aligned} \mathbf{B} &= (\mathcal{U}^*_{\mathbf{B}}, \mathcal{B}^*_{\mathbf{B}}) \qquad (28) \\ \mathcal{U}^*_{\mathbf{B}} &= \mathbb{R} \times \mathbb{R}^+ \times \{\emptyset\} \times (\mathbb{R}^3 \times \mathbb{R}^{2+}) \end{aligned}$$

$$\mathcal{B}^*_{\mathbf{B}} = \left\{ t, K_v, \begin{pmatrix} x(t) \\ Q_1(t) \\ Q_2(t) \\ P_1(t) \\ P_2(t) \end{pmatrix} \middle| \begin{array}{l} Q_1(t) = x(t) \cdot K_v \cdot (P_1(t) - P_2(t)) \\ Q_2(t) = -Q_1(t) \end{array} \right\} \qquad (29)$$

The valve coefficient K_v is a parameter and $P_1(t)/P_2(t), Q_1(t)/Q_2(t)$ denote the upstrem/downstream pressures and flow rates of the valve body. The complete behavioral representation of **A** *of the actuator is defined by :*

$$\begin{aligned} \mathbf{A} &= (\mathcal{U}^*_{\mathbf{A}}, \mathcal{B}^*_{\mathbf{A}}) \qquad (30) \\ \mathcal{U}^*_{\mathbf{A}} &= \mathbb{R}^+ \times \{\emptyset\} \times \{\emptyset\} \times \mathbb{R}^2 \\ \mathcal{B}^*_{\mathbf{A}} &= \left\{ t, \begin{pmatrix} x(t) \\ u(t) \end{pmatrix} \middle| \; x(t) = g(t, u) \right\} \end{aligned}$$

$u(t)$ denotes the control signal of the actuator. The behavioral model of the actuator is given by the characteristic law $f(\cdot)$ plotted in figure (3(b)). According to the definitions of **B** *and* **A***, and the definition of the composition relationship, the complete behavioral representation of* **V** *is given by:*

$$\mathcal{U}^*_{\mathbf{V}} = \mathbb{R}^+ \times \mathbb{R} \times \{\emptyset\} \times (\mathbb{R}^5 \times \mathbb{R}^{2+}).$$

$$\mathcal{B}^*_{\mathbf{V}} = \left\{ t, K_v, \begin{pmatrix} A1.u(t) \\ A1.x(t) \\ B1.x(t) \\ B1.Q_1(t) \\ B1.Q_2(t) \\ B1.P_1(t) \\ B1.P_2(t) \end{pmatrix} \middle| \; \mathcal{E}^*_{\mathbf{V}} \right\}$$

$$\mathcal{E}^*_{\mathbf{V}} = \begin{pmatrix} A1.x(t) = B1.x(t) \\ B1.Q_1(t) = B1.x(t) \cdot B1.K_v \cdot (B1.P_1(t) - B1.P_2(t)) \\ B1.Q_2(t) = -B1.Q_1(t) \\ A1.x(t) = g(t, A1.u) \end{pmatrix}, \qquad (31)$$

where $\mathcal{E}^*_{\mathbf{V}}$ denotes the complete system of behavioral equations of **V**.

Conjecture 3.3 *Given definitions 3.1 and 3.2, any partial behavioral representation of an object class* **O** *may be transformed into a complete representation if the behavioral representations of components and super-classes of* **O** *are known.*

3.3 Compositions of object-relationships

Composition and generalization relationships can be combined with each other.

- Let $\mathcal{C}(\cdot)$ be a composition application such that if $A \in \mathcal{C}_O$ then $\mathbf{O} = \mathcal{C}(\mathbf{A})$ means that $\mathbf{O}$ is composed of $\mathbf{A}$. 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 $\mathbf{A} \in \mathcal{H}_{\mathbf{O}}$ 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.

Two possible compositions of those object applications are possible.

- $\mathbf{O} = \mathcal{C} \circ \mathcal{H}(\mathbf{A})$ implies that $\mathbf{O}$ is an object class composed of an intermediate class $\mathbf{B} = \mathcal{H}(\mathbf{A})$ a specialised class of $\mathbf{A}$.
- $\mathbf{O} = \mathcal{H} \circ \mathcal{C}(\mathbf{A})$ implies that $\mathbf{O}$ is a specialised class of $\mathbf{B} = \mathcal{C}(\mathbf{A})$ which is composed of the object class $\mathbf{A}$.

In both cases, the complete behavioral representation of $\mathbf{O}$ is the same and is given by:

$$\mathbf{O} = (\mathcal{U}^*_{\mathbf{O}}, \mathcal{B}^*_{\mathbf{O}}) \tag{32}$$

$$\mathcal{U}^*_{\mathbf{O}} = \mathcal{T}_{\mathbf{O}} \times \mathcal{P}^*_{\mathbf{O}} \times \mathcal{W}^*_{\mathbf{O}} \times \mathcal{L}^*_{\mathbf{O}} \tag{33}$$

$$\mathcal{P}^*_{\mathbf{O}} = \mathcal{P}_{\mathbf{O}} \times \mathcal{P}^*_{\mathbf{B}} = \mathcal{P}_{\mathbf{O}} \times \mathcal{P}_{\mathbf{B}} \times \mathcal{P}^*_{\mathbf{A}} \tag{34}$$

$$\mathcal{W}^*_{\mathbf{O}} = \mathcal{W}_{\mathbf{O}} \times \mathcal{W}^*_{\mathbf{B}} = \mathcal{W}_{\mathbf{O}} \times \mathcal{W}_{\mathbf{B}} \times \mathcal{W}^*_{\mathbf{A}} \tag{35}$$

$$\mathcal{L}^*_{\mathbf{O}} = \mathcal{L}_{\mathbf{O}} \times \mathcal{L}^*_{\mathbf{B}} = \mathcal{L}_{\mathbf{O}} \times \mathcal{L}_{\mathbf{B}} \times \mathcal{L}^*_{\mathbf{A}} \tag{36}$$

$$\mathcal{B}^*_{\mathbf{O}} = \mathcal{B}_{\mathbf{O}} \cap \mathcal{B}^*_{\mathbf{B}} = \mathcal{B}_{\mathbf{O}} \cap \mathcal{B}_{\mathbf{B}} \cap \mathcal{B}^*_{\mathbf{A}} \tag{37}$$

This result is directly deduced from the definition of composition and generalization relationships, *c.f.* equations (14) and (17), applied successively to $\mathbf{O}$ and $\mathbf{B}$ by assuming that $\mathbf{B}$ has only one component ($\mathbf{A}$) and no super-class.

4 Object-oriented modeling of interconnected dynamic systems

The interconnection of dynamic systems may be regarded in the object-oriented paradigm as the composition of objects.

Definition 4.1 *Let* $\mathbf{S}$ *be the object class of a dynamic system defined over a time range* $\mathcal{T}_{\mathbf{S}}$ *and composed of interconnected components. Its partial behavioral representation is defined by:*

$$\mathbf{S} = (\mathcal{C}_{\mathbf{S}}, \mathcal{H}_{\mathbf{S}}, \mathcal{U}_{\mathbf{S}}, \mathcal{B}_{\mathbf{S}}) \tag{38}$$

$$\mathcal{H}_{\mathbf{S}} = \{\emptyset\} \tag{39}$$

$$\mathcal{C}_{\mathbf{S}} = \{C_1, \cdots, C_n\} \tag{40}$$

$$\mathcal{U}_{\mathbf{S}} = \mathcal{T}_{\mathbf{S}} \tag{41}$$

$$\mathcal{B}_{\mathbf{S}} = \{t \,|\, \exists w' \in (\mathcal{W}_{C_1} \times \cdots \times \mathcal{W}_{C_n}) \text{ with: } f(t, w') = 0\}\,, \tag{42}$$

where $C_1, \cdots, C_n$ are objects associated with the system components and $f(\cdot)$ contains all the interconnection equations between the external variables w' of the components.

Example 4.1 ***Electrical circuit.***

Let $\boldsymbol{C}$ be the object class of an electrical circuit depicted in figure 4(a) by the interconnection of two electrical components: a battery $\mathbf{B}$ *and a resistor* $\mathbf{R}$*. The partial behavioral representation of $\boldsymbol{C}$ is defined as follows:*

$$\mathbf{C} = (\mathcal{C}_{\mathbf{C}}, \mathcal{H}_{\mathbf{C}}, \mathcal{U}_{\mathbf{C}}, \mathcal{B}_{\mathbf{C}}) \tag{43}$$

$$\mathcal{H}_{\mathbf{S}} = \{\emptyset\} \tag{44}$$

$$\mathcal{C}_{\mathbf{S}} = \{B1 : \mathbf{B}, R1 : \mathbf{R}\} \tag{45}$$

$$\mathcal{U}_{\mathbf{S}} = \mathbb{R} \times \{\emptyset\} \times \{\emptyset\} \times \{\emptyset\} \tag{46}$$

$$\mathcal{B}_{\mathbf{S}} = \left\{ t \;\middle|\; \exists \begin{pmatrix} B1.U(t) \\ B1.i(t) \\ R1.U(t) \\ R1.i(t) \end{pmatrix} \; \textit{with:} \; \begin{bmatrix} B1.U(t) = R1.U(t) \\ B1.i(t) = R1.i(t) \end{bmatrix} \right\}$$

$\mathbf{R}$ *and* $\mathbf{B}$ *are given by:*

$$\mathbf{R} = (\mathcal{U}_R^*, \mathcal{B}_R^*) \tag{47}$$

$$\mathcal{U}_{\mathbf{R}}^* = \mathbb{R} \times \mathbb{R}^+ \times \{\emptyset\} \times \mathbb{R}^2 \tag{48}$$

$$\mathcal{B}_{\mathbf{R}}^* = \left\{ t, R, \begin{pmatrix} U(t) \\ i(t) \end{pmatrix} \middle| \; U(t) = R \cdot i(t) \right\}. \tag{49}$$

$$\mathbf{B} = (\mathcal{U}_B^*, \mathcal{B}_B^*) \tag{50}$$

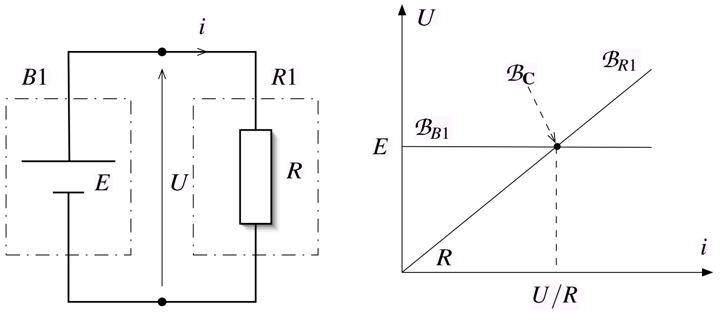
$$\mathcal{U}_{\mathbf{B}}^* = \mathbb{R} \times \mathbb{R}^+ \times \{\emptyset\} \times \mathbb{R}^2 \tag{51}$$

$$\mathcal{B}_{\mathbf{B}}^* = \left\{ t, E, \begin{pmatrix} U(t) \\ i(t) \end{pmatrix} \middle| \; U(t) = E \quad \forall i(t) \right\}. \tag{52}$$

According to the definition of the instantiation, composition and generalization relationships, the complete behavioral representation of $\mathbf{C}$ *is then given by:*

$$\begin{aligned} \mathcal{U}_{\mathbf{C}}^* &= \mathbb{R} \times \mathbb{R}^{2+} \times \{\emptyset\} \times \mathbb{R}^4. \\ \mathcal{B}_{\mathbf{C}}^* &= \left\{ t, \begin{pmatrix} B1.E \\ R1.R, \end{pmatrix}, \begin{pmatrix} B1.U(t) \\ B1.i(t) \\ R1.U(t) \\ R1.i(t) \end{pmatrix} \middle| \; \mathcal{E}_{\mathbf{C}}^* \right\} \\ \mathcal{E}_{\mathbf{C}}^* &= \begin{pmatrix} B1.U(t) = R1.U(t) \\ B1.i(t) = R1.i(t) \\ B1.U(t) = B1.E \quad \forall B1.i(t) \\ R1.U(t) = R1.R \cdot R1.i(t) \end{pmatrix}, \end{aligned} \tag{53}$$

As illustrated in figure 4(b), the complete behavioral model of $\mathbf{C}$*,* i.e. *the solution of $\mathcal{E}_{\mathbf{C}}^*$, may be regarded as the intersection of the behavioral models of its components $B1$ and $R1$,* c.f. *equation (17).*



(a) electrical circuit

(b) intersection of behavioral models

Figure 4: Interconnection of electrical components

5 Link between behavioral representations and Modelica©

Table 2 presents the implementation of a model class **O** into the object-oriented language Modelica [Elmqvist et al., 1999]. **O** is defined by its partial behavioral representation :

$$\mathbf{O} = (\mathcal{C}_{\mathbf{O}}, \mathcal{H}_{\mathbf{O}}, \mathcal{U}_{\mathbf{O}}, \mathcal{B}_{\mathbf{O}}) \tag{54}$$

$$\mathcal{H}_{\mathbf{O}} = \{\mathbf{A}\} \tag{55}$$

$$\mathcal{C}_{\mathbf{O}} = \{B1 : \mathbf{B}\} \tag{56}$$

$$\mathcal{U}_{\mathbf{O}} = \mathbb{R} \times \mathbb{R} \times \mathbb{R} \times \mathbb{R} \tag{57}$$

$$\mathcal{B}_{\mathbf{O}} = \{t, p, l, w \,|\, \exists w' \in (\mathcal{W}_{\mathbf{A}} \times \mathcal{W}_{\mathbf{B}}) \text{ with: } f(t, p, l, w, w') = 0\}\,. \tag{58}$$

O is an subclass of **A** and is composed of an object $B1$. **O** has one parameter p, one internal variable l and one external variable w. w' denotes the vector of external variables of **A** and $B1$. The instruction **time** corresponds to the time variable t. Note that the structure of the Modelica model is split up into four parts corresponding to the sets $\mathcal{H}_{\mathbf{O}}, \mathcal{C}_{\mathbf{O}}, \mathcal{U}_{\mathbf{O}}$ and $\mathcal{B}_{\mathbf{O}}$. Accordingly, an automatic Modelica code generator from behavioral representations is possible.

6 Conclusion

This paper presents a mathematical interpretation of the object-oriented modeling paradigm inspired from the Willems' behavioral approach of systems theory. The object modeling of interconnected dynamic systems is expressed as a set computation problem. This result is a first step that could enhance development of new identification methods of object-oriented models based on set computation techniques.

Table 2: Implementation of a partial behavioral representation into Modelica

```
class O
  // 1: INHERITANCE SET:          $\mathcal{H}_{\mathbf{O}} = \{\mathbf{A}\}$
  extend A
  // 2: COMPOSITION SET:          $\mathcal{C}_{\mathbf{O}} = \{B1 : \mathbf{B}\}$
  B $B1$;
  // 3: UNIVERSAL SET:            $\mathcal{U}_{\mathbf{O}} = (\mathcal{T}_{\mathbf{O}}, \mathcal{P}_{\mathbf{O}}, \mathcal{L}_{\mathbf{O}}, \mathcal{W}_{\mathbf{O}})$
  parameter p;                    $\mathcal{P}_{\mathbf{O}}$
  local Real l;                   $\mathcal{L}_{\mathbf{O}}$
  Real w;                         $\mathcal{W}_{\mathbf{O}}$
  // 4: BEHAVIORAL MODEL:         $\mathcal{B}_{\mathbf{O}}$
equation
  $f(\mathbf{time}, p, l, w, w') = 0$;
end O
```

References

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

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

[Borutzky, 1999b] Borutzky, W. (1999b). 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.

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

[Cellier, 1991] Cellier, F. E. (1991). *Continuous System Modeling*. Springer-Verlag.

[Elmqvist, 1978] 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).

[Elmqvist et al., 1999] Elmqvist, H., Brück, D., and Otter, M. (1999). *Dymola - Dynamic Modeling Laboratory. User's Manual*. Dynasim AB.

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

[Jaulin et al., 2001] Jaulin, L., Kieffer, M., Didrit, O., and Walter, E. (2001). *Applied Interval Analysis*. Springer.

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

[Otter and Elmqvist, 1997] Otter, O. and Elmqvist, H. (1997). Energy flows modeling of mechatronic systems via object diagrams. In *Proc. of the 2nd MATHMOD*, pages 705–710, Vienna.

[Polderman and Willems, 1998] Polderman, J. W. and Willems, J. C. (1998). *Introduction to Mathematical Systems Theory - A Behavioral Approach.* Texts in Applied Mathematics, 26. Springer.

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

[Willems, 1986] 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, 1991] Willems, J. C. (1991). Paradigms and puzzles in the theory of dynamical systems. *IEEE Transactions on Automatic Control*, 36(3):259–294.

Biography

Thierry Bastogne was born in 1968 in Vitry-le-François, France. He received the M.Sc. degree in Computer Engineering and Optoelectronics from Henri Poincaré Université, Nancy 1, France in 1992. From 1992 to 1993 he was an engineer in the Electro-Industrie company, Saint-Dizier, France. He received the M.Sc. and Ph.D. degrees in Automatic Control Engineering from Henri Poincaré Université in 1994 and 1997 respectively. Since 1998, he has been an assistant professor in the Automatic Control Department of Henri Poincaré Université and a member of the Research Center for Automatic Control of Nancy (CRAN). His current research interests are in issues of parameter estimation and system identification and their applications in systems biology and biomedical engineering.