

A note on reactive transitions and Reo connectors

Daniel Figueiredo¹ and Manuel A. Martins¹ and Luís S. Barbosa²

¹ CIDMA — Universidade de Aveiro
Aveiro, Portugal

daniel.figueiredo@ua.pt, martins@ua.pt

² HASLab — INESC TEC & QUANTALab, Universidade do Minho
UNU-EGOV, United Nations University
Portugal
lsb@di.uminho.pt

Abstract. The structure of a reactive transition system can to be modified on the fly by e.g. removing, reversing or adding new transitions. The topic has been studied by D. Gabbay and his collaborators in different contexts. In this paper we take their work a step further, introducing a suitable notion of bisimulation and obtaining a Hennessy-Milner theorem with respect to a hybrid logic in which transition properties can be expressed. Our motivation is to provide a characterisation of equivalence for such systems in order to exploit their possible roles in the formal description of software connectors in Reo, either from a behavioural (semantic) or spatial (syntactic) point of view.

This paper is dedicated to Farhad Arbab, on the occasion of his retirement, as a tribute of gratitude for his outstanding contributions to the field of systems' interaction and composition, his inspiring attitude, and generosity.

1 Introduction

Complex, distributed systems require reliable and yet flexible architectures. A clear separation between typical *loci* of computation (e.g. services or components) and the protocols that manage their interaction is at the heart of the so-called *exogenous coordination* models and, in particular, of Farhad Arbab's outstanding contributions to Software Engineering [1, 2]. Actually, Reo connectors mediate interaction, offering a powerful “glue-code” to express such protocols, while maintaining the envisaged separation of concerns. Moreover, Reo connectors are compositional, providing a very flexible approach to software composition. Among languages with a similar purpose, Reo is the only one that allows for propagation of mutual exclusion and synchrony requirements along the connector structure.

Different forms of transition systems have been used as semantic domains for connector behaviour [14], either directly (as in, e.g. constraint [5]) or Reo

automata [8]), or indirectly through mappings to process algebra formalisms [6, 15]. Typically, such systems are then regarded as (variants of) Kripke frames upon which (variants of) modal logics are interpreted providing a framework to reason about coordination semantics. This is well-known and will not be further detailed here.

From a different point of view, Reo connectors are syntactically represented as graphs of communication primitives (e.g. channels) whose nodes stand for interaction points. Edges are labelled with channel identifiers and types which classify their behaviour. Again such graphs can be regarded as Kripke frames expressing the spatial structure of coordination patterns. This perspective was introduced by N. Oliveira and L. S. Barbosa [17] when proposing an elementary framework for expressing reconfigurations of the interaction protocols, i.e. of the connector’s structure, as discussed, for instance, by C. Krause [16]. Reconfigurations in that sense may substitute, add or remove communication channels, or move communication interfaces between components, in order to restructure a complex interaction policy. The corresponding modal logic expresses properties of (spatial) connector structure.

From this perspective, the focus is placed on the interconnection structure, with no reference to the connector’s emerging behaviour. Examples of structural, or ‘syntactic’ properties are:

- i) every fifo_e channel from a node n is connected to at least a lossy channel,
- ii) node i is an output node of the connector.

In [17] it is required that a reconfiguration preserves such properties.

Often structural properties are to be formulated relatively to a particular node in the pattern. An example is given by property *ii*) above. In general, one may require, for instance, that all the channels incident to a specific node and their interconnections remain unchanged under a reconfiguration. This justifies the choice of hybrid logic [7, 9] to express such properties. In general, hybrid logic adds to a modal language the ability to name, or to explicitly refer to specific states of the underlying Kripke structure. This is done through the introduction of propositional symbols of a new sort, called nominals, each of which is true at exactly one possible state. The sentences are then enriched in two directions. On the one hand, nominals are used as simple sentences holding exclusively in the state they name. On the other hand, explicit reference to states is provided by a satisfaction operator $@$ such that $@_i\phi$ asserts the validity of ϕ at the state named i . In the logic described in next section, properties i) and ii) above are written³ as

$$@_n[\text{fifo}_e]\langle\text{lossy}\rangle\top$$

and

$$@_i[-]\perp$$

respectively, where $[-]\perp$ states the absence of outgoing channels from the node referred by nominal i .

³ As used in modal logics coming from process algebras, modalities are indexed by sets of labels, with symbol “-” standing for the whole set of those.

The starting point for this paper is that the description of coordination elements at any level (*behavioural* or *spatial*, *semantic* or *syntactic*) may be enriched, and become more expressive if the underlying transition system exhibits a reactive structure. The qualifier *reactive* classifies a frame, or a transition system, in which some transitions may inhibit others to occur. The study of this sort of structures, and of their corresponding logics, goes back to the seminal paper of Johan van Benthem [18] which introduces what was then called *sabotage logic*. In this language crossing a specifically annotated edge would erase an edge from the underlying Kripke frame. Other variants of reactivity encompass different effects, for example creating new edges [3], or reversing their direction [4]. The topic has received some attention along the last 10 years — see e.g. [12] for a detailed account.

This paper illustrates the use of reactive transition systems to specify Reo connectors in both perspectives, behavioural and spatial, mentioned above. However, to make this a useful feature in practice, one needs to have at hand the tools typically used to reason about transitions, in particular a notion of bisimulation, a logic and a Hennessy-Milner-like theorem relating model bisimilarity and logical equivalence. Such is actually the paper’s contribution, adding to the theory of reactive frames developed within the modal logic community.

The remaining of the paper is organised as follows. The next section introduces a hybrid logic for reactive transition systems and illustrates the application of such systems to connector modelling. Section 3 contains the core results of the paper, introducing a suitable notion of bisimulation and proving the corresponding Hennessy-Milner theorem for the logic. Finally, section 4 concludes and suggests a few topics for future work.

2 A hybrid logic for reactive transitions

Figure 1 illustrates how reactive transition systems may be used in the context of connector specification, leading to short and crisp descriptions. On the left hand side is depicted the structure of a merger-broadcaster connector intermediated by a lossy channel; edges in the graph are labelled with the type and identifier of constituent channels. The arrow connecting channel sync_b to sync_d inhibits a transmission on the latter whenever a data token has entered the circuit through the former.

The system on the right, on the other hand, expresses the semantics of a synchronous channel \leftrightarrow whose both ends can either receive or deliver data. Ignoring all the double arrows the behaviour of such a connector would simply be to accept a token at end a (represented by label in_a) and deliver it at b (out_b), or the other way round. As designed, however, a strict alternating discipline is enforced. Actually, crossing transition in_a has two side effects: inhibits itself and, at the same time, removes the inhibition affecting transition in_b which will be selected in the next acceptance round.

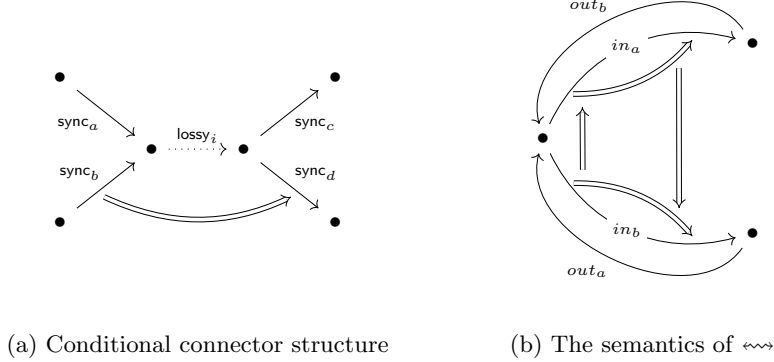


Fig. 1. Reactive transition in Reo connectors.

A (hybrid) logic to talk about this kind of systems is specified as follows. Note the presence of two kinds of modalities ($\langle a \rangle$ and \Box_P) as well as a hybrid satisfaction operator ($@_i$).

Definition 1. Given sets Π , NOM , and A , of atomic propositions, nominals, and labels respectively, the set of $\mathcal{H}\mathcal{L}_r$ -formulas is defined inductively as follows:

$$\varphi ::= i \mid p \mid \neg\varphi \mid \varphi \vee \varphi \mid @_i\varphi \mid \langle a \rangle\varphi \mid \Diamond_P\varphi$$

for any $i \in \text{NOM}$, $p \in \Pi$ and $a \in A$. As usual, connectives \perp , \top , \wedge , \rightarrow and \leftrightarrow for $a \in A$ and $\Box_P, [a]$ are introduced as abbreviations.

As discussed below, formulas are interpreted over paths. Modality $\Diamond_P\varphi$ and its dual, \Box_P , are blind for the path taken to arrive to the current state. They are necessary because in a reactive system the accessibility relation may change on the fly. Actually, differently from what happens in classical modal logic, the evaluation of a proposition p at a world w depends on the path taken to reach w . In general, $(W, R), w \models \langle a \rangle p \Leftrightarrow (W, R'), w' \models p$ for some edge w, w' labelled by a , where R' is the accessibility relation obtained after crossing the edge $(w, w') \in R$.

Example 1. Formula $[sync_a][lossy_i]\langle sync_d \rangle \top$ characterises a property valid for the system depicted in Fig. 1(a): the connector allows data to flow through channels $sync_a$, $lossy_i$, and $sync_d$ in sequence. If channel $sync_b$ was taken instead, it would no longer be possible to use $sync_d$ after $lossy_i$. The fact is captured by the formula $[sync_b][lossy_i][sync_d]\perp$. Finally both properties can be combined in one formula resorting to modality \Diamond_P : The formula

$$[sync_a][lossy_i](\langle sync_d \rangle \top \wedge \Diamond_P[sync_d]\perp)$$

records the possibility of data flowing through channel $sync_d$ for data items coming from $sync_a$, as well as the fact that there exists another possible edge leading to the same state, but which makes impossible to cross $sync_d$.

Formally, let W be a nonempty set of vertices (or states), A a set of labels, and denote by $(A \times W)^*$ the set of all nonempty finite sequences (i.e. *paths*) over $A \times W$. Then,

Definition 2. A reactive frame with labels is a set of finite paths $\Delta \subseteq W \times (A \times W)^*$ such that $(w) \in \Delta$ for any $w \in W$, and $(w_1, ((a_2, w_2), \dots, (a_n, w_n))) \in \Delta$ whenever $(w_1, ((a_2, w_2), \dots, (a_n, w_n), (a_{n+1}, w_{n+1}))) \in \Delta$, for every $n \geq 1$.

Path composition is denoted by juxtaposition; if no ambiguity arises, w and (a, w) will denote the corresponding singleton paths. Given a set W and a nonempty set of paths $\Delta \subseteq W \times (A \times W)^*$, function $t : \Delta \rightarrow W$ returns $t(w) = w$ and $t(w_1, \dots, (a_n, w_n)) = w_n$. In practice, t returns the last state in the path.

Definition 3. Let Π be a set of atomic propositions. A reactive model with labels is a triple (W, Δ, V) , where (W, Δ) is a reactive frame with labels, and $V : \Pi \cup \text{NOM} \rightarrow 2^\Delta$ is a function such that $\lambda \in V(p)$ iff $t(\lambda) \in V(p)$ for any $p \in \Pi$, $\lambda \in \Delta$ and $|V(i)| = 1$ for any $i \in \text{NOM}$.

We may now define how to evaluate $\mathcal{H}\mathcal{L}_r$ -formulas with respect to a reactive model with labels $M = (W, \Delta, V)$, at a path $\lambda \in \Delta$:

Definition 4. The validity of a $\mathcal{H}\mathcal{L}_r$ -formula is established recursively:

- $M, \lambda \models p$ iff $\lambda \in V(p)$, for any $p \in \Pi$
- $M, \lambda \models i$ iff $\lambda \in V(i)$, for any $i \in \text{NOM}$
- $M, \lambda \models \neg\varphi$ iff $M, \lambda \not\models \varphi$
- $M, \lambda \models \varphi \vee \psi$ iff $M, \lambda \models \varphi$ or $M, \lambda \models \psi$
- $M, \lambda \models @_i\varphi$ iff $M, \gamma \models \varphi$, where $V(i) = \{\gamma\}$
- $M, \lambda \models \langle a \rangle\varphi$ iff $\exists w \in W$, $\lambda(a, w) \in \Delta$ and $M, \lambda(a, w) \models \varphi$
- $M, \lambda \models \diamond_P\varphi$ iff $\exists \gamma \in \Delta$, $M, \gamma \models \varphi$ and $t(\lambda) = t(\gamma)$

Both modalities and the hybrid satisfaction operator are interpreted over paths rather than individual states. For modality \diamond_P note that a path $\gamma \in \Delta$ is considered accessible from another one $\lambda \in \Delta$ if and only if their final state coincides.

Example 2. Consider the reactive model $M = (W, \Delta, V)$ with labels depicted in Figure 2. The set of paths Δ is built as follows. From w_4 no move is possible, therefore, $w_4 \in \Delta$. From w_3 , one may move to w_4 through an edge labeled by a , thus w_3 and $(w_3, (a, w_4)) \in \Delta$. Similarly, starting from w_2 leads to w_4 , $(w_2, (a, w_3)), (w_2, (a, w_3), (a, w_4)) \in \Delta$. Finally, from w_1 one may move to w_3 , through an edge labeled by a . However, afterwards, it is not possible to go to w_4 because the edge (w_3, a, w_4) was inhibited when edge (w_1, a, w_3) was crossed. This is the effect represented by the double arrow. Therefore, $w_1, (w_1, (a, w_3)) \in \Delta$. Δ contains no other paths. Clearly, $M, w_2 \models \langle a \rangle \langle a \rangle \top$ but $M, w_1 \not\models \langle a \rangle \langle a \rangle \top$.

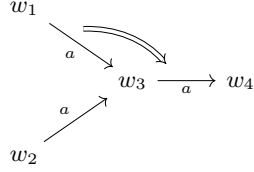


Fig. 2. A reactive frame (W, Δ) .

3 Bisimulation for reactive models with labels

This section characterises a notion of bisimulation to compare reactive models with labels. Its relationship with the modal equivalence induced by the logic above is discussed.

Definition 5. Let (W, Δ, V) and (W', Δ', V') be reactive models over a set of labels A . A relation $\mathcal{S} \subseteq \Delta \times \Delta'$ is an \mathcal{H} -bisimulation if and only if

(nom) for any $i \in \text{NOM}$, $w \in V(i)$ and $v \in V'(i)$ implies $(w, v) \in \mathcal{S}$

and, for all $\lambda \in \Delta, \lambda' \in \Delta'$, such that $(\lambda, \lambda') \in \mathcal{S}$,

(atom) $V(p)(\lambda) = V'(p)(\lambda')$, for all $p \in \Pi \cup \text{NOM}$

(A-zig) $\forall a \in A \left(\forall w \in W (\lambda(a, w) \in \Delta \Rightarrow \exists w' \in W', \lambda'(a, w') \in \Delta' \text{ such that } (\lambda(a, w), \lambda'(a, w')) \in \mathcal{S}) \right)$

(A-zag) $\forall a \in A \left(\forall w' \in W' (\lambda'(a, w') \in \Delta' \Rightarrow \exists w \in W, \lambda(a, w) \in \Delta \text{ such that } (\lambda(a, w), \lambda'(a, w')) \in \mathcal{S}) \right)$

(P-zig) $\forall \gamma \in \Delta (t(\lambda) = t(\gamma) \Rightarrow \exists \gamma' \in \Delta' (t(\lambda') = t(\gamma') \text{ and } (\gamma, \gamma') \in \mathcal{S}))$

(P-zag) $\forall \gamma' \in \Delta' (t(\lambda') = t(\gamma') \Rightarrow \exists \gamma \in \Delta (t(\lambda) = t(\gamma) \text{ and } (\gamma, \gamma') \in \mathcal{S}))$

Example 3. Consider the two systems depicted in Fig. 3. Propositions holding at each particular state are listed between square brackets; no nominals are considered. It is easy to verify that relation $\{((w_1), (v_1)), ((w_1, (a, w_2)), (v_1, (a, v_2))), ((w_2), (v_2)), ((w_2, (b, w_3)), (v_2, (b, v_2))), ((w_4), (v_1)), ((w_4, (a, w_3)), (v_1, (a, v_2))), ((w_3), (v_2)), ((w_3, (b, w_2)), (v_2, (b, v_2)))\}$ is an \mathcal{H} -bisimulation.

As expected, bisimilarity entails modal equivalence.

Theorem 1. Let (W, Δ, V) and (W', Δ', V') be two reactive models, let $\lambda \in \Delta, \lambda' \in \Delta'$ and let $\mathcal{S} \subseteq \Delta \times \Delta'$ be an \mathcal{H} -bisimulation. Then $(\lambda, \lambda') \in \mathcal{S}$ implies $M, \lambda \models \varphi \Leftrightarrow M', \lambda' \models \varphi$ for every formula $\mathcal{H}\mathcal{L}_r$ -formula φ .

Proof. The proof proceeds by induction over the structure of formulas. If $\varphi \in \Pi \cup \text{NOM}$, then $M, \lambda \models \varphi \Leftrightarrow M', \lambda' \models \varphi$ by definition of \mathcal{H} -bisimulation. The non-basic cases are presented below, under the hypothesis that $(\lambda, \lambda') \in \mathcal{S}$.

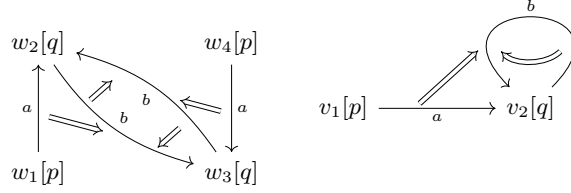


Fig. 3. Two bisimilar models.

- $M, \lambda \models \neg\varphi \Leftrightarrow M, \lambda \not\models \varphi \Leftrightarrow M', \lambda' \not\models \varphi \Leftrightarrow M', \lambda' \models \neg\varphi$.
- $M, \lambda \models \varphi \wedge \psi \Leftrightarrow M, \lambda \models \varphi$ and $M, \lambda \models \psi$
 $\Leftrightarrow M', \lambda' \models \varphi$ and $M', \lambda' \models \psi \Leftrightarrow M', \lambda' \models \varphi \wedge \psi$.
- $M, \lambda \models \langle a \rangle \varphi \Rightarrow \exists w \in W, \lambda(a, w) \in \Delta$ and $M, \lambda(a, w) \models \varphi$. By definition of bisimulation we conclude that $\exists w' \in W', \lambda'(a, w') \in \Delta'$ such that $\lambda(a, w)$ and $\lambda'(a, w')$ are bisimilar. The induction hypothesis entails $\exists w' \in W', \lambda'(a, w') \in \Delta'$ and $M', \lambda'(a, w') \models \varphi$. Finally, $M', \lambda' \models \langle a \rangle \varphi$. The reciprocal condition is proved analogously.
- $M, \lambda \models \Diamond_P \varphi \Rightarrow \exists \gamma \in \Delta, t(\gamma) = t(\lambda)$ and $M, \gamma \models \varphi$. Again, by definition of bisimulation, we conclude that $\exists \gamma' \in \Delta'$ with $t(\gamma') = t(\lambda')$ such that γ' and γ are bisimilar. Thus, by induction, $\exists \gamma' \in \Delta'$ with $t(\gamma') = t(\lambda')$ and $M', \gamma' \models \varphi$, from which we conclude that $M', \lambda' \models \Diamond_P \varphi$. The reciprocal condition is proved analogously.
- $M, \lambda \models @_i \varphi \Leftrightarrow M, \gamma \models \varphi$ such that $V(i) = \{\gamma\}$. From the definition of bisimulation we have $M', \gamma' \models \varphi$ such that $V'(i) = \{\gamma'\}$, and therefore $M', \lambda' \models @_i \varphi$. The reciprocal condition is proved analogously.

□

The reciprocal of Theorem 1 is only valid for a restricted class of models. In classical modal logic a condition stating image-finiteness is usually imposed. We resort here to a slightly more relaxed notion, that of a saturated model [4]. With this restriction, theorem 2 below explains how an \mathcal{H} -bisimulation relating paths indistinguishable by $\mathcal{H}\mathcal{L}_r$ -formulas can be built. In the sequel, for a relation $Z \subseteq \Delta \times \Delta$ on paths, notation $Z[\lambda]$ abbreviates the set $\{\gamma \in \Delta : \lambda Z \gamma\}$.

Definition 6. Let Σ be a set of formulas and $M = (W, \Delta, V)$ a reactive model with labels.

- Σ is satisfiable over a set of paths $\Lambda \subseteq \Delta$ if there is a path $\lambda \in \Lambda$ such that $M, \lambda \models \varphi$ for every $\varphi \in \Sigma$.
- Σ is finitely satisfiable over a set of paths $\Lambda \subseteq \Delta$ if, for every finite subset $\bar{\Sigma} \subseteq \Sigma$, there is a path $\lambda \in \Lambda$ such that $\lambda \models \varphi$ for every $\varphi \in \bar{\Sigma}$.

- A model is Z -saturated over a relation $Z \subseteq \Delta \times \Delta$, if, for all λ , every set Σ is satisfiable over $Z[\lambda]$ whenever Σ is finitely satisfiable over $Z[\lambda]$.

Definition 7. Given a reactive frame (W, Δ) and a set of labels A , relation $R_a \subseteq \Delta \times \Delta$, for each $a \in A$ is defined by $(\lambda, \gamma) \in R_a$ iff $\exists w \in W, \gamma = \lambda(a, w)$. Similarly, define relation $P \subseteq \Delta \times \Delta$ by $(\lambda, \gamma) \in P$ iff $t(\gamma) = t(\lambda)$.

In order to complete the Hennessy-Milner theorem for the presented logic, we state and prove the next theorem. It is not proved for all reactive models but comprise an embracing class of reactive models.

Theorem 2. Let M and M' be two P -saturated and $(R_a)_{a \in A}$ -saturated reactive-models with labels. A non-empty relation $\mathcal{S} \subseteq \Delta \times \Delta'$ such that $(\lambda, \lambda') \in \mathcal{S}$ iff for any formula φ , $M, \lambda \models \varphi \Leftrightarrow M', \lambda' \models \varphi$, is an \mathcal{H} -bisimulation.

Proof. Consider an arbitrary $a \in A$, suppose that $(\lambda, \gamma) \in R_a$, for some $\gamma \in \Delta$ and let $Sat(\gamma) = \{\varphi : M, \gamma \models \varphi\}$. Then, for each finite subset $\Sigma' \subseteq Sat(\gamma)$, $M, \lambda \models \langle a \rangle \bigwedge_{\varphi \in \Sigma'} \varphi$ holds and, therefore, $M', \lambda' \models \langle a \rangle \bigwedge_{\varphi \in \Sigma'} \varphi$. This means that $Sat(\gamma)$ is finitely satisfiable over $R_a(\lambda')$, and since M' is R_a -saturated, $Sat(\gamma)$ is satisfied over $R_a(\lambda')$. Thus, there exists a state γ' such that $(\lambda', \gamma') \in R_a$ and $(\gamma, \gamma') \in \mathcal{S}$. Analogously, if $(\lambda, \lambda') \in \mathcal{S}$ and $(\lambda', \gamma') \in R_a$, then there exists some $w \in W$ such that $(\lambda w, \lambda' w') \in \mathcal{S}$.

Suppose now that $(\lambda, \gamma) \in P$, for some $\gamma \in \Delta$ and consider $Sat(\gamma) = \{\varphi : M, \gamma \models \varphi\}$. Then, for each finite subset $\Sigma' \subseteq Sat(\gamma)$, $M, \lambda \models \diamond_P \bigwedge_{\varphi \in \Sigma'} \varphi$ and,

therefore, $M', \lambda' \models \diamond_P \bigwedge_{\varphi \in \Sigma'} \varphi$. This means that $Sat(\gamma)$ is finitely satisfiable over

$P_{\lambda'}$, and since M' is P -saturated, $Sat(\gamma)$ is satisfied over $P_{\lambda'}$. Again, there exists a state γ' such that $(\lambda', \gamma') \in P$ and $(\gamma, \gamma') \in \mathcal{S}$. Analogously, if $(\lambda, \lambda') \in \mathcal{S}$ and $(\lambda', \gamma') \in P$, then there exists some $\gamma \in \Delta$ such that $(\lambda, \gamma) \in P$ and $(\gamma, \gamma') \in \mathcal{S}$.

Now, let $i \in \text{NOM}$ such that $V(i) = \{\lambda\}$ and $V'(i) = \{\lambda'\}$ for some $\lambda \in \Delta, \lambda' \in \Delta'$. Let $(\gamma, \gamma') \in \mathcal{S} \neq \emptyset$, then, for any \mathcal{HL}_r -formula φ , $M, \gamma \models @_i \varphi \Leftrightarrow M', \gamma' \models @_i \varphi$ that semantically implies, $M, \lambda \models \varphi \Leftrightarrow M', \lambda' \models \varphi$. Since φ is arbitrary, $(\lambda, \lambda') \in \mathcal{S}$. Finally, if $(\lambda, \lambda') \in \mathcal{S}$, then we can trivially verify that, $\forall p \in \Pi \cup \text{NOM}, M, \lambda \models p \Leftrightarrow M', \lambda' \models p$ by definition. □

Clearly the theorem would fail for non R_a -saturated models. The following proposition gives a sufficient condition for a model to be R_a -saturated.

Proposition 1. Let $M = (W, \Delta, V)$ be a reactive model with labels and consider $R_a(\lambda)$ as defined above. If $|R_a(\lambda)| < \infty$, for any $\lambda \in \Delta$, then M is R_a -saturated.

Proof. Suppose $|R_a(\lambda)| < \infty$, for any $\lambda \in \Delta$, holds for M , but the model is not R_a -saturated. This means that there exists $\lambda \in \Delta$ and a set Σ of formulas such that Σ is finitely satisfiable over $R_a(\lambda)$ but not satisfiable over $R_a(\lambda)$.

Clearly, any formula $\varphi \in \Sigma$, $\{\varphi\}$ is satisfiable over $R_a(\lambda)$ which means that $\langle a \rangle \Sigma = \{\langle a \rangle \varphi : \varphi \in \Sigma\}$ is satisfied in λ . Since $|R_a(\lambda)| < \infty$, every path in $R_a(\lambda)$ can be enumerated as $\gamma_1, \dots, \gamma_n$. Since Σ is not satisfiable over $R_a(\lambda)$, there is a formula $\varphi_i \in \Sigma$ for each γ_i , $i \in \{1, \dots, n\}$, such that φ_i is not satisfied in γ_i . However, for any $i \in \{1, \dots, n\}$, $\langle a \rangle \varphi_i$ is satisfied in λ . Thus, the set $\Phi = \{\varphi_1, \dots, \varphi_n\} \subseteq \Sigma$ is finite and, therefore, satisfiable over $R_a(\lambda)$. This leads to a contradiction since each path $\gamma_i \in R_a(\lambda)$ does not verify $\varphi_i \in \Phi$. \square

An analogous result for relation P is obtained along similar lines, but additionally requiring the absence of cycles in the reactive model.

Our last results establish a connection between reactive models, as discussed in this paper, and the usual Kripke models. We start by making explicit how a reactive model arises from a classical one. Note that in the sequel $NOM = \emptyset$ because nominals, in the logic introduced here, bind paths, rather than states as in standard hybrid logic.

Definition 8. *A Kripke model $K = (W_K, R, V_W)$ induces a reactive model $M = (W, \Delta, V)$ as follows:*

- $W = W_K$
- Δ is the set of all possible paths generated by the accessibility relation R , i.e.:
 - $(w) \in \Delta$ for any $w \in W$
 - For any $n \geq 2$, $(w_0, \dots, w_n) \in \Delta$ whenever $\forall i \in \{1, n-1\}, (w_i, w_{i+1}) \in R$
- V is defined in order to be coherent with the notion of valuation for models with no reactivity:
 - $\forall p \in \Pi, \lambda \in V(p)$ iff $t(\lambda) \in V_K(p)$

Theorem 3. *Let $(W_K, R, V_K), (W'_K, R', V'_K)$ be Kripke models and $\mathcal{B} \subseteq W \times W'$ a bisimulation. Let (W, Δ, V) (respectively, (W', Δ', V')) be the induced reactive model with respect to (W, R, V) (respectively, (W', R', V')). Let the relation $\mathcal{S} \subseteq \Delta \times \Delta'$ be such that $(\lambda, \lambda') \in \mathcal{S}$ iff $t(\lambda) \mathcal{B} t(\lambda')$. Then \mathcal{S} is an \mathcal{H} -bisimulation of reactive models (with $NOM = \emptyset$).*

Proof. (nom) Trivial since $NOM = \emptyset$.

(A-zig) Let us consider $\lambda \in \Delta, \lambda' \in \Delta'$ and $w \in W$ such that $(\lambda, \lambda') \in \mathcal{S}$ and $\lambda w \in \Delta$. By definition of \mathcal{S} , we conclude that $(t(\lambda), t(\lambda')) \in \mathcal{B}$ and $(t(\lambda), w) \in R$. Therefore, since \mathcal{B} is a bisimulation, there exists $w' \in W'$ such that $(t(\lambda'), w') \in R'$ and $(w, w') \in \mathcal{B}$. Thus, $\exists w' \in W', (t(\lambda w), t(\lambda' w')) \in \mathcal{B}$ which implies $\exists w' \in W', (\lambda w, \lambda' w') \in \mathcal{S}$.

(A-zag) Analogous to A-zig.

(P-zig) Let us consider $\lambda, \gamma \in \Delta$ and $\lambda' \in \Delta'$ such that $t(\lambda) = t(\gamma)$ and $(\lambda, \lambda') \in \mathcal{S}$. Therefore $(t(\lambda), t(\lambda')) \in \mathcal{B}$ implies $(t(\gamma), t(\lambda')) \in \mathcal{B}$ and, thus, $(\gamma, \lambda') \in \mathcal{S}$. The result follows because, trivially, $t(\lambda) = t(\lambda')$.

(P-zag) Analogous to P-zig.

(*atom*) $M, \lambda \vDash_X p \Leftrightarrow M, t(\lambda) \vDash p$ for any $p \in \Pi$. Thus, if $(\lambda, \lambda') \in \mathcal{S}$, then $M, \lambda \vDash_X p \Leftrightarrow M, t(\lambda) \vDash p \Leftrightarrow M', t(\lambda') \vDash p \Leftrightarrow M', \lambda' \vDash_X p$

□

In the opposite direction a similar result pops out:

Theorem 4. Let $(W_K, R, V_K), (W'_K, R', V'_K)$ be Kripke models and $\mathcal{S} \subseteq \Delta \times \Delta'$ a bisimulation between paths of the corresponding induced reactive models. Define relation $\mathcal{B} \subseteq W \times W'$ by $(w, w') \in \mathcal{B}$ iff there exists $(\lambda, \lambda') \in \mathcal{S}$ such that $t(\lambda) = w$ and $t(\lambda') = w'$. Then \mathcal{B} is a bisimulation between the original Kripke models.

Proof. We prove the *zig* and *zag* conditions, as well as the semantic equivalence between atomic propositions.

(*zig*) Let us suppose $(w, w') \in \mathcal{B}$ and that there exists $v \in W$ such that $(w, v) \in R$. Then, there exists $(\lambda, \lambda') \in \mathcal{S}$ such that $t(\lambda) = w$ and $t(\lambda') = w'$. Furthermore, $\lambda v \in \Delta$. Since \mathcal{S} is a bisimulation, there exists $v' \in W'$ such that $\lambda' v' \in \Delta'$ and $(\lambda v, \lambda' v') \in \mathcal{S}$. Hence, $t(\lambda v) = v$, $t(\lambda' v') = v'$ and, therefore, $v \mathcal{B} v'$.

(*zag*) Analogous to *zig*.

(*atom*) Finally, we note that $M, \lambda \vDash_X p \Leftrightarrow M, t(\lambda) \vDash p$ for any $p \in \Pi$ because $X = \Pi$. If $(w, w') \in \mathcal{B}$, then there exists $(\lambda, \lambda') \in \mathcal{S}$ such that $t(\lambda) = w$ and $t(\lambda') = w'$. Because $M, \lambda \vDash_X p \Leftrightarrow M', \lambda' \vDash_X p$, we conclude that $M, t(\lambda) \vDash p \Leftrightarrow M', t(\lambda') \vDash p$, *i.e.*, $M, v \vDash p \Leftrightarrow M', v' \vDash p$. □

As explained above, the set of nominals was assumed to be empty in the context in which both theorems were formulated. However, if nominals were considered in the language of reactive models, the second theorem would remain valid, but not the first one, as non bisimilar paths may be bound by the same nominal.

4 Conclusions and future work

This paper indicates that *reactive* transition systems may play an interesting role in the formal description of software connectors in **Reo**, either from a behavioural (semantic) or spatial (syntactic) point of view. In the later sense, we are currently enriching our previous work on connector reconfiguration [17] to handle such reactive spatial descriptions of coordination patterns.

A preliminary step for those developments is a suitable characterisation of equivalence. Such is the focus of the technical contents of the present paper. We introduce a notion of bisimulation for reactive transition systems with labels and establish, under reasonable conditions, a Hennessy-Milner-like result. This adds to the quest of D. Gabbay and his collaborators for suitable (logic) tools to specify the dynamics of reactivity. We are currently extending our work to *switch graphs* [13], another sort of reactive frames with an interesting potential for describing coordination patterns, namely in the context of analogues to biological systems — a main topic in the first author's doctoral research [10, 11].

Acknowledgments. This work is a result of project “SmartEGOV / NORTE-01-0145-FEDER-000037”, supported by Norte Portugal Regional Operational Programme (NORTE 2020), under the PORTUGAL 2020 Partnership Agreement, through the European Regional Development Fund (EFDR). Additional support was provided by the European Regional Development Fund through the Operational Programme for Competitiveness and Internationalisation - COMPETE 2020 and by National Funds through the Portuguese funding agency, FCT - Fundação para a Ciência e a Tecnologia within project UID/MAT/04106/2013 at CIDMA. The first author is also supported by an Individual Doctoral Grant (reference number PD/BD/114186/2016).

References

1. Farhad Arbab. Reo: a channel-based coordination model for component composition. *Mathematical Structures in Computer Science*, 14(3):329–366, 2004.
2. Farhad Arbab. Abstract behavior types: a foundation model for components and their composition. *Sci. Comput. Program.*, 55(1-3):3–52, 2005.
3. Carlos Areces, Raul Fervari, and Guillaume Hoffmann. Swap logic. *Logic Journal of IGPL*, 22(2):309–332, 2014.
4. Carlos Areces, Raul Fervari, and Guillaume Hoffmann. Relation-changing modal operators. *Logic Journal of IGPL*, 23(4):601–627, 2015.
5. Christel Baier, Marjan Sirjani, Farhad Arbab, and Jan J. M. M. Rutten. Modeling component connectors in Reo by constraint automata. *Sci. Comput. Program.*, 61(2):75–113, 2006.
6. Marco A. Barbosa and Luís S. Barbosa. A perspective on service orchestration. *Sci. Comput. Program.*, 74(9):671–687, 2009.
7. P. Blackburn. Representation, reasoning, and relational structures: A hybrid logic manifesto. *Logic Journal of IGPL*, 8(3):339–365, 2000.
8. Marcello M. Bonsangue, Dave Clarke, and Alexandra Silva. Automata for context-dependent connectors. In John Field and Vasco Thudichum Vasconcelos, editors, *Coordination Models and Languages, 11th International Conference, COORDINATION 2009, Lisboa, Portugal, June 9-12, 2009. Proceedings*, volume 5521 of *Lecture Notes in Computer Science*, pages 184–203. Springer, 2009.
9. T. Brauner. *Hybrid Logic and its Proof-Theory*. Applied Logic Series. Springer, 2010.
10. Daniel Figueiredo. Relating bisimulations with attractors in boolean network models. In María Botón-Fernández, Carlos Martín-Vide, Sergio Santander-Jiménez, and Miguel A. Vega-Rodríguez, editors, *Algorithms for Computational Biology - Third International Conference, AlCoB 2016, Trujillo, Spain, June 21-22, 2016, Proceedings*, volume 9702 of *Lecture Notes in Computer Science*, pages 17–25. Springer, 2016.
11. Daniel Figueiredo, Manuel A. Martins, and Madalena Chaves. Applying differential dynamic logic to reconfigurable biological networks. *Mathematical Biosciences*, 291:10–20, 2017.
12. Dov Gabbay and Sérgio Marcelino. Modal logics of reactive frames. *Studia Logica*, 93(2):405–446, 2009.
13. Dov Gabbay and Sérgio Marcelino. Global view on reactivity: switch graphs and their logics. *Annals of Mathematics and Artificial Intelligence*, pages 1–32, 2012.
14. Sung-Shik T. Q. Jongmans and Farhad Arbab. Overview of thirty semantic formalisms for Reo. *Sci. Ann. Comp. Sci.*, 22(1):201–251, 2012.

15. Natallia Kokash, Christian Krause, and Erik P. de Vink. Reo + mCRL2: A framework for model-checking dataflow in service compositions. *Formal Asp. Comput.*, 24(2):187–216, 2012.
16. Christian Krause, Ziyang Maraikar, Alexander Lazovik, and Farhad Arbab. Modeling dynamic reconfigurations in Reo using high-level replacement systems. *Sci. Comput. Program.*, 76(1):23–36, 2011.
17. Nuno Oliveira and Luís S. Barbosa. Reasoning about software reconfigurations: The behavioural and structural perspectives. *Sci. Comput. Program.*, 110:78–103, 2015.
18. Johan van Benthem. An essay on sabotage and obstruction. In *Mechanizing Mathematical Reasoning*, pages 268–276. Springer, 2005.