



Ensuring Reachability by Design

Benoit Caillaud, Jean-Baptiste Raclet

► To cite this version:

Benoit Caillaud, Jean-Baptiste Raclet. Ensuring Reachability by Design. [Research Report] RR-7928, INRIA. 2012, pp.1-20. hal-00696151

HAL Id: hal-00696151

<https://inria.hal.science/hal-00696151>

Submitted on 11 May 2012

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Ensuring Reachability by Design

Benoît Caillaud, Jean-Baptiste Raclet

**RESEARCH
REPORT**

N° 7928

May 2012

Project-Team S4



Ensuring Reachability by Design

Benoît Caillaud*, Jean-Baptiste Raclet†

Project-Team S4

Research Report n° 7928 — May 2012 — 20 pages

Abstract: Modular design aims at decomposing systems as a set of distinct components that can be independently developed and then assembled all together. Interfaces are then attached to components; they abstract implementation details while exposing to the environment relevant information about cross-component interactions.

Whereas state-of-the-art on interfaces essentially consider independent implementability of safety properties, we consider in this paper reachability properties, which are in general not compositional. The approach we advocate consists in controlling the design flow of components, that is, the evolution of their interfaces through combinations and refinements, in order to ensure a reachability property by construction.

Modal specifications are widely acknowledged as a suitable specification formalism for interface-based design. In order to obtain the required expressivity, we extend them with marked states to model states to be reached. We then develop an algebra with both logical and structural composition operators ensuring reachability properties by design.

Key-words: component-based design; compositional reasoning; interface theories; modal specifications

* INRIA Rennes - Bretagne, Atlantique, Campus de Beaulieu, F-35042 Rennes cedex, France.
Benoit.Caillaud@inria.fr

† IRIT/CNRS, 118 Route de Narbonne, F-31062 Toulouse cedex 9, France. Jean-Baptiste.Raclet@irit.fr

Garantir l'atteignabilité par construction

Résumé : Les méthodes de conception modulaire procèdent par décomposition d'un système en un ensemble de composants qui seront, dans un premier temps, réalisés indépendamment les uns des autres, pour être par la suite combinés de façon à former un système complet. Il est alors utile d'associer aux composants des interfaces qui servent à abstraire les détails de réalisation, tout en exposant les aspects cruciaux concernant les interactions avec les autres composants et l'environnement. Alors que l'état de l'art sur les interfaces concerne en premier lieu l'implémentation indépendante de propriétés de sûreté, ce rapport porte sur la prise en compte de propriétés d'atteignabilité, qui sont en général non compositionnelles. L'approche défendue ici consiste à contrôler le flot de conception d'un système, c'est à dire, l'évolution des interfaces tout au long des étapes de composition et de raffinement, de telle manière qu'une propriété d'atteignabilité est garantie par construction. Les spécifications modales sont reconnues comme un formalisme d'interface approprié. Elles sont ici étendues par ajout de marquage des états, servant à modéliser l'atteignabilité de certains états. L'algèbre d'interfaces modales marquée, avec ses opérateurs de composition logique et structurelle, permet d'assurer par construction l'atteignabilité de certains états.

Mots-clés : conception par composants; raisonnement compositionnel; théories d'interfaces; spécifications modales

1 Introduction

In order to face the intrinsic complexity of automotive, aeronautic and consumer electronics embedded systems, but also of web-based service oriented architectures, modular design aims at organizing systems as a set of distinct components that can be developed independently and then assembled together. This is best achieved using interfaces which abstract superfluous implementation details of a component and expose cross-component protocol informations that are essential to a correct use of a component. Component reuse in different contexts is thus made possible, not only reducing design time, but also enabling the amortization of design costs over several different projects.

Component interoperability or compatibility is then a major issue: when can we safely compose two (or more) components? Compatibility is often considered at a signature level. In this simple case, interfaces consist in function or method types and compatibility consists in a type-checking, performed either at compile-time or at run-time. This paper deals with a richer notion of interfaces capable of capturing behavioral properties.

The first work on behavioral compatibility of interfaces has been proposed in [1]. This paper considers an automata-based formalism for interfaces in which transitions are labeled with output (produced by the component) or input (produced by the environment) actions. Then, a run-time error occurs whenever a component produces an output that is not accepted as input by one of its peers. The fact that a runtime error may occur does not necessarily lead to deem the interfaces incompatible. Indeed, the authors promotes an optimistic approach of composition in which two interfaces are compatible if there exists a restriction of the permitted actions of the environment in order to prevent the reachability of a runtime error. They show that this form of compatibility is preserved in the design flow provided alternating refinement [2] is used. More precisely, starting from initial interfaces whose product satisfies a particular *safety* property (i.e., a runtime error cannot be reached), they can be refined independently and then composed, their product will also satisfy the same safety property. This principle, called *independent implementability*, is of key importance [12] and enables the concurrent design of systems that are then assembled in a bottom-up manner.

This paper now studies the case of *reachability* properties and proposes results regarding their satisfaction by design. Basically, a reachability property states that some particular situation can be reached. Examples abound in practice. For instance, consider Service Oriented Architectures (SOA) formed of several interacting services; they should always have the possibility to reach a termination state, by delivering a response to all service activation. However, termination is in general not preserved by service composition. Although reachability properties are easy to verify in this context [4], model-checking may not be an appropriate solution. First, because it requires to construct the reachability graph of a system which may lead to a state explosion problem. Moreover, in case model-checking reveals a violation of the reachability property, designers have to iterate the design cycle by re-coding and re-validating their components, therefore extending time to market. The alternative approach advocated in this paper consists in controlling the design flow of components, that is, the evolution of interfaces through compositions and refinements, in order to ensure a reachability property by construction.

Now, what specification formalism capturing some behavioral aspects of components is convenient for interface-based design? Modal specifications [18, 16, 3] are widely acknowledged as a suitable proposal [14, 22, 23]. Basically, they consist in labeling interface transitions with modalities, either *must* if the transition has to be enabled in any refinement, or *may* if the transition is allowed. Recently, they have been extended to a timed setting [5] and also to a quantitative one [13]. In [14, 22, 23], modal specifications are shown to have many benefits comparing the specification formalism introduced in [1]; they are not only equipped with an optimistic compo-

sition operator and a refinement relation but also with a conjunction and a quotient operator. The role of these operators in a design flow will be motivated later in the paper. As reachability properties cannot be expressed, in general, with modal specifications, we first consider in this paper modal specifications enriched with marked states, in the same fashion as it is done in [6]. We show that, in this framework, we can develop a theory ensuring reachability properties by design.

2 Modeling with marked modal specifications

2.1 Background on automata

Let Σ be a finite alphabet of actions, a *deterministic* automaton over Σ is a tuple $\mathcal{M} = (R, r^0, \Sigma, \lambda, G)$ where R is a finite set of states, $r^0 \in R$ is the unique initial state, λ is a partial map from $R \times \Sigma$ to R called the *labeled transition map* and $G \subseteq R$ is a non-empty set of *marked* states. The set of *firable* actions from $r \in R$ is $ready(r) = \{a \in \Sigma \mid \lambda(r, a) \text{ is defined}\}$.

Transition map λ is extended to its transitive and reflexive closure: let ϵ denote the empty word, for all $r \in R$, $\lambda(r, \epsilon) = r$ and for all $u \in \Sigma^*$, $a \in \Sigma$, $r_1, r_2, r_3 \in R$, $\lambda(r_1, u) = r_2$ and $\lambda(r_2, a) = r_3$ imply $\lambda(r_1, u.a) = r_3$. Define $\mathcal{L}_{\mathcal{M}} = \{u \in \Sigma^* \mid \exists r' \in R, \lambda(r^0, u) = r'\}$ to be the *language* of $\mathcal{M}$. If $\lambda(r, u) = r'$ for some u then r' is said to be *reachable* from r . The natural projection of a word $v \in \mathcal{L}_{\mathcal{M}}$ on $\Sigma' \subseteq \Sigma$ is defined inductively with $pr_{\Sigma'}(\epsilon) = \epsilon$ and, if $v = u.a$, $pr_{\Sigma'}(v) = pr_{\Sigma'}(u).a$ when $a \in \Sigma'$ and $pr_{\Sigma'}(v) = pr_{\Sigma'}(u)$ otherwise.

Given $P \subseteq R$, define $pre^*(P)$ and $post^*(P)$ to be the set of states that are respectively coreachable and reachable from any state $r \in P$: it is the least set such that for $r \in P$, $r \in pre^*(P)$ and $r \in post^*(P)$ and for every $\lambda(r', a) = r''$, if $r'' \in pre^*(P)$ then $r' \in pre^*(P)$ and if $r' \in post^*(P)$ then $r'' \in post^*(P)$. With a slight abuse, we may write $pre^*(r)$ and $post^*(r)$ for $pre^*(\{r\})$ and $post^*(\{r\})$.

If modeling a service, it is desirable to set that a service session eventually terminates; this is often referred in SOC as weak termination. To capture this kind of requirement, we define *terminating* automata: an automaton $\mathcal{M}$ is said to be *terminating* whenever $R = pre^*(G)$ meaning that it is always possible to reach a marked state from any state of the automaton. In other words, $\mathcal{M}$ is terminating if and only if for any $u \in \mathcal{L}_{\mathcal{M}}$, there exists a v such that $uv \in \mathcal{L}_{\mathcal{M}}$ and $\lambda(r^0, uv) \in G$. In the temporal logic CTL, this property can be written $AG(EF G)$.

Given two automata $\mathcal{M}_1 = (R_1, r_1^0, \Sigma_1, \lambda_1, G_1)$ and $\mathcal{M}_2 = (R_2, r_2^0, \Sigma_2, \lambda_2, G_2)$, their product is the automaton $\mathcal{M}_1 \times \mathcal{M}_2 = (R_1 \times R_2, (r_1^0, r_2^0), \Sigma_1 \cup \Sigma_2, \lambda, G_1 \times G_2)$ where $\lambda((r_1, r_2), a)$ is defined as $(\lambda_1(r_1, a), r_2)$ for $a \in \Sigma_1 \setminus \Sigma_2$, $(r_1, \lambda_2(r_2, a))$ for $a \in \Sigma_2 \setminus \Sigma_1$ and $(\lambda_1(r_1, a), \lambda_2(r_2, a))$ for $a \in \Sigma_1 \cap \Sigma_2$. As a consequence, the language of $\mathcal{M}_1 \times \mathcal{M}_2$ is the mixed language $\mathcal{L}_{\mathcal{M}_1} \sqcap \mathcal{L}_{\mathcal{M}_2} = \{u \in (\Sigma_1 \cup \Sigma_2)^* \mid pr_{\Sigma_1}(u) \in \mathcal{L}_{\mathcal{M}_1} \text{ and } pr_{\Sigma_2}(u) \in \mathcal{L}_{\mathcal{M}_2}\}$.

2.2 Marked modal specifications

Following [6], we enrich modal specifications [18, 16, 3] with marked states in order to model *states to be reached*. For instance, if a designer specifies a service, this enables to represent session terminations. The obtained formalism allows to specify a (possibly infinite) set of automata called *implementations*.

Definition 1 (Marked Modal Specification) *A marked modal specification over Σ is a tuple $\mathcal{C} = (Q, q^0, \Sigma, \delta, must, may, F)$, where Q is a finite set of states, $q^0 \in Q$ is the unique initial state, $\delta : Q \times \Sigma \rightarrow Q$ is a partial labeled transition map; $must, may : Q \rightarrow 2^\Sigma$ map to each state q the set of required and allowed actions from q , $F \subseteq Q$ is a non-empty set of marked states.*

It is assumed that a transition is associated to any allowed action, that is for every state $q \in Q$ and every action $a \in \Sigma$, $a \in \text{may}(q)$ if and only if $\delta(q, a)$ is defined. The mapping $\text{may} : Q \rightarrow 2^\Sigma$ can thus be reconstructed from the transition relation δ . However, this distinction simplifies the definition of satisfaction and refinement relations and compositions operators.

In this paper, marked modal specifications are taken *deterministic*, that is: for any $a \in \Sigma$ and any state q there is at most one state q' such that $\delta(q, a) = q'$. The reason for this will be given later in Sec. 3.

The *underlying automata* associated to $\mathcal{C}$ is $\text{Un}(\mathcal{C}) = (Q, q^0, \Sigma, \delta, F)$. The language $\mathcal{L}_{\mathcal{C}}$ is then $\mathcal{L}_{\text{Un}(\mathcal{C})}$. As previously for automata, we extend δ to words by taking its transitive and reflexive closure. Moreover, we define $\text{pre}_M^*(P)$ and $\text{pre}_m^*(P)$ with $P \subseteq Q$ as the set of states that are coreachable from any state $q \in Q$ by following transitions labeled by required and allowed actions, respectively: $\text{pre}_m^*(P)$ corresponds to $\text{pre}^*(P)$ in $\text{Un}(\mathcal{C})$; $\text{pre}_M^*(P)$ is the least set such that for $r \in P$, $r \in \text{pre}_M^*(P)$ and for every $\lambda(r', a) = r''$ with $a \in \text{must}(r')$ and $r'' \in \text{pre}^*(P)$ then $r' \in \text{pre}_M^*(P)$. Last, $\text{post}_m^*(P)$ is $\text{post}^*(P)$ in $\text{Un}(\mathcal{C})$.

Any terminating automaton can be seen as a marked modal specification with no design choice left open, that is, for any state r , the optional action set $\text{may}(r) \setminus \text{must}(r)$ is empty. More formally, the *embedding* of a terminating automaton $\mathcal{M} = (R, r^0, \Sigma, \lambda, G)$ into the class of the marked modal specifications is $\text{Em}(\mathcal{M}) = (R, r^0, \Sigma, \lambda, \text{must}, \text{may}, G)$ with, for all $r \in R$, $\text{may}(r) = \text{must}(r) = \text{ready}(r)$. Now, the semantics of marked modal specifications is given in terms of terminating automata:

Definition 2 (Satisfaction) *A terminating automaton $\mathcal{M} = (R, r^0, \Sigma, \lambda, G)$ satisfies the marked modal specification $\mathcal{C} = (Q, q^0, \Sigma, \delta, \text{must}, \text{may}, F)$, denoted $\mathcal{M} \models \mathcal{C}$, if and only if there exists a simulation relation $\pi \subseteq R \times Q$ such that $(r^0, q^0) \in \pi$ and for all pairs $(r, q) \in \pi$:*

1. $\text{must}(q) \subseteq \text{ready}(r) \subseteq \text{may}(q)$;
2. $r \in G$ implies $q \in F$;
3. for every $a \in \Sigma$ and every $r' \in Q$, $\lambda(r, a) = r'$ implies $(r', \delta(q, a)) \in \pi$.

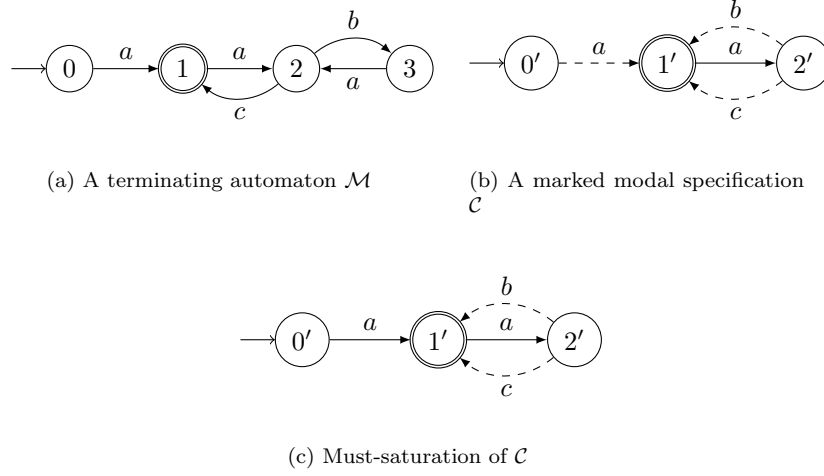
The set of models (or implementations) of $\mathcal{C}$ is denoted $\llbracket \mathcal{C} \rrbracket$. A marked modal specification is said *satisfiable* if and only if $\llbracket \mathcal{C} \rrbracket \neq \emptyset$. Two marked modal specifications $\mathcal{C}$ and $\mathcal{C}'$ are said *equivalent*, written $\mathcal{C} \equiv \mathcal{C}'$, if and only if they admit the same implementations: $\llbracket \mathcal{C} \rrbracket = \llbracket \mathcal{C}' \rrbracket$. Any unsatisfiable specification is mapped on a special specification marked modal specification denoted $\mathcal{C}_\perp$, with $\llbracket \mathcal{C}_\perp \rrbracket = \emptyset$.

Example 3 *Consider the terminating automaton $\mathcal{M}$ in Fig. 1(a) and the marked modal specification $\mathcal{C}$ in Fig. 1(b) where transitions from q labeled by a are dashed when $a \in \text{may}(q) \setminus \text{must}(q)$ and plain when $a \in \text{must}(q)$; marked states are double-circled. $\mathcal{M}$ satisfies $\mathcal{C}$ because of the simulation relation $\pi = \{(0, 0'), (1, 1'), (2, 2'), (3, 1')\}$.*

Observe that, in state $2'$, although none of the two outgoing transition is must, at least one of the two has to be present in any model in order to preserve the reachability of a marked state. Such restricted disjunction cannot be expressed with traditional unmarked modal specifications. Observe also that, according to the second item of the Def. 2, the reachability of a marked state may be delayed: $1'$ is marked, $(3, 1') \in \pi$ but 3 is not marked; however, a marked state can be eventually reached from 3 thanks to the state 1 .

According to Def. 2, only reachable states of $\mathcal{C}$ are semantically meaningful. We thus suppose from now on, and without loss of generality, that $\mathcal{C}$ is reachable, that is: $\forall q \in Q, q^0 \in \text{pre}^*(q)$.

A marked state $q \in F$ is said *delayable* if q can be reached again, that is, there exists a word $u \neq \epsilon$ such that $\delta(q, u) = q$; it is said *undelayable* otherwise. Denote by D the set of delayable states of a marked modal specification.

Figure 1: $\mathcal{M}$ is a model of $\mathcal{C}$

A marked state $q \in F$ is a bottleneck of $\mathcal{C}$ if it is the only marked states reachable from some state $q' \in Q$ that is, $\text{post}_m^*(q') \cap F = \{q\}$. Intuitively, this notion allows to identify the states that will be marked in any model of the specification.

Lemma 4 *Given a terminating automaton $\mathcal{M}$ and a marked modal specification $\mathcal{C}$ s.t. $\mathcal{M} \models \mathcal{C}$ then: $\mathcal{L}_{\mathcal{M}} \subseteq \mathcal{L}_{\mathcal{C}}$, and, for all $u \in \mathcal{L}_{\mathcal{M}}$, $(\lambda(r^0, u), \delta(q^0, u)) \in \pi$.*

Proof: The language inclusion is a direct consequence of the third item of Def. 2. Moreover, as modal specifications are assumed deterministic, the relation π is unique, when it exists, and thus, for any $u \in \mathcal{L}_{\mathcal{M}}$, $(\lambda(r^0, u), \delta(q^0, u)) \in \pi$. $\square$

The introduced semantics induces some simplifications in the structure of the marked modal specifications that we discuss now. At the end of this section, this will leads to the definition of an associated normal form.

Must-saturation. Observe that any terminating automaton model of the marked modal specification in Fig. 1 includes the starting transition labeled by a stemming from the initial state. It is thus a required transition that can be assigned a must modality in the specification. We therefore introduce the *must-saturation* of marked modal specifications.

Definition 5 (Must-saturation) *A marked modal specification is must-saturated if for all $q \notin F$ such that there is a unique $a \in \text{may}(q)$, we have $a \in \text{must}(q)$. Such a must-mapping is then said to be saturated.*

Lemma 6 *Any must-mapping can be saturated without changing the set of marked implementations.*

Proof: Let $(r, q) \in \pi$. If $q \notin F$ then $r \notin G$. As $\mathcal{M}$ is terminating, it is then possible to reach a marked state from r and: $\text{card}(\text{ready}(r)) \geq 1$. As $\text{card}(\text{may}(q)) = 1$, we have: $\text{card}(\text{ready}(r)) = 1$ for any $\mathcal{M}$ and it is not different than to have $\text{card}(\text{must}(q)) = 1$. $\square$

Consistency and attractability. Given a marked modal specification $\mathcal{C} = (Q, q^0, \Sigma, \delta, \text{must}, \text{may}, F)$ and a state $q \in Q$, $\mathcal{C}$ is said *consistent* in q if and only if $\text{must}(q) \subseteq \text{may}(q)$. $\mathcal{C}$ is said *attracted* in q if and only if $q \in \text{pre}_m^*(F)$.

Lemma 7 *If $\mathcal{M} \models \mathcal{C}$ then $\mathcal{C}$ is consistent and attracted in every state $\delta(q^0, u)$ with $u \in \mathcal{L}_{\mathcal{M}}$.*

Proof: Let $u \in \mathcal{L}_{\mathcal{M}}$, by Lem. 4 we have $(\lambda(r^0, u), \delta(q^0, u)) \in \pi$. Thus, from the first item of Def. 2, $\text{must}(\delta(q^0, u)) \subseteq \text{ready}(\lambda(r^0, u)) \subseteq \text{may}(\delta(q^0, u))$ and $\mathcal{C}$ is consistent in $\delta(q^0, u)$.

We now prove that $\mathcal{C}$ is attracted in $\delta(q^0, u)$ that is, $\delta(q^0, u) \in \text{pre}^*(F)$. Since $\mathcal{M}$ is terminating $\lambda(r^0, u) \in \text{pre}^*(G)$. As a result, there exists a suffix v of u such that $uv \in \mathcal{L}_{\mathcal{M}}$ and $\lambda(r^0, uv) \in G$. By Lem. 4, $uv \in \mathcal{L}_{\mathcal{C}}$ and $(\lambda(r^0, uv), \delta(q^0, uv)) \in \pi$. Now if $\lambda(r^0, uv) \in G$ then $\delta(q^0, uv) \in F$ by item 2 of Def. 2 and thus $\delta(q^0, u) \in \text{pre}^*(F)$. $\square$

As a consequence, only consistent and attracting states of $\mathcal{C}$ are semantically meaningful. This now leads us to define a reduced form:

Definition 8 (Reduced marked modal specification) $\mathcal{C}$ is reduced iff every state is reachable and it is consistent and attracted in every state $q \in Q$.

Proposition 9 (Reducibility) *Every satisfiable marked modal specification is equivalent to a reduced marked modal specification.*

Proof of this proposition is by construction of a reduced marked specification $\rho\mathcal{C}$ and then proving that $\mathcal{C}$ and $\rho\mathcal{C}$ are equivalent. This construction makes use of a pruning operation. We denote by $Q_{\Psi} \subseteq Q$ the set of all states $q \in Q$ such that q is inconsistent or unattracting, that is: $\text{must}(q) \not\subseteq \text{may}(q)$ or $q \notin \text{pre}^*(F)$.

Definition 10 (Reduction operation) *Given a marked modal specification $\mathcal{C} = (Q, q^0, \Sigma, \delta, \text{must}, \text{may}, F)$: if $q^0 \in \text{pre}_M^*(Q_{\Psi})$ then the reduction of $\mathcal{C}$ is $\mathcal{C}_{\perp}$; otherwise, it is the marked modal specification $(Q \setminus \text{pre}_M^*(Q_{\Psi}), q^0, \Sigma, \delta', \text{must}', \text{may}', F \setminus \text{pre}_M^*(Q_{\Psi}))$ where: $\delta'(r, a) = r'$ if and only if $\delta(r, a) = r'$ and $r, r' \notin \text{pre}_M^*(Q_{\Psi})$; as indicated in right after Def. 1, may' can be recovered from δ' whereas must' is the restriction of must to the domain $Q \setminus \text{pre}_M^*(Q_{\Psi})$.*

We now prove Prop. 9:

Proof: We first show : $\llbracket \mathcal{C} \rrbracket \subseteq \llbracket \rho\mathcal{C} \rrbracket$. We prove that $\mathcal{M} \models \rho\mathcal{C}$ can be established with the same simulation relation π that states $\mathcal{M} \models \mathcal{C}$:

- Let $(r, q) \in \pi \subseteq R \times Q$. Observe first that $q \notin Q_{\Psi}$ according to Lem. 7. Moreover, $q \notin \text{pre}_M^*(Q_{\Psi})$ otherwise, there would exist $q' \in Q_{\Psi}$ coreachable from q by following transitions labeled by *must*-actions and thus we would have $(r', q') \in \pi$ with $q' \in Q_{\Psi}$ for some $r' \in R$ which is not possible by Lem. 7. As a result, π stating $\mathcal{M} \models \mathcal{C}$ is more precisely defined over $R \times (Q \setminus \text{pre}_M^*(Q_{\Psi}))$.
- Moreover, we can show that if $\lambda(r, a)$ is defined and $q \notin \text{pre}_M^*(Q_{\Psi})$ then $\delta'(q, a)$ is also defined. By contradiction, suppose that $\delta'(q, a)$ is not defined then $\delta(q, a) \in \text{pre}_M^*(Q_{\Psi})$ by construction of δ' . However, $\lambda(r, a)$ defined implies $\delta(q, a)$ defined and $(\lambda(r, a), \delta(q, a)) \in \pi$ as $\mathcal{M} \models \mathcal{C}$ and, by Lem. 7, $\delta(q, a) \notin \text{pre}_M^*(Q_{\Psi})$ which is in contradiction with the previous sentence.

Consequence from the two previous items is that any pair $(r, q) \in \pi$ which states $\mathcal{M} \models \mathcal{C}$ can also be seen as pair of states of $\mathcal{M}$ and $\rho\mathcal{C}$. We now show that F' , must' and may' satisfies the constraints of a model relation:

- First, as $\mathcal{M} \models \mathcal{C}$, we have $r \in G$ implies $q \in F$. As previously shown, $q \notin \text{pre}_M^*(Q_\Psi)$ and thus $q \in F \setminus \text{pre}_M^*(Q_\Psi)$;
- Next, if $\text{must}(q) \subseteq \text{ready}(r)$ then, as $q \in Q \setminus \text{pre}_M^*(Q_\Psi)$, by construction we have $\text{must}'(q) = \text{must}(q)$ and thus $\text{must}'(q) \subseteq \text{ready}(r)$;
- Last, if $\text{ready}(r) \subseteq \text{may}(q)$, we can show that $\text{ready}(r) \subseteq \text{may}'(q)$. By contradiction, suppose $\text{ready}(r) \cap (\text{may}(q) \setminus \text{may}'(q)) \neq \emptyset$. Then there exists a for which $\lambda(r, a)$, $\delta(q, a)$ are defined and $q \notin \text{pre}_M^*(Q_\Psi)$ but $\delta'(q, a)$ is not defined. We have shown above that this situation is not possible.

As a result, $\mathcal{M} \models \rho\mathcal{C}$ which concludes the first part of the proof.

We now prove : $\llbracket \mathcal{C} \rrbracket \supseteq \llbracket \rho\mathcal{C} \rrbracket$. It is clear that the graph associated with $\text{Un}(\rho\mathcal{C})$ is a sub-graph of $\text{Un}(\mathcal{C})$ as $\rho\mathcal{C}$ is built from $\mathcal{C}$ by removing some edges. As a result, all pairs (r, q) in the simulation relation π stating that $\mathcal{M} \models \rho\mathcal{C}$ can also be seen as a pair of states of $\mathcal{M}$ and $\mathcal{C}$. We now show that π also defines a suitable simulation relation in order to establish $\mathcal{M} \models \mathcal{C}$ as G , may and must of $\mathcal{C}$ satisfies the constraints of a model relation. Let $(r, q) \in \pi$:

- If $r \in G$ then $q \in F \setminus \text{pre}_M^*(Q_\Psi)$ and thus, in particular, $q \in F$;
- By construction, $\text{must}'(q) = \text{must}(q)$ and $\text{may}'(q) \subseteq \text{may}(q)$. As a result, if $\text{must}'(q) \subseteq \text{ready}(r) \subseteq \text{may}'(q)$ then $\text{must}(q) \subseteq \text{ready}(r) \subseteq \text{may}(q)$.

As a result, $\mathcal{M} \models \mathcal{C}$ which concludes the proof. $\square$

Normal form. This now leads us to define the normal form of any marked modal specification:

Definition 11 (Normal form) A marked modal specification is in normal form if it is both must-saturated and reduced.

According to Lem. 6 and Prop. 9, any marked modal specification $\mathcal{C}$ can be put in normal form $\eta\mathcal{C}$ without altering its set of models. As a result, from now on, we always suppose that marked modal specifications are in normal form.

At this point, the reader may wonder why must-saturation, consistency and attractability are not fully part of the definition of marked modal specification (as it is the case for the consistency requirement in the original papers on unmarked modal specifications [18, 16]). The reason for this is because, in what follows, we propose composition operators on marked modal specifications and it is easier to define these constructions without trying to preserve these different requirements. Now if the combination of two marked modal specifications (which are now implicitly supposed to be in normal form) gives rise to the a specification violating one of the above requirements then a step of normalization has to be applied on the result in order to have an iterative process.

3 Refinement of marked modal specifications

A refinement relation aims at relating interfaces at different stages of their design. Basically, it should correspond to refine the set of allowed implementations of an interface. Moreover, we shall see later that refinement should entail *substitutability*, meaning that the substitution of an interface $\mathcal{C}_2$ by a refined version $\mathcal{C}_1$ must not impact the possible and actual cooperation with other components, that have been previously declared as legal for $\mathcal{C}_2$.

Definition 12 (Refinement) Given two marked modal specifications $\mathcal{C}_1 = (Q_1, q_1^0, \Sigma, \delta_1, \text{must}_1, \text{may}_1, F_1)$ and $\mathcal{C}_2 = (Q_2, q_2^0, \Sigma, \delta_2, \text{must}_2, \text{may}_2, F_2)$, $\mathcal{C}_1$ is a refinement of $\mathcal{C}_2$, noted $\mathcal{C}_1 \leq \mathcal{C}_2$, if and only if there exists a simulation relation $\Pi \subseteq Q_1 \times Q_2$ such that $(q_1^0, q_2^0) \in \Pi$ and, for all pairs $(q_1, q_2) \in \Pi$:

1. $\text{may}_1(q_1) \subseteq \text{may}_2(q_2)$ and $\text{must}_1(q_1) \supseteq \text{must}_2(q_2)$;

2. $q_1 \in F_1$ implies $q_2 \in F_2$;
3. for every $a \in \text{may}_1(q_1)$, we have: $(\delta_1(q_1, a), \delta_2(q_2, a)) \in \Pi$.

Intuitively, refining an interface corresponds to possibly changing a transition with a *may* modality into either a required or a proscribed transition while potentially delaying the reachability of a marked state. This relation is reflexive and transitive and is thus a preorder.

Theorem 13 *Given two marked modal specifications $\mathcal{C}_1$ and $\mathcal{C}_2$, $\mathcal{C}_1 \leq \mathcal{C}_2$ if and only if, $\llbracket \mathcal{C}_1 \rrbracket \subseteq \llbracket \mathcal{C}_2 \rrbracket$.*

Theorem 13 holds provided the marked modal specifications are deterministic. If nondeterminism is allowed, refinement becomes correct but not fully abstract (the implication from right to left in Theorem 13 is not true in general). This is discussed for *unmarked* modal specifications in [17]; their counterexample can be immediately adapted to our context. Moreover, as argued in [9], nondeterministic modal specifications are not really suitable to characterize a set of *deterministic* automata.

To prove this Theorem, the two following intermediate Lemmata are necessary. We first define the maximal implementation associated to a marked modal specification:

Lemma 14 (Maximal Implementation) *Given a marked modal specification $\mathcal{C} = (Q, q^0, \Sigma, \delta, \text{must}, \text{may}, F)$, the automaton $\mathcal{M}_{\mathcal{C}} = (Q, q^0, \Sigma, \delta, F)$ satisfies $\mathcal{C}$ and moreover, $\mathcal{L}_{\mathcal{M}} = \mathcal{L}_{\mathcal{C}}$.*

Proof: Take equality as the simulation relation. For all $q \in Q$, we have for $\mathcal{M}_{\mathcal{C}}$: $\text{ready}(q) = \text{may}(q)$. Since $\mathcal{C}$ is in normal form, we have $\text{must}(q) \subseteq \text{may}(q)$. Hence, $\text{must}(q) \subseteq \text{ready}(q) \subseteq \text{may}(q)$. Normal form implies that $Q = \text{pre}^*(F)$. Hence $\mathcal{M}_{\mathcal{C}}$ is a terminating automaton satisfying $\mathcal{C}$ and such that $\mathcal{L}_{\mathcal{M}} = \mathcal{L}_{\mathcal{C}}$. $\square$

Remark also that the simulation relation can be chosen to be a least relation defined as follows:

Definition 15 (Least Simulation Relation) *Given $\mathcal{C}_1 = (Q_1, q_1^0, \Sigma, \delta_1, \text{must}_1, \text{may}_1, F_1)$ and $\mathcal{C}_2 = (Q_2, q_2^0, \Sigma, \delta_2, \text{must}_2, \text{may}_2, F_2)$ two marked modal specifications, define $\Gamma \subseteq Q_1 \times Q_2$ to be the least relation such that:*

1. $(q_1^0, q_2^0) \in \Gamma$;
2. For all $(q_1, q_2) \in \Gamma$, if $a \in \text{may}_1(q_1) \cap \text{may}_2(q_2)$ then $(\delta_1(q_1, a), \delta_2(q_2, a)) \in \Gamma$.

Lemma 16 *Given two marked modal specifications $\mathcal{C}_1$ and $\mathcal{C}_2$, $\mathcal{C}_1$ refines $\mathcal{C}_2$ implies that any $(q_1, q_2) \in \Gamma$ satisfies the items of Def. 12.*

Proof: Assume $\mathcal{C}_1 \leq \mathcal{C}_2$, meaning that there exists Π satisfying the items of Def. 12. Lets prove that $\Gamma \subseteq \Pi$. The pair of initial states are in both relations: $(q_1^0, q_2^0) \in \Gamma \cap \Pi$. If $(q_1, q_2) \in \Gamma \cap \Pi$, then for every $a \in \text{may}_1(q_1) \subseteq \text{may}_2(q_2)$, $(\delta_1(q_1, a), \delta_2(q_2, a)) \in \Gamma \cap \Pi$. This proves that $\Gamma = \Gamma \cap \Pi$, and hence $\Gamma \subseteq \Pi$. This implies that any $(q_1, q_2) \in \Gamma$ also satisfies the items of Def. 12. $\square$

We can now prove Theorem 13:

Proof: $(\Rightarrow)$ Suppose that $\mathcal{M} \models \mathcal{C}_1$ and $\mathcal{C}_1 \leq \mathcal{C}_2$, meaning that there exists simulation relations $\pi \subseteq R \times Q_1$ and $\Pi \subseteq Q_1 \times Q_2$, witnesses of the satisfaction and refinement relations. Hence we assume that π satisfies the items of Def. 2 and the Π satisfies the items of Def. 12.

We now prove that $\mathcal{M} \models \mathcal{C}_2$. This is done by proving that relation $\pi \cdot \Pi = \{(r, q_2) \in R \times Q_2 \mid \exists q_1 \in Q_1 \text{ st. } (r, q_1) \in \pi \text{ and } (q_1, q_2) \in \Pi\}$ satisfies the items of Def. 2. $(r^0, q_1^0) \in \pi$

and $(q_1^0, q_2^0) \in \Pi$ imply $(r^0, q_2^0) \in \pi \cdot \Pi$. Given $(r, q_2) \in \pi \cdot \Pi$, there exists $q_1 \in Q_1$ such that $(r, q_1) \in \pi$ and $(q_1, q_2) \in \Pi$. Take any $q_1 \in Q_1$ such that $(r, q_1) \in \pi$ and $(q_1, q_2) \in \Pi$. (i) $\text{must}_2(q_2) \subseteq \text{must}_1(q_1) \subseteq \text{ready}(r) \subseteq \text{may}_1(q_1) \subseteq \text{may}_2(q_2)$; (ii) $r \in G$ implies $q_1 \in F_1$ implies $q_2 \in F_2$; And (iii) for every $a \in \Sigma$ and every $r' \in R$ such that $\lambda(r, a) = r'$, $(r', \delta_1(q_1, a)) \in \pi$ and $(\delta_1(q_1, a), \delta_2(q_2, a)) \in \Pi$. Hence $(r', \delta_2(q_2, a)) \in \pi \cdot \Pi$.

($\Leftarrow$) We proceed by contraposition; we prove that if $\mathcal{C}_1 \not\leq \mathcal{C}_2$ then there exists a model $\mathcal{M}$ of $\mathcal{C}_1$ that is not a model of $\mathcal{C}_2$. Consider the least simulation Γ . Since $\mathcal{C}_1 \not\leq \mathcal{C}_2$, (i) $\mathcal{C}_1$ is satisfiable and (ii) there exists $(q_1, q_2) \in \Gamma$ such that at least one of the items of Def. 12 is not satisfied:

- Assume $\text{may}_1(q_1) \not\subseteq \text{may}_2(q_2)$. By construction $\mathcal{M}_{\mathcal{C}_1} \models \mathcal{C}_1$, while $\mathcal{M}_{\mathcal{C}_1} \not\models \mathcal{C}_2$.
- Assume $\text{may}_1(q_1) \subseteq \text{may}_2(q_2)$ for all $(q_1, q_2) \in \Gamma$ and $\text{must}_1(q_1) \not\subseteq \text{must}_2(q_2)$. Hence there exists $a \in \text{must}_2(q_2) \setminus \text{must}_1(q_1)$. Construct $\mathcal{M} = (Q_1, q_1^0, \Sigma, \lambda, G_1)$ such that $\lambda(q, b) = q'$ of and only if $\delta_1(q, b) = q'$ and $q \neq q_1$ or $b \neq a$. Recall $\mathcal{C}_1$ is in normal form, hence $\mathcal{M} \models \mathcal{C}_1$. However, $a \in \text{must}_2(q_2)$ and $a \notin \text{ready}(q_1)$. Hence $\mathcal{M} \not\models \mathcal{C}_2$.
- Assume $\text{may}_1(q_1) \subseteq \text{may}_2(q_2)$ and $\text{must}_1(q_1) \supseteq \text{must}_2(q_2)$ for all $(q_1, q_2) \in \Gamma$. Assume $q_1 \in F_1$ and $q_2 \notin F_2$. $\mathcal{M}_{\mathcal{C}_1} \models \mathcal{C}_1$, while $\mathcal{M}_{\mathcal{C}_1} \not\models \mathcal{C}_2$.

□

When the left counterpart is ultimately refined, the refinement relation coincide with the implementation relation: given a terminating automaton $\mathcal{M}$ and a marked modal specification $\mathcal{C}$, $\mathcal{M} \models \mathcal{C}$ if and only if $\text{Em}(\mathcal{M}) \leq \mathcal{C}$.

4 Conjunction of marked modal specification

It is a current practice, when modeling complex systems, to associate several specifications with a same system, sub-system, or component, each of them describing a different aspect of it. These so-called *viewpoints* may be engineered independently, and possibly by different teams. It is then natural to question whether different viewpoints are not contradictory and how to realize all of them. This leads to define a conjunction operator. Moreover in [8], the authors point out that, during the design cycle, a designer may be tempted to merge two interfaces which share some similarities in order to use a same implementation for the two interfaces. More formally, this corresponds to look for a shared refinement of the interfaces, if it exists.

We now define a conjunction operator which enjoy the expected properties to solve the two above problems.

Definition 17 (Conjunction) Given two marked modal specifications $\mathcal{C}_1 = (Q_1, q_1^0, \Sigma, \delta_1, \text{must}_1, \text{may}_1, F_1)$ and $\mathcal{C}_2 = (Q_2, q_2^0, \Sigma, \delta_2, \text{must}_2, \text{may}_2, F_2)$, the conjunction of $\mathcal{C}_1$ and $\mathcal{C}_2$, noted $\mathcal{C}_1 \wedge \mathcal{C}_2$, is the normal form $\eta(\mathcal{C}_1 \& \mathcal{C}_2)$ of $\mathcal{C}_1 \& \mathcal{C}_2 = (Q, q^0, \Sigma, \delta, \text{must}, \text{may}, F)$ with:

1. $Q = Q_1 \times Q_2$ and $q^0 = (q_1^0, q_2^0)$;
2. for any $q_1 \in Q_1$, $q_2 \in Q_2$ and $a \in \Sigma$, $\delta((q_1, q_2), a) = (q'_1, q'_2)$ if and only if $\delta_1(q_1, a) = q'_1$ and $\delta_2(q_2, a) = q'_2$;
3. $\text{may}(q_1, q_2) = \text{may}_1(q_1) \cap \text{may}_2(q_2)$ and $\text{must}(q_1, q_2) = \text{must}_1(q_1) \cup \text{must}_2(q_2)$;
4. $(q_1, q_2) \in F$ if and only if $q_1 \in F_1$ and $q_2 \in F_2$.

Considering the manipulations done on the may/must-maps and on the transition map to obtain $\mathcal{C}_1 \& \mathcal{C}_2$, the must-saturation and the consistency may not be respected. We thus impose a normalization step in order to have an iterative process as explained at the end of Sec. 2.

Theorem 18 Given some marked modal specifications $\mathcal{C}_1, \mathcal{C}_2, \mathcal{C}_3$ and $\mathcal{C}$:

- $\llbracket \mathcal{C}_1 \wedge \mathcal{C}_2 \rrbracket = \llbracket \mathcal{C}_1 \rrbracket \cap \llbracket \mathcal{C}_2 \rrbracket$;

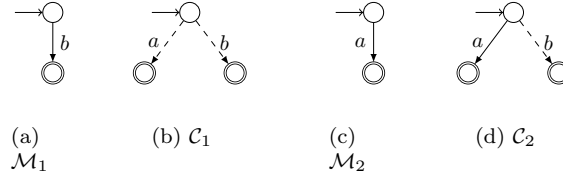


Figure 2: Reachability is not compositional

- $\mathcal{C}_1 \wedge \mathcal{C}_2$ is the greatest lower bound of $\mathcal{C}_1$ and $\mathcal{C}_2$ for the refinement relation: $\mathcal{C} \leq \mathcal{C}_1$ and $\mathcal{C} \leq \mathcal{C}_2$ iff $\mathcal{C} \leq \mathcal{C}_1 \wedge \mathcal{C}_2$;
- $\wedge$ is associative: $\mathcal{C}_1 \wedge (\mathcal{C}_2 \wedge \mathcal{C}_3) \equiv (\mathcal{C}_1 \wedge \mathcal{C}_2) \wedge \mathcal{C}_3$.

Proof: The proof of the second item is a direct consequence of the first one: if $\mathcal{C} \leq \mathcal{C}_i$ for $i \in \{1, 2\}$ then, by Theorem 13: $\llbracket \mathcal{C} \rrbracket \subseteq \llbracket \mathcal{C}_i \rrbracket$. As a result, we have: $\llbracket \mathcal{C} \rrbracket \subseteq \llbracket \mathcal{C}_1 \rrbracket \cap \llbracket \mathcal{C}_2 \rrbracket$. By the first item of Theorem 18, this is equivalent to: $\llbracket \mathcal{C} \rrbracket \subseteq \llbracket \mathcal{C}_1 \wedge \mathcal{C}_2 \rrbracket$. Last, we deduce from Theorem 13 that $\mathcal{C} \preceq \mathcal{C}_1 \wedge \mathcal{C}_2$. $\square$

5 Product of marked modal specifications

Reachability is not preserved by product in general. Fig. 2 shows a simple example: $\mathcal{M}_1 \models \mathcal{C}_1$ and $\mathcal{M}_2 \models \mathcal{C}_2$; however the product of $\mathcal{M}_1 \times \mathcal{M}_2$ is a single non-marked state, hence the reachability of a marked state is not possible.

This leads us to consider the following problem: given two marked modal specifications, can they be implemented concurrently i.e., such that the product of any model of the first specification with any model of the second one will always have the ability to reach a marked state of the product?

Similarly to [1], we distinguish a pessimistic from an optimistic view of composition and solve the previous problem in this two contexts.

First, in order to represent the cooperation between subsystems, a signature over Σ is now associated to any terminating automaton or marked modal specification over Σ :

Definition 19 (Signature) Given a set of actions Σ , a signature over Σ is a mapping $\mu : \Sigma \rightarrow \{?, !\}$ which associates to any action either $?$ when the action is an input or $!$ when it is an output.

Now, transitions are either labeled $!a$ (for $\mu(a) = !$) when the entity responsible for the occurrence of a is the system, or $?a$ (for $\mu(a) = ?$) if a stems from the environment of the system. The resulting formalism is thus suited to model protocols between a system and an unknown partner belonging to the system environment. Contrarily to the input/output automata of [20] and following the interface automata of [1], terminating automata and marked modal specifications are not required to be input-enabled, meaning that some actions $?b$ of the environment may not be permitted in some state q . More formally, this situation occurs in state q if there is no outgoing transition from q labeled by $?b$. This allows to restrict, from the point of view of the system, the behavior of its environment.

Example 20 Fig. 3(a) depicts the specification of a service which can receive requests $?r$ from an unidentified subsystem in its environment and then answers by producing $!a$ until it is stopped

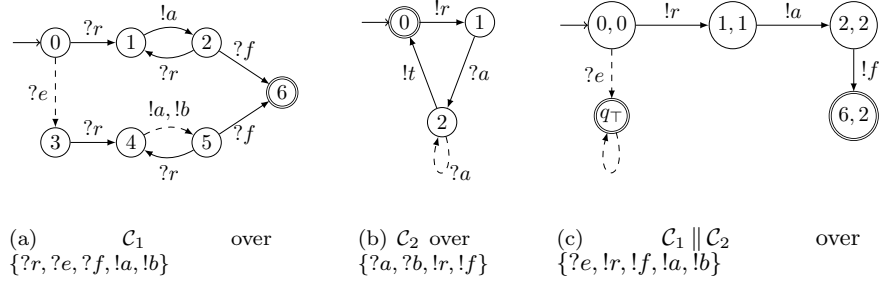


Figure 3: Example of composition

with $?f$. It may also produce $!b$ when set in an enhanced mode by $?e$. Fig. 3(b) depicts the specification of a client which expects to receive $?a$ as an answer to any request $!r$ and is ready to receive remitted $?a$. Although $?b$ is in the signature of $\mathcal{C}_2$, there is no transition labeled $?b$ meaning that the client rejects this inputs.

We write $\Sigma^?$ and $\Sigma^!$ for the set of input and output actions, respectively, thus forming a partition of Σ . A system is then *closed* if its associated signature is such that $\Sigma^? = \emptyset$ and *open* otherwise. In this paper, we assume that if $\mathcal{M} \models \mathcal{C}$ then the signature associated to $\mathcal{M}$ and $\mathcal{C}$ is identical. Similarly, if $\mathcal{C}_1 \leq \mathcal{C}_2$ then $\mathcal{C}_1$ and $\mathcal{C}_2$ have the same signature¹.

A first condition to product is the *composability* of signatures. Given two signatures μ_1 and μ_2 over Σ_1 and Σ_2 respectively, defining two partitions $(\Sigma_1^?, \Sigma_1^!)$ and $(\Sigma_2^?, \Sigma_2^!)$, they are *composable* if no output actions is shared: $\Sigma_1^! \cap \Sigma_2^! = \emptyset$. For composable signatures, we let the *communication* actions be the set $\Sigma_{co}(\mu_1, \mu_2) = (\Sigma_1^? \cap \Sigma_2^!) \cup (\Sigma_2^? \cap \Sigma_1^!)$ which corresponds to the shared actions on which a synchronization will be possible. The complement is called the set of *private* actions and is denoted $\Sigma_{pr}(\mu_1, \mu_2) = (\Sigma_1 \cup \Sigma_2) \setminus \Sigma_{co}(\mu_1, \mu_2)$.

Definition 21 (Product of signatures) *The product of two composable signatures μ_1 and μ_2 is $\mu = \mu_1 \times \mu_2$ defined over $\Sigma_1 \cup \Sigma_2$ such that: $\Sigma^? = (\Sigma_1^? \cup \Sigma_2^?) \setminus \Sigma_{co}(\mu_1, \mu_2)$ and $\Sigma^! = \Sigma_1^! \cup \Sigma_2^!$.*

The product of two terminating automata $\mathcal{M}_1$ and $\mathcal{M}_2$ with respective composable signatures μ_1 and μ_2 is then $\mathcal{M}_1 \times \mathcal{M}_2$ as defined in Sec. 2.1 with signature $\mu_1 \times \mu_2$.

5.1 Pessimistic composition of marked modal specifications

We first consider the case of *pessimistic*² composition; we define a sufficient and necessary condition such that two marked modal specification can be independently implemented, the product of any of their implementations being terminating.

This condition corresponds to the existence of a joint path to a marked state, for every reachable state of the product of arbitrary implementations. We then consider the less cooperative situation in which any optional behavior is disabled and check if such paths exist. However, the minimal behavior associated to a state of a marked modal specification is not unique in general. Consider $\mathcal{C}_1$ in Fig. 2(b), the minimal number of outgoing transition stemming from the initial

¹This assumption is taken to simplify the presentation. Refinement of signature as defined in [23] can be handled in the presented theory.

²The pessimistic view of this approach will be made clearer in the next section.

state among all the models of $\mathcal{C}_1$ is 1 and can be either a transition label by a or by b . To represent the different minimal possibilities, we thus use an intermediate structure called *minimal constraint automaton*. First we define the set of minimal constraints associated to a state:

Definition 22 (Minimal constraints) *For any state q of a marked modal specification $\mathcal{C}$ defined over Σ , we associate the set $\zeta(q) \in 2^{2^\Sigma}$ defined by:*

$$\zeta(q) = \begin{cases} \{ \text{must}(q) \} & \text{if } \text{must}(q) \neq \emptyset \\ \{ \{a\} \mid a \in \text{may}(q) \} & \text{if } \text{must}(q) = \emptyset \text{ and } q \notin F \\ \{ \emptyset \} & \text{if } \text{must}(q) = \emptyset \text{ and } q \in F \end{cases}$$

Definition 23 (Minimal constraints automaton) *Given a state q of a marked modal specification $\mathcal{C}$ over Σ , the minimal constraints automaton $\text{Min}(\mathcal{C}, q)$ is the automaton over Σ whose initial state is q ; its labeled transition map is λ_{Min} such that $\lambda_{\text{Min}}(q', a) = q''$ if and only if $a \in X$ with $X \in \zeta(q')$ and $\delta(q', a) = q''$; its set of final states G_{Min} is the set of undelayable bottlenecks of $\mathcal{C}$.*

We identify potential dead-ends, that is pairs of states of two marked modal specifications $\mathcal{C}_1$ and $\mathcal{C}_2$ to be composed from which no outgoing transition may be available in a product of two respective implementations:

Definition 24 (Dead-end) *Given q_1 and q_2 two states respectively from the marked modal specifications $\mathcal{C}_1$ and $\mathcal{C}_2$ defined over Σ_1 and Σ_2 , the pair (q_1, q_2) is a dead-end if:*

1. $q_1 \notin (F_1 \setminus D_1)$ or $q_2 \notin (F_2 \setminus D_2)$ and,
2. there exists $X_1 \in \zeta_1(q_1)$ and $X_2 \in \zeta_2(q_2)$ such that: $(X_1 \cup (\Sigma_2 \setminus \Sigma_1)) \cap (X_2 \cup (\Sigma_1 \setminus \Sigma_2)) = \emptyset$.

Example 25 *The minimal constraints associated to the initial states of $\mathcal{C}_1$ and $\mathcal{C}_2$ from Fig. 2 and defined over the same alphabet of actions $\{a, b\}$ are respectively $\{\{a\}, \{b\}\}$ and $\{\{a\}\}$. The pair formed by this two states is thus a dead-end as for $X_1 = \{b\}$ and $X_2 = \{a\}$, we have $X_1 \cap X_2 = \emptyset$.*

This now leads us to a definition of *exception state pairs* from which a joint path to a marked state pair cannot be ensured independently of the implementation choices to be made:

Definition 26 (Exception state pair) *Given q_1 and q_2 two states respectively from two marked modal specifications $\mathcal{C}_1$ and $\mathcal{C}_2$, the pair (q_1, q_2) is an exception, noted $\text{Ex}(q_1, q_2)$ if:*

- $\text{Min}(\mathcal{C}_1, q_1) \times \text{Min}(\mathcal{C}_2, q_2)$ is not terminating or,
- there exists a reachable dead-end³ (q'_1, q'_2) in $\text{Min}(\mathcal{C}_1, q_1) \times \text{Min}(\mathcal{C}_2, q_2)$.

Then we can define the following criterion characterizing marked modal specifications having compatible reachability:

Definition 27 (Compatible reachability) *Two marked modal specifications $\mathcal{C}_1$ and $\mathcal{C}_2$ have a compatible reachability, noted $\mathcal{C}_1 \sim_{\mathcal{T}} \mathcal{C}_2$, if there is no exception state pair that is reachable in $\text{Un}(\mathcal{C}_1) \times \text{Un}(\mathcal{C}_2)$.*

The soundness and the completeness of the previous definition is then stated by the following Theorem:

³As the set of states of $\text{Min}(\mathcal{C}_i, q_i)$ is a subset of these of $\mathcal{C}_i$, we can refer to (q'_1, q'_2) in $\text{Min}(\mathcal{C}_1, q_1) \times \text{Min}(\mathcal{C}_2, q_2)$ as a pair of states of $\mathcal{C}_1$ and $\mathcal{C}_2$ and then test if it is a dead-end in the sense of Def. 24.

Theorem 28 (Independent implementability) *Given two marked modal specifications $\mathcal{C}_1$ and $\mathcal{C}_2$, $\mathcal{C}_1 \sim_{\mathcal{T}} \mathcal{C}_2$ if and only if for any $\mathcal{M}_1 \models \mathcal{C}_1$ and $\mathcal{M}_2 \models \mathcal{C}_2$, the product $\mathcal{M}_1 \times \mathcal{M}_2$ is terminating.*

To prove this Theorem, we need the two following intermediate Lemmata.

Lemma 29 *For any $\mathcal{M} \models \mathcal{C}$ and pair $(r, q) \in \pi$, there exists a set $X \in \zeta(q)$ such that $X \subseteq \text{ready}(r)$.*

Proof: If $\text{must}(q) \neq \emptyset$ then $\zeta(q) = \{\text{must}(q)\}$. As $(r, q) \in \pi$, we have $\text{must}(q) \subseteq \text{ready}(r)$. Thus, for $X = \text{must}(q) \in \zeta(q)$, we have $X \subseteq \text{ready}(r)$.

If $\text{must}(q) = \emptyset$ and $q \notin F$ then $\zeta(q) = \{\{a\} \mid a \in \text{may}(q)\}$. As $q \notin F$, we have $r \notin G$ and there exists a path from r to a marked state. Thus, there exists at least an $a \in \text{may}(q)$ such that $\lambda(r, a)$ is defined. For $X = \{a\} \in \zeta(q)$, we then have $X \subseteq \text{ready}(r)$.

Last, if $\text{must}(q) = \emptyset$ and $q \in F$ then $\zeta(q) = \{\emptyset\}$. Thus, for $X = \emptyset \in \zeta(q)$, we have $X \subseteq \text{ready}(r)$. $\square$

Lemma 30 *If a state q of a marked modal specification $\mathcal{C}$ is a bottleneck and is not delayable then $\text{may}(q) = \emptyset$.*

Proof: Suppose that $\text{may}(q) \neq \emptyset$ then as $\mathcal{C}$ is supposed to be in normal form, there exists a marked state reachable from q . As q is a bottleneck, the only reachable marked state from q is q . However, as q is not delayable, q cannot be reached again. As a result, $\text{may}(q) = \emptyset$. $\square$

We can now prove Theorem 28:

Proof: $(\Rightarrow)$ We assume $\mathcal{C}_1 \sim_{\mathcal{T}} \mathcal{C}_2$ and we prove that, for $\mathcal{M}_1 \models \mathcal{C}_1$ and $\mathcal{M}_2 \models \mathcal{C}_2$, $\mathcal{M}_1 \times \mathcal{M}_2$ is terminating. We thus fix a $u \in \mathcal{L}_{\mathcal{M}_1} \sqcap \mathcal{L}_{\mathcal{M}_2}$ and we show that there exists a v such that $\lambda((r_1^0, r_2^0), uv) \in G_1 \times G_2$.

The word u corresponds to $u_1 = \text{pr}_{\Sigma_1}(u)$ in $\mathcal{M}_1$ and to $u_2 = \text{pr}_{\Sigma_2}(u)$ in $\mathcal{M}_2$. Let $r_1 = \lambda_1(r_1^0, u_1)$ and $r_2 = \lambda_2(r_2^0, u_2)$. By Lem. 4, $u_1 \in \mathcal{L}_{\mathcal{C}_1}$ and $u_2 \in \mathcal{L}_{\mathcal{C}_2}$. Let $q_1 = \delta_1(q_1^0, u_1)$ and $q_2 = \delta_2(q_2^0, u_2)$. As $\mathcal{C}_1 \sim_{\mathcal{T}} \mathcal{C}_2$ and as (q_1, q_2) is reached in $\text{Un}(\mathcal{C}_1) \times \text{Un}(\mathcal{C}_2)$ after u , (q_1, q_2) is not an exception state pair. Thus $\text{Min}(\mathcal{C}_1, q_1) \times \text{Min}(\mathcal{C}_2, q_2)$ is terminating and no potential dead-end is reachable in $\text{Min}(\mathcal{C}_1, q_1) \times \text{Min}(\mathcal{C}_2, q_2)$.

From Lem. 29, we know that u_1 and u_2 can be extended with actions belonging to the minimal constraints sets of $\mathcal{C}_1$ and $\mathcal{C}_2$ respectively. More precisely, we construct a word w in $\text{Min}(\mathcal{C}_1, q_1) \times \text{Min}(\mathcal{C}_2, q_2)$ such that $w = a_0 \dots a_n$ with $a_i \in (X_1 \cup (\Sigma_2 \setminus \Sigma_1)) \cap (X_2 \cup (\Sigma_1 \setminus \Sigma_2))$ where:

$$\begin{cases} X_1 \in \zeta_1(\delta_1(q_1^0, u_1 \cdot \text{pr}_{\Sigma_1}(a_0 \dots a_{i-1}))) \text{ and} \\ \quad X_1 \subseteq \text{ready}_1(\lambda_1(r_1^0, u_1 \cdot \text{pr}_{\Sigma_1}(a_0 \dots a_{i-1}))) \\ \\ X_2 \in \zeta_2(\delta_2(q_2^0, u_2 \cdot \text{pr}_{\Sigma_2}(a_0 \dots a_{i-1}))) \text{ and} \\ \quad X_2 \subseteq \text{ready}_2(\lambda_2(r_2^0, u_2 \cdot \text{pr}_{\Sigma_2}(a_0 \dots a_{i-1}))) \end{cases}$$

As no dead-end is reachable in $\text{Min}(\mathcal{C}_1, q_1) \times \text{Min}(\mathcal{C}_2, q_2)$, $(X_1 \cup (\Sigma_2 \setminus \Sigma_1)) \cap (X_2 \cup (\Sigma_1 \setminus \Sigma_2)) \neq \emptyset$ or a state in $(F_1 \setminus D_1) \times (F_2 \setminus D_2)$ is reached. In any case, we eventually reach a marked state as $\text{Min}(\mathcal{C}_1, q_1) \times \text{Min}(\mathcal{C}_2, q_2)$ is terminating. Let $w_1 = \text{pr}_{\Sigma_1}(w)$ and to $w_2 = \text{pr}_{\Sigma_2}(w)$ and let (q'_1, q'_2) be the marked state reached after w in $\text{Min}(\mathcal{C}_1, q_1) \times \text{Min}(\mathcal{C}_2, q_2)$. Note that, by construction, q'_1 and q'_2 are undelayable bottlenecks in $\mathcal{C}_1$ and $\mathcal{C}_2$. Now by Lem. 30, $\text{may}_1(q'_1) = \emptyset$ and $\text{may}_2(q'_2) = \emptyset$. As a result, $\lambda_i(r_i^0, u_i \cdot w_i) \in G_i$, for $i = 1, 2$ as $\mathcal{M}_i \models \mathcal{C}_i$ and $\mathcal{M}_i$ is terminating. Thus, $\lambda((r_1^0, r_2^0), uw) \in G_1 \times G_2$;

($\Leftarrow$) By contraposition, we prove that if $\mathcal{C}_1 \sim_{\mathcal{T}} \mathcal{C}_2$ then there exists $\mathcal{M}_1 \models \mathcal{C}_1$ and $\mathcal{M}_2 \models \mathcal{C}_2$ such that $\mathcal{M}_1 \times \mathcal{M}_2$ is not terminating.

If $\mathcal{C}_1 \sim_{\mathcal{T}} \mathcal{C}_2$ then there exists $u \in \mathcal{L}_{\mathcal{C}_1} \sqcap \mathcal{L}_{\mathcal{C}_2}$ such that the state (q_1, q_2) reached after u is an exception pair of state. Two cases are possible:

- $\text{Min}(\mathcal{C}_1, q_1) \times \text{Min}(\mathcal{C}_2, q_2)$ is not terminating, that is: there exists $v \in \mathcal{L}_{\text{Min}(\mathcal{C}_1, q_1)} \sqcap \mathcal{L}_{\text{Min}(\mathcal{C}_2, q_2)}$ for which there is no w such that $vw \in \mathcal{L}_{\text{Min}(\mathcal{C}_1, q_1)} \sqcap \mathcal{L}_{\text{Min}(\mathcal{C}_2, q_2)}$ and the state reached after vw is marked, that is, by construction, is a pair of bottleneck of $\mathcal{C}_1$ and $\mathcal{C}_2$.

We consider $\mathcal{M}_i \models \mathcal{C}_i$ for $i \in \{1, 2\}$ such that $\text{pr}_{\Sigma_i}(u) \in \mathcal{L}_{\mathcal{M}_i}$ and the possible suffix of $\text{pr}_{\Sigma_i}(u)$ are the words in $\mathcal{L}_{\text{Min}(\mathcal{C}_i, q_i)}$. We can choose the marked state of $\mathcal{M}_i$ such that there is no word w such that $\text{pr}_{\Sigma_i}(uvw)$ is marked in both $\mathcal{M}_1$ and $\mathcal{M}_2$ as a pair of bottleneck from $\mathcal{C}_1$ and $\mathcal{C}_2$ cannot be reached in $\text{Min}(\mathcal{C}_1, q_1) \times \text{Min}(\mathcal{C}_2, q_2)$. As a consequence, for $uv \in \mathcal{L}_{\mathcal{M}_1} \sqcap \mathcal{L}_{\mathcal{M}_2}$, there is no suffix w such that the state reached in $\mathcal{M}_1 \times \mathcal{M}_2$ after uvw is marked. As a result, $\mathcal{M}_1 \times \mathcal{M}_2$ is not terminating.

- (q'_1, q'_2) is a dead-end, reached in $\text{Min}(\mathcal{C}_1, q_1) \times \text{Min}(\mathcal{C}_2, q_2)$ after a word v , that is: $q'_1 \notin (F_1 \setminus D_1)$ or $q'_2 \notin (F_2 \setminus D_2)$, and there exists $X_1 \in \zeta_1(q'_1)$ and $X_2 \in \zeta_2(q'_2)$ such that: $(X_1 \cup (\Sigma_2 \setminus \Sigma_1)) \cap (X_2 \cup (\Sigma_1 \setminus \Sigma_2)) = \emptyset$.

We consider $\mathcal{M}_i \models \mathcal{C}_i$ for $i \in \{1, 2\}$ such that $\text{pr}_{\Sigma_i}(uv) \in \mathcal{L}_{\mathcal{M}_i}$ and such that $\lambda(r^0, \text{pr}_{\Sigma_i}(uv)) \in G$ if and only if $q'_i \in F_i \setminus D_i$ and $\text{ready}_i(\text{pr}_{\Sigma_i}(uv)) = X_i$. The automaton $\mathcal{M}_1 \times \mathcal{M}_2$ is not terminating as the pair of state reached after $uv \in \mathcal{L}_{\mathcal{M}_1} \sqcap \mathcal{L}_{\mathcal{M}_2}$ is not final and has no outgoing transition by construction. $\square$

We now define the product of two marked modal specifications with compatible reachability.

Definition 31 (Product) Given two marked modal specifications $\mathcal{C}_1 = (Q_1, q_1^0, \Sigma_1, \delta_1, \text{must}_1, \text{may}_1, F_1)$ and $\mathcal{C}_2 = (Q_2, q_2^0, \Sigma_2, \delta_2, \text{must}_2, \text{may}_2, F_2)$ with compatible reachability, the product $\mathcal{C}_1 \otimes \mathcal{C}_2$ is the normal form of the marked modal specification $(Q, q^0, \Sigma_1 \cup \Sigma_2, \delta, \text{must}, \text{may}, F)$ with:

1. $Q = Q_1 \times Q_2$ and $q^0 = (q_1^0, q_2^0)$;
2. for any $q_1 \in Q_1$, $q_2 \in Q_2$ and $a \in \Sigma_1 \cup \Sigma_2$, $\delta((q_1, q_2), a)$ is defined as $(\delta_1(q_1, a), \delta_2(q_2, a))$ for $a \in \Sigma_1 \cap \Sigma_2$, $(\delta_1(q_1, a), q_2)$ for $a \in \Sigma_1 \setminus \Sigma_2$ and $(q_1, \delta_2(q_2, a))$ for $a \in \Sigma_2 \setminus \Sigma_1$;
3. $\text{may}((q_1, q_2)) = (\text{may}_1(q_1) \cup (\Sigma_2 \setminus \Sigma_1)) \cap (\text{may}_2(q_2) \cup (\Sigma_1 \setminus \Sigma_2))$;
4. $\text{must}((q_1, q_2)) = (\text{must}_1(q_1) \cup (\Sigma_2 \setminus \Sigma_1)) \cap (\text{must}_2(q_2) \cup (\Sigma_1 \setminus \Sigma_2))$;
5. $(q_1, q_2) \in F$ if and only if $q_1 \in F_1$ and $q_2 \in F_2$.

Now, the product of any models $\mathcal{M}_1$ of $\mathcal{C}_1$ and $\mathcal{M}_2$ of $\mathcal{C}_2$ is model of $\mathcal{C}_1 \otimes \mathcal{C}_2$:

Proposition 32 Given two marked modal specifications $\mathcal{C}_1$ and $\mathcal{C}_2$, if $\mathcal{C}_1 \sim_{\mathcal{T}} \mathcal{C}_2$ then for any $\mathcal{M}_1 \models \mathcal{C}_1$ and $\mathcal{M}_2 \models \mathcal{C}_2$, $\mathcal{M}_1 \times \mathcal{M}_2 \models \mathcal{C}_1 \otimes \mathcal{C}_2$.

Proof: Let $u \in \mathcal{L}_{\mathcal{M}_1} \sqcap \mathcal{L}_{\mathcal{M}_2}$, we prove that:

1. $\text{must}(\delta((q_1^0, q_2^0), u)) \subseteq \text{ready}(\lambda((r_1^0, r_2^0), u)) \subseteq \text{may}(\delta((q_1^0, q_2^0), u))$;
2. if $\lambda((r_1^0, r_2^0), u) \in G$ then $\delta((q_1^0, q_2^0), u) \in F$.

Observe first that: $\text{ready}(\lambda((r_1^0, r_2^0), u)) = \text{ready}_1(\lambda_1(r_1^0, \text{pr}_{\Sigma_1}(u)))$ $\square$

Moreover, $\mathcal{C}_1 \otimes \mathcal{C}_2$ gives the most precise characterization of the behavior of the product of any models $\mathcal{M}_1$ of $\mathcal{C}_1$ and $\mathcal{M}_2$ of $\mathcal{C}_2$:

Proposition 33 Given two marked modal specifications $\mathcal{C}_1$ and $\mathcal{C}_2$, if $\mathcal{C}_1 \sim_{\mathcal{T}} \mathcal{C}_2$ and if there exists a marked modal specification $\mathcal{C}$ such that for any $\mathcal{M}_1 \models \mathcal{C}_1$ and $\mathcal{M}_2 \models \mathcal{C}_2$ we have $\mathcal{M}_1 \times \mathcal{M}_2 \models \mathcal{C}$ then $\mathcal{C}_1 \otimes \mathcal{C}_2 \leq \mathcal{C}$.

One important principle in modular and concurrent design of systems is the fact that a property checked on a primary version of some system artifacts remains true on any refined version of them. This is what allows to guarantee that the system parts corresponding to compatible interfaces can be designed concurrently. This is respected for compatible reachability:

Proposition 34 *For all marked modal specifications $\mathcal{C}_1$, $\mathcal{C}'_1$ and $\mathcal{C}_2$, if $\mathcal{C}_1 \sim_{\mathcal{T}} \mathcal{C}_2$ and $\mathcal{C}'_1 \preceq \mathcal{C}_1$ then $\mathcal{C}'_1 \sim_{\mathcal{T}} \mathcal{C}_2$ and $\mathcal{C}'_1 \otimes \mathcal{C}_2 \preceq \mathcal{C}_1 \otimes \mathcal{C}_2$.*

Proof: Let $\mathcal{M}_1$ and $\mathcal{M}_2$ be models of $\mathcal{C}'_1$ and $\mathcal{C}_2$. As $\mathcal{C}'_1 \preceq \mathcal{C}_1$, by Theorem 13, $\mathcal{M}_1$ is also a model of $\mathcal{C}_1$. Moreover, the product $\mathcal{M}_1 \times \mathcal{M}_2$ is terminating as $\mathcal{C}_1 \sim_{\mathcal{T}} \mathcal{C}_2$, by Theorem 28. As a result, by Theorem 28, $\mathcal{C}'_1 \sim_{\mathcal{T}} \mathcal{C}_2$. $\square$

Last, the product is a commutative and associative operator, meaning that interfaces can be assembled in any order without affecting the result.

Proposition 35 *The product of marked modal specifications is commutative and associative. Given three marked modal specifications $\mathcal{C}_1$, $\mathcal{C}_2$ and $\mathcal{C}_3$: $\mathcal{C}_1 \otimes \mathcal{C}_2 \equiv \mathcal{C}_2 \otimes \mathcal{C}_1$ and $\mathcal{C}_1 \otimes (\mathcal{C}_2 \otimes \mathcal{C}_3) \equiv (\mathcal{C}_1 \otimes \mathcal{C}_2) \otimes \mathcal{C}_3$.*

Remark 36 *If a bottleneck q_1 of $\mathcal{C}_1$ belongs to a cycle in $\text{Min}(\mathcal{C}_1, q_1)$ then for any q_2 of $\mathcal{C}_2$, $\text{Min}(\mathcal{C}_1, q_1) \times \text{Min}(\mathcal{C}_2, q_2)$ will not be terminating, that is $\mathcal{C}_1 \not\sim_{\mathcal{T}} \mathcal{C}_2$.*

5.2 Optimistic composition of marked modal specifications

Consider again $\mathcal{C}_1$ and $\mathcal{C}_2$ from Fig. 3. They do not have a compatible reachability as $(4, 1)$ is a reachable dead-end and thus an exception state pairs in the sense of Def. 26. It is however pessimistic to declare $\mathcal{C}_1$ and $\mathcal{C}_2$ as not composable. Indeed, the system potentially formed by any model of $\mathcal{C}_1$ and $\mathcal{C}_2$ would not be closed as the occurrence of $?e$ would still be under the control of the environment. Now by preventing the environment from producing $!e$ when $\mathcal{C}_1$ and $\mathcal{C}_2$ are in their initial state, the reachability of the exception state pairs $(4, 1)$ can be avoided⁴. In this section, let us now be optimistic and declare composable any $\mathcal{C}_1$ and $\mathcal{C}_2$ if there exists *at least one* environment, closing the system and preventing the reachability of *bad* states of $\mathcal{C}_1$ and $\mathcal{C}_2$ in which the reachability property cannot be guaranteed.

Definition 37 (Legal environment) *Given $\mathcal{M}$ and $\mathcal{E}$ two terminating automata, $\mathcal{E}$ is said to be a legal environment for $\mathcal{M}$, if and only if: the signature of $\mathcal{M}$ and $\mathcal{E}$ are composable; $\mathcal{M} \times \mathcal{E}$ is closed; $\text{Em}(\mathcal{M}) \sim_{\mathcal{T}} \text{Em}(\mathcal{E})$, that is $\mathcal{M} \times \mathcal{E}$ is terminating.*

Next, we identify *unusable* states from which the reachability of an exception state pair cannot be prevented regardless of the environment. This corresponds to pairs of state that can autonomously reach an exception state pair by a sequence of *output* actions:

Definition 38 (Unusable pair of states) *Given $\mathcal{C}_1$ and $\mathcal{C}_2$ two marked modal specifications with respective signature μ_1 and μ_2 , the set of unusable pair of states $U \subseteq Q_1 \times Q_2$ is the least set such that:*

- if $\text{Ex}(q_1, q_2)$ then $(q_1, q_2) \in U$;
- if, for $a \in \Sigma_{co}(\mu_1, \mu_2)$, $(\delta_1(q_1, a), \delta_2(q_2, a)) \in U$ then $(q_1, q_2) \in U$;
- if, for $a \in \Sigma_{pr}(\mu_1, \mu_2) \cap \Sigma_1^!$, $(\delta_1(q_1, a), q_2) \in U$ then $(q_1, q_2) \in U$;
- if, for $a \in \Sigma_{pr}(\mu_1, \mu_2) \cap \Sigma_2^!$, $(q_1, \delta_2(q_2, a)) \in U$ then $(q_1, q_2) \in U$.

⁴A similar pruning on the outputs of input complete specifications was proposed in [7], earlier than in [1].

A pair of states is said *usable* if it is not unusable.

Definition 39 (Optimistic compatible reachability) Two marked modal specifications $\mathcal{C}_1$ and $\mathcal{C}_2$ have an optimistic compatible reachability, noted $\mathcal{C}_1 \sim_{\mathcal{O}} \mathcal{C}_2$ if the pair of initial states (q_1^0, q_2^0) is usable, that is $(q_1^0, q_2^0) \notin U$.

This criterion is sound and complete as stated by the following Theorem:

Theorem 40 (Independent implementability) Given two marked modal specifications $\mathcal{C}_1$ and $\mathcal{C}_2$, $\mathcal{C}_1 \sim_{\mathcal{O}} \mathcal{C}_2$ if and only if for any $\mathcal{M}_1 \models \mathcal{C}_1$ and $\mathcal{M}_2 \models \mathcal{C}_2$ there exists a legal environment $\mathcal{E}$ for $\mathcal{M}_1 \times \mathcal{M}_2$.

Definition 41 (Optimistic product) Given two marked modal specifications $\mathcal{C}_1 = (Q_1, q_1^0, \Sigma_1, \delta_1, must_1, may_1, F_1)$ and $\mathcal{C}_2 = (Q_2, q_2^0, \Sigma_2, \delta_2, must_2, may_2, F_2)$ over composable signatures μ_1 and μ_2 and with optimistic compatible reachability, the optimistic product $\mathcal{C}_1 \parallel \mathcal{C}_2$ is the normal form of the marked modal specification $(Q, q^0, \Sigma_1 \cup \Sigma_2, \delta, must, may, F)$ over $\mu_1 \times \mu_2$ with:

- $Q = ((Q_1 \times Q_2) \setminus U) \cup \{q_{\top}\};$
- $q^0 = (q_1^0, q_2^0);$
- for any $q_1 \in Q_1, q_2 \in Q_2$ and $a \in \Sigma_1 \cup \Sigma_2$:
 - if $a \in \Sigma_{pr}(\mu_1, \mu_2) \cap \Sigma_1$: $\delta((q_1, q_2), a) = (\delta_1(q_1, a), q_2)$ if $(\delta_1(q_1, a), q_2) \notin U$ and $q_{\top}$ otherwise;
 - if $a \in \Sigma_{pr}(\mu_1, \mu_2) \cap \Sigma_2$: $\delta((q_1, q_2), a) = (q_1, \delta_2(q_2, a))$ if $(q_1, \delta_2(q_2, a)) \notin U$ and $q_{\top}$ otherwise;
 - if $a \in \Sigma_{co}(\mu_1, \mu_2)$: $\delta((q_1, q_2), a) = (\delta_1(q_1, a), \delta_2(q_2, a))$ if $(\delta_1(q_1, a), \delta_2(q_2, a)) \notin U$ and $q_{\top}$ otherwise;
- $may((q_1, q_2)) = (may_1(q_1) \cup (\Sigma_2 \setminus \Sigma_1)) \cap (may_2(q_2) \cup (\Sigma_1 \setminus \Sigma_2));$
- $must((q_1, q_2)) = (must_1(q_1) \cup (\Sigma_2 \setminus \Sigma_1)) \cap (must_2(q_2) \cup (\Sigma_1 \setminus \Sigma_2));$
- $may(q_{\top}) = \Sigma_1 \cup \Sigma_2$ and $must(q_{\top}) = \emptyset;$
- $(q_1, q_2) \in F$ if and only if $q_1 \in F_1$ and $q_2 \in F_2$; $q_{\top} \in F$.

This construction is very similar to the product $\otimes$ of Def. 31 except that unusable states are pruned away. Transitions labeled by an action $?a$ that would lead to an unusable state if performed by the environment are rerouted to a fresh state $q_{\top}$ on which all constraints on the future are relaxed.

Example 42 The optimistic product of $\mathcal{C}_1$ and $\mathcal{C}_2$ from is depicted in Fig. 3(c). The transition labeled by $?e$ is rerouted from the unusable $(3, 0)$ to $q_{\top}$.

The resulting $\mathcal{C}_1 \parallel \mathcal{C}_2$ is satisfied by the product of any models of $\mathcal{C}_1$ and $\mathcal{C}_2$:

Proposition 43 Given two marked modal specifications $\mathcal{C}_1$ and $\mathcal{C}_2$, if $\mathcal{C}_1 \sim_{\mathcal{O}} \mathcal{C}_2$ then for any $\mathcal{M}_1 \models \mathcal{C}_1$ and $\mathcal{M}_2 \models \mathcal{C}_2$, $\mathcal{M}_1 \times \mathcal{M}_2 \models \mathcal{C}_1 \parallel \mathcal{C}_2$.

The next proposition states that $\mathcal{C}_1 \parallel \mathcal{C}_2$ is the minimal marked modal specification w.r.t. refinement enjoying the independent implementability property:

Proposition 44 Given two marked modal specifications $\mathcal{C}_1$ and $\mathcal{C}_2$, if $\mathcal{C}_1 \sim_{\mathcal{O}} \mathcal{C}_2$ and if there exists a marked modal specification $\mathcal{C}$ such that for any $\mathcal{M}_1 \models \mathcal{C}_1$ and $\mathcal{M}_2 \models \mathcal{C}_2$ there exists a legal environment $\mathcal{E}$ for $\mathcal{M}_1 \times \mathcal{M}_2$ and $\mathcal{M}_1 \times \mathcal{M}_2 \models \mathcal{C}$, then $\mathcal{C}_1 \parallel \mathcal{C}_2 \leq \mathcal{C}$.

Optimistic compatible reachability is preserved by refinement hence allowing concurrent design of sub-systems. Moreover, the optimistic product is monotonic with respect to the refinement relation and is also associative which guarantees independence in the design flow.

Proposition 45 *For all marked modal specifications C_1 , C'_1 and C_2 , if $C_1 \sim_{\mathcal{O}} C_2$ and $C'_1 \preceq C_1$ then $C'_1 \sim_{\mathcal{O}} C_2$ and $C'_1 \parallel C_2 \preceq C_1 \parallel C_2$.*

Proposition 46 *The optimistic product of marked modal specifications is commutative and associative. Given three marked modal specifications C_1 , C_2 and C_3 : $C_1 \parallel C_2 \equiv C_2 \parallel C_1$ and $C_1 \parallel (C_2 \parallel C_3) \equiv (C_1 \parallel C_2) \parallel C_3$.*

6 Related works and conclusion

Marked modal specifications can be used to express, in a modular manner, that a system should be capable of reaching one or several marked states representing either the completion of a composition of services or the quiescence of a network of interacting agents. They improve the expressive power of deterministic modal specifications that corresponds to the conjunctive ν -calculus [11] which does not allow to capture reachability properties.

The same goal can be achieved with automata-theoretic specifications in which states are annotated with propositional formulas expressing implementation variants and, possibly, an obligation of progress. This is the case of *annotated automata* [24] and *operating guidelines* [21, 19]. While both formalisms have a product (or parallel) composition operator, they are missing the optimistic view of composition and also the conjunction operator that turns out to be instrumental as soon as components are described according to several distinct but interacting viewpoints [23].

The disjunctive variants of modal specifications [15, 10] allows to constraint progress and thus to inductively express reachability. However no implementation relations including marked states nor optimistic composition have been proposed for these variants of modal specifications.

Marked modal specifications look similar to the modal specifications with marked states introduced in [6]. However, these two formalisms are very different because the satisfaction relation in [6] admits implementations having final states corresponding to a state of the specification that is not final. This is appropriate in the context of supervisory control synthesis. However, this semantics does not seem well-suited to a specification algebra with a refinement preorder, which explains why a different satisfaction relation is used for marked modal specifications.

References

- [1] de Alfaro, L., Henzinger, T.A.: Interface automata. In: Proc. of the 9th ACM SIGSOFT Inter. Symp. on Foundations of Software Engineering (FSE'01). pp. 109–120. ACM Press (2001)
- [2] Alur, R., Henzinger, T.A., Kupferman, O., Vardi, M.Y.: Alternating refinement relations. In: Proc. of the 9th Inter. Conf. on Concurrency Theory (CONCUR'98). LNCS, vol. 1466, pp. 163–178. Springer (1998)
- [3] Antonik, A., Huth, M., Larsen, K.G., Nyman, U., Wasowski, A.: 20 years of modal and mixed specifications. Bulletin of the EATCS 1(94) (2008)
- [4] Bérard, B., Bidoit, M., Finkel, A., Laroussinie, F., Petit, A., Petrucci, L., Schnoebelen, Ph.: Systems and Software Verification. Model-Checking Techniques and Tools. Springer (2001)
- [5] Bertrand, N., Legay, A., Pinchinat, S., Raclet, J.B.: Modal event-clock specifications for timed component-based design. Science of Computer Programming (2011), in Press, Corrected Proof

- [6] Darondeau, P., Dubreil, J., Marchand, H.: Supervisory control for modal specifications of services. In: Workshop on Discrete Event Systems (WODES'10). pp. 428–435. Berlin, Germany (August 2010)
- [7] Dill, D.L.: Trace theory for automatic hierarchical verification of speed-independent circuits. ACM distinguished dissertations, MIT Press (1989)
- [8] Doyen, L., Henzinger, T.A., Jobstmann, B., Petrov, T.: Interface theories with component reuse. In: Proc. of the 8th Inter. Conf. on Embedded Software (EMSOFT'08). pp. 79–88. ACM Press (2008)
- [9] Fecher, H., de Frutos-Escrig, D., Lüttgen, G., Schmidt, H.: On the expressiveness of refinement settings. In: Proc. of the 3rd Inter. Conf. on Fundamentals of Software Engineering (FSEN'09). LNCS, vol. 5961, pp. 276–291. Springer (2009)
- [10] Fecher, H., Schmidt, H.: Comparing disjunctive modal transition systems with an one-selecting variant. J. Log. Algebr. Program. 77(1-2), 20–39 (2008)
- [11] Feuillade, G., Pinchinat, S.: Modal specifications for the control theory of discrete-event systems. Discrete Event Dynamic Systems 17(2), 181–205 (2007)
- [12] Henzinger, T.A., Sifakis, J.: The discipline of embedded systems design. IEEE Computer 40(10), 32–40 (2007)
- [13] Juhl, L., Larsen, K., Srba, J.: Modal transition systems with weight intervals. Journal of Logic and Algebraic Programming (2011), to appear.
- [14] Larsen, K.G., Nyman, U., Wasowski, A.: Modal I/O automata for interface and product line theories. In: Proc. of the 16th Euro. Symp. on Programming (ESOP'07). LNCS, vol. 4421, pp. 64–79. Springer (2007)
- [15] Larsen, K.G., Xinxin, L.: Equation solving using modal transition systems. In: Proc. of the 5th IEEE Symp. on Logic in Computer Science, LICS'90. pp. 108–117. IEEE Computer Society Press (1990)
- [16] Larsen, K.G.: Modal specifications. In: Automatic Verification Methods for Finite State Systems. LNCS, vol. 407, pp. 232–246. Springer (1989)
- [17] Larsen, K.G., Nyman, U., Wasowski, A.: On modal refinement and consistency. In: Proc. of the 18th Inter. Conf. on Concurrency Theory (CONCUR'07). pp. 105–119. Springer (2007)
- [18] Larsen, K.G., Thomsen, B.: A modal process logic. In: Proc. of the 3rd Annual Symp. on Logic in Computer Science (LICS'88). pp. 203–210. IEEE (1988)
- [19] Lohmann, N., Wolf, K.: Compact representations and efficient algorithms for operating guidelines. Fundam. Inform. 108(1-2), 43–62 (2011)
- [20] Lynch, N., Tuttle, M.R.: An introduction to Input/Output automata. CWI-quarterly 2(3), 219–246 (1989)
- [21] Massuthe, P., Schmidt, K.: Operating guidelines - an automata-theoretic foundation for the service-oriented architecture. In: QSIC. pp. 452–457. IEEE Computer Society (2005)
- [22] Raclet, J.B., Badouel, E., Benveniste, A., Caillaud, B., Legay, A., Passerone, R.: Modal interfaces: unifying interface automata and modal specifications. In: Proc. of the 9th Int. Conf. on Embedded Software (EMSOFT'09). pp. 87–96. ACM (2009)

- [23] Raclet, J.B., Benveniste, A., Caillaud, B., Legay, A., Passerone, R.: A modal interface theory for component-based design. *Fundam. Inform.* 107, 1–32 (2011)
- [24] Wombacher, A., Mahleko, B., Neuhold, E.J.: IPSI-PF - a business process matchmaking engine based on annotated finite state automata. *Inf. Syst. E-Business Management* 3(2), 127–150 (2005)



**RESEARCH CENTRE
RENNES – BRETAGNE ATLANTIQUE**

Campus universitaire de Beaulieu
35042 Rennes Cedex

Publisher
Inria
Domaine de Volveau - Rocquencourt
BP 105 - 78153 Le Chesnay Cedex
inria.fr

ISSN 0249-6399