Synthesis of component adaptor via quotient of specifications

Jean-Baptiste Raclet

INRIA Rennes, France, Jean-Baptiste.Raclet@irisa.fr

Supervisor: Eric Badouel, INRIA Rennes, France,
Keywords: Component-based System, Adaptor, Modal Specifications

Abstract. We address the problem of reuse of a component as the synthesis of an adaptor that moderate a given component in order to realize a prescribed behaviour. We propose a solution for this problem when behaviours are given by modal specifications. The adaptor synthesis algorithm relies on a quotient operation that computes the residual specification.

1 Introduction

Component-based design is acknowledged as an relevant approach to improving productivity in the design of complex software systems as it allows pre-designed reactive components to be reused in larger systems. Reuse can be addressed as wrapping a reactive component with given behaviour into an adaptor, so as to realize a global prescribed behaviour. It can thus be seen as a kind of extended supervisor synthesis with the main difference that adaptor are general transducers, hence they can both relabel sequences of component events to (longer or shorter) sequences of system events, and also contribute directly to sequences of system.

In this paper, we propose a solution to this problem when behaviours are modal specifications: given a component $\mathcal{C}_1$, model of a modal specification $\mathcal{S}_1$, and a modal specification $\mathcal{S}$ for the desired behaviour of the overall system, we compute the residual specification $\mathcal{S}/\mathcal{S}_1$ characterizing those components $\mathcal{C}_2$ that, when composed with $\mathcal{C}_1$, provide a system satisfying $\mathcal{S}$.

In section 2, we define modal specification. In section 3, we explain how to compute $\mathcal{S}/\mathcal{S}_1$ and how to build a suitable adaptor from such a modal specification.

2 Definition of modal specifications

A modal specification can be seen as a simple automaton with a modality on each labeled transition: we distinguish events that necessarily occur (must modality graphically represented by plain arrows) and events that possibly occur (may modality graphically represented by dashed arrows). Moreover, the absence of

a transition labeled by an event a from a state points out that this event is forbidden in this state. Because of the may modality, this model is well suited to represent partially specified systems.

An example of modal specification defined on the alphabet $\{a, b\}$ is depicted in fig.1(a); it specifies that, after the event a, the event a is available and b may occur.

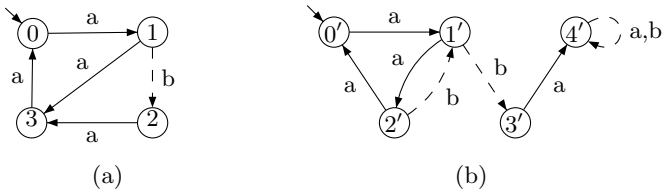

Fig. 1. Modal specifications

More formally, a modal specification $\mathcal{S}$ over an alphabet Σ is a tuple $(Q, q^0, \Delta, may, must)$ where Q is a non-empty and finite set of states and $q^0 \in Q$ is the initial state. The function $\Delta : Q \times \Sigma \to Q$ is a partial function describing a deterministic transition system. The functions $may, must : Q \to \mathcal{P}(\Sigma)$ are partial functions that types events: $a \in may(q)$ means that a is allowed from q; $a \in must(q)$ means that a is required from q; $a \notin may(q)$ means that a is forbidden in q. We require that for all state q, $must(q) \subseteq may(q)$ which expresses that anything that is required should also be allowed. Modal specifications are supposed well-formed: $a \in may(q)$ if and only if $\Delta(q, a)$ is defined.

In the sequel, we shall freely use the following notations:

- if $a \in must(q)$: we write $q \xrightarrow{a} q'$ for $\Delta(q, a) = q'$ and $q \xrightarrow{a}$ for $\exists q'.\Delta(q, a) = q'$
- if $a \in may(q) \setminus must(q)$: we write $q \overset{a}{\dashrightarrow} q'$ for $\Delta(q, a) = q'$
- if $a \notin may(q)$: we write $q \overset{a}{\not\to}$

Since the underlying transition system is deterministic, we adopt the notation $q \bullet u$ to denote the state q' reached from q by reading the word u. The partial function $\bullet : Q \times \Sigma^* \to Q$ is given by :

$q \bullet \epsilon \;\; = q$
$q \bullet a \;\; = q'$ iff $\Delta(q, a) = q'$
$q \bullet ua = q'$ iff $q \bullet u = q'' \;\wedge\; \Delta(q'', a) = q'$

Such a sequence u of events is called a trace of $\mathcal{S}$.

A model of a modal specification $\mathcal{S}$ is a deterministic automaton $\mathcal{C} = (R, r^0, \delta)$ with R a non-empty finite set of states, $r^0 \in R$ the initial state, $\delta : R \times \Sigma \to R$ a partial function of transition. We let $out(r)$ be the set of events a such that $\delta(r, a)$ is defined. As previously, we shall use $r \xrightarrow{a} r'$ for $\delta(r, a) = r'$. Moreover $r' = r \bullet u$ express that the state r' is reached from r after the sequence of events u. We note $\mathcal{C}_1 \times \mathcal{C}_2$ the synchronous product of $\mathcal{C}_1$ and $\mathcal{C}_2$.

The automaton $\mathcal{C}$ is a model of the modal specification $\mathcal{S}$ (noted $\mathcal{C} \models \mathcal{S}$) if there exists a relation $\sim \subseteq R \times Q$ such that:

– $r^0 \sim q^0$
– $\forall (r,q)$ such that $r \sim q$: $must(q) \subseteq out(r) \subseteq may(q)$ and furthermore, $\forall a \in out(r), \delta(r,a) \sim \Delta(q,a)$.

Thus, every trace u of a model $\mathcal{C}$ of $\mathcal{S}$ is also a trace of $\mathcal{S}$ and if $r = r^0 \bullet u$ and $q = q^0 \bullet u$ then $must(q) \subseteq out(r) \subseteq may(q)$. By determinacy, the relation $\sim$ is unique when it exists.

For technical reasons, we shall consider modal specifications that have states such that $must(q) \not\subseteq may(q)$. An event a that must occur ($a \in must(q)$) but that is also not allowed ($a \notin may(q)$) is said incoherently specified (noted $q \not\downarrow a$). Such a state can't belong to a simulation relation stating that $\mathcal{C} \models \mathcal{S}$ because this would imply, for a state q such that $r \sim q$ and $must(q) \not\subseteq may(q)$, on one hand that $r \xrightarrow{a}$ (as $a \in must(q)$) and, on the other hand, that $r \overset{a}{\not\to}$ (as $a \notin may(q)$). In the sequel, these particular modal specifications will be called pseudo-modal specifications. We can note two particular pseudo-modal specifications:
– $(\{q^0\}, q^0, \Delta, may, must)$ with, for all event $a \in \Sigma$, $q^0 \overset{a}{\dashrightarrow} q^0$ (ie. $must(q^0) = \emptyset$ and $may(q^0) = \Sigma$), admit every automaton over Σ as model.
– $(\{q^0\}, q^0, \Delta, may, must)$ with, for all event $a \in \Sigma$, $q^0 \not\downarrow a$ (ie. $must(q^0) = \Sigma$ and $may(q^0) = \emptyset$), has no model.

Remark 1: if $\mathcal{S}$ is a pseudo-modal specification, it is always possible to construct an equivalent modal specification (ie. a modal specification that has exactly the same models) using the algorithm 1. This algorithm consists in first, marking all states that can reach a state having incoherently specified event by a sequence of "must" transitions, and then pruning the pseudo-modal specification by cutting "may" transitions leading to marked states. For example, the pseudo-modal specification in fig.2(a) is equivalent to the modal specification in fig.2(b).

3 Quotient of modal specifications

We define the pseudo-modal specification $\mathcal{S}/\mathcal{S}_1$ where $\mathcal{S} = (Q, q^0, \Delta, may, must)$ and $\mathcal{S}_1 = (Q_1, q_1^0, \Delta_1, may_1, must_1)$ are modal specifications defined over Σ. The set of states $Q_/$ of $\mathcal{S}/\mathcal{S}_1$ is $(Q \times Q_1) \cup \{\top\}$ and the initial state is $q_/^0 = (q^0, q_1^0)$. The functions $\Delta_/$ and $may_/$ and $must_/$ are defined according to the following rules:
– for the state $\top$, we have : $\top \overset{a}{\dashrightarrow} \top, \forall a \in \Sigma$
– and for each pair (q, q_1):
  • if $q \xrightarrow{a} q'$ and $q_1 \xrightarrow{a} q_1'$ then $(q, q_1) \xrightarrow{a} (q', q_1')$
  • if $q \xrightarrow{a} q'$ and ($q_1 \overset{a}{\dashrightarrow} q_1'$ or $q_1 \overset{a}{\not\to}$) then $(q, q_1) \not\downarrow a$
  • if $q \overset{a}{\not\to}$ and ($q_1 \xrightarrow{a} q_1'$ or $q_1 \overset{a}{\dashrightarrow} q_1'$) then $(q, q_1) \overset{a}{\not\to}$
  • if $q \overset{a}{\not\to}$ and $q_1 \overset{a}{\not\to}$ then $(q, q_1) \overset{a}{\dashrightarrow} \top$
  • if $q \overset{a}{\dashrightarrow}$ and $q_1 \overset{a}{\not\to}$ then $(q, q_1) \overset{a}{\dashrightarrow} \top$
  • if $q \overset{a}{\dashrightarrow}$ and ($q_1 \xrightarrow{a} q_1'$ or $q_1 \overset{a}{\dashrightarrow} q_1'$) then $(q, q_1) \overset{a}{\dashrightarrow} (q', q_1')$

L(*set of states*), L'(*set of states*);

// initialisation
$\mathsf{L} \leftarrow \{q \mid \exists a.q \not\downarrow a\}$;

// marking
repeat
  $\mathsf{L}' \leftarrow \mathsf{L}$;
  forall q *in* $Q \setminus \mathsf{L}$ **do**
    if $\exists a.q \xrightarrow{a} q'$ *with* $q' \in \mathsf{L}$ **then**
      $\mathsf{L} = \mathsf{L} \cup \{q\}$;
    end
  end
until $\mathsf{L} \neq \mathsf{L}'$;

// pruning
forall q *in* $Q \setminus \mathsf{L}$ **do**
  if $\exists a.q \overset{a}{\dashrightarrow} q'$ *avec* $q' \in \mathsf{L}$ **then**
    $may(q) = may(q) \setminus \{a\}$;
    $\Delta(q, a) =$ undefined;
  end
end

Algorithm 1: Construction of a coherently specified modal specification

The quotient of the modal specification in fig.1(a) and fig.1(b) is depicted in fig.2(a).

Proposition 1. *Let $\mathcal{S}$ and $\mathcal{S}_1$ be two modal specifications over Σ, $\mathcal{C}_2$ is a model of $\mathcal{S}/\mathcal{S}_1$ if and only if, for all $\mathcal{C}_1$ model of $\mathcal{S}_1$, we have $\mathcal{C}_1 \times \mathcal{C}_2 \models \mathcal{S}$.*

Proof *($\Rightarrow$) We let $\sim_1$ and $\sim_2$ be the simulation relations that respectively state that $\mathcal{C}_1 \models \mathcal{S}_1$ and $\mathcal{C}_2 \models \mathcal{S}/\mathcal{S}_1$. We consider the least relation $\sim \subseteq (R \times R_1) \times Q$ such that :*

- $(r_1^0, r_2^0) \sim q^0$
- $\forall((r_1, r_2), q)$ *such that* $(r_1, r_2) \sim q$, $\forall a \in out(r_1, r_2)$, *we have* $(r_1', r_2') \sim q'$ *with* $(r_1, r_2) \xrightarrow{a} (r_1', r_2')$ *and* $\Delta(q, a) = q'$.

In order to prove that $\sim$ is a simulation stating that $\mathcal{C}_1 \times \mathcal{C}_2 \models \mathcal{S}$, we have to check that $(r_1, r_2) \sim q$ entails $must(q) \subseteq out(r_1, r_2) \subseteq may(q)$. For all $a \in out(r_1, r_2)$, $(r_1, r_2) \xrightarrow{a} (r_1', r_2')$ in $\mathcal{C}_1 \times \mathcal{C}_2$ implies $r_1 \xrightarrow{a} r_1'$ in $\mathcal{C}_1$ and $r_2 \xrightarrow{a} r_2'$ in $\mathcal{C}_2$. As $\mathcal{C}_1 \models \mathcal{S}_1$, there exists a state q_1 such that $r_1 \sim_1 q_1$. Moreover, according to the definition of the quotient and to the fact that $\mathcal{C}_2 \models \mathcal{S}/\mathcal{S}_1$, $r_2 \sim_2 (q, q_1)$. So $a \in out(r_1)$ and $a \in out(r_2)$ implies respectively that $a \in may(q_1)$ and $a \in may((q, q_1))$.

- *if $(q, q_1) \xrightarrow{a}$, necessarily $q \xrightarrow{a}$ and $a \in may(q)$*
- *if $(q, q_1) \overset{a}{\dashrightarrow} (q', q_1')$ then $q \overset{a}{\dashrightarrow}$ and $(q_1 \xrightarrow{a} q_1'$ or $q_1 \overset{a}{\dashrightarrow} q_1')$ and $a \in may(q)$*
- *if $(q, q_1) \overset{a}{\dashrightarrow} \top$ then $(q \overset{a}{\dashrightarrow}$ and $q_1 \overset{a}{\not\dashrightarrow})$ or $(q \overset{a}{\not\dashrightarrow}$ and $q_1 \overset{a}{\dashrightarrow})$; this situation is impossible because $a \in may(q_1)$*

By the previous case analysis, if $a \in out((r_1, r_2))$ *then* $a \in may(q)$*. Next we have to check that if* $a \in must(q)$ *then* $a \in out((r_1, r_2))$ *: for all* $a \in must(q)$*, as* $r_2 \sim_2 (q, q_1)$*, we necessarily have that* a *is coherently specified in* (q, q_1) *and* $q_1 \xrightarrow{a}$ *and* $(q, q_1) \xrightarrow{a}$ *hence* $r_1 \xrightarrow{a}$ *and* $(r_1, r_2) \xrightarrow{a}$*.*

(⇐) By contraposition, it suffices to prove that if $\mathcal{C}_2 \not\models \mathcal{S}/\mathcal{S}_1$ *then there exists an automaton* $\mathcal{C}_1$ *such that* $\mathcal{C}_1 \models \mathcal{S}_1$ *and* $\mathcal{C}_1 \times \mathcal{C}_2 \not\models \mathcal{S}$*.*

If $\mathcal{C}_2 \not\models \mathcal{S}/\mathcal{S}_1$ *then there exists a trace* $u = a_0 a_2 ... a_n$ *such that if we let* $u_i = a_0 a_2 ... a_i$ *for* $0 \leq i < n$ *then:*

- $must((q^0, q_1^0) \bullet u_i) \subseteq out(r_2^0 \bullet u_i) \subseteq may((q^0, q_1^0) \bullet u_i)$
- *and there is no incoherently specified event in each* $(q^0, q_1^0) \bullet u_i$*.*
- *but* $must((q^0, q_1^0) \bullet u) \not\subseteq out(r_2^0 \bullet u)$ *or* $out(r_2^0 \bullet u) \not\subseteq may((q^0, q_1^0) \bullet u)$ *or there exists an event* a *such that* $(q^0, q_1^0) \bullet u \not\downarrow a$

As $may(\top) = \Sigma$ *and* $must(\top) = \emptyset$*,* $(q^0, q_1^0) \bullet u \neq \top$*, hence,* $(q^0, q_1^0) \bullet u_i \neq \top$*. Furthermore, according to the definition of the quotient and to the absence of incoherently specified event for each* $(q^0, q_1^0) \bullet u_i$*,* u *is a trace common to* $\mathcal{S}$ *and* $\mathcal{S}_1$*. We build the automaton* $\mathcal{C}_1$ *model of* $\mathcal{S}_1$ *such that for every pair* $(r_1^0 \bullet u_i, q_1^0 \bullet u_i)$ *with* $(r_1^0 \bullet u_i) \sim_1 (q_1^0 \bullet u_i)$ *we have* $a_i \in out(r_1^0 \bullet u_i)$*. Thus,* u *is a trace of* $\mathcal{C}_1$ *and hence, a trace of* $\mathcal{C}_1 \times \mathcal{C}_2$*. Three possibilities can lead to* $\mathcal{C}_2 \not\models \mathcal{S}/\mathcal{S}_1$ *:*

- $\exists a$ *s.t.* $r_2^0 \bullet u \xrightarrow{a}$ *and* $(q^0, q_1^0) \bullet u \not\xrightarrow{a}$ *:*
  $(q^0, q_1^0) \bullet u \not\xrightarrow{a}$ *iff* $q^0 \bullet u \not\xrightarrow{a}$ *and* $(q_1^0 \bullet u \xrightarrow{a}$ *or* $q_1^0 \bullet u \dashrightarrow)$*. If* $r_1^0 \bullet u \xrightarrow{a}$ *in* $\mathcal{C}_1$ *then* $(r_1^0, r_2^0) \bullet u \xrightarrow{a}$ *in* $\mathcal{C}_1 \times \mathcal{C}_2$*. As* $q^0 \bullet u \not\xrightarrow{a}$*,* $\mathcal{C}_1 \times \mathcal{C}_2 \not\models \mathcal{S}$
- $\exists a$ *s.t.* $r_2^0 \bullet u \not\xrightarrow{a}$ *and* $(q^0, q_1^0) \bullet u \xrightarrow{a}$ *:*
  $(q^0, q_1^0) \bullet u \xrightarrow{a}$ *iff* $q^0 \bullet u \xrightarrow{a}$ *and* $q_1^0 \bullet u \xrightarrow{a}$ *but, as* $r_2^0 \bullet u \not\xrightarrow{a}$ *in* $\mathcal{C}_2$ *then* $(r_1^0, r_2^0) \bullet u \not\xrightarrow{a}$ *and as* $q^0 \bullet u \xrightarrow{a}$ *then* $\mathcal{C}_1 \times \mathcal{C}_2 \not\models \mathcal{S}$
- $\exists a$ *s.t.* $(q^0, q_1^0) \bullet u \not\downarrow a$ *:*
  It occurs when $q^0 \bullet u \xrightarrow{a}$ *and* $(q_1^0 \bullet u \dashrightarrow$ *or* $q_1^0 \bullet u \not\xrightarrow{a})$*. If* $r_1^0 \bullet u \not\xrightarrow{a}$ *in* $\mathcal{C}_1$ *then* $(r_1^0, r_2^0) \bullet u \not\xrightarrow{a}$ *in* $\mathcal{C}_1 \times \mathcal{C}_2$*. As* $q^0 \bullet u \xrightarrow{a}$*,* $\mathcal{C}_1 \times \mathcal{C}_2 \not\models \mathcal{S}$ □

The proposition 1 can be used to solve an adaptor synthesis problem: given a component $\mathcal{C}_1$ that satisfies $\mathcal{S}_1$, you can adapt it to satisfies $\mathcal{S}$ using a model of $\mathcal{S}/\mathcal{S}_1$. To optimize the reuse, one is interesting in having the least adaptor such that the desired behaviour is obtained. This can easily be computed when using modal specifications: if a modal specification $\mathcal{S}$ is satisfiable, its set of models is a distributive complete lattice for the inclusion of language. The least (respectively, the greatest) model $\mathcal{C}$ is the one such that for every pair (r, q) such that $r \sim q$ we have $out(r) = must(q)$ (respectively, $out(r) = may(q)$). For example, the least model of the modal specification in fig.2(b) is the automaton in fig.2(c).

To sum up, whatever be the component $\mathcal{C}_1$ model of $\mathcal{S}_1$ and a desired behaviour $\mathcal{S}$: compute $\mathcal{S}/\mathcal{S}_1$ (the complexity of this step is $O(|\mathcal{S}| \times |\mathcal{S}_1|)$); compute the modal specification equivalent to the pseudo-modal specification $\mathcal{S}/\mathcal{S}_1$ using the algorithm 1; compute the least model of the previous modal specification (if it exists, ie. if (q^0, q_1^0) has no incoherently specified event) by cutting every dashed transition. Then the obtained automaton $\mathcal{C}_2$ is the least adaptor, for every $\mathcal{C}_1$ model of $\mathcal{S}_1$, such that $\mathcal{C}_1 \times \mathcal{C}_2$ satisfies $\mathcal{S}$.

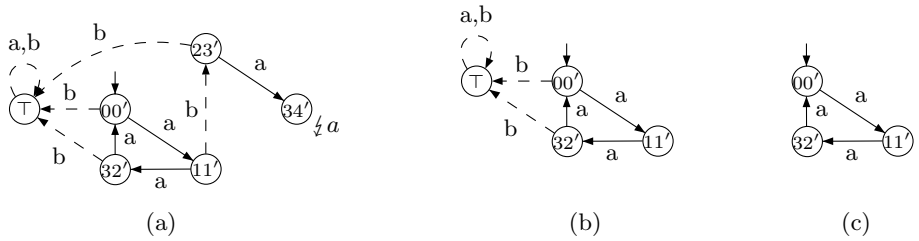

Fig. 2. From the residual specification to an adaptor

4 Conclusion and future work

In this paper, we have presented modal specifications, a simple and intuitive formalism to specify systems. Furthermore, we have define an adaptor synthesis algorithm for a component whose behaviour is expressed via a modal specification. This algorithm computes a residual specification by quotienting the desired behaviour and the known one to be reused.

Quotient of specifications given by Mu-calculus formulas has been studied in [AVW03]. The proposition 1 is true in this framework. A benefit of using Mu-calculus formulas instead of modal specifications is expressity (cf. [Feu05]). Nevertheless, the complexity of the quotient operation of Mu-calculus formulas is worse: the size of the automaton associated to the formula $\mathcal{S}/\mathcal{S}_1$ is $O(|\mathcal{S}| \times m')$ where $m' = 2^{O(m.\log(m))}$ and $m = O(|\mathcal{S}_1|)$ whereas for modal specifications, the size of $\mathcal{S}/\mathcal{S}_1$ is $O(|\mathcal{S}| \times |\mathcal{S}_1|)$. Furthermore, there is no notion of least model of a Mu-calculus formula so it is not possible to define a least adaptor.

An extension of this work could be to enrich modal specifications in order to add a mechanism to check the compatibility between two components (as in the work of [dAH01]).

A crucial aspect in component reuse is how to enforce the reuse of the given component during the adaptor synthesis. The product operation between the provided component and the synthesized adaptor can be used in this purpose. We expect to propose a solution to the problem of adaptor synthesis that consider the product operation as a parameter of the problem.

References

[AVW03] A. Arnold, A. Vincent, and I. Walukiewicz. Games for synthesis of controllers with partial observation. *Theorical Computer Science*, 303(1):7–34, 2003.

[dAH01] L. de Alfaro and T.A. Henzinger. Interface automata. In *Proceedings of the Ninth Annual Symposium on Foundations of Software Engineering*, pages 109–120. ACM Press, 2001.

[Feu05] Guillaume Feuillade. Modal specifications are a syntactic fragment of the mu-calculus. Research Report PI-1729, IRISA, 2005.