

Interface Theories for (A)synchronously Communicating Modal I/O-Transition Systems*

Sebastian S. Bauer Rolf Hennicker Stephan Janisch

Institut für Informatik
Ludwig-Maximilians-Universität München, Germany
{bauerse,hennicker,janisch}@pst.ifi.lmu.de

Interface specifications play an important role in component-based software development. An interface theory is a formal framework supporting composition, refinement and compatibility of interface specifications. We present different interface theories which use modal I/O-transition systems as their underlying domain for interface specifications: synchronous interface theories, which employ a synchronous communication schema, as well as a novel interface theory for asynchronous communication where components communicate via FIFO-buffers.

1 Introduction

The idea of an interface theory is to capture basic requirements that any formalism should obey which is intended to support the design of components and component systems. Since system development usually concerns two dimensions, a horizontal dimension where larger components are built from smaller ones, and a vertical dimension, where interface specifications are successively refined (and finally implemented), an interface theory requires concepts of composition, refinement and compatibility. Of course, it is important that the different dimensions of system development fit properly together. Therefore an interface theory requires (at least) that refinement is preserved by composition and that compatibility of interfaces is preserved by refinement, which is needed for independent implementability and reusability of components.

A formal notion of an *interface theory* was, to our knowledge, first proposed by de Alfaro and Henzinger in [2]. In their work, an interface theory consists of an interface algebra together with a component algebra thus distinguishing between interface specifications and component implementations. Later, in [3], the authors have introduced the term *interface language* which simplifies the approach by considering just interfaces with the requirements that incremental design and independent implementability is possible. Interface theory and interface language are abstract concepts which can be instantiated by concrete formalisms. The (abstract) notion of an interface theory we shall use hereafter is close to an interface language but further simplified by concentrating on the two rudimentary requirements mentioned above which guarantee independent implementability and which we want to study for particular interface theories supporting synchronous as well as asynchronous composition.

All interface theories studied in this work use modal I/O-transition systems (MIOs), introduced by Larsen et al. [11], [12], as underlying formalism for interface specifications. MIOs are well suited to describe behavioural properties of reactive components. They allow to distinguish between transitions which are optional or mandatory for refinements and thus support loose specification and stepwise development. We first summarize our previous work on interface theories [6] which was based on synchronous

*This research has been supported by the GLOWA-Danube project 01LW0602A2 sponsored by the German Federal Ministry of Education and Research.

composition of MIOs. We discuss strong and weak versions of refinement and compatibility and we show that both versions lead to an interface theory. Then we extend our previous work and consider asynchronous composition of MIOs which communicate via output queues. We introduce the notion of asynchronous compatibility which requires that each message put in the output queue of a MIO must eventually be taken by its communication partner which is related to the requirement of specified reception in communicating finite state machines [8]. We show that MIOs with asynchronous composition, asynchronous compatibility and weak refinement form again an interface theory. Finally, we discuss possibilities for verification and further directions of our work.

2 Interface Theories for MIOs with Synchronous Composition

In our study the abstract concept of an interface theory defines rudimentary properties that should be satisfied by any formal framework for interface specifications. Given a class \mathcal{A} of interface specifications, an interface theory includes a partial composition operator \otimes to combine specifications to larger ones. The composition operator is, in general, partial since it is not always syntactically meaningful to compose specifications. Interface specifications for which the composition is defined are called composable. Additionally, an interface theory must offer a refinement relation \leq to relate “concrete” and “abstract” specifications, and a compatibility relation \rightleftharpoons to express when two interface specifications describe components which can work properly together. In contrast to (syntactic) composability, compatibility has a semantic flavour related to the behaviour of components. To obtain an interface theory, three requirements must be satisfied. Obviously, compatible specifications must be syntactically composable. Moreover, refinement must be compositional in the sense that it must be preserved by the composition operator and, third, compatibility must be preserved by refinement.

Definition 1 (Interface Theory). *An interface theory is a tuple $(\mathcal{A}, \otimes, \leq, \rightleftharpoons)$ consisting of a class \mathcal{A} of interface specifications, a partial composition operator $\otimes : \mathcal{A} \times \mathcal{A} \rightarrow \mathcal{A}$, a reflexive and transitive refinement relation $\leq \subseteq \mathcal{A} \times \mathcal{A}$, and a symmetric compatibility relation $\rightleftharpoons \subseteq \mathcal{A} \times \mathcal{A}$, such that the following conditions are satisfied. Let $S, S', T, T' \in \mathcal{A}$ be interfaces.*

- (1) (Compatibility implies composability) *If $S \rightleftharpoons T$ then $S \otimes T$ is defined.*
- (2) (Compositional refinement) *If $S' \leq S$ and $T' \leq T$ and $S \otimes T$ is defined, then $S' \otimes T'$ is defined and $S' \otimes T' \leq S \otimes T$.*
- (3) (Preservation of compatibility) *If $S \rightleftharpoons T$ and $S' \leq S$ and $T' \leq T$, then $S' \rightleftharpoons T'$.*

Obviously, in a top-down design, the requirements for an interface theory expressed by conditions (1) to (3) support independent development of components and thus independent implementability in the sense of [3]. To a certain extent an interface theory supports also bottom-up design, where existing components can be reused as parts of a larger system architecture, as long as local refinements are correct and local interfaces fit into the context.

In the following we will study particular interface theories which all use modal I/O-transition systems (MIOs) as their underlying formalism for interface specifications. Modal I/O-transition systems have been introduced by Larsen et al. [11], [12] as a formalism to describe the behaviour of reactive, concurrent components. MIOs distinguish between may- and must-transitions, where the former model allowed behaviour, which may or may not be present in a refinement, whereas the latter model required behaviour to be preserved by any refinement. Thus MIOs support loose specifications and flexible notions of refinement.

Definition 2 (MIO). A modal I/O-transition system (MIO) $S = (states_S, start_S, act_S, \dashrightarrow_S, \longrightarrow_S)$ consists of a set of states $states_S$, an initial state $start_S \in states_S$, a set act_S of actions being the disjoint union of sets in_S , out_S and int_S of input, output and internal actions resp., a may-transition relation $\dashrightarrow_S \subseteq states_S \times act_S \times states_S$, and a must-transition relation $\longrightarrow_S \subseteq \dashrightarrow_S$, i.e. every required transition is also allowed. The set act_S of actions together with its partition into input, output and internal actions is called the signature of S .

As usual, we write $s \dashrightarrow_S^a s'$ instead of $(s, a, s') \in \dashrightarrow_S$, and similarly for must-transitions. A state $s \in states_S$ of S is called *reachable* if there exist may-transitions $s_0 \dashrightarrow_S^{a_0} s_1 \dashrightarrow_S^{a_1} \dots \dashrightarrow_S^{a_{n-1}} s_n$, $n \geq 0$, such that $s_n = s$. The class of modal I/O-transition systems is denoted by \mathcal{M} . It provides the underlying domain of specifications for all interface theories considered in the following.

Two MIOs $S, T \in \mathcal{M}$ are (*syntactically*) *composable* if their actions only overlap on complementary types, i.e. $act_S \cap act_T \subseteq (in_S \cap out_T) \cup (in_T \cap out_S)$. The *set of shared actions* $act_S \cap act_T$ is denoted by $shared(S, T)$. The *synchronous composition* of two composable MIOs S and T is defined as the usual product of transition systems with synchronization on shared actions which become internal in the product. A synchronization transition in the composition is a must-transition only if both of the single synchronized transitions were must-transitions.

Definition 3 (Synchronous composition). Let $S, T \in \mathcal{M}$ be two composable MIOs. The synchronous composition of S and T is the MIO $S \otimes_{sy} T = (states_S \times states_T, (start_S, start_T), act, \dashrightarrow, \longrightarrow)$ where the action alphabet act is the disjoint union of the input actions $(in_S \cup in_T) \setminus shared(S, T)$, the output actions $(out_S \cup out_T) \setminus shared(S, T)$, and the internal actions $int_S \cup int_T \cup shared(S, T)$. The transition relations are the smallest relations satisfying:

- for all $a \in shared(S, T)$,
 - if $s \dashrightarrow_S^a s'$ and $t \dashrightarrow_T^a t'$, then $(s, t) \dashrightarrow^a (s', t')$,
 - if $s \longrightarrow_S^a s'$ and $t \dashrightarrow_T^a t'$, then $(s, t) \longrightarrow^a (s', t')$,
- for all $a \in act_S \setminus shared(S, T)$,
 - if $s \dashrightarrow_S^a s'$, then $(s, t) \dashrightarrow^a (s', t)$ for all $t \in states_T$,
 - if $s \longrightarrow_S^a s'$, then $(s, t) \longrightarrow^a (s', t)$ for all $t \in states_T$,
- for all $a \in act_T \setminus shared(S, T)$,
 - if $t \dashrightarrow_T^a t'$, then $(s, t) \dashrightarrow^a (s, t')$ for all $s \in states_S$,
 - if $t \longrightarrow_T^a t'$, then $(s, t) \longrightarrow^a (s, t')$ for all $s \in states_S$.

The basic idea of *modal refinement* is that required (*must*) transitions of an abstract specification must also occur in the concrete specification. Conversely, allowed (*may*) transitions of the concrete specification must be allowed by the abstract specification. We distinguish between *strong modal refinement*, due to [12] and denoted by \leq_m , and *weak modal refinement*, due to [9] and denoted by \leq_m^* , which are both defined in terms of a simulation relation. While in the strong case every transition must be simulated “immediately”, weak refinement allows to abstract from transitions with internal actions. We only review the formal definition of the latter here. In the following, the successive execution of arbitrarily many internal must-transitions is denoted by \dashrightarrow^{τ^*} , and similarly for may-transitions.

Definition 4 (Weak modal refinement). Let S and T be MIOs with the same signature. S weakly modally refines T , written $S \leq_m^* T$, if there exists a relation $R \subseteq states_S \times states_T$ containing $(start_S, start_T)$ such that for all $(s, t) \in R$:

- (1) $\forall a \in in_T \cup out_T : t \xrightarrow{a}_T t' \implies \exists s \xrightarrow{\tau}_S^* \bar{s} \xrightarrow{a}_S \bar{s} \xrightarrow{\tau}_S^* s' \wedge (s', t') \in R,$
- (2) $\forall a \in int_T : t \xrightarrow{a}_T t' \implies \exists s \xrightarrow{\tau}_S^* s' \wedge (s', t') \in R,$
- (3) $\forall a \in in_S \cup out_T : s \xrightarrow{a}_S s' \implies \exists t \xrightarrow{\tau}_T^* \bar{t} \xrightarrow{a}_T \bar{t} \xrightarrow{\tau}_T^* t' \wedge (s', t') \in R,$
- (4) $\forall a \in int_S : s \xrightarrow{a}_S s' \implies \exists t \xrightarrow{\tau}_T^* t' \wedge (s', t') \in R.$

In conditions (2) and (4), a is an internal action which must be simulated by a sequence of arbitrarily many internal actions (denoted by $\xrightarrow{\tau}_S^*$, $\xrightarrow{\tau}_T^*$ resp.). This sequence may be empty but the important point is that the original transition with a must stay in the relation R .

Our notion of strong modal compatibility is inspired by [3] and [11]. Two MIOs S and T are *strongly modally compatible*, denoted by $S \rightleftharpoons_{sc} T$, if they are composable and if for each reachable state (s, t) in the composition $S \otimes_{sy} T$, if S may send out in state s an action shared with T , then T must be able to receive it in state t , and conversely. The difference to [3] and [11] is that we consider the “pessimistic” case, where MIOs should work properly together in *any* composable environment while the “optimistic” approach, pursued in [3] and [11], requires the existence of a (helpful) environment; for a discussion see [1].

Strong modal refinement is compositional w.r.t. the synchronous product [12] and preserves strong modal compatibility [6]. Thus we obtain a first interface theory. The detailed proof can be found in [7].

Theorem 1. $(\mathcal{M}, \otimes_{sy}, \leq_m, \rightleftharpoons_{sc})$ is an interface theory.

Weak modal refinement, however, does not preserve strong modal compatibility due to the possible insertion of internal transitions in the refinement; see [6] for a counterexample. Therefore, we have introduced in [6] a weak version of compatibility such that a communication partner can delay the reception of a message by performing some internal must-transitions before.

Definition 5 (Weak modal compatibility). *Two MIOs S and T are weakly modally compatible, denoted by $S \rightleftharpoons_{wc} T$, if they are composable and if for all reachable states (s, t) in $S \otimes_{sy} T$,*

- (1) $\forall a \in out_S \cap in_T : s \xrightarrow{a}_S s' \implies \exists t \xrightarrow{\tau}_T^* \bar{t} \xrightarrow{a}_T t',$
- (2) $\forall a \in out_T \cap in_S : t \xrightarrow{a}_T t' \implies \exists s \xrightarrow{\tau}_S^* \bar{s} \xrightarrow{a}_S s'.$

Since weak modal refinement is compositional w.r.t. the synchronous product [9] and preserves weak modal compatibility [6] we obtain a second interface theory. For a detailed proof see again [7].

Theorem 2. $(\mathcal{M}, \otimes_{sy}, \leq_m^*, \rightleftharpoons_{wc})$ is an interface theory.

All kinds of refinement and synchronous compatibility notions considered here are decidable for finite MIOs and can be efficiently computed in time polynomial in the size of the MIOs. For further variants of interface theories with synchronous composition and for an introduction of the MIO Workbench for refinement and compatibility checking see [6].

3 An Interface Theory for MIOs with Asynchronous Composition

In distributed applications, implemented, for instance, with a message-oriented middleware, usually an asynchronous communication pattern is used. To obtain an interface theory for this kind of systems we change the composition operator and focus on components which communicate via FIFO-buffered message queues. In Fig. 1 two asynchronously communicating MIOs S and T are schematically depicted: S sends a message n to T by putting it into a queue which stores the outputs of S , and then T can receive n by removing n from the queue. Obviously, there is a delay between sending and reception. Similarly,

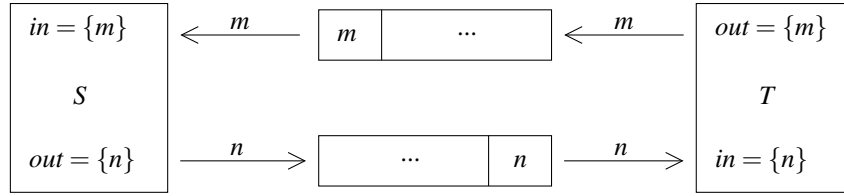


Figure 1: Asynchronously communicating MIOs

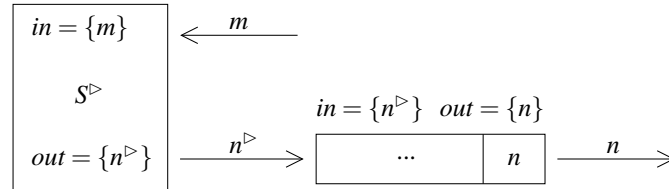


Figure 2: MIO with output queue

T can send a message m to S by using a second queue which stores the outputs of T . Technically, we enhance MIOs by output queues which are themselves modelled as MIOs. Given a MIO S and a distinguished subset $o \subseteq out$ of the output actions of S , the MIO S “with output queue for the messages in o ” is modelled by the synchronous product of a renamed version of S (where all $n \in o$ are renamed to n^\triangleright) and the “queue MIO” Q_o which is able to store messages of o . Fig. 2 shows the idea of this construction where S^\triangleright denotes the renamed version of S .

Definition 6 (MIO with output queue). *Given a set o of output actions, the queue MIO for o is $Q_o = (o^*, \varepsilon, act, \dashrightarrow, \longrightarrow)$ where the set of states o^* is the set of all finite strings over o , the initial state $\varepsilon \in o^*$ is the empty string, and the set of actions act is the disjoint union of input actions $in = \{n^\triangleright \mid n \in o\}$, output actions $out = o$ and with no internal action. Moreover, $\dashrightarrow = \longrightarrow$ and the must-transition relation \longrightarrow is the smallest relation such that*

- for all $n^\triangleright \in in$ and states $s \in o^* : s \xrightarrow{n^\triangleright} ns$,
- for all $n \in out (= o)$ and states $s \in o^* : sn \xrightarrow{n} s$.

Given a MIO S with actions $act_S = in_S \cup out_S \cup int_S$ and a distinguished set $o \subseteq out_S$ of output actions, the MIO S with output queue for o is given by the synchronous product $\Omega_o(S) = S_o^\triangleright \otimes_{sy} Q_o$ (where S_o^\triangleright denotes the renamed version of S where all $n \in o$ are renamed to n^\triangleright). Obviously, the product is well-defined since S_o^\triangleright and Q_o are composable.

By the rules of synchronous composition the input and the output actions of $\Omega_o(S)$ coincide with those of S ; an output n of $\Omega_o(S)$ means that the message n is either a free output of S or it is removed from the output queue of S . The synchronization actions n^\triangleright of $\Omega_o(S)$ express that the message n is put by S (more precisely by S_o^\triangleright) in the queue.

To define the asynchronous composition of two MIOs S and T , we assume again that S and T are composable. Then one can equip S with an output queue for those outputs o_S of S which can be received by T , i.e. which are shared actions. The other output actions of S remain free. Similarly T is equipped with an output queue for its shared output actions o_T . Obviously, since S and T are composable, $\Omega_{o_S}(S)$ and $\Omega_{o_T}(T)$ are composable as well. Hence, two composable MIOs S and T can be asynchronously composed by synchronously composing their extensions by output queues.

Definition 7 (Asynchronous composition). *Let S, T be two composable MIOs and $o_S = out_S \cap in_T$, $o_T = out_T \cap in_S$. The asynchronous composition of S and T is defined by $S \otimes_{as} T = \Omega_{o_S}(S) \otimes_{sy} \Omega_{o_T}(T)$.*

We consider two composable MIOs S and T to be asynchronously compatible, if for each reachable state in $S \otimes_{as} T$, if the output queue of S is not empty, then T must be able to take (i.e. input) the next removable element of the queue possibly after some internal must-transitions, and conversely. Obviously, due to the use of output queues (instead of input queues), this idea can be easily formalized with the help of weak modal compatibility as defined in the synchronous case.

Definition 8 (Asynchronous modal compatibility). *Two MIOs S and T are asynchronously modally compatible, denoted by $S \rightleftharpoons_{ac} T$, if they are composable and if, for $o_S = out_S \cap in_T$, $o_T = out_T \cap in_S$, $\Omega_{o_S}(S) \rightleftharpoons_{wc} \Omega_{o_S}(T)$.*

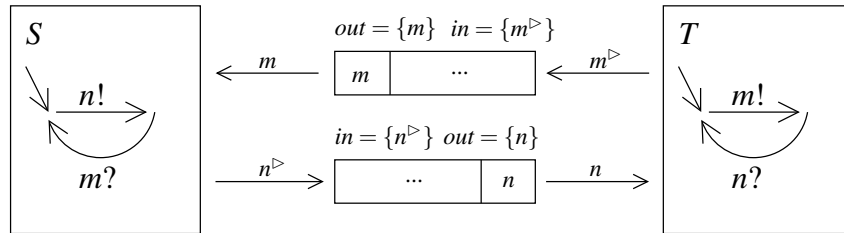


Figure 3: Example of asynchronously communicating MIOs

As a simple example consider the two MIOs S and T depicted in Fig. 3 where input actions are marked with “?” and output actions with “!”, i.e. $in_S = out_T = \{m\}$ and $out_S = in_T = \{n\}$. S has the transitions $start_S \xrightarrow{n} s \xrightarrow{m} start_S$, and T has the transitions $start_T \xrightarrow{m} t \xrightarrow{n} start_T$. S and T are asynchronously compatible, since each communication partner must take the provided message after it has put its own issued message in its queue (which is an internal must-transition in $\Omega_{o_S}(S)$ and $\Omega_{o_T}(T)$ resp.). Note that S and T are obviously neither strongly nor weakly modally compatible which shows the flexibility of the asynchronous compatibility concept. The other way round it is shown in [10] that, under certain conditions like input separated states, weak compatibility implies asynchronous compatibility.

The behaviour described by the asynchronous composition of MIOs coincides with the operational model of communicating finite state machines (CFSMs); see [8]. In [8] it is required that a system of CFSMs should be well-formed. One part of the well-formedness condition requires that executable receptions should be specified, which is just the strong version of the asynchronous compatibility notion used here. The other direction of the well-formedness condition requires that specified receptions should be executable. This corresponds to a kind of “input” compatibility which we have not considered here, since, in general, it would not be necessary that any service offered by a component must actually be used. Another difference to CFSMs is that we consider a binary (asynchronous) composition operator but allow open systems, while in the CFSM approach closed networks of CFSMs are considered.

To obtain an interface theory with asynchronous composition we still have to choose an appropriate refinement notion. After a closer look it becomes obvious that refinement is not really related to the communication paradigm, since refinement concerns the vertical dimension of software development moving from abstract to more concrete abstraction levels, whereas composition is related to the horizontal dimension where larger systems are constructed from smaller ones and where the underlying communication schema is crucial. Hence, we can simply reuse the powerful notion of weak modal refinement which leads to an interface theory for MIOs with asynchronous composition.

Theorem 3. $(\mathcal{M}, \otimes_{as}, \leq_m^*, \rightleftharpoons_{ac})$ is an interface theory.

Proof. The proof relies on the previous results for the synchronous case, since the asynchronous notions have been defined in terms of the synchronous ones. As a first observation, we show that for any two MIOs S and S' and for any subset o of output actions of S and of S' ,

$$S' \leq_m^* S \implies \Omega_o(S') \leq_m^* \Omega_o(S). \quad (\text{I})$$

Since weak modal refinement is compositional, by Thm. 2, $S' \leq_m^* S$ implies $S' \otimes_{\text{sy}} Q_o \leq_m^* S \otimes_{\text{sy}} Q_o$. Hence, by definition, $\Omega_o(S') = S'_o \triangleright \otimes_{\text{sy}} Q_o \leq_m^* S_o \triangleright \otimes_{\text{sy}} Q_o = \Omega_o(S)$.

We can now prove that the conditions (1) - (3) of an interface theory are satisfied.

- (1) Asynchronously compatible MIOs are, by definition, composable.
- (2) Compositionality of refinement: Assume that $S' \leq_m^* S$, $T' \leq_m^* T$ and that $S \otimes_{\text{as}} T$ is defined, i.e. S and T are composable. Since weak modal refinement \leq_m^* does not change signatures, S' and T' are composable as well, i.e. $S' \otimes_{\text{as}} T'$ is defined.

We have to show that $S' \otimes_{\text{as}} T' \leq_m^* S \otimes_{\text{as}} T$ which means, by definition,

$$\Omega_{o_{S'}}(S') \otimes_{\text{sy}} \Omega_{o_{T'}}(T') \leq_m^* \Omega_{o_S}(S) \otimes_{\text{sy}} \Omega_{o_T}(T) \quad (\text{II})$$

where $o_{S'} = \text{out}_{S'} \cap \text{in}_{T'}$, $o_{T'} = \text{out}_{T'} \cap \text{in}_{S'}$, $o_S = \text{out}_S \cap \text{in}_T$, and $o_T = \text{out}_T \cap \text{in}_S$. First, $S' \leq_m^* S$ implies that S and S' have the same signature; the same holds for T and T' . Therefore, $o_{S'} = o_S$ and $o_{T'} = o_T$. By (I), $S' \leq_m^* S$ and $T' \leq_m^* T$ implies $\Omega_{o_S}(S') \leq_m^* \Omega_{o_S}(S)$ and $\Omega_{o_T}(T') \leq_m^* \Omega_{o_T}(T)$, respectively. Then, (II) follows from compositionality of \leq_m^* w.r.t. synchronous composition \otimes_{sy} , see Thm. 2, taking into account $o_{S'} = o_S$ and $o_{T'} = o_T$.

- (3) Preservation of compatibility under refinement: Assume that $S \rightleftharpoons_{\text{ac}} T$, $S' \leq_m^* S$ and $T' \leq_m^* T$. By definition, $S \rightleftharpoons_{\text{ac}} T$ means $\Omega_{o_S}(S) \rightleftharpoons_{\text{wc}} \Omega_{o_T}(T)$. From (I) we know that $S' \leq_m^* S$ implies $\Omega_{o_S}(S') \leq_m^* \Omega_{o_S}(S)$ and $T' \leq_m^* T$ implies $\Omega_{o_T}(T') \leq_m^* \Omega_{o_T}(T)$. By Thm. 2, $\rightleftharpoons_{\text{wc}}$ is preserved under \leq_m^* and therefore $\Omega_{o_S}(S') \rightleftharpoons_{\text{wc}} \Omega_{o_T}(T')$. Thus $\Omega_{o_{S'}}(S') \rightleftharpoons_{\text{wc}} \Omega_{o_{T'}}(T')$, since $o_{S'} = o_S$ and $o_{T'} = o_T$ as above. This means, by definition, $S' \rightleftharpoons_{\text{ac}} T'$. □

4 Conclusion

We have studied interface theories based on modal I/O-transition systems (MIOs) with synchronous and with asynchronous composition. We have chosen MIOs as the underlying domain for interface specifications since they allow for a flexible refinement notion. In the synchronous case, if the underlying MIOs are finite, strong and weak refinement as well as strong and weak compatibility are decidable and can be efficiently checked with the MIO Workbench; see [6] and [13]. In the asynchronous case, the buffering mechanism used for communication may lead to infinite state spaces. Concerning refinement it is, however, still possible to derive weak refinements between composed specifications with infinite state spaces, say $S' \otimes_{\text{as}} T' \leq_m^* S \otimes_{\text{as}} T$, from local refinements $S' \leq_m^* S$ and $T' \leq_m^* T$ and the latter can be decided if the local MIOs are finite. This is an important consequence of the interface theory with asynchronous composition. The situation is different, if we consider the verification of asynchronous compatibility which is, in general, not decidable due to the potentially infinite output queues. We are currently working on criteria for asynchronous compatibility, which are decidable and powerful at the same time, and on the integration of such criteria into the MIO Workbench. As an outcome of our theoretical work, we want to apply the results to provide a solid basis for modelling hierarchical and

asynchronously communicating components in the context of the Unified Modeling Language (UML). At the same time we are also interested in interface theories for components with local data states [5, 4] and for timed systems.

Acknowledgement. An important input for this study was the suggestion of Alexander Knapp to use output queues (instead of input queues) for the formalization of asynchronous compatibility. We are grateful to Alexander for this very valuable hint.

References

- [1] Luca de Alfaro & Thomas A. Henzinger (2001): *Interface automata*. *SIGSOFT Softw. Eng. Notes* 26, pp. 109–120.
- [2] Luca de Alfaro & Thomas A. Henzinger (2001): *Interface Theories for Component-Based Design*. In: Thomas A. Henzinger & Christoph M. Kirsch, editors: *EMSOFT, LNCS 2211*, Springer, pp. 148–165.
- [3] Luca de Alfaro & Thomas A. Henzinger (2005): *Interface-based Design*. In: Manfred Broy, Johannes Grünbauer, David Harel & C. A. R. Hoare, editors: *Engineering Theories of Software-intensive Systems, NATO Science Series: Mathematics, Physics, and Chemistry* 195, Springer, pp. 83–104.
- [4] Sebastian S. Bauer, Rolf Hennicker & Michel Bidoit (2010): *A Modal Interface Theory with Data Constraints*. In: *SBMF 2010, LNCS*, Springer. To appear.
- [5] Sebastian S. Bauer, Rolf Hennicker & Stephan Janisch (2010): *Behaviour Protocols for Interacting Stateful Components*. *Electr. Notes Theor. Comput. Sci.* 263, pp. 47–66.
- [6] Sebastian S. Bauer, Philip Mayer, Andreas Schroeder & Rolf Hennicker (2010): *On Weak Modal Compatibility, Refinement, and the MIO Workbench*. In: Javier Esparza & Rupak Majumdar, editors: *TACAS, LNCS 6015*, Springer, pp. 175–189.
- [7] Sebastian S. Bauer, Philip Mayer, Andreas Schroeder & Rolf Hennicker (2010): *On Weak Modal Compatibility, Refinement, and the MIO Workbench*. Technical Report 1001, Institut für Informatik, Ludwig-Maximilians-Universität München.
- [8] Daniel Brand & Pitro Zafropulo (1983): *On Communicating Finite-State Machines*. *J. ACM* 30(2), pp. 323–342.
- [9] Hans Hüttel & Kim Guldstrand Larsen (1989): *The Use of Static Constructs in A Modal Process Logic*. In: Albert R. Meyer & Michael A. Taitslin, editors: *Logic at Botik, LNCS 363*, Springer, pp. 163–180.
- [10] Stephan Janisch (2010): *Behaviour and Refinement of Port-Based Components with Synchronous and Asynchronous Communication*. Ph.D. thesis, Institut für Informatik, Ludwig-Maximilians-Universität München.
- [11] Kim Guldstrand Larsen, Ulrik Nyman & Andrzej Wasowski (2007): *Modal I/O Automata for Interface and Product Line Theories*. In: Rocco De Nicola, editor: *ESOP, LNCS 4421*, Springer, pp. 64–79.
- [12] Kim Guldstrand Larsen & Bent Thomsen (1988): *A Modal Process Logic*. In: *LICS*, IEEE Computer Society, pp. 203–210.
- [13] MIO-Workbench. <http://www.miowb.net/>.