Noname manuscript No.
(will be inserted by the editor)

A Generalised Theory of Interface Automata, Component Compatibility and Error

Sascha Fendrich · Gerald Lüttgen

Received: date / Accepted: date

Abstract Interface theories allow system designers to reason about the composability and compatibility of concurrent system components. Such theories often extend both de Alfaro and Henzinger's *Interface Automata* and Larsen's *Modal Transition Systems*, which leads, however, to several issues that are undesirable in practice: an unintuitive treatment of specified unwanted behaviour, a binary compatibility concept that does not scale to multi-component assemblies, and compatibility guarantees that are insufficient for software product lines.

In this article we show that communication mismatches are central to all these problems and, thus, the ability to represent such errors semantically is an important feature of an interface theory. Accordingly, we present the *error-aware* interface theory EMIA, where the above shortcomings are remedied by introducing explicit *fatal error states*. In addition, we prove via a Galois insertion that EMIA is a conservative generalisation of the established MIA (Modal Interface Automata) theory.

1 Introduction

Today's software systems are increasingly composed from off-the-shelf components. Hence, software developers desire to detect incompatibilities between components early. This is supported by *interface theories* [2,7,8,10,14,15,19,26,30,32], which may serve as specification theories for component-based design [15,2,9,24], software product lines [26], web services [5] and the Internet of Things [29]. Interface theories may also be employed as contract languages or behavioural type theories when transitioning from software design to implementation [1,20].

Many interface theories [2,7,26,30,32] extend de Alfaro and Henzinger's *Interface Automata* (IA) [14,15] and Larsen's *Modal Transition Systems* (MTS) [25,28]. In order to express compatibility assumptions of components on the communication behaviour of their environment, IA divides an interface's action alphabet into input actions ('?'), output actions ('!') and an internal action τ. A *communication mismatch*, or error, arises between parallelly composed components P and Q, if P may

This is a post-peer-review, pre-copyedit version of an article published in Acta Informatica. The final authenticated version is available online at: `https://doi.org/10.1007/s00236-018-0319-8`. An extended abstract of this article appeared in [19] and this article also provides a foundation of the first author's dissertation [18]. Research has been supported by the DFG (German Research Foundation) under grants LU-1748/3-1&2.

Software Technologies Research Group
University of Bamberg
96045 Bamberg, Germany
E-mail: sascha.fendrich@swt-bamberg.de · gerald.luettgen@swt-bamberg.de

issue an output $a!$ while Q is not ready to receive the input $a?$ in its current state. Orthogonally, MTS permits one to specify required and optional behaviour. Taking stepwise decisions on the optional behaviour allows for a component-based, incremental design, which is supported by a compositional refinement preorder.

1.1 Shortcomings of Related Work

Unfortunately, interface theories combining IA and MTS [2,7,26,30,32] have several issues that impact their practical use.

Issue A: Forbidden inputs are preserved by the resp. refinement preorder but are largely ignored by parallel composition, so that behaviour that is forbidden in one component may be re-introduced in the composed system if another component defies this prohibition. This unintuitive treatment of communication mismatches and, in particular, unwanted behaviour, is dangerous for safety-critical applications.

Issue B: Pairwise binary compatibility of multiple components is neither necessary nor sufficient for their overall compatibility when being considered as a multi-component assembly, even if parallel composition is associative. To address this, Hennicker and Knapp [22] have introduced *assembly theories* that extend interface theories by a separate level of assemblies where multi-component compatibility is checked. However, these assemblies have to be re-interpreted as interfaces to be of further use.

Issue C: Optional behaviour, modelled via may-transitions as in MTS, may be employed to express variability inherent in software product lines. In current interface theories, two product families may be considered compatible only if all products of one family are compatible with all products of the other. However, one would prefer a more detailed set of guarantees, such that one may distinguish if all, some or none of the product lines' products are compatible [26] in order to compute compatible subfamilies.

Issue D: MTS and MTS-based interface theories have some subtle differences wrt. their treatment of modalities, resulting in different composition concepts: in MTS, components unanimously agree on transitions of their composition; in interface theories, an error arises if the components' requirements do not match. Each theory makes a global choice of a composition concept, which is tightly bound to a respective compatibility notion and does not allow one to mix different compatibility and composition concepts that are suitable for the application at hand.

We illustrate the compatibility problems of current interface theories by means of an example highlighting Issue A; this and the other issues are further discussed in Sec. 3. Consider a driving assistance system that enables a car to drive into and out of a garage autonomously. Such a system must communicate with the garage in order to make it open and close its door.

Fig. 1 shows specifications G and C of the garage's and the car's interfaces, resp. In figures, we use capital letters in italic font followed by a colon to represent (names of) interface specifications, small letters in italics for states of a specification, and normal font for transition labels. Initial states are marked by a small arrow usually placed at the top left. The alphabets of transition labels are written as $A = I/O$, where I and O are the sets of input and output labels, resp.

Starting in state g_0, the garage is ready to receive a passage request (rqstPass?). After such a request, the garage opens its door (openDoor!), waits for a car driving in or out (drive?) and, finally, closes the door (closeDoor!) again. The car starts in state c_0 waiting for a user's request

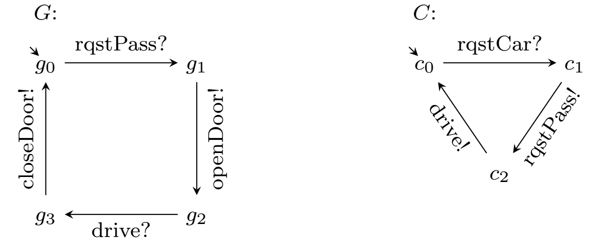

Fig. 1 Example of a driving assistant system including a garage G and a car C, where $A_G := \{\text{drive?}, \text{rqstPass?}\}/\{\text{closeDoor!}, \text{openDoor!}\}$ and $A_C := \{\text{rqstCar?}\}/\{\text{drive!}, \text{rqstPass!}\}$.

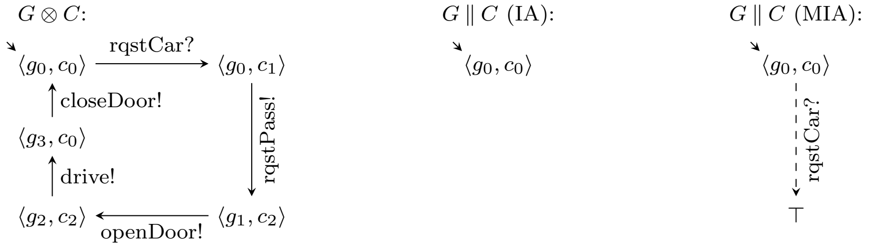

Fig. 2 Parallel product in IA or MIA (left), and parallel composition in IA (middle) and MIA (right) of the components depicted in Fig. 1, where $A_{G\otimes C} = A_{G\|C} := \{\text{rqstCar?}\}/\{\text{drive!}, \text{closeDoor!}, \text{openDoor!}, \text{rqstPass!}\}$.

(rqstCar?). Upon receiving such a request, the car requests passage from the garage (rqstPass!) and then drives into or out of the garage (drive!), reaching state c_0 again.

Specifications G and C have a communication mismatch due to the drive!-transition at state c_2 and the fact that no drive?-transition is specified at state g_1. Hence, in the parallel product $G \otimes C$ shown in Fig. 2 (left), state $\langle g_1, c_2\rangle$ is considered illegal. In interface theories with a *pessimistic* notion of compatibility, e.g., [2,30], the parallel composition of G and C is undefined, because the illegal state $\langle g_1, c_2\rangle$ is reachable from the initial state $\langle g_0, c_0\rangle$. *Optimistic* theories, e.g., [7,8,10,14,15, 26,30,32], assume a helpful environment that tries to steer away from communication mismatches by controlling the composed system via its input transitions. A state is optimistically illegal if a communication mismatch is reachable via uncontrollable actions, i.e., output or τ-transitions. The parallel composition $G \parallel C$ is obtained from $G \otimes C$ by removing all illegal states. In our example, state $\langle g_1, c_2\rangle$ is illegal, just as state $\langle g_0, c_1\rangle$ from which $\langle g_1, c_2\rangle$ is reachable by an output (rqstPass!). This pruning leaves a single state $\langle g_0, c_0\rangle$ with no transitions; all other states are unreachable. The rqstCar?-transition at state $\langle g_0, c_0\rangle$, which would allow one to reach illegal states when triggered by the environment, is also removed. However, in order to ensure compositionality of refinement, rqstCar? must be permitted with arbitrary behaviour afterwards (cf. [7]); IA-based refinement [14, 15,30] allows this implicitly for all unspecified inputs (Fig. 2, middle). In MTS-based interface theories, where unspecified transitions represent forbidden behaviour, compositionality is achieved by replacing pruned behaviour by an explicit optional transition to a special, universally refinable state $\top$ (Fig. 2, right) that semantically stands for arbitrary behaviour [7].

Due to this possibility of introducing arbitrary behaviour in case of a communication mismatch, stepwise refinement may re-introduce behaviour that has previously been removed due to the mismatch. Hence, optimistic theories accept a car driving into or out of the garage before the door is opened as a valid implementation of $G\|C$. This contradicts G's sensible constraint that driving in or out is only permitted after the door has been opened, i.e., the meaning of a car crashing into the door can simply be 'refined' to not being an error. In other words, the assumptions and guarantees expressible in current interface theories are insufficient for expressing unwanted behaviour.

Bujtor and Vogler [8] have shown that keeping or removing illegal states on a purely syntactic level are equivalent for IA wrt. preserving compatibility. In this spirit, current interface theories [2, 7, 8, 14, 15, 26, 30, 32] eliminate erroneous behaviour either by regarding it as undefined (pessimistic) or by pruning (optimistic); all errors are considered semantically equivalent. Due to this equivalence, theories combining IA and MTS cannot remove illegal states completely but must replace them by a special, arbitrarily refinable behaviour as mentioned above. However, because optional transitions (i.e., may-transitions) and disjunctive transitions allow for underspecification in MTS-based interface theories, one may distinguish potential errors that can be resolved by a suitable refinement from actual, unresolvable errors that arise when an output is required and the corresponding input is forbidden. That is, specifications based on MTS contain more information wrt. compatibility, which we make explicit in *Error-aeare Modal Interface Automata* (EMIA). EMIA guarantees that compatible specifications have only compatible implementations, potential errors have both compatible and erroneous implementations, and actual errors have only erroneous implementations (cf. Sec. 3.3, Issue C).

1.2 Contributions and Organisation

This article shows that communication mismatches are central to Issues A–D above. Hence, the ability to represent such errors semantically is an important feature that is missing in current interface theories. In Sec. 2 we present the core of our interface theory *Error-aware Modal Interface Automata* (EMIA), for which we remedy Issues A–D by making communication mismatches explicit in the form of *fatal error states* and by employing an *error-preserving refinement preorder* and an *error-aware parallel composition*. In contrast, current interface theories [2, 7, 8, 10, 14, 15, 26, 30, 32] remove such information about the causes and possible resolutions of communication mismatches.

In Sec. 3 we show that a Galois insertion [13] renders our refined semantics a conservative extension of the arguably most general interface theory to date, MIA (Modal Interface Automata) [7]. We also revisit the introductory example in terms of EMIA, and discuss how fatal error states solve Issues A–D. The resulting specification theory tightly integrates MTS, interface theories and assembly theories, and allows system designers to combine the different composition concepts of these theories within a single interface specification.

In Sec. 4 we discuss the logical operators conjunction and disjunction for EMIA, as is typical for interface theories. In addition, we introduce an underapproximation of implication for EMIA and show that implication cannot be fully supported in MTS-based interface theories. We discuss the Galois insertion between MIA and EMIA wrt. these logical operators and illustrate how they enable system designers to combine operational and declarative specification styles.

Sec. 5 discusses the standard process algebraic operators hiding, restriction and alphabet extension. In addition, we present a quotienting operator that is adjoint to parallel composition and allows for reasoning in the context of component reuse. The impact of the Galois insertion on these operators is discussed, and the usage of the operators in the design process is demonstrated by means of an example.

In summary, EMIA generalises previous interface theories by means of a semantic representation of communication mismatches. This results in a better theoretical understanding of the concept of error and yields a more flexible interface theory that resolves several practical issues (Issues A–D) with previous interface theories.

1.3 Added Value of this Journal Version

Compared to the extended abstract that appeared in [19], this article contains the proofs of all technical results, more explanations and examples as well as extended discussions, e.g., of assembly theories (cf. Sec. 3).

In addition, this article covers more operators: (a) a quotient operator adjoint to parallel composition, including the consideration of quotienting under the Galois insertion and additional properties of quotienting when compared to other interface theories such as antitonicity in the divisor and a De Morgan-like law (cf. Sec. 5); (b) a hiding and a restriction operator, and (c) a discussion of implication and negation, which are rarely considered in interface theories, including a proof that MTS-based interface theories are not closed under implication and negation (cf. Sec. 4).

As another new contribution of this article, we have generalised the main theory EMIA, which now supports *universal states* and allows for a more uniform presentation of MIA and EMIA that simplifies some of the proofs (cf. Sec. 2). In [19] universal states were not necessary for presenting the main idea of fatal error states. We now include universal states for several reasons. Firstly, because we here include the quotient operator $/\!\!/$: the quotient $P /\!\!/ D$ is the maximal specification Q satisfying $Q \parallel D \sqsubseteq P$. An action that is not used by D may be implemented with arbitrary subsequent behaviour in such a maximal Q. This arbitrary behaviour is exactly the meaning of universal states. Secondly, as universal states are already necessary in MIA to make the refinement preorder compositional when pruning errors, allowing universal states in EMIA yields a more uniform presentation of the two theories. As a side benefit, the Galois insertion of MIA into EMIA becomes simpler because an infinite disjunction is not needed anymore (cf. Sec. 3).

2 Error-aware Modal Interface Automata

Our interface theory *Error-aware Modal Interface Automata* (EMIA), which we present in this section, is equipped with a *parallel composition operator* modelling concurrency and communication, a *conjunction operator* permitting the specification of a component from different perspectives, a *disjunction operator* for providing alternatives, a *quotienting operator* allowing for component reuse, and a *compositional refinement preorder* enabling the substitution of an interface by a more concrete version. In addition to these standard requirements on interface theories, EMIA solves Issues A–D of Sec. 1. We achieve this by introducing *fatal error states*, which represent unresolvable incompatibilities between interfaces. This enables EMIA to deal with errors on a semantic level, since forbidden behaviour can be modelled by input transitions leading to a fatal error state.

Definition 1 (Error-aware Modal Interface Automata) An *Error-aware Modal Interface Automaton* (EMIA) is a tuple $P := (S_P, I_P, O_P, \longrightarrow_P, \dashrightarrow_P, S_P^0, E_P, U_P)$, where S_P is the set of states, I_P, O_P are the disjoint alphabets of input and output actions not including the silent action τ (we define $A_P := I_P \cup O_P$ and $\Omega_P := O_P \cup \{\tau\}$), $\longrightarrow_P \subseteq S_P \times (A_P \cup \{\tau\}) \times \mathfrak{P}(S_P)$ is the disjunctive must-transition relation ($\mathfrak{P}$ denotes the power set operator), $\dashrightarrow_P \subseteq S_P \times (A_P \cup \{\tau\}) \times S_P$ is the may-transition relation, $S_P^0 \subseteq S_P$ is the set of initial states, $E_P \subseteq S_P$ is the set of fatal error states and $U_P \subseteq S_P$ is the set of universal states, if the following conditions hold:

E1. For all $\alpha \in A_P \cup \{\tau\}$ and $p \xrightarrow{\alpha} P'$, we have $\forall p' {\in} P'.\, p \overset{\alpha}{\dashrightarrow} p'$, (syntactic consistency)
E2. States in $E_P \cup U_P$ have no outgoing transitions, (sink condition)
E3. $E_P \cap U_P = \emptyset$. (exclusive markings)

If $S_P^0 = \emptyset$, then P is called *inconsistent* and often denoted as $\bot$. An EMIA P is an *implementation* (i.e., an LTS) if $|S_P^0| = 1$, $|P'| = 1$ for all $p \xrightarrow{\alpha} P'$ and, for each $p \overset{\alpha}{\dashrightarrow} p'$, there is a transition $p \xrightarrow{\alpha} \{p'\}$.

In the following we often omit the index P when referring to components of an EMIA P, e.g., we write I for I_P. Similarly, we write, e.g., I_1 instead of I_{P_1} for EMIA P_1. In addition, we let i, o, a, ω

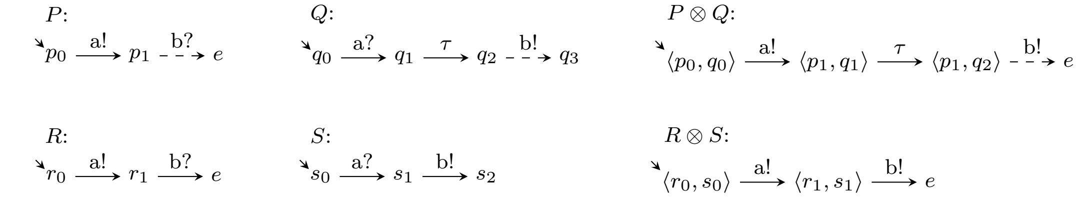

Fig. 3 Examples of EMIAs (cf. Def. 1), error-aware modal refinement (cf. Def. 3), parallel composition (cf. Def. 5) and compositionality (cf. Thm. 6), where $A_P = A_R = \{\text{b?}\}/\{\text{a!}\}$, $A_Q = A_S = \{\text{a?}\}/\{\text{b!}\}$, $A_{P\otimes Q} = A_{R\otimes S} = \emptyset/\{\text{a!}, \text{b!}\}$ and $R \sqsubseteq_{\text{e}} P$, $S \sqsubseteq_{\text{e}} Q$, $R \otimes S \sqsubseteq_{\text{e}} P \otimes Q$. States e indicate fatal error states.

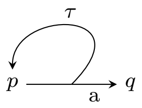

Fig. 4 Example of a DMTS that may lead to an infinite unfolding.

and α stand for representatives of the alphabets I, O, A, Ω and $A \cup \{\tau\}$, resp.; we write $A = I/O$ when highlighting inputs I and outputs O in an alphabet A. In the context of weak transitions that abstract from τs, we use the notation $\hat{\alpha}$, where $\hat{\alpha} := a$ if $\alpha = a \neq \tau$ and $\hat{\alpha} := \epsilon$ if $\alpha = \tau$. In figures, we often refer to an action a as a? if $a \in I$, and a! if $a \in O$. Must-transitions (may-transitions) are drawn using solid, possibly splitting arrows (dashed arrows); any depicted must-transition also implicitly represents the underlying may-transition(s) due to syntactic consistency. For notational convenience, we let $p \xrightarrow{a} p'$, $p \not\xrightarrow{a}$ and $p \overset{a}{\not\dashrightarrow}$ denote $p \xrightarrow{a} \{p'\}$, $\nexists P'.\, p \xrightarrow{a} P'$ and $\nexists p'.\, p \overset{a}{\dashrightarrow} p'$, resp.

Several examples of EMIAs are shown in Fig. 3. We only discuss EMIA P here: starting from the initial state p_0, implementations of P are required to enable action a! as an output. Afterwards, an optional input b? leading to a fatal error state e is specified, i.e., b? is permitted but not required in an implementation.

Definition 2 (Weak Transition Relations) Let P be an EMIA. We define *weak* must- and may-transition relations, $\Longrightarrow$ and $=\!=\!>$ resp., as the smallest relations satisfying the following conditions, where we use $P' \overset{\hat{\alpha}}{\Longrightarrow} P''$ as a shorthand for $\forall p{\in}P'.\, \exists P_p.\, p \overset{\hat{\alpha}}{\Longrightarrow} P_p$ and $P'' = \bigcup_{p\in P'} P_p$:

WT1. $p \overset{\epsilon}{\Longrightarrow} \{p\}$ for all $p \in P$,
WT2. $p \xrightarrow{\tau} P'$ and $P' \overset{\hat{\alpha}}{\Longrightarrow} P''$ implies $p \overset{\hat{\alpha}}{\Longrightarrow} P''$,
WT3. $p \xrightarrow{a} P'$ and $P' \overset{\epsilon}{\Longrightarrow} P''$ implies $p \overset{a}{\Longrightarrow} P''$,
WT4. $p \overset{\epsilon}{=\!=\!>} p$,
WT5. $p \overset{\epsilon}{=\!=\!>} p'' \overset{\tau}{\dashrightarrow} p'$ implies $p \overset{\epsilon}{=\!=\!>} p'$,
WT6. $p \overset{\epsilon}{=\!=\!>} p'' \overset{\alpha}{\dashrightarrow} p''' \overset{\epsilon}{=\!=\!>} p'$ implies $p \overset{\alpha}{=\!=\!>} p'$.

We write $\xrightarrow{a}\overset{\epsilon}{\Longrightarrow}$ for transitions that are built up according to WT3 and call them *trailing-weak* must-transitions. Similarly, $\overset{a}{\dashrightarrow}\overset{\epsilon}{=\!=\!>}$ stands for trailing-weak may-transitions.

This definition is adopted from MIA [7]. Examples of weak transitions may be found in EMIA Q in Fig. 3; we only highlight two examples here: by Rule WT3, there is a weak must-transition $q_0 \overset{\text{a?}}{\Longrightarrow} \{q_2\}$ and, by Rule WT6, a weak may-transition $q_1 \overset{\text{b!}}{=\!=\!>} q_3$.

Our *error-aware modal refinement preorder* $\sqsubseteq_{\text{e}}$ corresponds to standard modal refinement from MTS [25, 28] but reflects *and* preserves fatal error states. Intuitively, $P \sqsubseteq_{\text{e}} Q$ for an implementation P and a specification Q, enforces that P's may-transitions are permitted by Q while for any of Q's disjunctive must-transitions at least one of the branches is implemented by P.

In contrast to DMTS [28], we require that all branches of a disjunctive transition have the same label and call this restricted formalism dMTS. This is sufficient for our purposes and does away with potential technical complications of parallel composition in the presence of τ-transitions. The usual way of defining parallel composition on DMTS, e.g., as is done in [3], is by unfolding each disjunctive must-transition into its set of possible implementation variants, i.e., selections of transition branches. The parallel composition of two components is then obtained by forming all pairwise products of the components' implementation variants. The unfolding operation corresponds to a transformation of a conjunctive normal form into a disjunctive normal form and is, thus, only a change of representation. However, in order to define weak transitions in the unfolded representation, one has to unfold the τ-closure of each transition. This might result in an infinite unfolding, at least when not being careful. For example, consider the DMTS shown in Fig. 4. When unfolding state p, we get the possible implementations $\{\{(a, q)\}, \{(\tau, p)\}, \{(a, q), (\tau, p)\}\}$. However, this set is not necessarily τ-closed, i.e., we replace the τ-transitions by the unfolding of their target states. Due to the τ-loop, a τ-transition is re-introduced in each unfolding step, yielding an infinite unfolding process. To our knowledge, it is an open problem whether there is a solution for weak transitions in DMTS.

Definition 3 (Error-aware Modal Refinement) Let P and Q be EMIAs with equal alphabets, i.e., $I_P = I_Q$ and $O_P = O_Q$. A relation $\mathcal{R} \subseteq S_P \times S_Q$ is an *error-aware modal refinement* relation if, for all $\langle p, q \rangle \in \mathcal{R}$, $q \notin U_Q$ implies

R1. $p \in E_P$ iff $q \in E_Q$,

R2. $p \notin U_P$,

R3. $q \xrightarrow{i} Q'$ implies $\exists P'.\, p \xrightarrow{i}\overset{\epsilon}{\Rightarrow} P'$ and $\forall p' \in P'\ \exists q' \in Q'.\, \langle p', q' \rangle \in \mathcal{R}$,

R4. $q \xrightarrow{\omega} Q'$ implies $\exists P'.\, p \overset{\hat{\omega}}{\Longrightarrow} P'$ and $\forall p' \in P'\ \exists q' \in Q'.\, \langle p', q' \rangle \in \mathcal{R}$,

R5. $p \overset{i}{\dashrightarrow} p'$ implies $\exists q'.\, q \overset{i}{\dashrightarrow}\overset{\epsilon}{=\,=\!\!>} q'$ and $\langle p', q' \rangle \in \mathcal{R}$,

R6. $p \overset{\omega}{\dashrightarrow} p'$ implies $\exists q'.\, q \overset{\hat{\omega}}{=\,=\!\!>} q'$ and $\langle p', q' \rangle \in \mathcal{R}$.

We write $p \sqsubseteq_{\mathrm{e}} q$ if there is an error-aware modal refinement relation $\mathcal{R}$ with $\langle p, q \rangle \in \mathcal{R}$, and $P \sqsubseteq_{\mathrm{e}} Q$ if, for each $p \in S_P^0$, there is a $q \in S_Q^0$ with $p \sqsubseteq_{\mathrm{e}} q$. If $p \sqsubseteq_{\mathrm{e}} q$ and $q \sqsubseteq_{\mathrm{e}} p$, we employ the symbol $p \sqsupseteq\sqsubseteq_{\mathrm{e}} q$, and similar for EMIAs P, Q.

In a pure EMIA setting we may relax Rule R5 by permitting leading τ-transitions in Q. Due to the employed pruning operation this relaxation would break compositionality wrt. parallel composition when MIA or any other IA-based interface theory is involved [7]. Therefore, we do without this relaxation in order to make EMIA and MIA more comparable. In particular, we employ the same pruning operation for establishing the Galois insertion between these two theories.

In Fig. 3, EMIA R refines EMIA P by implementing the specified b?-may-transition. Because the target state of this transition is specified as a fatal error state e, R is also required to target a fatal error state. Analogously, EMIA S refines EMIA Q where, in addition S's a?-must-transition is matched by a weak transition in Q.

Lemma 4 ($\sqsubseteq_{\mathrm{e}}$ is a Preorder) *Error-aware modal refinement $\sqsubseteq_{\mathrm{e}}$ is reflexive and transitive.*

Proof (sketch) Reflexivity is easy because the identity relation is an isomorphism, which trivially satisfies all refinement conditions. The proof of transitivity closely follows the proof in [7]; therefore, we only sketch the proof idea: given EMIAs P, Q, R with refinement relations $\mathcal{R}_{PQ}$ and $\mathcal{R}_{QR}$, we have to show that $\mathcal{R} := \{\langle p, r \rangle \mid \exists q{\in}S_Q.\, \langle p, q \rangle \in \mathcal{R}_{PQ} \wedge \langle q, r \rangle \in \mathcal{R}_{QR}\}$ is an error-aware modal refinement relation. It is easy to see that conditions R1 and R2 hold transitively. If a relation satisfies conditions R3 through R6, then it also satisfies the same conditions with weak transitions in the premises, e.g., a rule R3' of the form $q \xrightarrow{i}\overset{\epsilon}{\Rightarrow} Q'$ implies $\exists P'.\, p \xrightarrow{i}\overset{\epsilon}{\Rightarrow} P'$ and $\forall p'{\in}P'\ \exists q'{\in}Q'.\, \langle p', q' \rangle \in \mathcal{R}$. Hence, these conditions also follow transitively. □

IA's parallel composition operator synchronises input and output transitions to τ-transitions. In contrast, we define a multicast parallel composition, where an output can synchronise with multiple input transitions as in MI [32] and MIA [7].

Definition 5 (Parallel Composition) Let P and Q be EMIAs. We call P and Q *composable* if $O_P \cap O_Q = \emptyset$. If P and Q are composable, the *multicast parallel composition* $P \otimes Q$ is defined by $S_{P\otimes Q} := S_P \times S_Q$, $I_{P\otimes Q} := (I_P \cup I_Q) \setminus O_{P\otimes Q}$, $O_{P\otimes Q} := O_P \cup O_Q$, $S^0_{P\otimes Q} := S^0_P \times S^0_Q$, $E_{P\otimes Q} := (E_P \times S_Q) \cup (S_P \times E_Q)$, $U_{P\otimes Q} := ((S_P \setminus E_P) \times U_Q) \cup (U_P \times (S_Q \setminus E_Q))$, and the transition relations are given by the following rules:

P1. $\langle p, q\rangle \xrightarrow{\alpha} P' \times \{q\}$ if $p \xrightarrow{\alpha} P'$ and $\alpha \notin A_Q$,
P2. $\langle p, q\rangle \xrightarrow{\alpha} \{p\} \times Q'$ if $\alpha \notin A_P$ and $q \xrightarrow{\alpha} Q'$,
P3. $\langle p, q\rangle \xrightarrow{a} P' \times Q'$ if $p \xrightarrow{a} P'$ and $q \xrightarrow{a} Q'$ for some $a \in A_P \cap A_Q$.
P4. $\langle p, q\rangle \overset{\alpha}{\dashrightarrow} \langle p', q\rangle$ if $p \overset{\alpha}{\dashrightarrow} p'$ and $\alpha \notin A_Q$,
P5. $\langle p, q\rangle \overset{\alpha}{\dashrightarrow} \langle p, q'\rangle$ if $\alpha \notin A_P$ and $q \overset{\alpha}{\dashrightarrow} q'$,
P6. $\langle p, q\rangle \overset{a}{\dashrightarrow} \langle p', q'\rangle$ if $p \overset{a}{\dashrightarrow} p'$ and $q \overset{a}{\dashrightarrow} q'$ for some $a \in A_P \cap A_Q$.

We also write $p \otimes q$ for $\langle p, q\rangle$.

Hence, an error in one component implies an error in the overall system, whereas universal behaviour in one component extends to the overall system only in absence of errors.

Several aspects of parallel composition are illustrated in EMIA $P \otimes Q$ of Fig. 3. Firstly, P and Q synchronise on their common actions a and b. Secondly, the τ-transition specified in Q is interleaved in $P \otimes Q$. Third, the composition of the fatal error state e with the regular state q_2 is again a fatal error state.

IA-based interface theories usually define a communication mismatch for p at q as a situation where an action $a \in O_P \cap I_Q$ is permitted at p and not required at q. In EMIA, such an optional input transition, which may be refined to required or forbidden behaviour, is expressed as a disjunctive must-transition containing a fatal error state in its set of target states. For example, optional $a?$-transitions from q to states q_1 and q_2 are modelled as $q \xrightarrow{a?} \{q_1, q_2, q_3\}$ for some fatal error state $q_3 \in E_Q$.

It is easy to see that parallel composition is associative and commutative. Further, $\sqsubseteq_{\mathrm{e}}$ is a precongruence wrt. $\otimes$:

Theorem 6 (Compositionality) *If P_1, P_2 and Q are EMIAs such that $P_1 \sqsubseteq_{\mathrm{e}} P_2$ and P_2, Q are composable, then P_1 and Q are composable and $P_1 \otimes Q \sqsubseteq_{\mathrm{e}} P_2 \otimes Q$.*

Proof We write I_P, O_P and A_P for the equal alphabets of P_1 and P_2. Composability is trivial. We show that $\mathcal{R} := \{\langle p_1 \otimes q, p_2 \otimes q\rangle \mid p_1 \sqsubseteq_{\mathrm{e}} p_2\}$ is an error aware modal refinement relation. For $\langle p_1 \otimes q, p_2 \otimes q\rangle \in \mathcal{R}$ with $p_2 \otimes q \notin U_{P\otimes Q}$, we consider the following cases:

R1 $p_1 \otimes q \notin E_{P_1 \otimes Q}$ iff (by Def. 5) $p_1 \notin E_{P_1} \wedge q \notin E_Q$ iff (by $p_1 \sqsubseteq_{\mathrm{e}} p_2$ and R1) $p_2 \notin E_{P_2} \wedge q \notin E_Q$ iff (by Def. 5) $p_2 \otimes q \notin E_{P_2 \otimes Q}$.

R2 We consider two cases:

1. $\langle p_2, q\rangle \in E_{P_2 \otimes Q}$: As shown for Case R1, we have $\langle p_1, q\rangle \in E_{P_1 \otimes Q}$ and, by Def. 5 and E3, $\langle p_1, q\rangle \notin U_{P_1 \otimes Q}$.
2. $\langle p_2, q\rangle \notin E_{P_2 \otimes Q}$: By $\langle p_2, q\rangle \notin U_{P_2 \otimes Q}$, we have $p_2 \notin U_{P_2}$ and $q \notin U_Q$. Then, $p_1 \sqsubseteq_{\mathrm{e}} p_2$ implies $p_1 \notin U_{P_1}$, hence, $\langle p_1, q\rangle \notin U_{P_1 \otimes Q}$.

R3 Let $p_2 \otimes q \xrightarrow{i} R$ due to one of P1, P2 or P3:

P1 $R = P'_2 \times \{q\}$ for some transition $p_2 \xrightarrow{i} P'_2$. By $p_1 \sqsubseteq_{\mathrm{e}} p_2$, there is a $p_1 \overset{i}{\rightarrow}\overset{\epsilon}{\Rightarrow} P'_1$ such that, for all $p'_1 \in P'_1$, there is a $p'_2 \in P'_2$ with $p'_1 \sqsubseteq_{\mathrm{e}} p'_2$. Thus, we have $\langle p'_1 \otimes q, p'_2 \otimes q\rangle \in \mathcal{R}$, and P1 implies $p_1 \otimes q \overset{i}{\rightarrow}\overset{\epsilon}{\Rightarrow} P'_1 \times \{q\}$.

- **P2** $R = \{p_2\} \times Q'$ for some $q \xrightarrow{i} Q'$. By P2 we have $p_1 \otimes q \xrightarrow{i} \{p_1\} \times Q'$, and $p_1 \sqsubseteq_{\mathrm{e}} p_2$ implies $\langle p_1 \otimes q', p_2 \otimes q' \rangle \in \mathcal{R}$ for all $q' \in Q'$.
- **P3** $R = P_2' \times Q'$ due to $p_2 \xrightarrow{i} P_2'$ and $q \xrightarrow{i} Q'$. The argument is analogous to that of case P1, when replacing the application of P1 by P3 in the last step.

R4 Analogous to R3.

R5 Let $p_1 \otimes q \overset{i}{\dashrightarrow} p_1' \otimes q'$ due to one of the rules P4, P5 or P6:

- **P4** $q' = q$ for a transition $p_1 \overset{i}{\dashrightarrow} p_1'$. By $p_1 \sqsubseteq_{\mathrm{e}} p_2$, there is a $p_2 \overset{i}{\dashrightarrow}\overset{\epsilon}{\Rightarrow} p_2'$ such that $p_1' \sqsubseteq_{\mathrm{e}} p_2'$. Thus, we have $\langle p_1' \otimes q, p_2' \otimes q \rangle \in \mathcal{R}$, and P4 implies $p_2 \otimes q \overset{i}{\dashrightarrow}\overset{\epsilon}{\Rightarrow} p_2' \otimes q$.
- **P5** $p_1' = p_1$ for some $q \overset{i}{\dashrightarrow} q'$. By P5, we have $p_2 \otimes q \overset{i}{\dashrightarrow} p_2 \otimes q'$, and $p_1 \sqsubseteq_{\mathrm{e}} p_2$ implies $\langle p_1 \otimes q', p_2 \otimes q' \rangle \in \mathcal{R}$.
- **P6** $R = P_1' \times Q'$ due to $p_1 \overset{i}{\dashrightarrow} P_1'$ and $q \overset{i}{\dashrightarrow} Q'$. The argument is similar to that of case P4, where the application of P4 is replaced by P6 in the last step.

R6 Analogous to R5. □

Fig. 3 also illustrates compositionality. Because $R \sqsubseteq_{\mathrm{e}} P$ and $S \sqsubseteq_{\mathrm{e}} Q$, we also have $R \otimes S \sqsubseteq_{\mathrm{e}} P \otimes Q$.

3 Relation to other Interface Theories

Because IA-based interface theories prune errors, it is important to investigate the relation between such error-pruning interface theories and our error-preserving EMIA theory. We do this for MIA [7] because it is the most general IA-based interface theory to date in that it is nondeterministic rather than deterministic and optimistic rather than pessimistic, thus subsuming MI [32] and MIO [2] (wrt. strong compatibility), resp. As an aside, the interface theory MIO [2] employs standard non-disjunctive modal transitions and departs from IA by supporting pessimistic instead of optimistic compatibility. Several notions of pessimistic compatibility have been defined for MIO, e.g., strong compatibility, where an output must be received immediately, and weak compatibility, where receiving an output may be delayed through unbounded buffered communication channels.

In this section we establish a Galois insertion between MIA and EMIA, i.e., a Galois connection $\langle \gamma, \alpha \rangle$ for which $\alpha \circ \gamma = \mathrm{id}_{\mathsf{MIA}}$ [13] (up to $\sqsupseteq\sqsubseteq_{\mathrm{e}}$). Recall that states from which a communication mismatch is reachable via output or τ-transitions are called illegal. Intuitively, α abstracts from EMIAs by considering all illegal states to be equivalent, and γ concretises MIAs as EMIAs without any loss of information.

3.1 Error-abstracted Modal Interface Automata

We slightly generalise the MIA theory given in [7,19] in order to obtain a more uniform presentation of MIA and EMIA, which also simplifies some of the proofs when compared to the conference version of this paper [19].

Definition 7 (Error-abstracted Modal Interface Automata [7]) An EMIA P is called an *error-abstracted Modal Interface Automaton* (MIA) if

M1. For all $p \in S_P \setminus (E_P \cup U_P)$, $i \in I_P$, there is a $p' \in S_P$ with $p \overset{i}{\dashrightarrow} p'$, (input enabledness)
M2. If $p \overset{i}{\dashrightarrow} p'$, then there is a P' with $p' \in P'$ and $p \xrightarrow{i} P'$, (input must)
M3. States $E_P \cup U_P$ only appear as target states of input transitions. (error abstraction)

We write EMIA' for the collection of EMIAs satisfying M1 and M2, and MIA for the collection of MIAs. With $\sqsubseteq_{\mathrm{m}}$ we denote the restriction of $\sqsubseteq_{\mathrm{e}}$ to MIAs.

Note that input enabledness and the input must condition do not restrict our definition of MIA because a transition may target states in E_P and U_P. The purpose of these conditions is to distinguish the error-aware parallel composition of MIA from the unanimous parallel composition of MTS in our unified model because the latter mode of composition is not supported by MIA.

It is easy to see that EMIA' is closed under $\otimes$:

Lemma 8 *If $P, Q \in \mathsf{EMIA}'$, then $P \otimes Q \in \mathsf{EMIA}'$.*

Proof As a direct consequence of Def. 5, if $p \in S_P$ and $q \in S_Q$ are input enabled and satisfy the input must condition, then $p \otimes q$ satisfies both conditions, too. □

In order to make parallel composition on MIA respect error abstraction, we need to consider the reachability of illegal states:

Definition 9 (Backward Closure) Let P be an EMIA, $B \subseteq A_P \cup \{\tau\}$ and $S \subseteq S_P$. The *B-backward closure of S in P* is the smallest set $\mathrm{bcl}_P^B(S) \subseteq S_P$ s.t. $S \subseteq \mathrm{bcl}_P^B(S)$ and, for all $\alpha \in B$ and $p' \in \mathrm{bcl}_P^B(S)$, if $p \overset{\alpha}{\dashrightarrow} p'$, then $p \in \mathrm{bcl}_P^B(S)$.

Definition 10 (Illegal States) The set of *illegal states* of an EMIA P is defined as $\mathrm{ill}_P := \mathrm{bcl}_P^{\Omega}(E_P \cup U_P) \setminus (E_P \cup U_P)$.

The set $\mathrm{ill}_{P \otimes Q}$ of an EMIA composition $P \otimes Q$ corresponds to the set of illegal states in IA, MI and MIA. In contrast to these theories, EMIA requires one to match transitions of such states during refinement. The resulting refinement relation is comparable to other refinement preorders for error-free interfaces, but is more detailed for erroneous ones. Indeed, MIA can be seen as an abstraction of EMIA, where all states in $\mathrm{ill}_{P \otimes Q}$ are deemed equivalent (cf. Thm. 18). For example, the interfaces $P\colon p_0 \xrightarrow{i?} p_1 \xrightarrow{o_1!} \mathsf{e}_P$ and $Q\colon q_0 \xrightarrow{i?} q_1 \xrightarrow{o_2!} \mathsf{e}_Q$ are equivalent in MIA because after receiving input i, both may reach an error autonomously, whereas EMIA distinguishes P and Q according to the different behaviours ($o_1!$ vs. $o_2!$) that lead to an error.

Definition 11 (Error Abstraction) The *error abstraction* of an EMIA $P \in \mathsf{EMIA}'$ is the EMIA $\alpha(P) := (S_{\alpha(P)}, I_P, O_P, \longrightarrow_{\alpha(P)}, \dashrightarrow_{\alpha(P)}, S^0_{\alpha(P)}, E_P, U_{\alpha(P)})$ with $S_{\alpha(P)} := (S_P \setminus \mathrm{ill}_P) \uplus \{\top_{\alpha(P)}\}$ and $U_{\alpha(P)} := U_P \uplus \{\top_{\alpha(P)}\}$ (where $\uplus$ denotes the disjoint union). If $S^0_P \cap \mathrm{ill}_P \neq \emptyset$, then $S^0_{\alpha(P)} := (S^0_P \cap S_{\alpha(P)}) \cup \{\top_{\alpha(P)}\}$, else $S^0_{\alpha(P)} := S^0_P \cap S_{\alpha(P)}$. The transitions of $\alpha(P)$ are obtained from P by replacing all $i?$-transitions leading from a state p to states in ill_P by $p \xrightarrow{i?} \top_{\alpha(P)}$ and the underlying may-transition.

Obviously, $P \sqsubseteq_{\mathrm{e}} \alpha(P)$ and $\alpha(P) \in \mathsf{MIA}$ for all $P \in \mathsf{EMIA}'$. Further, α is monotonic:

Lemma 12 (Monotonicity of α) *The map α defined in Def. 11 is monotonic wrt. $\sqsubseteq_{\mathrm{e}}$.*

Proof Let $\mathcal{R}$ be an error-aware modal refinement relation between EMIAs P and Q. We show that the relation $\mathcal{R}_\alpha := (\mathcal{R} \cap (S_{\alpha P} \times S_{\alpha Q})) \cup (S_{\alpha P} \times U_{\alpha Q})$ is an error-aware modal refinement relation between αP and αQ. Let $\langle p, q\rangle \in \mathcal{R}_\alpha$. In case $q \in U_{\alpha Q}$, the definition of refinement is trivially satisfied, so we can assume $q \notin U_{\alpha Q}$. Hence, by definition of $\mathcal{R}_\alpha$, we may assume $\langle p, q\rangle \in \mathcal{R}$ and distinguish the following cases:

R1, R2 Because $\mathcal{R}$ is an error-aware modal refinement relation, $\langle p, q\rangle \in \mathcal{R}$ implies that R1 and R2 are satisfied trivially.

R3 Let $q \xrightarrow{i}_{\alpha Q} Q'_\alpha$. We consider two cases:

1. The transition is due to a transition $q \overset{i}{\dashrightarrow}_Q q'$ with $q' \in \mathrm{ill}_Q$, i.e., $Q'_\alpha = \{\top_{\alpha Q}\}$: Any P'_α is a possible implementation of Q'_α.

2. The transition is due to a transition $q \xrightarrow{i}_Q Q'$: Because all transitions into ill_Q are replaced in Def. 11, we know that $Q'_\alpha = Q'$ and that none of these target states is in ill_Q or $E_Q \cup U_Q$. By $\langle p, q\rangle \in \mathcal{R}$, there is a $p \overset{i}{\rightarrow}\overset{\epsilon}{\Rightarrow}_P P'$ such that P' matches Q'. With the same argument as before, we may conclude that $P'_\alpha := P'$ matches Q'_α.

R4 Similar to R3(2), where $\overset{i}{\rightarrow}\overset{\epsilon}{\Rightarrow}$ is replaced by $\overset{\omega}{\Longrightarrow}$.

R5 Let $p \overset{i}{\dashrightarrow}_{\alpha P} p'$. If $p' \neq \top_{\alpha P}$, then $p \overset{i}{\dashrightarrow}_P p'$ and, due to $\langle p, q\rangle \in \mathcal{R}$, there is a $q \overset{i}{\dashrightarrow}\overset{\epsilon}{\Rightarrow}_Q q'$ such that $\langle p', q'\rangle \in \mathcal{R}$. There are two cases:

1. $\exists q'' \in \mathrm{ill}_Q.\, q \overset{i}{\dashrightarrow}_Q q''$: By definition of α we have $\mathrm{ill}_Q \cap S_{\alpha Q} = \emptyset$; thus, $q'' \notin S_{\alpha Q}$. Hence, it follows from $q \in S_{\alpha Q}$ that $q \overset{i}{\dashrightarrow}_Q \top_{\alpha Q}$ by definition of α, and $\langle p', \top_{\alpha Q}\rangle \in \mathcal{R}_\alpha$ is obvious.
2. $\forall q'' \in \mathrm{ill}_Q.\, q \overset{i}{\not\dashrightarrow}\, {}_Q q''$: The definition of α implies $q' \in S_{\alpha Q}$ and $q \overset{i}{\dashrightarrow}\overset{\epsilon}{\Rightarrow}_{\alpha Q} q'$. Therefore, $\langle p', q'\rangle \in \mathcal{R}_\alpha$.

If $p' = \top_{\alpha P}$, then there is a $p'' \in \mathrm{ill}_P$ with $p \overset{i}{\dashrightarrow}_P p''$. By $\langle p, q\rangle \in \mathcal{R}$, there exists a $q'' \in \mathrm{ill}_Q$ such that $q \overset{i}{\dashrightarrow}\overset{\epsilon}{\Rightarrow}_Q q''$ and $\langle p'', q''\rangle \in \mathcal{R}$. Thus, $q \overset{i}{\dashrightarrow}_{\alpha Q} \top_{\alpha Q}$, and $\langle \top_{\alpha P}, \top_{\alpha Q}\rangle \in \mathcal{R}_\alpha$ is trivial.

R6 Analogous to R5 with $\overset{i}{\dashrightarrow}\overset{\epsilon}{\Rightarrow}$ and $\overset{i}{\dashrightarrow}$ replaced by $\overset{\omega}{\Rightarrow}$ and $\overset{\omega}{\dashrightarrow}$, resp., and where we always have $p' \neq \top_{\alpha P}$ and only Case 2 applies (otherwise, we would have $q \in \mathrm{ill}_Q$). □

Now we can define MIA parallel composition and show that it is compositional.

Definition 13 (MIA Parallel Composition [7]) For composable MIAs P and Q, the *parallel product* is given by $P \otimes Q$ as defined in Def. 5. The *MIA-parallel composition* is defined as the MIA $P \parallel Q := \alpha(P \otimes Q)$.

Due to Lem. 8 and $\alpha(P)$ being a MIA, $P \parallel Q$ is also a MIA.

Lemma 14 (Compositionality of $\parallel$) *If P_1, P_2 and Q are MIAs such that $P_1 \sqsubseteq_{\mathrm{e}} P_2$ and P_2, Q are composable, then P_1, Q are composable and $P_1 \otimes Q \sqsubseteq_{\mathrm{e}} P_2 \otimes Q$.*

Proof Composability is obvious. By Thm. 6 and Lem. 12, we have $P_1 \parallel Q = \alpha(P_1 \otimes Q) \sqsubseteq_{\mathrm{e}} \alpha(P_2 \otimes Q) = P_2 \parallel Q$. □

Lemma 15 (α is Homomorphic wrt. Parallel Composition) *The mapping α defined in Def. 11 is homomorphic wrt. parallel composition, i.e., $\alpha(P \otimes Q) \sqsupseteq\sqsubseteq_{\mathrm{m}} \alpha(P) \parallel \alpha(Q)$.*

Proof First, observe that $\alpha(P \otimes Q)$ and $\alpha(P) \parallel \alpha(Q)$ have the same state set $S := S_{\alpha(P \otimes Q)} = S_{\alpha(P) \parallel \alpha(Q)}$ because the same pruning operation is used in α and in MIA's parallel composition operator (see also [7,8]).

"$\sqsubseteq_{\mathrm{m}}$": We show that the relation $\mathcal{R} := \mathrm{id}_S \cup (S_{\alpha(P \otimes Q)} \times \{\top_{\alpha(P) \parallel \alpha(Q)}\})$ is a MIA-refinement relation. Let $\langle s, t\rangle \in \mathcal{R}$. If $t = \top_{\alpha(P) \parallel \alpha(Q)}$, there is nothing to show. Thus, we assume $s = t$ and distinguish the following cases:

R1, R2 From $s = t$, one directly concludes R1 and R2.

R3 Let $s = \langle p, q\rangle \in S_{\alpha(P \parallel Q)}$. A transition $\langle p, q\rangle \xrightarrow{i}_{\alpha(P) \parallel \alpha(Q)} S'$ is due to one of the rules P1, P2 or P3:

P1 $S' = P' \times \{q\}$ for some $p \xrightarrow{i}_{\alpha(P)} P'$ and $i \notin A_{\alpha(Q)}$: because this transition has neither been pruned nor replaced by a may-transition to $\top_{\alpha(P) \parallel \alpha(Q)}$, the same transition also exists in $\alpha(P \parallel Q)$.

P2 $S' = \{p\} \times Q'$ for some $q \xrightarrow{i}_{\alpha(Q)} Q'$ and $i \notin A_{\alpha(P)}$: Analogous to P1.

P3 $S' = P' \times Q'$ for some $p \xrightarrow{i}_{\alpha(P)} P'$ and $q \xrightarrow{i}_{\alpha(Q)} Q'$: Similar to P1 and P2.

R4 Analogous to R3.

R5 Let $\langle p,q\rangle \overset{i}{\dashrightarrow}_{\alpha(P\|Q)} s''$. In case $s'' = \top_{\alpha(P)\|\alpha(Q)}$, then this transition is due to a replacement of a transition $\langle p,q\rangle \overset{i}{\dashrightarrow}_{\alpha(P\|Q)} s'$ by $\top_{\alpha(P)\|\alpha(Q)}$. In case $s'' \neq \top_{\alpha(P)\|\alpha(Q)}$, by choosing $s' = s''$, we also have a transition $\langle p,q\rangle \overset{i}{\dashrightarrow}_{\alpha(P\|Q)} s'$. In both cases, this transition is due to one of the rules P4 through P6, which all result in a similar line of argument. In case of P4 we have $s' = \langle p',q\rangle$, $p \overset{i}{\dashrightarrow}_{\alpha(P)} p'$ and $a \notin A_Q$. By the definition of α, there must be a p'' such that $p \overset{i}{\dashrightarrow}_{P} p''$. By P4, $\langle p,q\rangle \overset{i}{\dashrightarrow}_{P\|Q} \langle p'',q\rangle$ and, thus, also $\langle p,q\rangle \overset{i}{\dashrightarrow}_{\alpha(P\|Q)} \langle p'',q\rangle$.

R6 Analogous to R5.

Direction "$\sqsupseteq_{\mathrm{m}}$" can be shown dually. □

3.2 The Galois Insertion

The Galois insertion between MIA and EMIA consists of a concretisation $\gamma\colon \mathsf{MIA} \to \mathsf{EMIA}'$ and an abstraction $\alpha\colon \mathsf{EMIA}' \to \mathsf{MIA}$ such that $\langle\gamma,\alpha\rangle$ is a Galois connection and $(\alpha\circ\gamma)(Q) \sqsupseteq\sqsubseteq_{\mathrm{m}} Q$. As presented in Sec. 3.1, the main idea behind α is to consider the states ill_P as equivalent. Each equivalence class of EMIAs resulting from this abstraction has a greatest element wrt. the refinement preorder, and α assigns each EMIA in such a class the greatest element of the class, which turns out to be a MIA. Vice versa, γ is the identical embedding of MIA into EMIA, such that a MIA represents its equivalence class.

Definition 16 (Concretisation Function from MIA to EMIA') The concretisation function $\gamma : \mathsf{MIA} \to \mathsf{EMIA}'$ is defined as $\gamma(P) := P$.

Obviously, γ is monotonic:

Lemma 17 (Monotonicity of γ) *The map γ defined in Def. 16 is monotonic wrt. $\sqsubseteq_{\mathrm{e}}$ and $\sqsubseteq_{\mathrm{m}}$.*

The monotonicity of α and γ is key to the proof that α and γ form a Galois insertion:

Theorem 18 (Galois Insertion) *The maps $\alpha\colon \mathsf{EMIA}' \to \mathsf{MIA}$ and $\gamma\colon \mathsf{MIA} \to \mathsf{EMIA}'$ defined in Defs. 11 and 16 form a Galois insertion between MIA and EMIA' up to $\sqsupseteq\sqsubseteq_{\mathrm{m}}$, i.e., $P \sqsubseteq_{\mathrm{e}} \gamma(Q)$ iff $\alpha(P) \sqsubseteq_{\mathrm{m}} Q$ and $(\alpha\circ\gamma)(Q) \sqsupseteq\sqsubseteq_{\mathrm{m}} Q$.*

Proof First, observe that $\alpha\circ\gamma = \mathrm{id}_{\mathsf{MIA}}$ (up to $\sqsupseteq\sqsubseteq_{\mathrm{m}}$). Second, the extensivity of α implies that $\gamma\circ\alpha$ is extensive, i.e., $P \sqsubseteq_{\mathrm{e}} (\gamma\circ\alpha)(P)$. Third, we show that α and γ form a Galois connection, i.e., $P \sqsubseteq_{\mathrm{e}} \gamma(Q)$ iff $\alpha(P) \sqsubseteq_{\mathrm{m}} Q$. Direction "$\Rightarrow$" holds due to $\alpha\circ\gamma = \mathrm{id}_{\mathsf{MIA}}$ and the monotonicity of α: $P \sqsubseteq_{\mathrm{e}} \gamma(Q) \Rightarrow \alpha(P) \sqsubseteq_{\mathrm{m}} (\alpha\circ\gamma)(Q) \sqsupseteq\sqsubseteq_{\mathrm{m}} Q$. Direction "$\Leftarrow$" follows from the monotonicity of γ, the extensivity of $\gamma\circ\alpha$ and the transitivity of $\sqsubseteq_{\mathrm{e}}$ by the following chain of implications: $\alpha(P) \sqsubseteq_{\mathrm{m}} Q \;\Rightarrow\; (\gamma\circ\alpha)(P) \sqsubseteq_{\mathrm{e}} \gamma(Q) \;\Rightarrow\; P \sqsubseteq_{\mathrm{e}} \gamma(Q)$. □

The extensivity of α makes γ non-homomorphic wrt. parallel composition; however, γ satisfies the inequality $\gamma(P \,\|\, Q) \sqsupseteq_{\mathrm{e}} \gamma(P) \otimes \gamma(Q)$ for MIAs P, Q. Although this follows directly from the definition of γ, we can prove a more general fact:

Lemma 19 *Let K and L be preorders, $\cdot$ a binary operation on K resp. L. If $\langle\gamma,\alpha\rangle$ is a Galois insertion between K and L such that α is homomorphic wrt. $\cdot$, then $\gamma(k\cdot k') \sqsupseteq \gamma(k)\cdot\gamma(k')$.*

Proof $\gamma(k)\cdot\gamma(k') \sqsubseteq (\gamma\circ\alpha)(\gamma(k)\cdot\gamma(k')) \equiv \gamma((\alpha\circ\gamma)(k)\cdot(\alpha\circ\gamma)(k')) \equiv \gamma(k\cdot k')$. □

Corollary 20 *Let P and Q be MIAs. Then, $\gamma(P \,\|\, Q) \sqsupseteq_{\mathrm{e}} \gamma(P) \otimes \gamma(Q)$.*

Proof By Thm. 18 and Lemma 15, we can apply Lem. 19. □

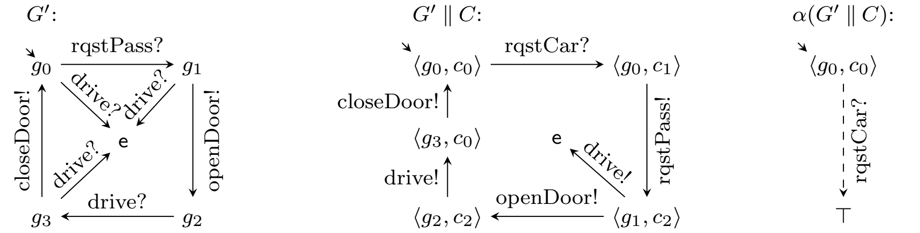
G':
rqstPass?
g_0
g_1
drive?
drive?
closeDoor!
openDoor!
e
drive?
drive?
g_3
g_2
$G' \parallel C$:
rqstCar?
$\langle g_0, c_0 \rangle$
$\langle g_0, c_1 \rangle$
closeDoor!
rqstPass!
$\langle g_3, c_0 \rangle$
e
drive!
drive!
openDoor!
$\langle g_2, c_2 \rangle$
$\langle g_1, c_2 \rangle$
$\alpha(G' \parallel C)$:
$\langle g_0, c_0 \rangle$
rqstCar?
$\top$

Fig. 5 Driving assistant system in EMIA and its Galois abstraction, where $A_{G'} := \{\text{drive?}, \text{rqstPass?}\}/\{\text{closeDoor!}, \text{openDoor!}\}$ and $A_{G'\parallel C} = A_{\alpha(G'\parallel C)} := \{\text{rqstCar?}\}/\{\text{closeDoor!}, \text{drive!}, \text{openDoor!}, \text{rqstPass!}\}$.

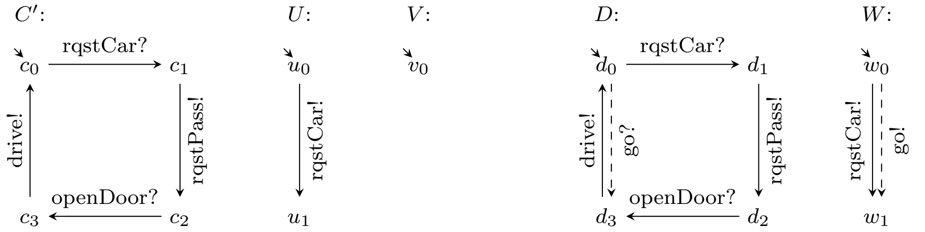

Fig. 6 Corrected car C', user interfaces U, V, and product families D and W, where $A_{C'} := \{\text{openDoor?}, \text{rqstCar?}\}/\{\text{drive!}, \text{rqstPass!}\}$, $A_U = A_V := \emptyset/\{\text{rqstCar!}\}$, $A_D := \{\text{go?}, \text{openDoor?}, \text{rqstCar?}\}/\{\text{drive!}, \text{rqstPass!}\}$ and $A_W := \emptyset/\{\text{go!}, \text{rqstCar!}\}$.

3.3 Discussion

In this section we illustrate how the fatal error states employed in EMIA solve Issues A–D presented in Sec. 1. In particular, we establish that EMIA treats unwanted behaviour more intuitively (Issue A), that EMIA, in contrast to MIA, is an assembly theory (Issue B), that EMIA provides better support for specifying product families (Issue C), and that EMIA unifies the composition concepts of MTS and interface theories (Issue D). We do this mostly along the example of Sec. 1 and also use this example to demonstrate the Galois abstraction from EMIA to MIA.

3.3.1 Issue A: Unwanted Behaviour

In EMIA, the garage's constraint that a car shall not drive in or out in state g_1 would be specified by a drive?-transition to a fatal error state e, which represents an unresolvable error as is illustrated in specification G' in Fig. 5. In the resulting parallel composition $G' \parallel C$, also shown in Fig. 5, driving in or out too early in state $\langle g_1, c_2 \rangle$, when the door is still closed, leads to the fatal error state e, where the car crashes into the door. This information is not removed and cannot be redefined to not being an accident by refining $G' \parallel C$. Keeping this information is essential for pinning down the location and the cause of the error within the specification. Because G' forbids action drive? between rqstPass? and openDoor! but allows drive? after openDoor!, we can infer that specification C must be aware of action openDoor! in order to be compatible with G'. This way, a software design tool based on EMIA can propose possible specification changes to the designer. For example, the tool may propose to add action openDoor? to the car's alphabet and to insert an openDoor?-transition between rqstPass! and drive!, so as to avoid the fatal error state e that is reachable from $\langle g_1, c_2 \rangle$. The resulting specification is shown as C' in Fig 6.

As an aside, Fig. 5 (right) illustrates the abstraction function α of the Galois insertion between MIA and EMIA. We have $\text{ill}_{G'\parallel C} = \{\langle g_1, c_2 \rangle, \langle g_0, c_1 \rangle\}$ (cf. Sec. 3). The rqstCar?-must-transition at $\langle g_0, c_0\rangle$ leading to $\mathrm{ill}_{G'\|C}$ is replaced by a rqstCar?-transition to $\top_{\alpha(G'\|C)}$. Due to α being a homomorphism wrt. $\|$, this result corresponds exactly to the MIA shown in Fig. 2 (right).

3.3.2 Issue B: Multi-component Assemblies

When adding the specification of a simple user interface, shown as U in Fig. 6, as a third component to the specifications G and C of Fig. 1, the three components G, C and U are pairwise optimistically compatible. However, the composed system $G \| C \| U$ is incompatible, because the mismatch for action drive! is reachable from the initial state $\langle g_0, c_0, u_0\rangle$. A different but related problem arises in pessimistic theories: the user interface specification V in Fig. 6 promises to never request a car. The components G and C are pessimistically incompatible and $(G \| C) \| V$ is undefined. However, $G \| (C \| V)$ is a perfectly valid composition. In other words, pairwise compatibility is neither necessary nor sufficient for compatibility of multiple components, i.e., IA, MI, MIO and MIA are not by themselves assembly theories.

To lift their interface theory MIO to an assembly theory, Hennicker and Knapp propose an enrichment EMIO of MIO by error states similar to our fatal errors [22]. However, they do not develop EMIO into a full interface theory: EMIOs are only employed to describe the result of a multi-component parallel composition and to check the communication safety of such an assembly, i.e., the absence of communication mismatches. In addition, refinement is lifted to assemblies by providing an error-preserving refinement relation for EMIOs, which is similar to error-aware modal refinement. However, no further operations like parallel composition or conjunction are defined for assemblies. Instead, EMIO forms a second layer on top of MIO, and an EMIO is re-interpreted as MIO via an encapsulation function that removes all error-information. In contrast to this loose integration, EMIA provides a uniform and tight integration of interfaces and assemblies by directly including its canonical assembly theory in the sense of [22]. In particular, EMIA does not need two separate refinement relations for interfaces and assemblies.

Translating the above examples of assemblies with U and V into EMIA, the composition $G' \| C \| U$ resembles $G' \| C$ (Fig. 5), except that action rqstCar is an output instead of an input. Further, $(G' \| C) \| V$ and $G' \| (C \| V)$ are equivalent in EMIA. In both examples, compatibility is checked via reachability of fatal error states in the composed system. However, it is up to the system designer to decide which error behaviour yields an incompatibility, i.e., compatibility is not necessarily a global concept as is the case for optimistic and pessimistic compatibility.

In order to establish the above results, we recap the definition of *assembly theory* by Hennicker and Knapp [22], with the following generalisation: in Hennicker and Knapp's definition of an interface theory, an interface cannot contain errors by itself and, thus, a single interface is always communication safe. EMIA additionally allows one to specify erroneous interfaces, which should not be considered communication safe. Therefore, we introduce a *communication safety predicate* on interfaces and generalise Conds. A1 and A3 below accordingly.

Definition 21 (Assembly Theory [22]) Let $\mathfrak{I} := (\mathcal{I}, \mathrm{cs}, \|, \sqsubseteq)$ be an interface theory, where $\mathcal{I}$ is a collection of interfaces, $\mathrm{cs} \subseteq \mathcal{I}$ is a communication safety predicate, $\|$ is a (binary) parallel composition operator, and $\sqsubseteq$ is the refinement preorder. A tuple $\mathfrak{A} := (\mathcal{A}, \mathrm{cs}, \varphi, \preceq)$ consisting of a collection of *assemblies* $\mathcal{A} := \{\langle I_k\rangle_{k\in K} \mid 0 < |K| < \infty \text{ and } I_k, I_l \in \mathcal{I} \text{ composable for } k \neq l\}$, a *communication safety predicate* $\mathrm{cs} \subseteq \mathcal{A}$, a partial *encapsulation operation* $\varphi : \mathfrak{A} \rightharpoonup \mathfrak{I}$ and an *assembly refinement relation* $\preceq \subseteq \mathcal{A} \times \mathcal{A}$ is called an *assembly theory over* $\mathfrak{I}$ if, for all $A, B, A_1, \ldots, A_n, B_1, \ldots, B_n \in \mathcal{A}$ (where $n \in \mathbb{N}$) and $I, J \in \mathcal{I}$, we have:

- **A1.** $\mathrm{cs}(\langle I\rangle)$ iff $\mathrm{cs}(I)$,
- **A2.** if $\mathrm{cs}(A)$, then $\varphi(A)$ is defined,
- **A3.** if $\varphi(\langle I\rangle)$ is defined, then $\varphi(\langle I\rangle) = I$,
- **A4.** $\preceq$ is reflexive and transitive,
- **A5.** $I \sqsubseteq J$ implies $\langle I\rangle \preceq \langle J\rangle$,

A6. if $A = A_1 \uplus \cdots \uplus A_n$ and $\mathrm{cs}(A_k)$ for $k = 1, \ldots, n$, then $\langle \varphi(A_1), \ldots, \varphi(A_n) \rangle \in \mathcal{A}$,
A7. if $A = A_1 \uplus \cdots \uplus A_n$, $\mathrm{cs}(A_k)$ for $k = 1, \ldots, n$ and $\mathrm{cs}(\langle \varphi(A_1), \ldots, \varphi(A_n) \rangle)$, then $\varphi(A) = \varphi(\langle \varphi(A_1), \ldots, \varphi(A_n) \rangle)$,
A8. if $A \preceq B$ and $\mathrm{cs}(B)$, then $\mathrm{cs}(A)$,
A9. if $A \preceq B$ and $\mathrm{cs}(B)$, then $\varphi(A) \sqsubseteq \varphi(B)$,
A10. if $A = A_1 \uplus \cdots \uplus A_n$, $B = B_1 \uplus \cdots \uplus B_n$, $\mathrm{cs}(\langle \varphi(B_1), \ldots \varphi(B_n) \rangle)$, as well as $\mathrm{cs}(B_k)$ and $A_k \preceq B_k$ for $k = 1, \ldots, n$, then $A \preceq B$.

Intuitively, the encapsulation $\varphi(A)$ of an assembly A represents the composition of A's components as an interface. Therefore, an assembly theory is called *canonical* if there is a strong correspondence between φ and $\|$. We write $\prod_{k \in K}$ for the generalisation of $\|$ to assemblies.

Definition 22 (Canonical Assembly Theory [22]) An assembly theory is called *canonical* if the following conditions hold:

1. $\mathrm{cs}(\langle I_k \rangle_{k \in K})$ iff, for all $l \in K$, I_l and $\prod_{k \in K \setminus \{l\}} I_k$ are compatible,
2. $\varphi(\langle I_k \rangle_{k \in K}) = \prod_{k \in K} \langle I_k \rangle$ if $\mathrm{cs}(\langle I_k \rangle_{k \in K})$, and undefined otherwise.

It is straightforward to define a canonical assembly theory over EMIA:

Definition 23 (Assembly Theory over EMIA) Let $\mathfrak{I}_{\mathsf{EMIA}} := (\mathsf{EMIA}, \mathrm{cs}, \otimes, \sqsubseteq_{\mathrm{e}})$ with $\mathrm{cs}(I)$ iff $S^0_I \cap \mathrm{bcl}^{\Omega}_I(E_I \cup U_I) = \emptyset$. We define $\mathfrak{A}_{\mathsf{EMIA}} := (\mathcal{A}, \mathrm{cs}, \varphi, \preceq)$ with $\mathcal{A} := \{\langle I_k \rangle_{k \in K} \mid 0 < |K| < \infty$ and $I_k, I_l \in \mathsf{EMIA}$ composable for $k \neq l\}$, $\mathrm{cs}(A)$ iff $S^0_{\varphi(A)} \cap \mathrm{bcl}^{\Omega}_{\varphi(A)}(E_{\varphi(A)} \cup U_{\varphi(A)}) = \emptyset$, $\varphi(\langle I \rangle) := I$ and $\varphi(\langle I_1, \ldots, I_n \rangle) := I_1 \otimes \cdots \otimes I_n$, and $A \preceq B$ iff $\varphi(A) \sqsubseteq_{\mathrm{e}} \varphi(B)$.

Lemma 24 $\mathfrak{A}_{\mathsf{EMIA}}$ *is an assembly theory over* $\mathfrak{I}_{\mathsf{EMIA}}$.

Proof A1 holds by definition. A2 is trivial because φ is defined for all assemblies. A3 holds by definition. A4 is trivial because $\sqsubseteq_{\mathrm{e}}$ is reflexive and transitive. A5 holds by definition. A6 and A7 are trivial due to the associativity of EMIA parallel composition. A8 holds by definition of $\sqsubseteq_{\mathrm{e}}$. A9 holds by definition of $\preceq$. A10 holds due to the compositionality of $\sqsubseteq_{\mathrm{e}}$. □

$\mathfrak{A}_{\mathsf{EMIA}}$ obviously satisfies the first condition of Def. 22. It almost satisfies the second condition, except that instead of being undefined in the 'otherwise'-branch, an erroneous interface results from the composition. We can either artificially set such a result to undefined in order to match the definition exactly, or argue that undefinedness is only necessary here because interface theories in [22] do not support the specification of erroneous interfaces (and, thus, one may change that definition accordingly). In both cases we have:

Theorem 25 (Assembly Theory) $\mathfrak{A}_{\mathsf{EMIA}}$ *is a canonical assembly theory over* $\mathfrak{I}_{\mathsf{EMIA}}$.

Because the encapsulation function ϕ directly corresponds to $\otimes$ and $\preceq$ to $\sqsubseteq_{\mathrm{e}}$, $\mathfrak{I}_{\mathsf{EMIA}}$ includes its own assembly theory $\mathfrak{A}_{\mathsf{EMIA}}$.

In addition, one can obviously show that EMIA constitutes an assembly theory for MIA:

Definition 26 (Assembly Theory over MIA) Let $\mathfrak{I}_{\mathsf{MIA}} := (\mathsf{MIA}, \mathrm{cs}, \|, \sqsubseteq_{\mathrm{m}})$ with $\mathrm{cs}(I)$ for all $I \in \mathsf{MIA}$. We define $\mathfrak{A}_{\mathsf{MIA}} := (\mathcal{A}, \mathrm{cs}, \varphi, \preceq)$ with $\mathcal{A} := \{\langle I_k \rangle_{k \in K} \mid 0 < |K| < \infty$ and $I_k, I_l \in \mathsf{MIA}$ composable for $k \neq l\}$, $\mathrm{cs}(\langle I_k \rangle_{k \in K})$ iff $S^0_{I_1 \otimes \ldots \otimes I_K} \cap \mathrm{bcl}^{\Omega}_{I_1 \otimes \ldots \otimes I_K}(E_{I_1 \otimes \ldots \otimes I_K} \cup U_{I_1 \otimes \ldots \otimes I_K}) = \emptyset$, $\varphi(\langle I \rangle) := I$ and $\varphi(\langle I_1, \ldots, I_n \rangle) := I_1 \| \cdots \| I_n$, and $A \preceq B$ iff $\varphi(A) \sqsubseteq_{\mathrm{m}} \varphi(B)$.

Lemma 27 $\mathfrak{A}_{\mathsf{MIA}}$ *is a canonical assembly theory over* $\mathfrak{I}_{\mathsf{MIA}}$.

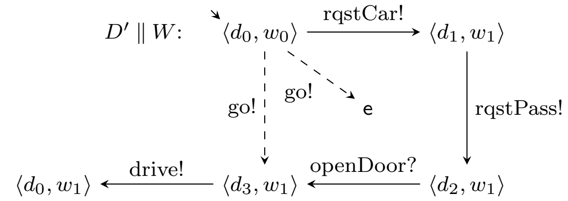
$D' \parallel W$:
$\langle d_0, w_0\rangle$
rqstCar!
$\langle d_1, w_1\rangle$
go!
go!
e
rqstPass!
$\langle d_0, w_1\rangle$
drive!
$\langle d_3, w_1\rangle$
openDoor?
$\langle d_2, w_1\rangle$

Fig. 7 Composition of product lines D' and W in EMIA, where $A_{D' \| W} := \{\text{openDoor?}\}/\{\text{drive!}, \text{go!}, \text{rqstCar!}, \text{rqstPass!}\}$.

3.3.3 Issue C: Software Product Lines

Consider specifications D and W of a car and a user interface product family, resp., both of which are shown in Fig. 6. These specifications allow product variations of a car and a user interface, which enable drivers to initiate the automatic driving assistance manually (go!), e.g., when parking in a different garage that is not equipped with an automatic door opener. Obviously, a user interface that provides this feature is incompatible with a car that does not, i.e., although some product combinations of D and W are compatible, some of them are not. Hence, D and W are incompatible, and no information that might help finding compatible product combinations is provided in current interface theories (see also the discussion about actual and potential errors in Sec. 1). In EMIA, the optional go?-transition at state d_0 would be modelled as a disjunctive go?-must-transition from d_0 to $\{d_3, \mathsf{e}\}$, for a fatal error state e. We refer to this specification as D'. The specified error information is still present in the parallel composition of D' and W, so that one may derive additional conditions on the go-transitions. These conditions result in compatible refinements of D' and W, which describe compatible sub-families of the original product families. For example, refining the optional go?-transition into a mandatory one in D', or removing the optional go!-transition in W; both result in appropriate restrictions to sub-families. The necessary error information is present in the EMIA parallel composition of D' and W (cf. Fig. 7).

3.3.4 Issue D: Unifying Composition Concepts

MTS and interface theories combining IA with MTS share many aspects of the modality semantics wrt. refinement. However, the meaning of may- and must-modalities differs wrt. parallel composition. Required and forbidden actions never cause an error in a parallel composition in MTS: either all components *unanimously* agree on implementing an action, or the action is forbidden in the composed system. The possibility to disagree on transitions enables an environment to control all transitions of an MTS, such that they may be interpreted as input-transitions from an interface theoretic view. However, the MTS parallel composition is not directly applicable to output actions, because these cannot be controlled by the environment. Consequently, previous interface theories have adopted an IA-like *error-aware* parallel composition that is tightly bound to a global compatibility concept. In contrast, EMIA's explicit error representation allows for a *local* description of compatibility that is independent of composition. Thus, EMIA unifies unanimous and error-aware parallel composition, i.e., it permits the mixing of these composition concepts within a specification. As an aside, note that EMIA collapses to MTS when considering input actions only.

The traditional interface-theoretic notions of optimistic and pessimistic compatibility may still be expressed in EMIA. Two composable EMIAs P, Q are *optimistically compatible* if and only if $S^0_{P\otimes Q} \cap \mathrm{bcl}^{\Omega}_{P\otimes Q}(E_{P\otimes Q} \cup U_{P\otimes Q}) = \emptyset$. Further, P and Q are *pessimistically compatible* if and only if $S^0_{P\otimes Q} \cap \mathrm{bcl}^{A\cup\{\tau\}}_{P\otimes Q}(\mathrm{ill}_{P\otimes Q}) = \emptyset$. As explained in Issue A, the error-information is not removed, i.e., in an optimistic variant of EMIA one cannot introduce unwanted behaviour as is the case in previous optimistic theories.

4 Logical Operators

Besides parallel composition, EMIA provides logical operators on interfaces, namely conjunction and disjunction. An implication operator is partially supported, and we show that a full support is impossible within MTS-based theories.

4.1 Conjunction and Disjunction

Perspective-based specification is concerned with specifying a system component from separate perspectives s.t. the component satisfies each of these perspective specifications. For example, each requirement for a component might describe a perspective. The component's overall specification is the most general specification refining all perspective specifications, i.e., it is the greatest lower bound wrt. the refinement preorder. This conjunction operator is defined in two stages:

Definition 28 (Conjunctive Product) Let P, Q be EMIAs with equal alphabets. The *conjunctive product* of P and Q is $P \,\&\, Q := (S_{P\&Q}, I, O, \longrightarrow_{P\&Q}, \dashrightarrow_{P\&Q}, S^0_{P\&Q}, E_{P\&Q}, U_{P\&Q})$ with $S_{P\&Q} := S_P \times S_Q$, $S^0_{P\&Q} := S^0_P \times S^0_Q$, $E_{P\&Q} := (E_P \times (E_Q \cup U_Q)) \cup ((E_P \cup U_P) \times E_Q)$, $U_{P\&Q} := U_P \times U_Q$, and the transition relations are given by the following rules:

C1. $\langle p,q\rangle \xrightarrow{i} \{\langle p',q'\rangle \mid p' \in P',\ q \overset{i}{\dashrightarrow}\overset{\epsilon}{\Rightarrow} q'\}$ if $p \xrightarrow{i} P'$ and $q \overset{i}{\dashrightarrow}\overset{\epsilon}{\Rightarrow}$,

C2. $\langle p,q\rangle \xrightarrow{i} \{\langle p',q'\rangle \mid p \overset{i}{\dashrightarrow}\overset{\epsilon}{\Rightarrow} p',\ q' \in Q'\}$ if $p \overset{i}{\dashrightarrow}\overset{\epsilon}{\Rightarrow}$ and $q \xrightarrow{i} Q'$,

C3. $\langle p,q\rangle \xrightarrow{\omega} \{\langle p',q'\rangle \mid p' \in P',\ q \overset{\omega}{\Longrightarrow} q'\}$ if $p \xrightarrow{\omega} P'$ and $q \overset{\omega}{\Longrightarrow}$,

C4. $\langle p,q\rangle \xrightarrow{\omega} \{\langle p',q'\rangle \mid p \overset{\omega}{\Longrightarrow} p',\ q' \in Q'\}$ if $p \overset{\omega}{\Longrightarrow}$ and $q \xrightarrow{\omega} Q'$,

C5. $\langle p,q\rangle \xrightarrow{\alpha} P' \times \{q\}$ if $p \xrightarrow{\alpha} P'$ and $q \in U_Q$,

C6. $\langle p,q\rangle \xrightarrow{\alpha} \{p\} \times Q'$ if $p \in U_P$ and $q \xrightarrow{\alpha} Q'$,

C7. $\langle p,q\rangle \overset{i}{\dashrightarrow} \langle p',q'\rangle$ if $p \overset{i}{\dashrightarrow}\overset{\epsilon}{\Rightarrow} p'$ and $q \overset{i}{\dashrightarrow}\overset{\epsilon}{\Rightarrow} q'$,

C8. $\langle p,q\rangle \overset{\omega}{\dashrightarrow} \langle p',q'\rangle$ if $p \overset{\omega}{\Longrightarrow} p'$ and $p \overset{\omega}{\Longrightarrow} q'$,

C9. $\langle p,q\rangle \overset{\tau}{\dashrightarrow} \langle p',q\rangle$ if $p \overset{\tau}{\Longrightarrow} p'$,

C10. $\langle p,q\rangle \overset{\tau}{\dashrightarrow} \langle p,q'\rangle$ if $q \overset{\tau}{\Longrightarrow} q'$,

C11. $\langle p,q\rangle \overset{\alpha}{\dashrightarrow} \langle p',q\rangle$ if $p \overset{\alpha}{\dashrightarrow} p'$ and $q \in U_Q$,

C12. $\langle p,q\rangle \overset{\alpha}{\dashrightarrow} \langle p,q'\rangle$ if $p \in U_P$ and $q \overset{\alpha}{\dashrightarrow} q'$,

A state $\langle p,q\rangle$ of $P \& Q$ is a candidate for refining both p and q. Because $\langle p,q\rangle$ cannot simultaneously require and forbid the same action a or be at once fatal and non-fatal, some states p and q do not have a common refinement. In such cases, $\langle p,q\rangle$ is called (logically) *inconsistent* and has to be removed from the candidates, including the removal of all states that require transitions leading to inconsistent states. In order to be the greatest common refinement of p and q, a state $\langle p,q\rangle$ may only be erroneous if p and q are erroneous or universal. This explains the definition of $E_{P\&Q}$ which obviously must exclude $U_{P\&Q}$.

Definition 29 (Conjunction) The set $\mathcal{F}_\& \subseteq S_{P\&Q}$ of *logically inconsistent* states is defined as the smallest set satisfying the following rules:

CF1. $\langle p,q\rangle \in (E_P \times (S_Q \setminus (E_Q \cup U_Q))) \cup ((S_P \setminus (E_P \cup U_P)) \times E_Q)$ implies $\langle p,q\rangle \in \mathcal{F}_\&$,

CF2. $\langle p,q\rangle \notin E_{P\&Q} \cup U_{P\&Q}$, $p \xrightarrow{i}$ and $q \overset{i}{\not\dashrightarrow}$ implies $\langle p,q\rangle \in \mathcal{F}_\&$,

CF3. $\langle p,q\rangle \notin E_{P\&Q} \cup U_{P\&Q}$, $p \overset{i}{\not\dashrightarrow}$ and $q \xrightarrow{i}$ implies $\langle p,q\rangle \in \mathcal{F}_\&$,

CF4. $\langle p,q\rangle \notin E_{P\&Q} \cup U_{P\&Q}$, $p \xrightarrow{\omega}$ and $q \overset{\omega}{\not\Longrightarrow}$ implies $\langle p,q\rangle \in \mathcal{F}_\&$,

CF5. $\langle p,q\rangle \notin E_{P\&Q} \cup U_{P\&Q}$, $p \overset{\omega}{\not\Longrightarrow}$ and $q \xrightarrow{\omega}$ implies $\langle p,q\rangle \in \mathcal{F}_\&$,

CF6. $\langle p,q\rangle \xrightarrow{\alpha} R$ and $R \subseteq \mathcal{F}_\&$ implies $\langle p,q\rangle \in \mathcal{F}_\&$.

The *conjunction* $P \wedge Q$ is obtained from $P \& Q$ by deleting all states in $\mathcal{F}_{\&}$. This deletes all transitions exiting deleted states and removes all deleted states from targets of must-transitions.

Fatal error states are excluded in Rules CF2 through CF5 because we do not care about consistency for these states. Note that the states in E and $\mathcal{F}_{\&}$ are different in nature: E-states represent states with possible but unwanted behaviour, whereas $\mathcal{F}_{\&}$-states represent contradictory specifications that are impossible to implement.

In order to prove that conjunction is the greatest lower bound wrt. the refinement preorder $\sqsubseteq_{\mathrm{e}}$, we need the notion of a witness along the lines of [7]:

Definition 30 (Witness [7]) Let P and Q be EMIAs with equal alphabets. A set $W \subseteq S_P \times S_Q$ is a *witness* of $P \,\&\, Q$ if, for all $\langle p, q\rangle \in W$, the following conditions hold:

W1. $p \in E_P$ implies $q \in E_Q \cup U_Q$,
W2. $q \in E_Q$ implies $p \in E_P \cup U_P$,
W3. $p \xrightarrow{o}_P$ implies $q \overset{o}{=\!=\!\!>}_Q$ or $q \in E_Q \cup U_Q$,
W4. $q \xrightarrow{o}_Q$ implies $p \overset{o}{=\!=\!\!>}_P$ or $p \in E_P \cup U_P$,
W5. $p \xrightarrow{i}_P$ implies $q \overset{i}{-\!\!\to}\overset{\epsilon}{=\!\!>}_Q$ or $q \in E_Q \cup U_Q$,
W6. $q \xrightarrow{i}_Q$ implies $p \overset{i}{-\!\!\to}\overset{\epsilon}{=\!\!>}_P$ or $p \in E_P \cup U_P$,
W7. $\langle p, q\rangle \xrightarrow{\alpha} R'$ implies $R' \cap W \neq \emptyset$.

We instantiate the concept of a witness concretely as follows:

Lemma 31 (Concrete Witness [7]) *Let P, Q, R be EMIAs with equal alphabets.*

1. *For any witness W of $P \,\&\, Q$, we have $W \cap \mathcal{F}_{\&} = \emptyset$.*
2. *The set $W := \{\langle p, q\rangle \in S_P \times S_Q \mid \exists r \in S_R.\, r \sqsubseteq_{\mathrm{e}} p$ and $r \sqsubseteq_{\mathrm{e}} q\}$ is a witness of $P \,\&\, Q$.*

Proof Claim 1 is obvious, so we only prove Claim 2:

W1 By R1, we get $p \in E_P$ implies $r \in E_R$ implies $q \in E_Q \cup U_Q$.
W2 Symmetrically to W1.
W3 If $q \in E_Q$, then W2 applies and there is nothing to show. Otherwise, let $p \xrightarrow{o}_P$. By $r \sqsubseteq_{\mathrm{e}} p$, there is a transition $r \xrightarrow{o}_R$ and, by syntactic consistency and $r \sqsubseteq_{\mathrm{e}} q$, a $q \overset{o}{=\!=\!\!>}_Q$.
W4 Symmetrically to W3.
W5 Analogous to W3 when replacing $\xrightarrow{o}$ and $\overset{o}{=\!=\!\!>}$ with $\xrightarrow{i}$ and $\overset{i}{-\!\!\to}\overset{\epsilon}{=\!\!>}$, resp.
W6 Symmetrically to W5.
W7 Let $\langle p, q\rangle \in W$ due to r s.t. $\langle p, q\rangle \xrightarrow{\omega} R'$ because of C3. By $r \sqsubseteq_{\mathrm{e}} p$, there is a matching $r \overset{\omega}{\Longrightarrow}_R R'$. For all $r' \in R'$, by syntactic consistency, we have a transition $r \overset{\omega}{=\!=\!\!>}_R r'$, such that $r \sqsubseteq_{\mathrm{e}} q$ implies the existence of a transition $q \overset{\omega}{=\!=\!\!>}_Q q'$ with $r' \sqsubseteq_{\mathrm{e}} q'$. Hence, there is a $\langle p', q'\rangle \in R' \cap W$ due to r'. The case of inputs is shown analogously. □

Next, we show that $\wedge$ is indeed conjunction:

Proposition 32 ($\wedge$ is And) *If P and Q are EMIAs with equal alphabets, then (i) an R with $R \sqsubseteq_{\mathrm{e}} P$ and $R \sqsubseteq_{\mathrm{e}} Q$ exists iff P and Q are consistent. Further, if P and Q are consistent, then, for any R, (ii) $R \sqsubseteq_{\mathrm{e}} P$ and $R \sqsubseteq_{\mathrm{e}} Q$ iff $R \sqsubseteq_{\mathrm{e}} P \wedge Q$.*

Proof (i) "$\Rightarrow''$ follows from Lemma 31.

(i), (ii) "$\Leftarrow$": Let $R \sqsubseteq_{\mathrm{e}} P \wedge Q$. We prove that $\mathcal{R} := \{\langle r, p\rangle \mid \exists q.\, r \sqsubseteq_{\mathrm{e}} p \wedge q\}$ is an error-aware modal refinement relation. By choosing $S_R^0 := \big\{r \in S_R \bigm| \exists p \wedge q \in S_{P\wedge Q}^0.\, \langle r, p \wedge q\rangle \in \mathcal{R}\big\}$ we may conclude (i) "$\Leftarrow$". Let $\langle r, p\rangle \in \mathcal{R}$ due to q. The proof follows closely the lines of [7] and proceeds as follows:

R1 If $r \in E_R$, then $p \wedge q \in E_{P\wedge Q}$; thus, $p \in E_P$.

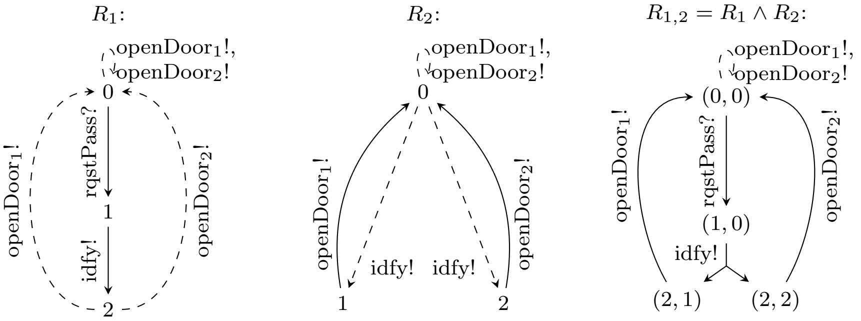

Fig. 8 Example of conjunction with the common alphabet $\{\text{rqstPass?}\}/\{\text{idfy!}, \text{openDoor}_1!, \text{openDoor}_2!\}$.

R3, R4 Let $p \xrightarrow{\alpha}_P P'$, then we have $q \overset{\alpha}{\Longrightarrow}_Q$ and $p \wedge q \xrightarrow{\alpha} \{p' \wedge q' \mid p' \in P',\ q \overset{\alpha}{\Longrightarrow}_Q q',\ p' \wedge q' \text{ defined}\}$. By $r' \sqsubseteq_{\mathrm{e}} p' \wedge q'$ we get a matching $r \xrightarrow{\alpha}_R R'$, i.e., $\forall r' \in R'\, \exists p' \in P'.\, \langle r', p' \rangle \in \mathcal{R}$. (In case of inputs, $\overset{\alpha}{\Longrightarrow}$ must be replaced by $\overset{\alpha}{\dashrightarrow}\overset{\epsilon}{\Longrightarrow}$.)

R5, R6 Let $r \overset{\alpha}{\dashrightarrow} r'$. By $r \sqsubseteq_{\mathrm{e}} p \wedge q$, there is a $p \wedge q \overset{\alpha}{\Longrightarrow} p' \wedge q'$ such that $r' \sqsubseteq_{\mathrm{e}} p' \wedge q'$; thus, $\langle r', p' \rangle \in \mathcal{R}$ due to q'. (In case of inputs, $\overset{\alpha}{\Longrightarrow}$ must be replaced by $\overset{\alpha}{\dashrightarrow}\overset{\epsilon}{\Longrightarrow}$.)

(ii) "$\Rightarrow$": We show that $\mathcal{R} := \{\langle r, p \wedge q \rangle \mid r \sqsubseteq_{\mathrm{e}} p \text{ and } r \sqsubseteq_{\mathrm{e}} q\}$ is an error-aware modal refinement relation.

R1 Obvious.
R3, R4, R5, R6 As above, the proof closely follows the lines of [7]. □

It is easy to also define a disjunction operator, which may be employed for specifying alternative implementations:

Definition 33 (Disjunction) For a family of EMIAs $\mathcal{P} := (P_j)_{j \in J}$ with equal alphabets, we define the *disjunction* of $\mathcal{P}$ as the following EMIA:
$\bigvee_{j \in J} P_j := (\biguplus_{j \in J} S_{P_j}, I, O, \biguplus_{j \in J} \longrightarrow_{P_j}, \biguplus_{j \in J} \dashrightarrow_{P_j}, \biguplus_{j \in J} S^0_{P_j}, \biguplus_{j \in J} E_{P_j})$.

Proposition 34 ($\vee$ is Or) *If P_j, for $j \in J$, and R are EMIAs with equal alphabets, then $\bigvee_{j \in J} P_j \sqsubseteq_{\mathrm{e}} R$ iff $P_j \sqsubseteq_{\mathrm{e}} R$ for all $j \in J$.*

Proof Let P_j $(j \in J)$ and R be EMIAs with equal alphabets and w.l.o.g. disjoint state sets S_j and S_R, and let $P_j \sqsubseteq_{\mathrm{e}} R$ due to the error-aware modal refinement relation $\mathcal{R}_j$. Because, in general, the union of error-aware modal refinement relations is an error-aware modal refinement relation, $(\bigcup_{j \in J} \mathcal{R}_j) \cup \mathcal{R}_Q$ is an error-aware modal refinement relation, too. Vice versa, if $\bigvee_{j \in J} P_j \sqsubseteq_{\mathrm{e}} R$ due to an error-aware modal refinement relation $\mathcal{R}$, then, for any $j \in J$, $\mathcal{R}_j := \mathcal{R} \cap (S_j \times S_R)$ is a suitable error-aware modal refinement relation, showing $P_j \sqsubseteq_{\mathrm{e}} R$. □

4.2 Example

In this section we illustrate how conjunction may be employed for perspective-based specification. Consider a double garage for which we want to specify a single controller operating both garage doors appropriately according to an identification of the requesting car. We state two requirements for such a controller, each of which may be considered as a separate perspective on the controller:

R_1: After a passage request, the garage shall identify the car and may then open one of the doors.
R_2: After the car is identified, the garage shall open either Door 1 or Door 2.

A representation of these requirements as EMIAs is shown in Fig. 8. In Specification R_1, the rqstPass?-transition from state 0 to state 1 is the entrance condition that may be triggered by a car in order to request passage. Upon such a request, the garage must identify (idfy!) the car, and may then open Door 1 or Door 2. Requirement R_2 specifies that after an identification, either Door 1 or Door 2 must be opened, i.e., the choice of door is a result of the identification. The overall specification must satisfy both requirements simultaneously; hence, we use conjunction in order to construct the greatest lower bound $R := R_1 \wedge R_2$, which is also shown in Fig. 8. Notably, the combination of nondeterminism and modalities of action idfy! yields a disjunctive must-transition in the conjunction.

4.3 Implication and Negation

In addition to conjunction and disjunction, it would be useful to define further logical operators like implication and negation. Implication $\rightarrow$, as an adjoint to conjunction, is defined by the condition $X \sqsubseteq_{\mathrm{e}} P \rightarrow C$ iff $X \wedge P \sqsubseteq_{\mathrm{e}} C$. In particular, we have $P \sqsubseteq_{\mathrm{e}} C$ iff $P \rightarrow C \sqsupseteq\sqsubseteq_{\mathrm{e}} \top$. Negation arises as the special case $\neg P := P \rightarrow \bot$. A straightforward way of defining implication is by setting $P \rightarrow C := \bigvee\{X \mid X \wedge P \sqsubseteq_{\mathrm{e}} C\}$. However, this declarative definition is impractical due to the infinite disjunction. Unfortunately, we can show that DMTS and, thus, any MTS-based interface theory is not closed under negation, so that an operational construction of implication and negation in the spirit of the other operators is impossible (cf. Thm. 35).

In a trace-based setting similar to deterministic IA, Dill argues that safety properties are not closed under negation and, therefore, a negation operator does not exist in his setting [16]. However, it is unclear to what extent this argument applies to an MTS-based setting where must-transitions express a limited form of liveness.

Gössler and Raclet [21] introduced an underapproximation $\rightsquigarrow$ of implication for deterministic MTS. This *sub-implication* satisfies $X \sqsubseteq P \rightsquigarrow C \implies X \wedge P \sqsubseteq C$, but the reverse direction does not hold in general. Specifications X and P are called *non-conflicting*—a concept introduced as *independence* in [27]—if $\mathcal{F}_{\&} = \emptyset$. For non-conflicting specifications X and P, Gössler and Raclet show that the reverse direction also holds, i.e., $X \wedge P \sqsubseteq C \implies X \sqsubseteq P \rightsquigarrow C$. In particular, $P \sqsubseteq C$ is such an X, i.e., P and $P \sqsubseteq C$ are non-conflicting. However, they do not consider that non-conflicting X and P satisfy $X \wedge P \sqsubseteq C$ only if P and C, as well as X and C are also non-conflicting. This undermines the purpose of disjunctive must-transitions to provide a choice between alternatives because, in a non-conflicting conjunction, all alternatives must be preserved. Further, as P and $P \sqsubseteq \bot$ are non-conflicting, $P \rightsquigarrow \bot \sqsupseteq\sqsubseteq \bot$ for all consistent specifications P, which renders negation completely useless. These issues significantly restrict the usability of sub-implication.

Gössler and Raclet also show that an implication operator does not exist for MTS. Because their counterexample does not work in the nondeterministic setting of EMIA, we provide a more general argument:

Theorem 35 (DMTS and Negation) *Disjunctive Modal Transition Systems (DMTS) are not closed under negation; hence, conjunction has no adjoint in DMTS.*

Proof DMTS have been shown to be equally expressive as Hennessy-Milner-Logic with greatest fixed points (νHML) if the number of initial states is required to be finite [4]. If DMTS were closed under negation, then least fixed points would be expressible by defining $\mu X.\, \phi(X) := \neg\nu X.\, \neg\phi(\neg X)$, and DMTS would be as expressive as the modal μ-calculus, which is strictly more expressive that νHML. □

A concrete example that illustrates the difficulty with negation is a specification S over alphabet $\{a\}$ with a single state s and a looping transition $s \xrightarrow{a} s$. The negation of S comprises all implementations that are inconsistent with S, i.e., all implementations that include a finite chain of

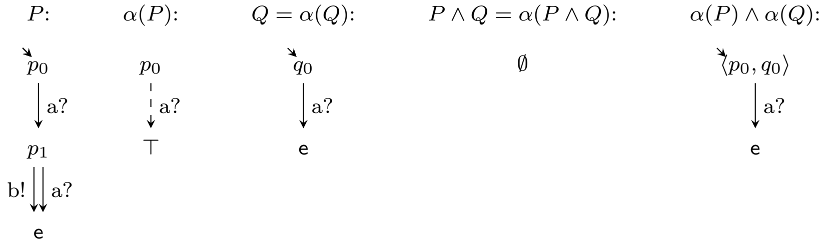
P:
p_0
a?
p_1
b!
a?
e
$\alpha(P)$:
p_0
a?
$\top$
$Q = \alpha(Q)$:
q_0
a?
e
$P \wedge Q = \alpha(P \wedge Q)$:
$\emptyset$
$\alpha(P) \wedge \alpha(Q)$:
$\langle p_0, q_0 \rangle$
a?
e

Fig. 9 Example of EMIAs P, Q with $\alpha(P \wedge Q) \not\sqsupseteq_{\mathrm{m}} \alpha(P) \wedge \alpha(Q)$, with the common alphabet $\{a?\}/\{b!\}$.

a-transitions, e.g., $T\colon t_0 \xrightarrow{a} t_1 \xrightarrow{a} t_2$. In some sense, sub-implication captures the non-negative part of implication, which explains the relation to non-conflicting conjunctions.

In order to extend the proposed sub-implication to nondeterministic interfaces, one may employ an exponential construction similar to the one presented in [4] for the quotient; however, this would lead to similar complications (see also Sec. 5). But this would be done for a questionable underapproximation, so we leave this matter for future work.

4.4 Logical Operators under Galois Insertion

In this subsection we briefly discuss how the Galois insertion relates MIA and EMIA wrt. the logical operators presented above. First, we show that MIA is closed under conjunction.

Lemma 36 (MIA-Conjunction [7]) *If P and Q are MIAs with equal alphabets, then their conjunction $P \wedge Q$ is also a MIA.*

Proof It is easy to see that the conjunctive product & (cf. Def. 28) preserves the properties M1 through M3. Hence, it remains to show that the pruning of inconsistent states also preserves these properties. A state $p \wedge q$ may only have a disabled input i if all i-may-transitions lead to states in $\mathcal{F}_{\&}$. Then, $p \wedge q$ would be inconsistent due to CF6 because $P \,\&\, Q$ satisfies M1 and M2. Therefore, $p \wedge q$ must be input enabled. The same line of reasoning applies to M2. Property M3 is trivial. □

Hence, conjunction is the greatest lower bound wrt. $\sqsubseteq_{\mathrm{m}}$ when restricted to MIAs. The map α is not homomorphic wrt. conjunction: although $\alpha(P \wedge Q) \sqsubseteq_{\mathrm{m}} \alpha(P) \wedge \alpha(Q)$ holds for $P, Q \in \mathsf{EMIA}'$ because α is monotonic, the converse direction "$\sqsupseteq_{\mathrm{m}}$" does not hold in general, because MIA's replacement of illegal states by $\top$ must be reproduced by α. An example of EMIAs P and Q with $\alpha(P \wedge Q) \not\sqsupseteq_{\mathrm{m}} \alpha(P) \wedge \alpha(Q)$ is shown in Fig 9. State p_1 of specification P is in ill_P due to the b!-transition. Therefore, α prunes p_1 and replaces it by a universal state $\top$ in $\alpha(P)$. The conjunction $P \wedge Q$ is inconsistent because P's regular state p_1 is conjoined with Q's fatal error state e, and the a?-must-transition propagates this inconsistency back to the initial state. In the abstract setting, both the error and the inconsistency are avoided resulting in a regular and consistent initial state that is trivially refined by $P \wedge Q$.

It is obvious that MIAs are closed under disjunction and that α is homomorphic wrt. disjunction. Further, α respects implication although we cannot define implication operationally:

Lemma 37 (Abstraction Respects Implication) *If $X \sqsubseteq_{\mathrm{e}} P \to C$, then $\alpha(X) \sqsubseteq_{\mathrm{e}} \alpha(P \to C)$.*

Proof $X \sqsubseteq_{\mathrm{e}} P \to C \iff X \wedge P \sqsubseteq_{\mathrm{e}} C \implies \alpha(X) \wedge \alpha(P) \sqsubseteq_{\mathrm{e}} \alpha(C) \iff \alpha(X) \sqsubseteq_{\mathrm{e}} \alpha(P) \to \alpha(C)$. □

5 Quotient and Standard Process Algebraic Operators

In this section, we discuss the quotient operator, which is adjoint to parallel composition, and the standard process algebraic operations hiding, restriction and alphabet extension.

5.1 Quotient

The quotient operation is adjoint to parallel composition. It equips the theory with the possibility of component synthesis which allows for component reuse and incremental, component-based design. Given EMIAs P and D, the quotient of P over D is the coarsest EMIA Q such that *the defining inequality of the quotient*, $Q \otimes D \sqsubseteq_{\mathrm{e}} P$, holds. We denote the quotient by $P \mathbin{/\!\!/} D$ if it exists. In the following, P is the *dividend* (one may think of it as an overall system specification), D the *divisor* (an already implemented component) and Q the *quotient* (the synthesised completion of D).

We define the quotient for a restricted set of EMIAs, namely where the specification P has no τs and where the divisor D is may-deterministic and without τs. We call D *may-deterministic* if $d \overset{\alpha}{\dashrightarrow} d'$ and $d \overset{\alpha}{\dashrightarrow} d''$ implies $d' = d''$ for all d, d', d'' and α. Due to syntactic consistency, a may-deterministic EMIA has no disjunctive must-transitions, i.e., the target sets of must-transitions are singletons. In principle, the determinism requirement on D may be omitted as Beneš et al. [4] do for DMTS. However, this comes with an exponential blowup in the quotient size and significant complications when adapting to interface theories as we already discuss in [7] for MIA.

Like several other operators, we define the quotient in two stages, where we write $\mathrm{may}_P(p, \alpha)$ for $\{p' \in P \mid p \overset{\alpha}{\dashrightarrow}_P p'\}$. Regarding the choice of the input and output alphabets in the following definition we adopt the one by Chilton et al. [11] and Raclet et al. [32]. Alternative choices are discussed in [7].

Definition 38 (Pre-quotient) Let P and D be τ-free EMIAs with $A_D \subseteq A_P$ and $O_D \subseteq O_P$. The *pre-quotient of* P *over* D is defined as the EMIA $P \oslash D := (S_P \times S_D \cup \{\top\}, I, O, \longrightarrow, \dashrightarrow, S_P^0 \times S_D^0, E, U)$, where $I := I_P \cup O_D$, $O := O_P \setminus O_D$, $E := E_P \times (S_D \setminus E_D)$ and $U := (U_P \times S_D) \cup (E_P \times E_D) \cup \{\top\}$. The transition relations of a state $\langle p, d \rangle$ are defined by the following rules:

PQ1. $\langle p, d \rangle \xrightarrow{a} P' \times \{d\}$ if $p \xrightarrow{a} P'$ and $a \notin A_D$,
PQ2. $\langle p, d \rangle \xrightarrow{a} P' \times D'$ if $p \xrightarrow{a} P'$ and $d \xrightarrow{a} D'$,
PQ3. $\langle p, d \rangle \overset{a}{\dashrightarrow} \langle p', d \rangle$ if $p \overset{a}{\dashrightarrow}$ and $a \notin A_D$,
PQ4. $\langle p, d \rangle \overset{a}{\dashrightarrow} \langle p', d' \rangle$ if $p \overset{a}{\dashrightarrow}$ and $d \overset{a}{\dashrightarrow} d'$,
PQ5. $\langle p, d \rangle \overset{a}{\dashrightarrow} \top$ if $p \not\xrightarrow{a}$ and $d \overset{a}{\not\dashrightarrow}$.

A state $q = \langle p, d \rangle$ in $P \oslash D$ encodes the condition that q should be the coarsest state wrt. $\sqsubseteq_{\mathrm{e}}$ such that q composed in parallel with d refines p. The purpose of the new state $\top$ is to ensure that U is nonempty, in order to have a universal target state in Rule PQ5. In case U is nonempty anyway, an arbitrary state from U may replace $\top$. With this in mind, we now justify the choices of E and U and the rules of Def. 38 intuitively. A formal proof is given in Lem. 40 and Thm. 41 below.

An error state $q \in E$ of the quotient satisfies $q \parallel d \sqsubseteq_{\mathrm{e}} p$ for some state $d \in S_D$ if and only if $p \in E_P \cup U_P$. However, if $d \in E_D$ or $p \in U_P$, then nothing is required for q to satisfy $q \parallel d \sqsubseteq_{\mathrm{e}} p$ and, hence, q has to be universal instead of erroneous in order to ensure the maximality of the quotient. This justifies the choices of E and U.

Rule PQ1 is necessary due to the following consideration. If P has an a-must-transition where a is unknown to D, then this can only originate from an a-must-transition in the quotient Q that we wish to construct. To be most permissive, each $p' \in P'$ must have a match in $Q \otimes D$. The corresponding consideration is true for Rule PQ3, which also ensures syntactic consistency for Rule PQ1.

Rule PQ2 is obvious in the light of the choice of alphabet in Def. 38. Because $P \oslash D$ has all actions of P and D in its alphabet, it also needs an a-must-transition to produce such a transition at $(p, d) \otimes d$. Here, Rule PQ4 is the companion rule for guaranteeing syntactic consistency.

Rule PQ5 makes $P \oslash D$ as coarse as possible. The input a-may-transitions introduced here just disappear in $(P \oslash D) \otimes D$ since a is blocked by D.

It is easy to see, that $P \oslash D$ is indeed an EMIA. Up to now we have only defined the pre-quotient. Considering a candidate pair (p, d), it may be impossible that p is refined by a state resulting from a parallel composition with d; this depends, e.g., on the modalities and the labels of the transitions leaving p and d. We call such pairs *divisionally inconsistent states* and remove them from the pre-quotient. For example, consider states $p \xrightarrow{a}$ and $d \overset{a}{\dashrightarrow}$ such that $d \not\xrightarrow{a}$; no parallel composition with d refines p. While may-transitions can be refined by removing them and disjunctive transitions can be refined to subsets of their targets in order to prevent the reachability of inconsistent states, all states having a must-transition to only inconsistent states must also be removed.

Definition 39 (Quotient) Let $P \oslash D$ be the pre-quotient of P over D. The set $\mathcal{F}_\oslash \subseteq S_P \times S_D$ of *divisionally inconsistent states* is defined as the least set satisfying the following rules:

QF1. $p \notin U_P \cup E_P$ and $d \in U_D \cup E_D$ implies $\langle p, d\rangle \in \mathcal{F}_\oslash$,
QF2. $p \xrightarrow{a}_P$ and $d \not\xrightarrow{a}_D$ and $a \in A_D$ implies $\langle p, d\rangle \in \mathcal{F}_\oslash$,
QF3. $(p, d) \xrightarrow{a}_{P\oslash D} R'$ and $R' \subseteq \mathcal{F}_\oslash$ implies $\langle p, d\rangle \in \mathcal{F}_\oslash$.

The *quotient* $P /\!\!/ D$ is obtained from $P \oslash D$ by deleting all states in $\mathcal{F}_\oslash$. This also removes any may- or must-transition exiting a deleted state and any may-transition entering a deleted state; in addition, deleted states are removed from targets of disjunctive must-transitions. If $(p, d) \in S_{P/\!\!/D}$, then we write $p /\!\!/ d$. If $S^0_{P/\!\!/D}$ is empty, then the quotient of P over D is inconsistent.

Rule QF1 captures the division by universal states and error states. A state $d \in U_D \cup E_D$ in parallel with any state is either universal or an error state, and does not refine $p \notin U_P \cup E_P$. Rule QF2 is obvious since (p, d) cannot ensure that $p \xrightarrow{a}_P$ is matched if d has no a-must-transition, as an a-may-transition or a forbidden action a at d can in no case compose to a refinement of a must-transition at p. Rule QF3 propagates back all inconsistencies that cannot be avoided by refining.

Since $P \oslash D$ is an EMIA and since syntactic consistency and the special states are preserved by pruning, $P /\!\!/ D$ is an EMIA, too. If the target set of a disjunctive must-transition became empty due to pruning, i.e., $R' \subseteq \mathcal{F}_\oslash$, Rule QF3 would be applicable and the source state and its must-transition are deleted.

We show next that the quotient operation above yields the coarsest EMIA satisfying the defining inequality. For this proof, the following lemma ensures that errors and inconsistencies of $/\!\!/$ are preserved across refinement:

Lemma 40 *Let P, D, Q be EMIAs such that P is τ-free, D is τ-free and may-deterministic, $A_D \subseteq A_P$, $O_D \subseteq O_P$, $O_Q = O_P \setminus O_D$ and $I_Q = I_P \cup O_D$. Further, let p, d, q be states in P, D, Q, resp. Then, the following statements hold:*

1. *If $q \otimes d \sqsubseteq_{\mathrm{e}} p$, then $\langle p, d\rangle \notin \mathcal{F}_\oslash$.*
2. *If $q \sqsubseteq_{\mathrm{e}} p /\!\!/ d$ and $p \notin U_P \cup E_P$, then $q \otimes d \notin E_{Q\otimes D}$.*

Proof We write $\longrightarrow_\otimes$, $\longrightarrow_\oslash$ and $\longrightarrow_{/\!\!/}$ as shorthands for $\longrightarrow_{Q\otimes D}$, $\longrightarrow_{P\oslash D}$ and $\longrightarrow_{P/\!\!/D}$, resp., and analogously for may-transitions.

Claim 1: We show that $(q \otimes d \sqsubseteq_{\mathrm{e}} p) \wedge (\langle p, d\rangle \in \mathcal{F}_\oslash)$ implies a contradiction. We prove this by induction on the rules of Def. 39, where our induction hypothesis is formalised as $H(p, d) \equiv \forall q.\, (q \otimes d \sqsubseteq_{\mathrm{e}} p \wedge \langle p, d\rangle \in \mathcal{F}_\oslash) \implies \bot$.

QF1 $p \notin E_P \cup U_P$ and $d \in E_D \cup U_D$: By Def. 5, we have $q \otimes d \in E_{Q\otimes D} \cup U_{Q\otimes D}$, and $q \otimes d \sqsubseteq_{\mathrm{e}} p$ implies $p \in E_P \cup U_P$ which contradicts our assumption.

QF2 $p \xrightarrow{a}$, $d \not\xrightarrow{a}$ and $a \in A_D$: By $q \otimes d \sqsubseteq_{\mathrm{e}} p$, we have $q \otimes d \xrightarrow{a}_\otimes$, which can only be due to P2 or P3; thus, $d \xrightarrow{a}$, which is a contradiction.

QF3 $(p, d) \xrightarrow{a}_\oslash R'$ with $R' \subseteq \mathcal{F}_\oslash$: By induction hypothesis $H(p', d')$ holds for all $(p', d') \in R'$. The transition is due to one of the rules PQ1 and PQ2:

PQ1 $p \xrightarrow{a} P'$, $a \notin A_D$ and $R' = P' \times \{d\}$: By $q \otimes d \sqsubseteq_{\mathrm{e}} p$, we have $q \otimes d \xrightarrow{a}_\otimes Q' \times \{d\}$ for some Q' such that $\forall q' {\in} Q'.\, \exists p' {\in} P'.\, q' \otimes d \sqsubseteq_{\mathrm{e}} p'$. Since $(p', d) \in R' \subseteq \mathcal{F}_\oslash$, $H(p', d)$ implies a contradiction.

PQ2 $p \xrightarrow{a} P'$, $d \xrightarrow{a} \{d'\}$ and $R' = P' \times \{d'\}$: By $q \otimes d \sqsubseteq_{\mathrm{e}} p$, there is a Q' with $q \xrightarrow{a} Q'$ and $\forall q' {\in} Q'.\, \exists p' {\in} P'.\, q' \otimes d' \sqsubseteq_{\mathrm{e}} p'$. Due to $(p', d') \in R' \subseteq \mathcal{F}_{\oslash}$ we can derive a contradiction from $H(p', d')$.

Claim 2: We show that $(q \sqsubseteq_{\mathrm{e}} p \mathbin{/\!\!/} d) \wedge (p \notin E_P) \wedge \langle q, d\rangle \in E_{Q \otimes D}$ implies a contradiction. By Def. 5, there are two cases for $\langle q, d\rangle \in E_{Q \otimes D}$:

A $\langle q, d\rangle \in E_Q \times S_D$: By $q \sqsubseteq_{\mathrm{e}} p \mathbin{/\!\!/} d$, we have $p \mathbin{/\!\!/} d \in E_{P /\!\!/ D}$. Hence, $p \in E_P$, which is a contradiction.
B $\langle q, d\rangle \in S_Q \times E_D$: By QF1, $\langle p, d\rangle \in \mathcal{F}_{\oslash}$, which contradicts $q \sqsubseteq_{\mathrm{e}} p \mathbin{/\!\!/} d$. □

Now, we can show that $/\!\!/$ is indeed a quotient operator wrt. $\otimes$:

Theorem 41 ($/\!\!/$ is a Quotient Operator wrt. $\otimes$) *Let P, D and Q be EMIAs such that P is τ-free, D is τ-free and may-deterministic, $A_D \subseteq A_P$, $O_D \subseteq O_P$, $O_Q = O_P \setminus O_D$ and $I_Q = I_P \cup O_D$. Then, $Q \sqsubseteq_{\mathrm{e}} P \mathbin{/\!\!/} D$ iff $Q \otimes D \sqsubseteq_{\mathrm{e}} P$.*

Proof We use the same shorthands as in Lem. 40.

"⇒": We show that $\mathcal{R} := \{(q{\otimes}d, p) \in S_{Q \otimes D} \times S_P \mid q \sqsubseteq_{\mathrm{e}} p \mathbin{/\!\!/} d \text{ or } p \in U_P\}$ is an error-aware modal refinement relation. We only have to consider a $(q{\otimes}d, p) \in \mathcal{R}$ with $p \notin U_P$. Note that Cases R4 and R6 are mostly analogous to Cases R3 and R5, resp.

R1 $q \otimes d \notin E_{Q \otimes D}$ iff (by Def. 5) $q \notin E_Q \wedge d \notin E_D$ iff (by $q \sqsubseteq_{\mathrm{e}} p \mathbin{/\!\!/} d$) $p \mathbin{/\!\!/} d \notin E_{P /\!\!/ D} \wedge d \notin E_D$. Def. 38 implies $p \notin E_P$. Vice versa, $p \notin E_P$ implies $p \mathbin{/\!\!/} d \notin E_{P /\!\!/ D}$. Now, $q \sqsubseteq_{\mathrm{e}} p \mathbin{/\!\!/} d$ implies that $p \mathbin{/\!\!/} d$ is consistent, hence, $d \notin E_D$ and, due to Def. 5, $q \otimes d \notin E_{Q \otimes D}$.

R2 By Def. 38, $p \notin U_P$ implies $p \mathbin{/\!\!/} d \notin U_{P /\!\!/ D}$. Due to $q \sqsubseteq_{\mathrm{e}} p \mathbin{/\!\!/} d$, we have $q \notin U_Q$. Now, QF1 implies $d \notin U_D$, hence, $q \otimes d \notin U_{Q \otimes D}$.

R3 $p \xrightarrow{i} P'$ for $i \in I_P$:

1. If $i \in A_D$ and $d \xrightarrow{i} \{d'\}$, then PQ2 implies $(p, d) \xrightarrow{i}_{\oslash} P' \times \{d'\}$. In $P \mathbin{/\!\!/} D$, the target set might only be a subset $P'' \times \{d'\}$ of $P' \times \{d'\}$. By $q \sqsubseteq_{\mathrm{e}} p \mathbin{/\!\!/} d$, we have $q \xrightarrow{i} Q'$ for some Q' such that $\forall q' {\in} Q'.\, \exists p' {\in} P''.\, q' \sqsubseteq_{\mathrm{e}} p' \mathbin{/\!\!/} d'$, whence $(q'{\otimes}d', p') \in \mathcal{R}$. Now, by P3, there is a transition $(q, d) \xrightarrow{i}_{\otimes} Q' \times \{d'\}$.
2. If $i \in A_D$ and $d \overset{i}{\not\longrightarrow}$, then $(p, d) \in \mathcal{F}_{\oslash}$ by QF2, which is impossible since $p /\!\!/ d$ is consistent.
3. If $i \notin A_D$, the proof is analogous to Case 1 with $d = d'$, when replacing PQ2 by PQ1 and P3 by P1.

R4 $p \xrightarrow{o} P'$ for $o \in O_P$: Here, the same arguments as for R3 apply.

R5 $q \otimes d \overset{i}{\dashrightarrow}_{\otimes} q' \otimes d'$ and $i \in I_P = I_{Q \otimes D}$: This transition is due to one of the rules P4 or P6. Rule P5 is impossible as $A_Q = A_P \supseteq A_D$.

P4 $q \overset{i}{\dashrightarrow} q'$ and $i \notin A_D$: We have $d = d'$, and $q \sqsubseteq_{\mathrm{e}} p \mathbin{/\!\!/} d$ implies $p \mathbin{/\!\!/} d \overset{i}{\dashrightarrow}_{/\!\!/} p' \mathbin{/\!\!/} d''$ for some p', d'' such that $q' \sqsubseteq_{\mathrm{e}} p' \mathbin{/\!\!/} d''$. Since $i \notin A_D$, we get $d = d''$ and $p \overset{i}{\dashrightarrow} p'$ by PQ3. We have $(q'{\otimes}d', p') \in \mathcal{R}$ since $q' \sqsubseteq_{\mathrm{e}} p' \mathbin{/\!\!/} d'$.

P6 $q \overset{i}{\dashrightarrow} q'$ and $d \overset{i}{\dashrightarrow} d'$: Since $q \sqsubseteq_{\mathrm{e}} p \mathbin{/\!\!/} d$, we conclude $p \mathbin{/\!\!/} d \overset{i}{\dashrightarrow}_{/\!\!/} p' \mathbin{/\!\!/} d''$ for some p', d'' with $q' \sqsubseteq_{\mathrm{e}} p' \mathbin{/\!\!/} d''$. This can be due to PQ3 or PQ4; in both cases we have $p \overset{i}{\dashrightarrow} p'$. Due to may-determinism, $d'' = d'$ and, since $q' \sqsubseteq_{\mathrm{e}} p' \mathbin{/\!\!/} d'$, we have $(q'{\otimes}d', p') \in \mathcal{R}$.

R6 $q \otimes d \overset{o}{\dashrightarrow}_{\otimes}$ and $o \in O_P = O_{Q \otimes D}$: The proof proceeds analogous to the one of R5.

"⇐": We show that $\mathcal{R} := \{(q, p \mathbin{/\!\!/} d) \in Q \times (P \mathbin{/\!\!/} D) \mid q \otimes d \sqsubseteq_{\mathrm{e}} p \text{ or } p \mathbin{/\!\!/} d \in U_{P /\!\!/ D}\}$ is an error-aware modal refinement relation. It suffices to consider some $(q, p \mathbin{/\!\!/} d) \in \mathcal{R}$ with $p \mathbin{/\!\!/} d \notin U_{P /\!\!/ D}$.

R1 $q \in E_Q$ implies (by Def. 5) $q \otimes d \in E_{Q \otimes D}$ iff (by $q \otimes d \sqsubseteq_{\mathrm{e}} p$) $p \in E_P$ iff (by Def. 38) $p \mathbin{/\!\!/} d \in E_{P \oslash D}$. For the reverse direction, it remains to show that the first implication can be reversed, i.e., that $d \notin E_D$. By $q \otimes d \in E_{Q \otimes D}$ and $q \otimes d \sqsubseteq_{\mathrm{e}} p$, we have $p \in E_P \cup U_P$. Hence, $p \mathbin{/\!\!/} d \notin U_{P /\!\!/ D}$ implies and $d \notin E_D$.

R2 By Def. 38, $p \mathbin{/\!\!/} d \notin U_{P /\!\!/ D}$ implies $p \notin U_P$ and $\langle p, d\rangle \notin E_P \times E_D$. There are two cases:

1. $p \in E_P$ and $d \notin E_D$: $q \otimes d \sqsubseteq_{\mathrm{e}} p$ implies $q \otimes d \in E_{Q\otimes D}$, hence, $q \in E_Q$.
2. $p \notin E_P$: $q \otimes d \sqsubseteq_{\mathrm{e}} p$ implies $q \otimes d \notin E_{Q\otimes D} \cup U_{Q\otimes D}$.

In both cases we conclude $q \notin U_Q$.

R3 $p \mathbin{/\!\!/} d \xrightarrow{i}_{/\!\!/} R' \subseteq P' \times \{d'\}$ for $i \in I_{P/\!\!/D}$, where $(p,d) \xrightarrow{i}_{\oslash} P' \times \{d'\}$ is due to one of the rules PQ1 or PQ2, and R' consists of the consistent states of $P' \times \{d'\}$. In the following, we use $A_P = A_Q$ throughout.

PQ1 $p \xrightarrow{i} P'$, $d = d'$ and $i \notin A_D$: By $q \otimes d \sqsubseteq_{\mathrm{e}} p$, we have a transition $q \otimes d \xrightarrow{i}_{\otimes} Q' \times \{d''\}$ for some Q', d'' with $\forall q' {\in} Q'.\, \exists p' {\in} P'.\, q' \otimes d'' \sqsubseteq_{\mathrm{e}} p'$. Since $i \notin A_D$, this transition can only be due to Rule P1, hence, $q \xrightarrow{i} Q'$ and $d'' = d$. By Lem. 40, $q' \otimes d \sqsubseteq_{\mathrm{e}} p'$ implies that $p' \mathbin{/\!\!/} d \notin \mathcal{F}_{\oslash}$, hence, $p' \mathbin{/\!\!/} d \in R'$.

PQ2 $p \xrightarrow{i} P'$ and $d \xrightarrow{i} d'$: By $q \otimes d \sqsubseteq_{\mathrm{e}} p$, we get $q \otimes d \xrightarrow{i}_{\otimes} Q' \times \{d'\}$ for some Q' such that $\forall q' {\in} Q'.\, \exists p' {\in} P'.\, q' \otimes d' \sqsubseteq_{\mathrm{e}} p'$. The transition must result from P3, and the rest of the proof is as in PQ1.

R4 $p \mathbin{/\!\!/} d \xrightarrow{o}_{/\!\!/} R'$ with $o \in O_{P/\!\!/D} = O_P \setminus O_D$: The same arguments as for R3 apply.

R5 $q \overset{i}{\dashrightarrow} q'$ for $i \in I_Q$:

1. $i \notin A_D$: By P4, we have $(q,d) \overset{i}{\dashrightarrow}_{\otimes} (q',d)$. There is a transition $p \overset{i}{\dashrightarrow} p'$ for some p' with $q' \otimes d \sqsubseteq_{\mathrm{e}} p'$, because of $q \otimes d \sqsubseteq_{\mathrm{e}} p$. By PQ3, we have $(p,d) \overset{i}{\dashrightarrow}_{\oslash} (p',d)$, and Lem. 40 implies the consistency of $p' \mathbin{/\!\!/} d$, hence $p \mathbin{/\!\!/} d \overset{i}{\dashrightarrow}_{/\!\!/} p' \mathbin{/\!\!/} d$.
2. $i \in A_D$ and $d \overset{i}{\not\dashrightarrow}$: Due to $d \overset{i}{\not\longrightarrow}_D$ and QF2, we have $p \overset{i}{\not\longrightarrow}_P$. Hence, PQ5 yields $p \mathbin{/\!\!/} d \overset{i}{\dashrightarrow} \top$, and $\langle q', \top \rangle \in \mathcal{R}$ is trivial.
3. $i \in A_D$ and $d \overset{i}{\dashrightarrow} d'$: By P6, a transition $(q,d) \overset{i}{\dashrightarrow}_{\otimes} (q',d')$ exists. The proof proceeds as for Case 1, except for using PQ4 instead of PQ3.

R6 $q \overset{o}{\dashrightarrow} q'$ for $o \in O_Q$:

1. $o \in A_D$, $d \overset{o}{\dashrightarrow} d'$ for some d': By P6, we have $(q,d) \overset{o}{\dashrightarrow}_{\otimes} (q',d')$ and, by $q \otimes d \sqsubseteq_{\mathrm{e}} p$, we obtain $p \overset{o}{\dashrightarrow} p'$ for some p' with $q' \otimes d' \sqsubseteq_{\mathrm{e}} p'$. Applying PQ4, we get $(p,d) \overset{o}{\dashrightarrow}_{\oslash} (p',d')$. Lem. 40 implies the consistency of $p' \mathbin{/\!\!/} d'$, hence, $p \mathbin{/\!\!/} d \overset{o}{\dashrightarrow}_{/\!\!/} p' \mathbin{/\!\!/} d'$.
2. $o \in A_D$, $d \overset{o}{\not\dashrightarrow}$: Analogous to case R5(2).
3. $o \notin A_D$: $q \otimes d \overset{o}{\dashrightarrow}_{\otimes} q' \otimes d$ by P4. Due to $q \otimes d \sqsubseteq_{\mathrm{e}} p$, there is a $p \overset{o}{\dashrightarrow} p'$ for some p' with $q' \otimes d \sqsubseteq_{\mathrm{e}} p'$. The rest follows as in the proof of Case 1, applying PQ3 instead of PQ4. □

From this theorem we may also conclude that $/\!\!/$ is monotonic wrt. $\sqsubseteq_{\mathrm{e}}$ in the left argument and antitonic wrt. the right argument.

Lemma 42 (Monotonicity to the Left of $/\!\!/$ wrt. $\sqsubseteq_{\mathrm{e}}$) *Let P_1, P_2, D be EMIAs with $P_1 \sqsubseteq_{\mathrm{e}} P_2$. If P_1, P_2 are τ-free and D is τ-free and may-deterministic, then $P_1 \mathbin{/\!\!/} D \sqsubseteq_{\mathrm{e}} P_2 \mathbin{/\!\!/} D$.*

Proof By Thm. 41, $X \sqsubseteq_{\mathrm{e}} P_1 \mathbin{/\!\!/} D$ implies $X \otimes D \sqsubseteq_{\mathrm{e}} P_1$. Applying the assumption $P_1 \sqsubseteq_{\mathrm{e}} P_2$ and transitivity of $\sqsubseteq_{\mathrm{e}}$, we conclude that $X \otimes D \sqsubseteq_{\mathrm{e}} P_2$. Thm. 41 implies $X \sqsubseteq_{\mathrm{e}} P_2 \mathbin{/\!\!/} D$. By reflexivity of $\sqsubseteq_{\mathrm{e}}$ we may substitute $P_1 \mathbin{/\!\!/} D$ for X. □

Lemma 43 (Antitonicity to the Right of $/\!\!/$ wrt. $\sqsubseteq_{\mathrm{e}}$) *Let P, D_1, D_2 be EMIAs with $D_1 \sqsubseteq_{\mathrm{e}} D_2$. If P is τ-free and D_1, D_2 are τ-free and may-deterministic, then $P \mathbin{/\!\!/} D_1 \sqsupseteq_{\mathrm{e}} P \mathbin{/\!\!/} D_2$.*

Proof By Thm. 41, $X \sqsubseteq_{\mathrm{e}} P \mathbin{/\!\!/} D_2$ implies $X \otimes D_2 \sqsubseteq_{\mathrm{e}} P$. Our assumption $D_1 \sqsubseteq_{\mathrm{e}} D_2$ and compositionality imply $X \otimes D_1 \sqsubseteq_{\mathrm{e}} X \otimes D_2$. Transitivity of $\sqsubseteq_{\mathrm{e}}$ yields $X \otimes D_1 \sqsubseteq_{\mathrm{e}} P$ which, by Thm. 41, implies $X \sqsubseteq_{\mathrm{e}} P \mathbin{/\!\!/} D_1$. By reflexivity of $\sqsubseteq_{\mathrm{e}}$, we may substitute $P \mathbin{/\!\!/} D_2$ for X. □

As a direct consequence of Thm. 32, Thm. 34 and Lem. 43, we get a De Morgan-like law for the quotient:

Corollary 44 (De Morgan-like Law for $/\!\!/$) *Let P, Q and R be EMIAs, then $P \mathbin{/\!\!/} (Q \vee R) \sqsubseteq_{\mathrm{e}} (P \mathbin{/\!\!/} Q) \wedge (P \mathbin{/\!\!/} R)$.*

5.2 Hiding, Restriction and Alphabet Extension

We now introduce operators for scoping actions, namely *hiding* [23] and *restriction* [31], as is usual in process algebra. In our setting, outputs are under the control of the system; when disconnected, they are still performed but the signal is no longer sent to the outside, i.e., the action is internal. In contrast, inputs are only performed because of an outside stimulus. Disconnecting an input rather blocks it and, therefore, we introduce a restriction operator for inputs. The same idea is used in the IA-setting of [12], but hiding and restriction are combined into a single operation.

Definition 45 (Hiding) Let $P = (S_P, I_P, O_P, \longrightarrow_P, \dashrightarrow_P, S^0_P, E_P, U_P)$ be an EMIA and L a set of actions with $L \cap I_P = \emptyset$. We define P *hiding* L as the EMIA $P \,/\, L := (S_P, I_P, O \setminus L, \longrightarrow_{P/L}, \dashrightarrow_{P/L}, S^0_P, E_P, U_P)$, where all transition labels $o \in L$ are replaced by τ.

Definition 46 (Restriction) Let $P = (S_P, I_P, O_P, \longrightarrow_P, \dashrightarrow_P, S^0_P, E_P, U_P)$ be an EMIA and L a set of actions with $L \cap O_P = \emptyset$. We define P *restricted in* L as the EMIA $P \setminus L := (S_P, I_P \setminus L, O_P, \longrightarrow_{P\setminus L}, \dashrightarrow_{P\setminus L}, S^0_P, E_P, U_P)$, where all transitions with a label contained in L are removed.

Observe that hiding and restriction yield well-defined EMIAs.

Lemma 47 (Weak Must-Transitions under Hiding) *Let P be a MIA, $L \cap I_P = \emptyset$ and $o \in L \cap O_P$. If $p \overset{o}{\Longrightarrow}_P P'$, then $p \overset{\epsilon}{\Longrightarrow}_{P/L} P'$.*

Proof By induction on the definition of $p \overset{o}{\Longrightarrow}_P P'$. If $p \overset{o}{\Longrightarrow}_P P'$ is due to WT3 of Def. 2, then the claim is obvious. Otherwise, $p \overset{o}{\Longrightarrow}_P P'$ is due to some $p \overset{\tau}{\longrightarrow}_P \bar{P}$ and $\bar{P} \overset{o}{\Longrightarrow}_P P'$ according to WT2. By induction hypothesis, we have $\bar{p} \overset{\epsilon}{\Longrightarrow}_{P/L} P_{\bar{p}}$ for each $\bar{p} \in \bar{P}$ and $P' = \bigcup_{\bar{p} \in \bar{P}} P_{\bar{p}}$. By WT2, we obtain $p \overset{\epsilon}{\Longrightarrow}_{P/L} P'$. □

As desired, EMIA-refinement is a precongruence wrt. hiding and restriction:

Proposition 48 *Let P, Q be EMIAs with equal alphabets and $P \sqsubseteq_{\mathrm{e}} Q$.*

1. *$P \,/\, L \sqsubseteq_{\mathrm{e}} Q \,/\, L$ for any set L of actions with $L \cap I = \emptyset$.*
2. *$P \setminus L \sqsubseteq_{\mathrm{e}} Q \setminus L$ for any set L of actions with $L \cap O = \emptyset$.*

Proof Since $P \sqsubseteq_{\mathrm{e}} Q$, there is an EMIA-refinement relation $\mathcal{R}$ with $\langle p, q \rangle \in \mathcal{R}$. We show that $\mathcal{R}$ is also an EMIA-refinement relation for $P \,/\, L \sqsubseteq_{\mathrm{e}} Q \,/\, L$ and $P \setminus L \sqsubseteq_{\mathrm{e}} Q \setminus L$. The only interesting case concerns hiding and Rule R4 of Def. 3, i.e., $q \overset{\tau}{\longrightarrow}_{Q/L} Q'$ due to $q \overset{o}{\longrightarrow}_Q Q'$ for $o \in O \cap L$. The latter is matched by a transition $p \overset{o}{\Longrightarrow}_P P'$ with $\forall p' \in P' \, \exists q' \in Q'. \, (p', q') \in \mathcal{R}$. By Lem. 47, this yields $p \overset{\epsilon}{\Longrightarrow}_{P/L} P'$. □

Originally, IA employs a parallel composition with immediate hiding [14]. This can easily be expressed by combining our parallel composition and the hiding operator, such that $P \mid Q = (P \parallel Q) \,/\, S$, where S is the set of synchronising actions. However, the immediate hiding weakens the associativity of this composition operation. We omit the details here, because they are presented in [7] for MIA and may directly be adopted to EMIA.

We close this section with a remark on alphabet extension. Conjunction, disjunction and refinement are defined for EMIAs with equal alphabets. For perspective-based specification, it is of interest to consider EMIAs with different alphabets [7]. Following the lines of MI and MIA, the operations on EMIAs can be lifted to different alphabets by extending the alphabets of the operands by their mutually foreign actions. When a specification's alphabet is extended, the least possible assumptions should be made on a new action a, while the same specification wrt. known actions should hold before and after a. This can be achieved by adding an optional a-loop to each state. For output actions this is straightforward, but the exact meaning of optional input transitions depends

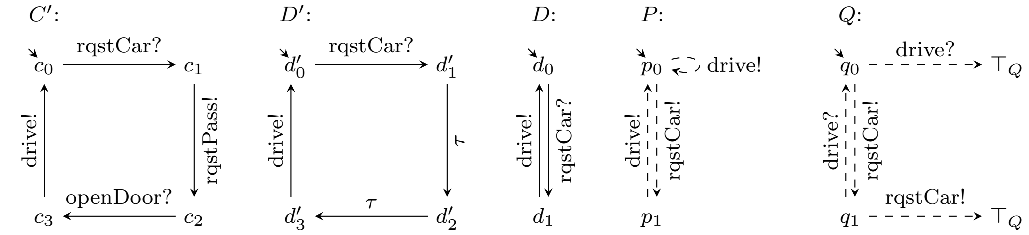

Fig. 10 Synthesis of a user interface Q from a given component D and a global specification P, where $A_{C'} := \{\text{openDoor?}, \text{rqstCar?}\}/\{\text{drive!}, \text{rqstPass!}\}$, $A_{D'} = A_D := \{\text{rqstCar?}\}/\{\text{drive!}\}$, $A_P := \emptyset/\{\text{rqstCar!}, \text{drive!}\}$ and $A_Q := \{\text{drive?}\}/\{\text{rqstCar!}\}$.

on the desired composition concept (cf. Sec. 1, Issue D). Therefore, a separate alphabet extension operator has to be defined for unanimous, broadcast and error-sensitive parallel composition. Alternatively, a mixed extension combining different composition concepts for each state and each new action is also possible. Besides this, there is nothing surprising to expect from alphabet extension, and we leave out the formal definition here for brevity.

5.3 Process Algebraic Operators and Quotient under Galois Insertion

The Galois insertion between MIA and EMIA translates the process algebraic operators presented above from EMIA to MIA. For hiding and restriction this relation is trivial; therefore, we only present the translation of the quotient.

Lemma 49 (Abstraction Respects Quotienting) *If $Q \sqsubseteq_e P \mathbin{/\!/} D$ for EMIAs P, Q and D, then $\alpha(Q) \sqsubseteq_e \alpha(P) \mathbin{/\!/} \alpha(D)$.*

Proof We have $Q \sqsubseteq_e P \mathbin{/\!/} D \overset{\text{Def. 41}}{\iff} Q \otimes D \sqsubseteq_e P \overset{\text{Lem. 12}}{\implies} \alpha(Q \otimes D) \sqsubseteq_e \alpha(P) \overset{\text{Lem. 15}}{\iff} \alpha(Q) \parallel \alpha(D) \sqsubseteq_e \alpha(P) \overset{\text{Def. 41}}{\iff} \alpha(Q) \sqsubseteq_e \alpha(P) \mathbin{/\!/} \alpha(D)$. □

Substituting $P \mathbin{/\!/} D$ for Q in Lemma 49 yields $\alpha(P \mathbin{/\!/} D) \sqsubseteq_e \alpha(P) \mathbin{/\!/} \alpha(D)$.

Lemma 50 (MIA Quotient) *Let P and D be MIAs. We have $P \mathbin{/\!/} D \sqsupseteq\sqsubseteq_e \alpha(\gamma(P) \mathbin{/\!/} \gamma(D))$, i.e., MIA is closed under quotienting.*

Proof By Lem. 49, Thm. 18, Def. 16 and extensivity of α (again Thm. 18), we get the following chain of inequalities: $\alpha(\gamma(P) \mathbin{/\!/} \gamma(D)) \sqsubseteq_e \alpha(\gamma(P)) \mathbin{/\!/} \alpha(\gamma(D)) \sqsupseteq\sqsubseteq_e P \mathbin{/\!/} D \sqsupseteq\sqsubseteq_e \gamma(P) \mathbin{/\!/} \gamma(D) \sqsubseteq_e \alpha(\gamma(P) \mathbin{/\!/} \gamma(D))$. By transitivity, all inequalities are equalities. □

5.4 Example

We reconsider the corrected driving assistant system of Sec. 3.3. For illustration purposes we simplified the user interface U presented in Fig. 6 as much as possible, making it impractical as it may only be used once. Now, we demonstrate how to employ quotienting in order to synthesise a useful specification of a user interface.

Starting with the corrected car C', which we repeat in Fig. 10, the actions rqstPass and openDoor are internal to the communication between the car and the garage and are invisible to the user interface. Hence, from the user interface's perspective, the car looks like specification $D' \sqsupseteq\sqsubseteq_e D$, which is obtained from C' by hiding actions rqstPass! and openDoor?. We consider specification D as an already given implementation, which we want to reuse in order to synthesise a specification of the user interface. To this end, the composition $D \otimes U$ of the car D and its user interface U must satisfy the global specification P which requires that, after some request, the car may drive and new requests are blocked until the drive is completed. A specification Q of the user interface may now be synthesised from P and D by quotienting, i.e., $Q := P \mathbin{/\!\!/} D$. Note, that drive? is an input action in Q. The two transitions leading to universal states (drive? in q_0 and rqstCar! in q_1) are only due to the maximality of Q. They disappear in the parallel composition with D. It is easy to see that the defining inequality $Q \otimes D \sqsubseteq_{\mathrm{e}} P$ is satisfied. The example also shows that, in general, we do not have equality of $(P \mathbin{/\!\!/} D) \otimes D$ and P.

6 Conclusions and Future Work

Our interface theory EMIA is a uniformly integrated specification framework that is applicable at different levels of abstraction, e.g., component-based design and product line specification. EMIA bridges the gaps between MTS [28], interface theories [2,6,7,8,10,14,15,26,30,32] and assembly theories [22]. It is based on a concept of *error-awareness*, whereby EMIA's refinement preorder reflects *and* preserves fatal error states. While recent interface theories [7,32] considered the problem of how to enforce required behaviour, our finer-grained error semantics also solves the dual and previously open problem of how to forbid unwanted behaviour.

We proved that EMIA is related to the IA-based interface theory MIA [7] via a Galois insertion, rendering MIA into an abstraction of EMIA. In the abstract theory, errors may be considered as models of unknown behaviour for which no guarantees can be made, while in EMIA errors model unwanted behaviour for which we know that it must not be implemented. This difference between EMIA and related interface theories can be captured in a more concise way when considering error states axiomatically. In related theories [7,32], an error state e satisfies the laws $e \parallel q = e$, meaning that a composed system is in an erroneous state if a component is, and $e \sqsubseteq p \;\Rightarrow\; p = e$, meaning that an error cannot be introduced when refining an ordinary state. In EMIA, the additional law $p \sqsubseteq e \;\Rightarrow\; p = e$ is satisfied, i.e., refinement cannot redefine an erroneous situation to be non-erroneous.

Regarding future work we intend to add alphabet extension and extend quotienting to nondeterministic divisors. Furthermore, we wish to capture the differences and commonalities of different interface theories via axiomatisations. We also plan to implement EMIA in a formal methods tool, e.g., Mica [9], the MIO-Workbench [2] or MoTraS [24], and to further develop EMIA as a behavioural type theory for the Go Programming Language [20]. Such tools would enable us to evaluate EMIA on larger, more realistic examples, e.g., the docking system studied in the context of IA in [17].

Acknowledgements We are grateful to Ferenc Bujtor, Rance Cleaveland, Rolf Hennicker, Kim G. Larsen, Michael Mendler, Thomas Streicher, Walter Vogler, Diedrich Wolter and the anonymous reviewers for their helpful suggestions.

References

1. S. S. Bauer, A. David, R. Hennicker, K. G. Larsen, A. Legay, U. Nyman, and A. Wasowski. Moving from specifications to contracts in component-based design. In *Fundamental Approaches to Software Engineering (FASE)*, volume 7212 of *LNCS*, pages 43–58. Springer, 2012.
2. S. S. Bauer, P. Mayer, A. Schroeder, and R. Hennicker. On weak modal compatibility, refinement, and the MIO Workbench. In *Tools and Algorithms for the Construction and Analysis of Systems (TACAS)*, volume 6015 of *LNCS*, pages 175–189. Springer, 2010.
3. N. Beneš, I. Černa, and J. Křetiínský. Disjunctive modal transition systems and generalized LTL model checking. Technical Report FIMU-RS-2010-12, Faculty of Informatics, Masaryk University Brno, 2010.

4. N. Beneš, B. Delahaye, U. Fahrenberg, J. Křetínský, and A. Legay. Hennessy-Milner logic with greatest fixed points as a complete behavioural specification theory. In *Concurrency Theory (CONCUR)*, volume 8052 of *LNCS*, pages 76–90. Springer, 2013.
5. D. Beyer, A. Chakrabarti, T. A. Henzinger, and Sanjit A. Seshia. An application of web-service interfaces. In *Intl. Conf. on Web Services (ICWS)*, pages 831–838. IEEE, 2007.
6. F. Bujtor, S. Fendrich, G. Lüttgen, and W. Vogler. Nondeterministic modal interfaces. In *Theory and Practice of Computer Science (SOFSEM)*, volume 8939 of *LNCS*, pages 152–163. Springer, 2015.
7. F. Bujtor, S. Fendrich, G. Lüttgen, and W. Vogler. Nondeterministic modal interfaces. *Theoretical Computer Science*, 642:24–53, 2016.
8. F. Bujtor and W. Vogler. Error-pruning in interface automata. In *Theory and Practice of Computer Science (SOFSEM)*, volume 8327 of *LNCS*, pages 162–173. Springer, 2014.
9. B. Caillaud. Mica: A modal interface compositional analysis library, 2011. Online, last accessed 27 Jan. 2017, `http://www.irisa.fr/s4/tools/mica/`.
10. T. Chen, C. Chilton, B. Jonsson, and M. Z. Kwiatkowska. A compositional specification theory for component behaviours. In *Programming Languages and Systems (ESOP)*, volume 7211 of *LNCS*, pages 148–168. Springer, 2012.
11. C. Chilton. *An Algebraic Theory of Componentised Interaction.* PhD thesis, Oxford University, 2013.
12. C. Chilton, B. Jonsson, and M. Kwiatkowska. An algebraic theory of interface automata. Technical Report RR-13-02, Oxford University, 2013.
13. P. Cousot and R. Cousot. Abstract interpretation: A unified lattice model for static analysis of programs by construction or approximation of fixpoints. In *Principles of Programming Languages (POPL)*, pages 238–252. ACM, 1977.
14. L. de Alfaro and T. A. Henzinger. Interface automata. In *Foundations of Software Engineering (FSE)*, pages 109–120. ACM, 2001.
15. L. de Alfaro and T. A. Henzinger. Interface-based design. In *Engineering Theories of Software-Intensive Systems*, volume 195 of *NATO Science*, pages 83–104. Springer, 2005.
16. David L. Dill. *Trace Theory for Automatic Hierarchical Verification of Speed-Independent Circuits.* MIT-Press, 1989.
17. M. Emmi, D. Giannakopoulou, and C. S. Păsăreanu. Assume-guarantee verification for interface automata. In *Formal Methods (FM)*, volume 5014 of *LNCS*, pages 116–131. Springer, 2008.
18. S. Fendrich. *Modal Interface Theories for Specifying Component-based Systems.* PhD thesis, Bamberg University, 2017.
19. S. Fendrich and G. Lüttgen. A generalised theory of interface automata, component compatibility and error. In *Integrated Formal Methods (iFM)*, volume 9681 of *LNCS*, pages 160–175. Springer, 2016.
20. J. Gareis. Prototypical Integration of the Modal Interface Automata Theory in Google Go. Master's thesis, Bamberg University, 2015.
21. G. Goessler and J.-B. Raclet. Modal contracts for component-based design. In *Software Engineering and Formal Methods (SEFM)*, pages 295–303. IEEE, 2009.
22. R. Hennicker and A. Knapp. Moving from interface theories to assembly theories. *Acta Informatica*, 52(2-3):235–268, 2015.
23. C. A. R. Hoare. *Communicating Sequential Processes.* Prentice-Hall, 1985.
24. J. Křetínský and S. Sickert. MoTraS: A tool for modal transition systems and their extensions. In *Automated Technology for Verification and Analysis (ATVA)*, volume 8172 of *LNCS*, pages 487–491. Springer, 2013.
25. K. G. Larsen. Modal specifications. In *Automatic Verification Methods for Finite State Systems*, volume 407 of *LNCS*, pages 232–246. Springer, 1989.
26. K. G. Larsen, U. Nyman, and A. Wasowski. Modal I/O automata for interface and product line theories. In *Programming Languages and Systems (ESOP)*, volume 4421 of *LNCS*, pages 64–79. Springer, 2007.
27. K. G. Larsen, B. Steffen, and C. Weise. A constraint oriented proof methodology based on modal transition systems. In *Tools and Algorithms for the Construction and Analysis of Systems (TACAS)*, volume 1019 of *LNCS*, pages 17–40. Springer, 1995.
28. K. G. Larsen and L. Xinxin. Equation solving using modal transition systems. In *Logic in Computer Scienc (LICS)*, pages 108–117. IEEE, 1990.
29. M. Lohstroh and E. A. Lee. An interface theory for the Internet of Things. In *Software Engineering and Formal Methods (SEFM)*, volume 9276 of *LNCS*, pages 20–34. Springer, 2015.
30. G. Lüttgen, W. Vogler, and S. Fendrich. Richer interface automata with optimistic and pessimistic compatibility. *Acta Informatica*, 52(4-5):305–336, 2015.
31. R. Milner. *Communication and Concurrency.* Prentice Hall, 1989.
32. J.-B. Raclet, E. Badouel, A. Benveniste, B. Caillaud, A. Legay, and R. Passerone. A modal interface theory for component-based design. *Fund. Inform.*, 108(1-2):119–149, 2011.