BAMBERGER BEITRÄGE

ZUR WIRTSCHAFTSINFORMATIK UND ANGEWANDTEN INFORMATIK

ISSN 0937-3349

Nr. 98

A Generalised Theory of Interface Automata, Component Compatibility and Error

Sascha Fendrich   Gerald Lüttgen

March 2016

FAKULTÄT WIRTSCHAFTSINFORMATIK UND ANGEWANDTE INFORMATIK

OTTO-FRIEDRICH-UNIVERSITÄT BAMBERG

A Generalised Theory of Interface Automata, Component Compatibility and Error

Sascha Fendrich Gerald Lüttgen

March 2016

Abstract Interface theories allow systems 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 paper 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.

This technical report is a long version of [13]. Research has been supported by the DFG (German Research Foundation) under grant LU-1748/3-1.

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* [10, 11, 2, 5, 6, 8, 18, 21, 22], which may serve as specification theories for component-based design [11, 2, 7, 16], software product lines [18], web services [4] and the Internet of Things [20]. Interface theories may also be employed as contract languages or behavioural type theories when transitioning from software design to implementation [1, 14].

Many interface theories [2, 5, 18, 21, 22] extend de Alfaro and Henzinger's *Interface Automata* (IA) [10, 11] and Larsen's *Modal Transition Systems* (MTS) [17, 19]. In order to express compatibility assumptions of components on the communication behaviour of their environment, IA divides the action alphabet of an interface into input ('?'), output ('!') and an internal action τ. A *communication mismatch*, or error, arises between parallelly composed components P and Q, if P may 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.

Unfortunately, interface theories combining IA and MTS have several issues that impact their practical use. *Issue (A):* Forbidden inputs are preserved by the resp. refinement preorder but are widely ignored by parallel composition, such 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 does not guarantee their overall compatibility when being considered as a multi-component assembly, and vice versa, even if parallel composition is associative. To address this, Hennicker and Knapp [15] 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 [18]. *Issue (D):* MTS and MTS-based interface theories have some subtle differences wrt. 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.

This paper 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. We illustrate this in Sec. 2 by an example wrt. Issue (A). In Sec. 3 we present our interface theory *Error-aware Modal Interface Automata* (EMIA), for which we remedy Issues (A)–(D) by making communication mismatches explicit in form of *fatal error states* and by employing an *error-aware refinement preorder*. In contrast, current interface theories [10, 11, 2, 5, 6, 8, 18, 21, 22] remove such information about the causes and possible resolutions of communication mismatches. As is typical for interface theories, EMIA also includes conjunction and disjunction operators, which enables systems designers to employ both operational and declarative aspects within a specification. In Sec. 4 we show that a Galois insertion [9] renders our refined semantics a conservative extension of the arguably most general interface theory to date, MIA (Modal Interface Automata) [5]. Sec. 5 revisits the example of Sec. 2 in terms of EMIA, and discusses the role of fatal error states in solving Issues (A)–(D). The resulting specification theory tightly integrates Modal Transition Systems, interface theories, and assembly theories and allows systems designers to combine the different composition concepts of these theories within a single interface specification.

2 Motivating Example

In this section we discuss compatibility problems of current interface theories by means of an illustrative example highlighting Issue (A). 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. In Fig. 1 we show specifications G and C of the garage's and the car's interfaces, 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 (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

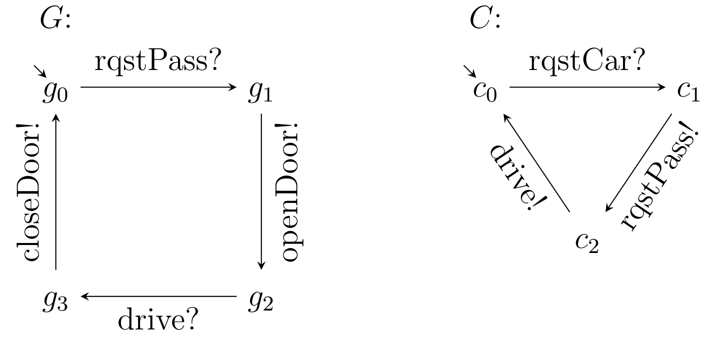

Figure 1: Example of a driving assistant system including a garage G and a car C.

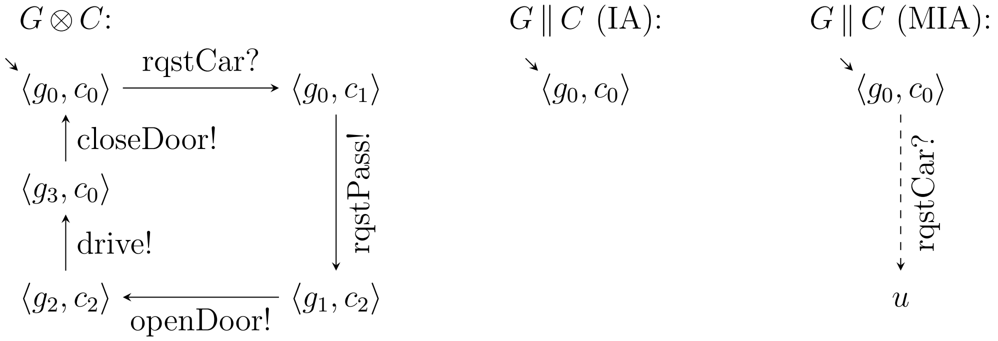
$G \otimes C$:
rqstCar?
$\langle g_0, c_0 \rangle$
$\langle g_0, c_1 \rangle$
closeDoor!
$\langle g_3, c_0 \rangle$
rqstPass!
drive!
$\langle g_2, c_2 \rangle$
$\langle g_1, c_2 \rangle$
openDoor!
$G \parallel C$ (IA):
$\langle g_0, c_0 \rangle$
$G \parallel C$ (MIA):
$\langle g_0, c_0 \rangle$
rqstCar?
u

Figure 2: Parallel product in IA or MIA (left), and parallel composition in IA (middle) and MIA (right) of the components depicted in Fig. 1.

considered illegal. In *pessimistic* theories, e.g., [2, 21], 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., [10, 11, 5, 6, 8, 18, 21, 22], 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. 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. [5]); IA-based refinement [10, 11, 21] 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 u (Fig. 2, right) [5].

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 because it provoked 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 [6] 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 [10, 11, 2, 5, 6, 18, 21, 22] eliminate erroneous behaviour either by regarding it as undefined (pessimistic) or by pruning (optimistic); all errors are treated 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 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. 5, Issue (C)).

3 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, 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 $P := (S_P, I_P, O_P, \longrightarrow_P, \dashrightarrow_P, S^0_P, D_P)$, 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, $S^0_P \subseteq S_P$ is the set of initial states, and $D_P \subseteq S_P$ is the set of fatal error states. We also adopt* syntactic consistency *from MTS, i.e., for all $\alpha \in A_P \cup \{\tau\}$ and $p \xrightarrow{\alpha} P'$, we have $\forall p' \in P.\, p \overset{\alpha}{\dashrightarrow} p'$.*

Our definition of weak transitions is adopted from the one in MIA [5]:

Definition 2 (Weak Transition Relations)**.** *Let P be an EMIA. We define* weak *must- and may-transition relations, $\Longrightarrow$ and $=\!=\!\!\Rightarrow$ 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. $p \overset{\epsilon}{=\!=\!\!\Rightarrow} p$,
WT5. $p \overset{\epsilon}{=\!=\!\!\Rightarrow} p'' \overset{\tau}{\dashrightarrow} p'$ *implies* $p \overset{\epsilon}{=\!=\!\!\Rightarrow} p'$,
WT6. $p \overset{\epsilon}{=\!=\!\!\Rightarrow} p'' \overset{\alpha}{\dashrightarrow} p''' \overset{\epsilon}{=\!=\!\!\Rightarrow} p'$ *implies* $p \overset{\alpha}{=\!=\!\!\Rightarrow} p'$.

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

Our *error-aware modal refinement preorder* $\sqsubseteq_{\text{EA}}$ corresponds to standard modal refinement from MTS [17, 19] but reflects *and* preserves fatal error states. Intuitively, $P \sqsubseteq_{\text{EA}} Q$ 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 [19], 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 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. If τ-loops are involved, this may result in an infinite unfolding—even in case of finite DMTS—because a different implementation may be chosen in each iteration of the loop.

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 (EA-refinement) if, for all $\langle p, q\rangle \in \mathcal{R} \setminus (D_P \times D_Q)$, the following conditions hold:*

R1. $p \notin D_P$ *and* $q \notin D_Q$,
R2. $q \xrightarrow{i} Q'$ *implies* $\exists P'.\, p \overset{i}{\to}\overset{\epsilon}{\Rightarrow} P'$ *and* $\forall p' \in P'\ \exists q' \in Q'.\, \langle p', q'\rangle \in \mathcal{R}$,
R3. $q \xrightarrow{\omega} Q'$ *implies* $\exists P'.\, p \overset{\omega}{\Longrightarrow} P'$ *and* $\forall p' \in P'\ \exists q' \in Q'.\, \langle p', q'\rangle \in \mathcal{R}$,
R4. $p \overset{i}{\dashrightarrow} p'$ *implies* $\exists q'.\, q \overset{i}{\dashrightarrow}\overset{\epsilon}{\Rrightarrow} q'$ *and* $\langle p', q'\rangle \in \mathcal{R}$,
R5. $p \overset{\omega}{\dashrightarrow} p'$ *implies* $\exists q'.\, q \overset{\omega}{\Rrightarrow} q'$ *and* $\langle p', q'\rangle \in \mathcal{R}$.

We write $p \sqsubseteq_{\text{EA}} q$ if there is an EA-refinement $\mathcal{R}$ with $\langle p, q\rangle \in \mathcal{R}$, and $P \sqsubseteq_{\text{EA}} Q$ if, for each $p \in S^0_P$, there is a $q \in S^0_Q$ with $p \sqsubseteq_{\text{EA}} q$. If $p \sqsubseteq_{\text{EA}} q$ and $q \sqsubseteq_{\text{EA}} p$, we employ the symbol $p \sqsupseteq\sqsubseteq_{\text{EA}} q$, and similar for EMIAs P, Q.

The refinement relation $\sqsubseteq_{\text{EA}}$ is reflexive and transitive and, hence, a preorder. Moreover, we have $p \in D_P$ iff $q \in D_Q$ for all $\langle p, q\rangle \in \mathcal{R}$ due to R1. Optional input-transitions, which may be refined to required or forbidden behaviour, are expressed as a disjunctive must-transition containing a fatal error state in its set of target states. For example, optional $a?$-transitions from a state p_0 to states p_1 and p_2 are modelled as $p_0 \xrightarrow{a?} \{p_1, p_2, p_3\}$ for some fatal error state $p_3 \in D_P$.

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 [22] and MIA [5]. We leave out MIA's separate hiding due to space constraints.

Definition 4 (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 *$P \,\|\, Q$ is defined by $S_{P\,\|\,Q} := S_P \times S_Q$, $I_{P\,\|\,Q} := (I_P \cup I_Q) \setminus O_{P\,\|\,Q}$, $O_{P\,\|\,Q} := O_P \cup O_Q$, $S^0_{P\,\|\,Q} := S^0_P \times S^0_Q$, $D_{P\,\|\,Q} := (D_P \times S_Q) \cup (S_P \times D_Q)$, 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 $p \,\|\, q$ *for* $\langle p, q\rangle$.

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 a situation is modelled with the help of an a?-must-transition from q to a target set Q' that includes some fatal error state $q' \in D_Q$, as explained above. Parallel composition is associative and commutative. Further, $\sqsubseteq_{\mathrm{EA}}$ is a precongruence wrt. $\|$:

Proposition 5 (Compositionality)**.** *If P_1, P_2, Q are EMIAs s.t. $P_1 \sqsubseteq_{\mathrm{EA}} P_2$ and P_2, Q are composable, then P_1 and Q are composable and $P_1 \,\|\, Q \sqsubseteq_{\mathrm{EA}} P_2 \,\|\, Q$.*

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 $\mathcal{R} := \{\langle p_1 \,\|\, q, p_2 \,\|\, q\rangle \mid p_1 \sqsubseteq_{\mathrm{EA}} p_2\}$ is an EA-refinement relation. For $\langle p_1 \,\|\, q, p_2 \,\|\, q\rangle \in \mathcal{R}$, we consider the following cases:

R1 $p_1 \,\|\, q \notin D_{P_1 \,\|\, Q}$ iff (by Def. 4) $p_1 \notin D_{P_1}$, $q \notin D_Q$ iff (by $p_1 \sqsubseteq_{\mathrm{EA}} p_2$) $p_2 \notin D_{P_2}$, $q \notin D_Q$ iff (by Def. 4) $p_2 \,\|\, q \notin D_{P_2 \,\|\, Q}$.

R2 Let $p_1 \,\|\, q \xrightarrow{i} R$ due to one of P1, P2 or P3:

P1 $R = P_1' \times \{q\}$ for some transition $p_1 \xrightarrow{i} P_1'$. For each $p_1' \in P_1'$, there is, due to syntactic consistency, a $p_1 \overset{i}{\dashrightarrow} p_1'$ and, by $p_1 \sqsubseteq_{\mathrm{EA}} p_2$, a $p_2 \overset{i}{\dashrightarrow}\overset{\epsilon}{\Rightarrow} p_2'$ s.t. $p_1' \sqsubseteq_{\mathrm{EA}} p_2'$. Thus, we have $\langle p_1' \,\|\, q, p_2' \,\|\, q\rangle \in \mathcal{R}$, and P4 implies $p_2 \,\|\, q \overset{i}{\dashrightarrow}\overset{\epsilon}{\Rightarrow} p_2' \,\|\, q$.

P2 $R = \{p_1\} \times Q'$ for some $q \xrightarrow{i} Q'$. By P2 we have $p_2 \,\|\, q \xrightarrow{i} \{p_2\} \times Q'$, and $p_1 \sqsubseteq_{\mathrm{EA}} p_2$ implies $\langle p_1 \,\|\, q', p_2 \,\|\, q'\rangle \in \mathcal{R}$ for all $q' \in Q'$.

P3 $R = P_1' \times Q'$ due to $p_1 \xrightarrow{i} P_1'$ and $q \xrightarrow{i} Q'$. The argument is similar to that of Case P1, where P4 is replaced by P6.

R3 Analogous to R2.

R4 Let $p_1 \,\|\, q \overset{i}{\dashrightarrow} p_1' \,\|\, q'$ due to one of the rules P4, P5 or P6:

P4 $q' = q$ for a transition $p_1 \overset{i}{\dashrightarrow} p_1'$. By $p_1 \sqsubseteq_{\mathrm{EA}} p_2$, there is a $p_2 \overset{i}{\dashrightarrow}\overset{\epsilon}{\Rightarrow} p_2'$ such that $p_1' \sqsubseteq_{\mathrm{EA}} p_2'$. Thus, we have $\langle p_1' \,\|\, q, p_2' \,\|\, q\rangle \in \mathcal{R}$, and P4 implies $p_2 \,\|\, q \overset{i}{\dashrightarrow}\overset{\epsilon}{\Rightarrow} p_2' \,\|\, q$.

P5 $p_1' = p_1$ for some $q \overset{i}{\dashrightarrow} q'$. By P5 we have $p_2 \,\|\, q \overset{i}{\dashrightarrow} p_2 \,\|\, q'$ and $p_1 \sqsubseteq_{\mathrm{EA}} p_2$ implies $\langle p_1 \,\|\, q', p_2 \,\|\, q'\rangle \in \mathcal{R}$, for all $q' \in Q'$.

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.

R5 Analogous to R4. □

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 6 (Conjunctive Product)**.** *Let P, Q be EMIAs with equal alphabets. The* conjunctive product *of P and Q is $P \mathbin{\&} Q := (S_{P\&Q}, I, O, \longrightarrow_{P\&Q}, \dashrightarrow_{P\&Q}, S^0_{P\&Q}, D_{P\&Q})$ with $S_{P\&Q} := S_P \times S_Q$, $S^0_{P\&Q} := S^0_P \times S^0_Q$, $D_{P\&Q} := D_P \times D_Q$, 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. $\langle p, q\rangle \xrightarrow{\omega} \{\langle p', q'\rangle \mid p' \in P',\ q \overset{\omega}{\Rightarrow} q'\}$ *if* $p \xrightarrow{\omega} P'$ *and* $q \overset{\omega}{\Rightarrow}$,
C4. $\langle p, q\rangle \xrightarrow{\omega} \{\langle p', q'\rangle \mid p \overset{\omega}{\Rightarrow} p',\ q' \in Q'\}$ *if* $p \overset{\omega}{\Rightarrow}$ *and* $q \xrightarrow{\omega} Q'$,
C5. $\langle p, q\rangle \overset{i}{\dashrightarrow} \langle p', q'\rangle$ *if* $p \overset{i}{\dashrightarrow}\overset{\epsilon}{\Rightarrow} p'$ *and* $q \overset{i}{\dashrightarrow}\overset{\epsilon}{\Rightarrow} q'$,
C6. $\langle p, q\rangle \overset{\omega}{\dashrightarrow} \langle p', q'\rangle$ *if* $p \overset{\omega}{\Rightarrow} p'$ *and* $p \overset{\omega}{\Rightarrow} q'$,
C7. $\langle p, q\rangle \overset{\tau}{\dashrightarrow} \langle p', q\rangle$ *if* $p \overset{\tau}{\Rightarrow} p'$,
C8. $\langle p, q\rangle \overset{\tau}{\dashrightarrow} \langle p, q'\rangle$ *if* $q \overset{\tau}{\Rightarrow} q'$.

A state $\langle p, q\rangle$ of $P \mathbin{\&} Q$ 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 *inconsistent* and has to be removed from the candidates, including the removal of all states that require transitions leading to inconsistent states.

Definition 7 (Conjunction)**.** *The set $F \subseteq S_{P\&Q}$ of* logically inconsistent *states is defined as the smallest set satisfying the following rules:*

F1. $\langle p, q\rangle \in (D_P \times (S_Q \setminus D_Q)) \cup ((S_P \setminus D_P) \times D_Q)$
implies $\langle p, q\rangle \in F$,
F2. $\langle p, q\rangle \notin D_{P\,\&\,Q}$, $p \xrightarrow{i}$ *and* $q \overset{i}{\not\dashrightarrow}$ *implies* $\langle p, q\rangle \in F$,
F3. $\langle p, q\rangle \notin D_{P\,\&\,Q}$, $p \overset{i}{\not\dashrightarrow}$ *and* $q \xrightarrow{i}$ *implies* $\langle p, q\rangle \in F$,
F4. $\langle p, q\rangle \notin D_{P\,\&\,Q}$, $p \xrightarrow{\omega}$ *and* $q \overset{\omega}{\not\Rightarrow}$ *implies* $\langle p, q\rangle \in F$,
F5. $\langle p, q\rangle \notin D_{P\,\&\,Q}$, $p \overset{\omega}{\not\Rightarrow}$ *and* $q \xrightarrow{\omega}$ *implies* $\langle p, q\rangle \in F$,
F6. $\langle p, q\rangle \xrightarrow{\alpha} R$ *and* $R \subseteq F$ *implies* $\langle p, q\rangle \in F$.

The conjunction $P \wedge Q$ *is obtained from* $P \,\&\, Q$ *by deleting all states in* F. *This deletes all transitions exiting deleted states and removes all deleted states from targets of must-transitions. If* $S^0_{P\wedge Q} = \emptyset$, *then* P *and* Q *are called inconsistent.*

Fatal states are excluded in Rules F2 through F5 because we do not care about consistency for fatal error states. Note that the states in D and F are different in nature: D-states represent states with possible but unwanted behaviour. F-states represent contradictory specifications that are impossible to implement. It is easy to show the following lemma:

Lemma 8. *Let* P *and* Q *be EMIAs and* $\mathcal{R} \subseteq S_P \times S_Q$ *an EA-refinement relation. For all* $\langle p, q\rangle \in \mathcal{R}$, *we have* $p \in D_P$ *iff* $q \in D_Q$.

Proof. Lemma 8 is a direct consequence of Def. 1. □

In order to prove that conjunction is the greatest lower bound wrt. the refinement preorder $\sqsubseteq_{\mathrm{EA}}$, we need the notion of a witness along the lines of [5]:

Definition 9 (Witness)**.** *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. $p \in D_P$ *iff* $q \in D_Q$,
W2. $p \xrightarrow{o}_P$ *implies* $q \overset{o}{\Rightarrow}_Q$ *or* $q \in D_Q$,
W3. $q \xrightarrow{o}_Q$ *implies* $p \overset{o}{\Rightarrow}_P$ *or* $p \in D_P$,
W4. $p \xrightarrow{i}_P$ *implies* $q \overset{i}{\dashrightarrow}\overset{\epsilon}{\Rightarrow}_Q$ *or* $q \in D_Q$,
W5. $q \xrightarrow{i}_Q$ *implies* $p \overset{i}{\dashrightarrow}\overset{\epsilon}{\Rightarrow}_P$ *or* $p \in D_P$,
W6. $\langle p, q\rangle \xrightarrow{\alpha} R'$ *implies* $R' \cap W \neq \emptyset$ *or* $\langle p, q\rangle \in D_{P\,\&\,Q}$.

Lemma 10 (Concrete Witness [5])**.** *Let* P, Q, R *be EMIAs with equal alphabets.*

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

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

W1 By Lemma 8 we get $p \in D_P$ iff $r \in D_R$ iff $q \in D_Q$.

W2 If $q \in D_Q$, then W1 applies and there is nothing to show. Let $p \xrightarrow{o}$. By $r \sqsubseteq_{\mathrm{EA}} p$, there is a transition $r \xrightarrow{o}$ and, by syntactic consistency and $r \sqsubseteq_{\mathrm{EA}} q$, a $q \overset{o}{=\!=\!>}$.

W3 Symmetrically to W2.

W4 Analogous to W2 when replacing $\xrightarrow{o}$ and $\overset{o}{=\!=\!>}$ with $\xrightarrow{i}$ and $\overset{i}{\dashrightarrow}\overset{\epsilon}{=\!=\!>}$, resp.

W5 Symmetrically to W4.

W6 Let $\langle p, q\rangle \in W$ due to r s.t. $\langle p, q\rangle \xrightarrow{\omega} R'$ because of C3. By $r \sqsubseteq_{\mathrm{EA}} p$, 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 $r \sqsubseteq_{\mathrm{EA}} q$ implies the existence of a transition $q \overset{\omega}{=\!=\!>}_Q q'$ with $r' \sqsubseteq_{\mathrm{EA}} q'$. Hence, there is a $\langle p', q'\rangle \in R' \cap W$ due to r'. The case of inputs is shown analogously. □

Proposition 11 ($\wedge$ is And). *If P and Q are EMIAs with equal alphabets, then (i) $\exists R.\, R \sqsubseteq_{\mathrm{EA}} P$ and $R \sqsubseteq_{\mathrm{EA}} Q$ iff P and Q are consistent. Further, if P and Q are consistent, then, for any R, (ii) $R \sqsubseteq_{\mathrm{EA}} P$ and $R \sqsubseteq_{\mathrm{EA}} Q$ iff $R \sqsubseteq_{\mathrm{EA}} P \wedge Q$.*

Proof. (i) "⇒″ follows from Lemma 10.

(i), (ii) "⇐": Let $R \sqsubseteq_{\mathrm{EA}} P \wedge Q$. We prove that $\mathcal{R} := \{\langle r, p\rangle \mid \exists q.\, r \sqsubseteq_{\mathrm{EA}} p \wedge q\}$ is an EA-refinement relation. Then, we may conclude (i) "⇐" by choosing $S^0_R := \big\{r \in S_R \,\big|\, \exists p \wedge q \in S^0_{P\wedge Q}.\, \langle r, p \wedge q\rangle \in \mathcal{R}\big\}$. Let $\langle r, p\rangle \in \mathcal{R}$ due to q. The proof follows closely the lines of [5] and proceeds as follows:

R1 If $r \in D_R$, then $p \wedge q \in D_{P\wedge Q}$; thus, $p \in D_P$.

R2, R3 Let $p \xrightarrow{\alpha} P'$, then we have $q \overset{\alpha}{=\!=\!>}$ and $p \wedge q \xrightarrow{\alpha} \{p' \wedge q' \mid p' \in P',\ q \overset{\alpha}{=\!=\!>}_Q q',\ p' \wedge q' \text{ defined}\}$. By $r' \sqsubseteq_{\mathrm{EA}} p' \wedge q'$ 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, $\overset{\alpha}{=\!=\!>}$ must be replaced by $\overset{\alpha}{\dashrightarrow}\overset{\epsilon}{=\!=\!>}$.)

R4, R5 Let $r \overset{\alpha}{\dashrightarrow} r'$. By $r \sqsubseteq_{\mathrm{EA}} p \wedge q$, there is a $p \wedge q \overset{\alpha}{=\!=\!>} p' \wedge q'$ such that $r' \sqsubseteq_{\mathrm{EA}} p' \wedge q'$; thus, $\langle r', p'\rangle \in \mathcal{R}$ due to q'. (In case of inputs, $\overset{\alpha}{=\!=\!>}$ must be replaced by $\overset{\alpha}{\dashrightarrow}\overset{\epsilon}{=\!=\!>}$.)

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

R1 Obvious.

R2, R3, R4, R5 As above, the proof closely follows the lines of [5]. □

As a standard category theoretic result, Prop. 11 implies that $\wedge$ is associative:

Corollary 12 (Associativity of $\wedge$)**.** *Conjunction is strongly associative, i.e., for all EMIAs P, Q, and R, if one of $P \wedge (Q \wedge R)$ and $(P \wedge Q) \wedge R$ is defined, then both are defined and $P \wedge (Q \wedge R) \sqsupseteq\sqsubseteq_{\mathrm{EA}} (P \wedge Q) \wedge R$.*

We close this section with a remark on alphabet extension. Refinement, conjunction and disjunction are defined for EMIAs with equal alphabets. When it comes to perspective-based specification, it is of interest to consider EMIAs with different alphabets [5]. 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 extending the alphabet of a specification, 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 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 localised extension combining different composition concepts is also possible. Besides this, there is nothing surprising to expect from alphabet extension, and we leave out the formal definition here for brevity.

4 Relation to other Interface Theories

The majority of IA-based interface theories prune errors. Therefore, it is important to investigate the relation between such error-pruning interface theories and our non-pruning EMIA theory. We do this for MIA [5] 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 [22] and MIO [2] (wrt. strong compatibility), resp. We establish here a Galois insertion between MIA and EMIA, i.e., a Galois connection $\langle \gamma, \alpha \rangle$ for which $\alpha \circ \gamma = \mathrm{id}_{\mathsf{MIA}}$ [9] (up to $\sqsupseteq\sqsubseteq_{\mathrm{MIA}}$). 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. Note that γ is different from the error-completion presented in [23] that is motivated by algorithmic considerations only. Error-completion preserves an interface's semantics when replacing missing inputs by transitions to an error state. In contrast, EMIA refines the semantics of MIA by retaining error states.

Definition 13 (MIA [5])**.** Modal Interface Automata *(MIA) are defined like EMIAs (cf. Def. 1), except that, instead of D_P, there is a* universal state u_P *that is only permitted as target of input may-transitions.*

An important difference between fatal error states and u_P is revealed in the different notion of refinement. While EMIA employs a variant of modal refinement [19] that preserves and reflects fatal error states, MIA adopts (ordinary) modal refinement in general but provides the possibility to employ IA-refinement where necessary. This is achieved by state u_P, which may be refined arbitrarily.

Definition 14 (MIA-Refinement [5])**.** *Let P and Q be MIAs with equal alphabets. A relation $\mathcal{R} \subseteq S_P \times S_Q$ is a* MIA-refinement *relation if, for all $\langle p, q\rangle \in \mathcal{R} \setminus (S_P \times \{u_Q\})$, the rules of Def. 3 hold when replacing R1 by: MR1. $p \neq u_P$.*

Parallel composition of MIAs is defined through reachability of illegal states:

Definition 15 (Backward Closure)**.** *Let P be a MIA or EMIA and $S \subseteq S_P$. The* Ω-backward closure of S in P *is the smallest set* $\mathrm{bcl}_P^\Omega(S) \subseteq S_P$ *s.t.* $S \subseteq \mathrm{bcl}_P^\Omega(S)$ *and, for all $\omega \in \Omega_P$ and $p' \in \mathrm{bcl}_P^\Omega(S)$, if $p \overset{\omega}{\dashrightarrow} p'$, then $p \in \mathrm{bcl}_P^\Omega(S)$.*

Definition 16 (MIA-Parallel Composition [5])**.** *For composable MIAs P and Q, the* parallel product $P \otimes Q$ *is defined by ignoring fatal error states in Def. 4. We say that there is a* communication mismatch *for p at q, in symbols* $\mathrm{mis}(p, q)$*, if there is an $a \in O_P \cap I_Q$ with $p \overset{a}{\dashrightarrow}$ and $q \overset{a}{\not\dashrightarrow}$. The set of* illegal states *is defined as $E_{P\otimes Q} := \mathrm{bcl}_{P\otimes Q}^\Omega(\{\langle p, q\rangle \mid \mathrm{mis}(p, q)$ or $\mathrm{mis}(q, p)\} \cup (S_P \times \{u_Q\}) \cup (\{u_P\} \times S_Q))$. The* parallel composition $P \,\|\, Q$ *is the MIA given by the state set $S_{P\,\|\,Q} := (S_{P\otimes Q} \setminus E_{P\otimes Q}) \cup \{u_{P\,\|\,Q}\}$, the alphabets $I_{P\,\|\,Q} := I_{P\otimes Q}$ and $O_{P\,\|\,Q} := O_{P\otimes Q}$, and the transition relations obtained from $P \otimes Q$ by replacing all $i?$-transitions of states $\langle p, q\rangle$ having an $i?$-transition to $E_{P\otimes Q}$ by a transition $\langle p, q\rangle \overset{i}{\dashrightarrow} u_{P\,\|\,Q}$. If $S^0_{P\otimes Q} \subseteq E_{P\otimes Q}$, then $S^0_{P\,\|\,Q} := \{u_{P\,\|\,Q}\}$, else $S^0_{P\,\|\,Q} := S^0_{P\otimes Q} \setminus E_{P\otimes Q}$.*

The set $\mathrm{bcl}_P^\Omega(D_P) \setminus D_P$ of an EMIA P corresponds roughly to the set of illegal states in IA, EIO, 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 $\mathrm{bcl}_P^\Omega(D_P) \setminus D_P$ are deemed equivalent (cf. Thm. 26).

Definition 17 (MIA-Conjunction [5])**.** *Let P and Q be MIAs with equal alphabets. The* MIA-conjunctive product *is defined by ignoring fatal error states in Def. 6 and adding the following rules for u:*

CE1. $\langle p, u_Q \rangle \xrightarrow{\alpha} P' \times \{u_Q\}$
if $p \xrightarrow{\alpha} P'$,
CE2. $\langle u_P, q \rangle \xrightarrow{\alpha} \{u_P\} \times Q'$
if $q \xrightarrow{\alpha} Q'$,
CE3. $\langle p, u_Q \rangle \overset{\alpha}{\dashrightarrow} \langle p', u_Q \rangle$ *if* $p \overset{\alpha}{\dashrightarrow} p'$,
CE4. $\langle u_P, q \rangle \overset{\alpha}{\dashrightarrow} \langle u_P, q' \rangle$ *if* $q \overset{\alpha}{\dashrightarrow} q'$.

The MIA-conjunction *is obtained from the* MIA-conjunctive product *by pruning logically inconsistent states according to Rules F2 through F6 of Def. 7.*

An input i forbidden at state p is modelled as a missing transition in MIA and, equivalently, as an i-must-transition from p to a fatal error state in EMIA. Hence, a MIA's behaviour can be modelled by an EMIA where non-fatal states are input-enabled. We write EMIA' for the collection of such EMIAs.

The Galois insertion between MIA and EMIA consists of a concretisation $\gamma : \mathsf{MIA} \to \mathsf{EMIA}'$ and an abstraction $\alpha : \mathsf{EMIA}' \to \mathsf{MIA}$ s.t. $\langle \gamma, \alpha \rangle$ is a Galois connection and $(\alpha \circ \gamma)(Q) \sqsupseteq\sqsubseteq_{\mathrm{MIA}} Q$. The main idea behind α is to consider the states $\mathrm{bcl}^{\Omega}_{P}(D_P) \setminus D_P$ as equivalent, thus, yielding equivalence classes of EMIAs; α assigns a MIA to each of these equivalence classes. Vice versa, γ assigns to each MIA the disjunction of an equivalence class of EMIAs.

Definition 18 (Abstraction Function from EMIA' to MIA)**.** *Let* $P \in \mathsf{EMIA}'$ *and* $C_P := \mathrm{bcl}^{\Omega}_{P}(D_P) \setminus D_P$. *The* MIA-abstraction *of* P *is the MIA* $\alpha(P) := (S_{\alpha(P)}, I_P, O_P, \longrightarrow_{\alpha(P)}, \dashrightarrow_{\alpha(P)}, S^0_{\alpha(P)}, u_{\alpha(P)})$ *with state sets* $S_{\alpha(P)} := (S_P \setminus (C_P \cup D_P)) \dot{\cup} \{u_{\alpha(P)}\}$ *and* $S^0_{\alpha(P)} := S^0_P \cap S_{\alpha(P)}$. *The transitions of* $\alpha(P)$ *are obtained from* P *by replacing all* $i?$*-transitions leading from a state* p *to states in* C_P *by* $p \overset{i?}{\dashrightarrow} u_{\alpha(P)}$. *The kernel equivalence* $\equiv_\alpha \subseteq \mathsf{EMIA}' \times \mathsf{EMIA}'$, *which is defined by* $P \equiv_\alpha Q$ *iff* $\alpha(P) \sqsupseteq\sqsubseteq_{\mathrm{MIA}} \alpha(Q)$ *and has equivalence classes* $[P]_\alpha$, *yields a canonical bijection* $\bar{\alpha} : \mathsf{EMIA}'/{\equiv_\alpha} \to \mathsf{MIA}$.

Lemma 19 (Monotonicity of α)**.** *The map* α *defined in Def. 18 is monotonic.*

Proof. Let $\mathcal{R}$ be an EA-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 a MIA-refinement relation between αP and αQ. Let $\langle p, q \rangle \in \mathcal{R}_\alpha$. In case $q = u_{\alpha Q}$, the definition of MIA-refinement is trivially satisfied, so we can assume $q \neq u_{\alpha Q}$. Hence, by definition of $\mathcal{R}_\alpha$, we also have $\langle p, q \rangle \in \mathcal{R}$ and may distinguish the following cases:

MR1 $\langle p, q \rangle \in \mathcal{R}$ implies $p \neq u_{\alpha P}$ as universal states do not occur in EMIA.

R2 Let $q \xrightarrow{i}_{\alpha Q} Q'_\alpha$ due to some $q \xrightarrow{i}_{Q} Q'$. Due to the replacement of transitions to C_Q in Def. 18, we know that $Q'_\alpha = Q'$ and that none of these target states is in C_Q or D_Q. By $\langle p, q \rangle \in \mathcal{R}$, there is a $p \xrightarrow{i}\overset{\epsilon}{\Rightarrow}_P P'$ such that P' matches Q'. With the same argument as before, we may conclude that $P'_\alpha := P'$ matches Q'_α.

R3 Analogous to R2, where $\xrightarrow{i}\overset{\epsilon}{\Rightarrow}$ is replaced by $\overset{\omega}{\Longrightarrow}$.

R4 Let $p \overset{i}{\dashrightarrow}_{\alpha P} p'$. If $p' \neq u_{\alpha P}$, 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. $\exists q'' \in C_Q.\, q \overset{i}{\dashrightarrow}_Q q''$: By definition of α we have $C_Q \cap S_{\alpha Q} = \emptyset$; thus, $q'' \notin S_{\alpha Q}$. Hence, it follows from $q \in S_{\alpha Q}$ that $q \overset{i}{\dashrightarrow}_Q u_{\alpha Q}$ by definition of α, and $\langle p', u_{\alpha Q}\rangle \in \mathcal{R}_\alpha$ is obvious.
2. $\forall q'' \in C_Q.\, q \overset{i}{\not\dashrightarrow}{}_Q q''$: 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 $p' = u_{\alpha P}$, then there is a $p'' \in C_P$ with $p \overset{i}{\dashrightarrow}_P p''$. By $\langle p, q\rangle \in \mathcal{R}$, there exists a $q'' \in C_Q$ such that $q \overset{i}{\dashrightarrow}\overset{\epsilon}{\Rightarrow}_Q q''$ and $\langle p'', q''\rangle \in \mathcal{R}$. Thus, $q \overset{i}{\dashrightarrow} u_{\alpha Q}$, and $\langle u_{\alpha P}, u_{\alpha Q}\rangle \in \mathcal{R}_\alpha$ is trivial.

R5 Analogous to R4 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 $p' \neq u_{\alpha P}$ and only Case 2 applies (otherwise, we would have $q \in C_Q$). □

Lemma 20 (α is Homomorphic wrt. $\|$). *The mapping α defined in Def. 18 is homomorphic wrt. parallel composition, i.e., $\alpha(P \,\|\, Q) \sqsupseteq\sqsubseteq_{\mathrm{MIA}} \alpha(P) \,\|\, \alpha(Q)$.*

Proof. First, observe that $\alpha(P \,\|\, Q)$ and $\alpha(P) \,\|\, \alpha(Q)$ have the same state set $S := S_{\alpha(P \,\|\, Q)} = S_{\alpha(P) \,\|\, \alpha(Q)}$ because the same pruning operation is used in α and in MIA's parallel composition operator (see also [5, 6]).

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

MR1 From $s = t \neq u_{\alpha(P) \,\|\, \alpha(Q)}$ one directly concludes $s \neq u_{\alpha(P) \,\|\, \alpha(Q)}$.

R2 Let $s = \langle p, q\rangle \in S_{\alpha(P \,\|\, Q)}$. A transition $\langle p, q\rangle \xrightarrow{i}_{\alpha(P) \,\|\, \alpha(Q)} S'$ 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 $u_{\alpha(P) \,\|\, \alpha(Q)}$, the same transition also exists in $\alpha(P \,\|\, Q)$.

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.

R3 Analogous to R2.

R4 Let $\langle p, q\rangle \overset{i}{\dashrightarrow}_{\alpha(P\,\|\,Q)} s''$. In case $s'' = u_{\alpha(P)\,\|\,\alpha(Q)}$, then this transition is due to a replacement of a transition $\langle p, q\rangle \overset{i}{\dashrightarrow}_{\alpha(P\,\|\,Q)} s'$ by $u_{\alpha(P)\,\|\,\alpha(Q)}$. In case $s'' \neq u_{\alpha(P)\,\|\,\alpha(Q)}$, 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 $\langle p, q\rangle \overset{i}{\dashrightarrow}_{\alpha(P\,\|\,Q)} \langle p'', q\rangle$.

R5 Analogous to R4.

Direction "$\sqsupseteq_{\text{MIA}}$" can be shown dually. □

In order to define the concretisation function γ we need a disjunction operator:

Definition 21 (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:*
$\bigvee_{j\in J} P_j := (\dot{\bigcup}_{j\in J} S_{P_j}, I, O, \dot{\bigcup}_{j\in J} \longrightarrow_{P_j}, \dot{\bigcup}_{j\in J} \dashrightarrow_{P_j}, \dot{\bigcup}_{j\in J} S^0_{P_j}, \dot{\bigcup}_{j\in J} D_{P_j})$.

Proposition 22 ($\vee$ is Or). *If P_j, for $j \in J$, and R are EMIAs with equal alphabets, then $\bigvee_{j\in J} P_j \sqsubseteq_{\text{EA}} R$ iff $P_j \sqsubseteq_{\text{EA}} R$ 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 $P_j \sqsubseteq_{\text{EA}} R$ due to EA-refinement relation $\mathcal{R}_j$. Because, in general, the union of EA-refinement relations is an EA-refinement relation, $(\bigcup_{j\in J} \mathcal{R}_j) \cup \mathcal{R}_Q$ is an EA-refinement relation, too. Vice versa, if $\bigvee_{j\in J} P_j \sqsubseteq_{\text{EA}} R$ due to an EA-refinement relation $\mathcal{R}$, then, for any $j \in J$, $\mathcal{R}_j := \mathcal{R} \cap (S_j \times S_R)$ is a suitable EA-refinement relation showing $P_j \sqsubseteq_{\text{EA}} R$. □

Disjunction on MIAs is defined analogously by ignoring fatal error states and replacing u_P and u_Q by $u_{P\vee Q}$. Obviously, α is homomorphic wrt. disjunction.

Definition 23 (Concretisation Function from MIA to EMIA'). *The concretisation function $\gamma : \mathsf{MIA} \to \mathsf{EMIA}'$ is defined as $\gamma(P) := \bigvee \bar{\alpha}^{-1}(P)$.*

Lemma 24 (Monotonicity of γ). *The map γ defined in Def. 23 is monotonic.*

Proof. Let $P \sqsubseteq_{\text{MIA}} Q$ be MIAs. By $\forall P' \in \bar{\alpha}^{-1}(P)\ \exists Q' \in \bar{\alpha}^{-1}(Q).\, P' \sqsubseteq_{\text{EA}} Q'$ we conclude $\bigvee \bar{\alpha}^{-1}(P) \sqsubseteq_{\text{EA}} \bigvee \bar{\alpha}^{-1}(Q)$, i.e., $\gamma(P) \sqsubseteq_{\text{EA}} \gamma(Q)$. □ □

Lemma 25. *Let P and Q be MIAs. Then, $\gamma(P\,\|\,Q) \sqsupseteq_{\text{EA}} \gamma(P)\,\|\,\gamma(Q)$.*

Proof. "$\sqsupseteq_{\text{EA}}$": By Thm. 26 and Lemma 20 we derive this chain of inequalities: $\gamma(P)\,\|\,\gamma(Q) \sqsubseteq_{\text{EA}} (\gamma \circ \alpha)(\gamma(P)\,\|\,\gamma(Q)) \sqsupseteq\!\sqsubseteq_{\text{EA}} \gamma((\alpha \circ \gamma)(P)\,\|(\alpha \circ \gamma)(Q)) \sqsupseteq\!\sqsubseteq_{\text{EA}} \gamma(P\,\|\,Q)$. □

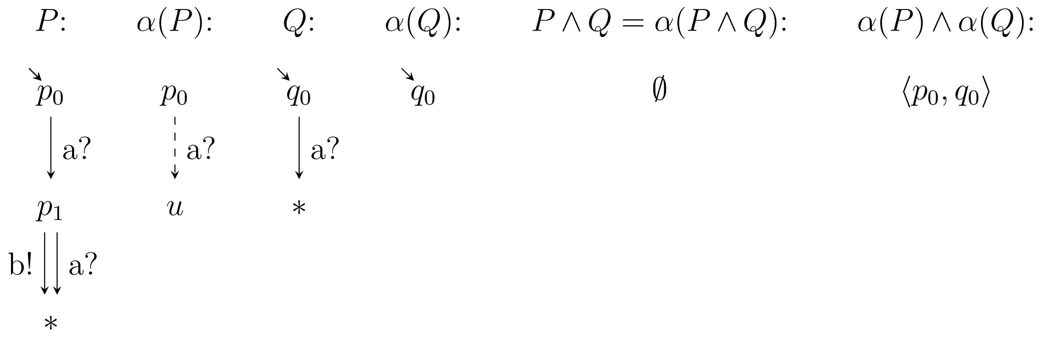

Figure 3: Example of EMIAs P, Q with $\alpha(P \wedge Q) \not\sqsupseteq_{\mathrm{MIA}} \alpha(P) \wedge \alpha(Q)$.

The monotonicity of the mappings α and γ defined in Defs. 18 and 23 is key to the proof of our main result that α and γ form a Galois insertion:

Theorem 26 (Galois Insertion)**.** *The maps* $\alpha\colon \mathsf{EMIA}' \to \mathsf{MIA}$ *and* $\gamma\colon \mathsf{MIA} \to \mathsf{EMIA}'$ *defined in Defs. 18 and 23 form a Galois insertion between* MIA *and* EMIA' *up to* $\sqsupseteq\sqsubseteq_{\mathrm{MIA}}$*, i.e.,* $P \sqsubseteq_{\mathrm{EA}} \gamma(Q)$ *iff* $\alpha(P) \sqsubseteq_{\mathrm{MIA}} Q$ *and* $(\alpha \circ \gamma)(Q) \sqsupseteq\sqsubseteq_{\mathrm{MIA}} Q$.

Proof. First, we show that $\alpha \circ \gamma = \mathrm{id}_{\mathsf{MIA}}$ (up to $\sqsupseteq\sqsubseteq_{\mathrm{MIA}}$):

$$\begin{aligned}
(\alpha \circ \gamma)(Q) &= \alpha\Big(\bigvee \bar{\alpha}^{-1}(Q)\Big) && \text{(by Def. 23)}\\
&\sqsupseteq\sqsubseteq_{\mathrm{MIA}} \bigvee \big\{\alpha(Q') \mid Q' \in \bar{\alpha}^{-1}(Q)\big\} && \text{(by homomorphicity of } \alpha)\\
&\sqsupseteq\sqsubseteq_{\mathrm{MIA}} \bigvee \{Q\} && \text{(by definition of } \bar{\alpha}^{-1})\\
&\sqsupseteq\sqsubseteq_{\mathrm{MIA}} Q.
\end{aligned}$$

Second, we prove that $\gamma \circ \alpha$ is extensive: From $P \in \bar{\alpha}^{-1}(\alpha(P))$ and Def. 23 we conclude $P \sqsubseteq_{\mathrm{EA}} \bigvee \bar{\alpha}^{-1}(\alpha(P)) = (\gamma \circ \alpha)(P)$. Third, we show that α and γ form a Galois connection, i.e., $P \sqsubseteq_{\mathrm{EA}} \gamma(Q)$ iff $\alpha(P) \sqsubseteq_{\mathrm{MIA}} Q$. Direction "$\Rightarrow$" holds due to $\alpha \circ \gamma = \mathrm{id}_{\mathsf{MIA}}$ and the monotonicity of α: $P \sqsubseteq_{\mathrm{EA}} \gamma(Q) \Rightarrow \alpha(P) \sqsubseteq_{\mathrm{MIA}} (\alpha \circ \gamma)(Q) \sqsupseteq\sqsubseteq_{\mathrm{MIA}} Q$. Direction "$\Leftarrow$" follows from the monotonicity of γ, the extensivity of $\gamma \circ \alpha$ and the transitivity of $\sqsubseteq_{\mathrm{EA}}$ by the following chain of implications: $\alpha(P) \sqsubseteq_{\mathrm{MIA}} Q \;\Rightarrow\; (\gamma \circ \alpha)(P) \sqsubseteq_{\mathrm{EA}} \gamma(Q) \;\Rightarrow\; P \sqsubseteq_{\mathrm{EA}} \gamma(Q)$. □

The map α is not homomorphic wrt. conjunction: $\alpha(P \wedge Q) \sqsubseteq_{\mathrm{MIA}} \alpha(P) \wedge \alpha(Q)$ holds for $P, Q \in \mathsf{EMIA}'$ because α is monotonic. However, the converse direction "$\sqsupseteq_{\mathrm{MIA}}$" does not hold in general, because MIA's replacement of illegal states by u—which must be reproduced by α—is a non-continuous operation. An example of EMIAs P and Q with $\alpha(P \wedge Q) \not\sqsupseteq_{\mathrm{MIA}} \alpha(P) \wedge \alpha(Q)$ is shown in Fig 3. State p_1 of specification P is in C_P due to the b!-transition. Therefore, α prunes p_1 and

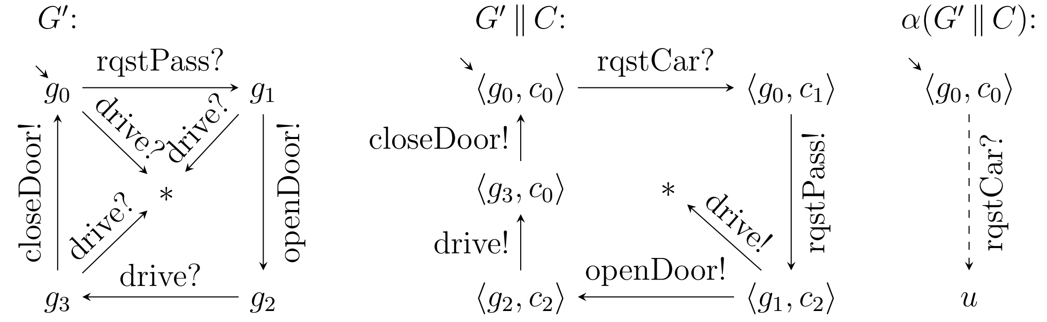
G':
rqstPass?
g_0
g_1
drive?
drive?
closeDoor!
openDoor!
$*$
drive?
drive?
g_3
g_2
$G' \parallel C$:
rqstCar?
$\langle g_0, c_0 \rangle$
$\langle g_0, c_1 \rangle$
closeDoor!
$\langle g_3, c_0 \rangle$
$*$
drive!
rqstPass!
drive!
openDoor!
$\langle g_2, c_2 \rangle$
$\langle g_1, c_2 \rangle$
$\alpha(G'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

Figure 4: Driving assistant system in EMIA and its Galois abstraction.

replaces it by a universal state u 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 $*$, 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 $P \wedge Q$.

The discontinuity of α also makes γ non-homomorphic wrt. parallel composition; however, γ satisfies the inequality $\gamma(P \parallel Q) \sqsupseteq_{\mathrm{EA}} \gamma(P) \parallel \gamma(Q)$ for MIAs P, Q.

5 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. 2 and also use this example to demonstrate the Galois abstraction from EMIA to MIA.

Issue (A)

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 $*$, which represents an unresolvable error as is illustrated in specification G' in Fig. 4. In the resulting parallel composition $G' \parallel C$, also shown in Fig. 4, 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 $*$, 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

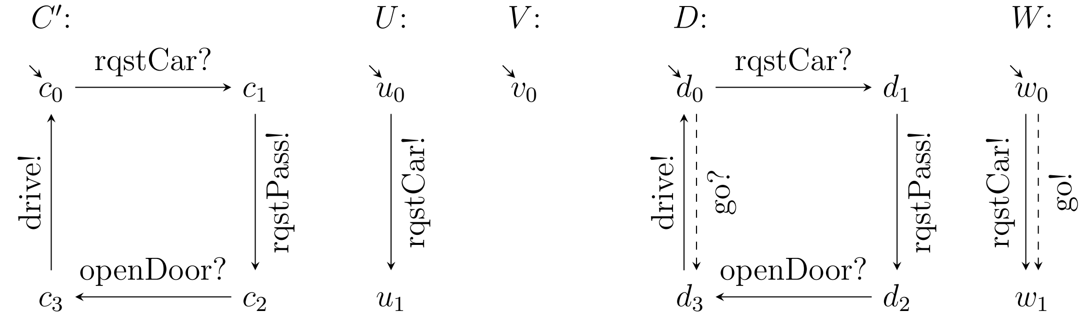

Figure 5: Corrected car C', user interfaces U, V, and product families D and W.

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 $*$ that is reachable from $\langle g_1, c_2\rangle$. The resulting specification is shown as C' in Fig 5.

Galois abstraction:

Fig. 4 (right) illustrates the abstraction function α of the Galois insertion between MIA and EMIA. We have $C_{G' \,\|\, C} := \mathrm{bcl}^{\Omega}_{G' \,\|\, C}(D_{G' \,\|\, C}) \setminus D_{G' \,\|\, C} = \{\langle g_1, c_2\rangle, \langle g_0, c_1\rangle\}$ (cf. Sec. 4). The rqstCar?-must-transition at $\langle g_0, c_0\rangle$ leading to $C_{G' \,\|\, C}$ is replaced by a rqstCar?-may-transition to $u_{\alpha(G' \,\|\, C)}$. Due to α being a homomorphism wrt. $\|$, this result corresponds exactly to the MIA shown in Fig. 2 (right).

Issue (B)

When adding the specification of a simple user interface, shown as U in Fig. 5, 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 $\langle g_0, c_0, u_0\rangle$. In other words, MIA is not by itself an assembly theory. A different but related problem arises in pessimistic theories: the user interface specification V in Fig. 5 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.

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 [15]. 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. In addition, refinement is lifted to assemblies by providing an error-preserving refinement relation for EMIOs, which is similar to EA-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 [15]. 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 $G' \parallel C \parallel U$ resembles $G' \parallel C$ (Fig. 4), except that action rqstCar is an output instead of an input. Further, $(G' \parallel C) \parallel V$ and $G' \parallel (C \parallel V)$ are equivalent in EMIA. In both examples, compatibility is checked via reachability of fatal error states. 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 [15], with the following difference: 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 27 (Assembly Theory [15]). *Let* $\mathfrak{I} := (\mathcal{I}, \mathrm{cs}, \parallel, \sqsubseteq)$ *be an interface theory, where* $\mathcal{I}$ *is a collection of interfaces,* $\mathrm{cs} \subseteq \mathcal{I}$ *is a communication safety predicate,* $\parallel$ *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 $\mathcal{A} := \{\langle I_k \rangle_{k \in K} \mid 0 < |K| < \infty$ *and* $I_k, I_l \in \mathcal{I}$ *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* $A, B, A_1, \ldots, A_n, B_1, \ldots, B_n \in \mathcal{A}$ *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* $A = A_1\dot{\cup}\ldots\dot{\cup}A_n$ *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* $A = A_1\dot{\cup}\ldots\dot{\cup}A_n$, $\mathrm{cs}(A_k)$ *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* $A = A_1\dot{\cup}\ldots\dot{\cup}A_n$, $B = B_1\dot{\cup}\ldots\dot{\cup}B_n$, $\mathrm{cs}(\langle\varphi(B_1), \ldots \varphi(B_n)\rangle)$, *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 28 (Canonical Assembly Theory [15]). *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 29 (Assembly Theory over EMIA). *Let* $\mathfrak{I}_{\mathsf{EMIA}} := (\mathsf{EMIA}, \mathrm{cs}, \|, \sqsubseteq_{\mathrm{EA}})$ *with* $\mathrm{cs}(I)$ *iff* $S^0_I \cap \mathrm{bcl}^\Omega_I(D_I) = \emptyset$. *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* $S^0_{\varphi(A)} \cap \mathrm{bcl}^\Omega_{\varphi(A)}(D_{\varphi(A)}) = \emptyset$, $\varphi(\langle I\rangle) := I$ *and* $\varphi(\langle I_1, \ldots, I_n\rangle) := I_1 \| \ldots \| I_n$, *and* $A \preceq B$ *iff* $\varphi(A) \sqsubseteq_{\mathrm{EA}} \varphi(B)$.

Showing that $\mathfrak{A}_{\mathsf{EMIA}}$ is an assembly theory is easy:

Lemma 30. $\mathfrak{A}_{\mathsf{EMIA}}$ *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 $\sqsubseteq_{\mathrm{EA}}$ 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 $\sqsubseteq_{\mathrm{EA}}$. A9 holds by definition of $\preceq$. A10 holds due to the compositionality of $\sqsubseteq_{\mathrm{EA}}$. □ □

$\mathfrak{A}_{\mathsf{EMIA}}$ obviously satisfies the first condition of Def. 28. 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

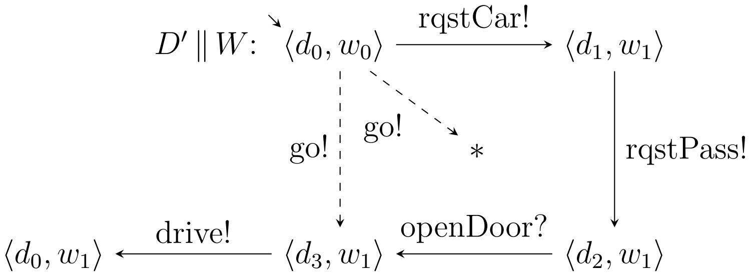
$D' \parallel W$:
$\langle d_0, w_0 \rangle$
rqstCar!
$\langle d_1, w_1 \rangle$
go!
go!
$*$
rqstPass!
$\langle d_0, w_1 \rangle$
drive!
$\langle d_3, w_1 \rangle$
openDoor?
$\langle d_2, w_1 \rangle$

Figure 6: Composition of product lines D' and W in EMIA.

that undefinedness is only necessary here because interface theories in [15] do not support the specification of erroneous interfaces (and, thus, one may change the definition accordingly). In both cases we have:

Theorem 31 (Assembly Theory)**.** *EMIA induces a canonical assembly theory.*

Because ϕ directly corresponds to $\|$ and $\preceq$ to $\sqsubseteq_{\mathrm{EA}}$, $\mathfrak{I}_{\mathsf{EMIA}}$ directly includes $\mathfrak{A}_{\mathsf{EMIA}}$. Because encapsulation directly corresponds to $\|$ and the assembly refinement preorder to $\sqsubseteq_{\mathrm{EA}}$, EMIA directly includes its canonical assembly theory.

Issue (C)

Consider specifications D and W of a car and a user interface product family, resp., both of which are shown in Fig. 5. 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. 2). In EMIA, the optional go?-transition at state d_0 would be modelled as a disjunctive go?-must-transition from d_0 to $\{d_3, *\}$, for a fatal error state $*$. 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. 6).

Issue (D)

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 does not directly scale 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.

6 Conclusions

Our interface theory EMIA is a uniformly integrated specification framework that is applicable at different levels of abstraction, e.g., component-based design, product line specification and programming with behavioural types. EMIA bridges the gaps between MTS [19], interface theories [10, 11, 2, 5, 6, 8, 18, 21, 22] and assembly theories [15]. It is based on a concept of *error-awareness*, whereby EMIA's refinement preorder reflects *and* preserves fatal error states. While recent interface theories [5, 22] 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 [5] 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 [5, 22], an error state e satisfies the laws $e \,\|\, q = e$, 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., refining cannot redefine an erroneous situation to be non-erroneous.

Regarding future work we intend to add alphabet extension and quotienting, and wish to capture differences and commonalities of different interface theories via axiomatisations. We also plan to implement EMIA in a formal methods tool, e.g., Mica [7], the MIO-Workbench [2] or MoTraS [16], and to adapt EMIA as a behavioural type theory for the Go Programming Language [14]. 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 [12].

Acknowledgements. We are grateful to Ferenc Bujtor, Walter Vogler 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 Křetiínský. Disjunctive modal transition systems and generalized ltl model checking. Technical Report FIMU-RS-2010-12, Faculty of Informatics, Masaryk University Brno, 2010.

[4] Dirk Beyer, Arindam Chakrabarti, Thomas A. Henzinger, and Sanjit A. Seshia. An application of web-service interfaces. In *Intl. Conf. on Web Services (ICWS)*, pages 831–838. IEEE, 2007.

[5] 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. An extended version of this paper, with a corrected definition of weak refinement, has been submitted to the TCS journal.

[6] 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.

[7] B. Caillaud. Mica: A modal interface compositional analysis library, 2011. online, accessed 2 Dec. 2015.

[8] 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.

[9] 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.

[10] L. de Alfaro and T. A. Henzinger. Interface automata. In *Foundations of Software Engineering (FSE)*, pages 109–120. ACM, 2001.

[11] 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.

[12] 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.

[13] S. Fendrich and G. Lüttgen. A generalised theory of interface automata, component compatibility and error. In *Integrated Formal Methods (iFM)*, LNCS. Springer, 2016.

[14] Johannes Gareis. Prototypical Integration of the Modal Interface Automata Theory in Google Go. Master's thesis, Bamberg University, Germany, 2015.

[15] R. Hennicker and A. Knapp. Moving from interface theories to assembly theories. *Acta Informatica*, 52(2-3):235–268, 2015.

[16] Jan Křetínský and Salomon 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.

[17] K. G. Larsen. Modal specifications. In *Automatic Verification Methods for Finite State Systems*, volume 407 of *LNCS*, pages 232–246. Springer, 1989.

[18] 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.

[19] K. G. Larsen and L. Xinxin. Equation solving using modal transition systems. In *Logic in Computer Scienc (LICS)*, pages 108–117. IEEE, 1990.

[20] Marten Lohstroh and Edward 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.

[21] G. Lüttgen, W. Vogler, and S. Fendrich. Richer interface automata with optimistic and pessimistic compatibility. *Acta Informatica*, 52(4-5):305–336, 2015.

[22] 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.

[23] S. Tripakis, C. Stergiou, M. Broy, and E. A. Lee. Error-completion in interface theories. In *Model Checking Software (SPIN)*, volume 7976 of *LNCS*, pages 358–375. Springer, 2013.

Bamberger Beiträge zur Wirtschaftsinformatik

Nr. 1 (1989) Augsburger W., Bartmann D., Sinz E.J.: Das Bamberger Modell: Der Diplom-Studiengang Wirtschaftsinformatik an der Universität Bamberg (Nachdruck Dez. 1990)

Nr. 2 (1990) Esswein W.: Definition, Implementierung und Einsatz einer kompatiblen Datenbankschnittstelle für PROLOG

Nr. 3 (1990) Augsburger W., Rieder H., Schwab J.: Endbenutzerorientierte Informationsgewinnung aus numerischen Daten am Beispiel von Unternehmenskennzahlen

Nr. 4 (1990) Ferstl O.K., Sinz E.J.: Objektmodellierung betrieblicher Informationsmodelle im Semantischen Objektmodell (SOM) (Nachdruck Nov. 1990)

Nr. 5 (1990) Ferstl O.K., Sinz E.J.: Ein Vorgehensmodell zur Objektmodellierung betrieblicher Informationssysteme im Semantischen Objektmodell (SOM)

Nr. 6 (1991) Augsburger W., Rieder H., Schwab J.: Systemtheoretische Repräsentation von Strukturen und Bewertungsfunktionen über zeitabhängigen betrieblichen numerischen Daten

Nr. 7 (1991) Augsburger W., Rieder H., Schwab J.: Wissensbasiertes, inhaltsorientiertes Retrieval statistischer Daten mit EISREVU / Ein Verarbeitungsmodell für eine modulare Bewertung von Kennzahlenwerten für den Endanwender

Nr. 8 (1991) Schwab J.: Ein computergestütztes Modellierungssystem zur Kennzahlenbewertung

Nr. 9 (1992) Gross H.-P.: Eine semantiktreue Transformation vom Entity-Relationship-Modell in das Strukturierte Entity-Relationship-Modell

Nr. 10 (1992) Sinz E.J.: Datenmodellierung im Strukturierten Entity-Relationship-Modell (SERM)

Nr. 11 (1992) Ferstl O.K., Sinz E. J.: Glossar zum Begriffsystem des Semantischen Objektmodells

Nr. 12 (1992) Sinz E. J., Popp K.M.: Zur Ableitung der Grobstruktur des konzeptuellen Schemas aus dem Modell der betrieblichen Diskurswelt

Nr. 13 (1992) Esswein W., Locarek H.: Objektorientierte Programmierung mit dem Objekt-Rollenmodell

Nr. 14 (1992) Esswein W.: Das Rollenmodell der Organsiation: Die Berücksichtigung aufbauorganisatorische Regelungen in Unternehmensmodellen

Nr. 15 (1992) Schwab H. J.: EISREVU-Modellierungssystem. Benutzerhandbuch

Nr. 16 (1992) Schwab K.: Die Implementierung eines relationalen DBMS nach dem Client/Server-Prinzip

Nr. 17 (1993) Schwab K.: Konzeption, Entwicklung und Implementierung eines computergestützten Bürovorgangssystems zur Modellierung von Vorgangsklassen und Abwicklung und Überwachung von Vorgängen. Dissertation

Nr. 18 (1993) Ferstl O.K., Sinz E.J.: Der Modellierungsansatz des Semantischen Objektmodells

Nr. 19 (1994) Ferstl O.K., Sinz E.J., Amberg M., Hagemann U., Malischewski C.: Tool-Based Business Process Modeling Using the SOM Approach

Nr. 20 (1994) Ferstl O.K., Sinz E.J.: From Business Process Modeling to the Specification of Distributed Business Application Systems - An Object-Oriented Approach -. 1st edition, June 1994

Ferstl O.K., Sinz E.J. : Multi-Layered Development of Business Process Models and Distributed Business Application Systems - An Object-Oriented Approach -. 2nd edition, November 1994

Nr. 21 (1994) Ferstl O.K., Sinz E.J.: Der Ansatz des Semantischen Objektmodells zur Modellierung von Geschäftsprozessen

Nr. 22 (1994) Augsburger W., Schwab K.: Using Formalism and Semi-Formal Constructs for Modeling Information Systems

Nr. 23 (1994) Ferstl O.K., Hagemann U.: Simulation hierarischer objekt- und transaktionsorientierter Modelle

Nr. 24 (1994) Sinz E.J.: Das Informationssystem der Universität als Instrument zur zielgerichteten Lenkung von Universitätsprozessen

Nr. 25 (1994) Wittke M., Mekinic, G.: Kooperierende Informationsräume. Ein Ansatz für verteilte Führungsinformationssysteme

Nr. 26 (1995) Ferstl O.K., Sinz E.J.: Re-Engineering von Geschäftsprozessen auf der Grundlage des SOM-Ansatzes

Nr. 27 (1995) Ferstl, O.K., Mannmeusel, Th.: Dezentrale Produktionslenkung. Erscheint in CIM-Management 3/1995

Nr. 28 (1995) Ludwig, H., Schwab, K.: Integrating cooperation systems: an event-based approach

Nr. 30 (1995) Augsburger W., Ludwig H., Schwab K.: Koordinationsmethoden und -werkzeuge bei der computergestützten kooperativen Arbeit

Nr. 31 (1995) Ferstl O.K., Mannmeusel T.: Gestaltung industrieller Geschäftsprozesse

Nr. 32 (1995) Gunzenhäuser R., Duske A., Ferstl O.K., Ludwig H., Mekinic G., Rieder H., Schwab H.-J., Schwab K., Sinz E.J., Wittke M: Festschrift zum 60. Geburtstag von Walter Augsburger

Nr. 33 (1995) Sinz, E.J.: Kann das Geschäftsprozeßmodell der Unternehmung das unternehmensweite Datenschema ablösen?

Nr. 34 (1995) Sinz E.J.: Ansätze zur fachlichen Modellierung betrieblicher Informationssysteme - Entwicklung, aktueller Stand und Trends -

Nr. 35 (1995) Sinz E.J.: Serviceorientierung der Hochschulverwaltung und ihre Unterstützung durch workflow-orientierte Anwendungssysteme

Nr. 36 (1996) Ferstl O.K., Sinz, E.J., Amberg M.: Stichwörter zum Fachgebiet Wirtschaftsinformatik. Erscheint in: Broy M., Spaniol O. (Hrsg.): Lexikon Informatik und Kommunikationstechnik, 2. Auflage, VDI-Verlag, Düsseldorf 1996

Nr. 37 (1996) Ferstl O.K., Sinz E.J.: Flexible Organizations Through Object-oriented and Transaction-oriented Information Systems, July 1996

Nr. 38 (1996) Ferstl O.K., Schäfer R.: Eine Lernumgebung für die betriebliche Aus- und Weiterbildung on demand, Juli 1996

Nr. 39 (1996) Hazebrouck J.-P.: Einsatzpotentiale von Fuzzy-Logic im Strategischen Management dargestellt an Fuzzy-System-Konzepten für Portfolio-Ansätze

Nr. 40 (1997) Sinz E.J.: Architektur betrieblicher Informationssysteme. In: Rechenberg P., Pomberger G. (Hrsg.): Handbuch der Informatik, Hanser-Verlag, München 1997

Nr. 41 (1997) Sinz E.J.: Analyse und Gestaltung universitärer Geschäftsprozesse und Anwendungssysteme. Angenommen für: Informatik ’97. Informatik als Innovationsmotor. 27. Jahrestagung der Gesellschaft für Informatik, Aachen 24.-26.9.1997

Nr. 42 (1997) Ferstl O.K., Sinz E.J., Hammel C., Schlitt M., Wolf S.: Application Objects – fachliche Bausteine für die Entwicklung komponentenbasierter Anwendungssysteme. Angenommen für: HMD – Theorie und Praxis der Wirtschaftsinformatik. Schwerpunkheft ComponentWare, 1997

Nr. 43 (1997): Ferstl O.K., Sinz E.J.: Modeling of Business Systems Using the Semantic Object Model (SOM) – A Methodological Framework - . Accepted for: P. Bernus, K. Mertins, and G. Schmidt (ed.): Handbook on Architectures of Information Systems. International Handbook on Information Systems, edited by Bernus P., Blazewicz J., Schmidt G., and Shaw M., Volume I, Springer 1997

Ferstl O.K., Sinz E.J.: Modeling of Business Systems Using (SOM), 2^{nd} Edition. Appears in: P. Bernus, K. Mertins, and G. Schmidt (ed.): Handbook on Architectures of Information Systems. International Handbook on Information Systems, edited by Bernus P., Blazewicz J., Schmidt G., and Shaw M., Volume I, Springer 1998

Nr. 44 (1997) Ferstl O.K., Schmitz K.: Zur Nutzung von Hypertextkonzepten in Lernumgebungen. In: Conradi H., Kreutz R., Spitzer K. (Hrsg.): CBT in der Medizin – Methoden, Techniken, Anwendungen -. Proceedings zum Workshop in Aachen 6. – 7. Juni 1997. 1. Auflage Aachen: Verlag der Augustinus Buchhandlung

Nr. 45 (1998) Ferstl O.K.: Datenkommunikation. In. Schulte Ch. (Hrsg.): Lexikon der Logistik, Oldenbourg-Verlag, München 1998

Nr. 46 (1998) Sinz E.J.: Prozeßgestaltung und Prozeßunterstützung im Prüfungswesen. Erschienen in: Proceedings Workshop „Informationssysteme für das Hochschulmanagement“. Aachen, September 1997

Nr. 47 (1998) Sinz, E.J.:, Wismans B.: Das „Elektronische Prüfungsamt“. Erscheint in: Wirtschaftswissenschaftliches Studium WiSt, 1998

Nr. 48 (1998) Haase, O., Henrich, A.: A Hybrid Respresentation of Vague Collections for Distributed Object Management Systems. Erscheint in: IEEE Transactions on Knowledge and Data Engineering

Nr. 49 (1998) Henrich, A.: Applying Document Retrieval Techniques in Software Engineering Environments. In: Proc. International Conference on Database and Expert Systems Applications. (DEXA 98), Vienna, Austria, Aug. 98, pp. 240-249, Springer, Lecture Notes in Computer Sciences, No. 1460

Nr. 50 (1999) Henrich, A., Jamin, S.: On the Optimization of Queries containing Regular Path Expressions. Erscheint in: Proceedings of the Fourth Workshop on Next Generation Information Technologies and Systems (NGITS'99), Zikhron-Yaakov, Israel, July, 1999 (Springer, Lecture Notes)

Nr. 51 (1999) Haase O., Henrich, A.: A Closed Approach to Vague Collections in Partly Inaccessible Distributed Databases. Erscheint in: Proceedings of the Third East-European Conference on Advances in Databases and Information Systems – ADBIS'99, Maribor, Slovenia, September 1999 (Springer, Lecture Notes in Computer Science)

Nr. 52 (1999) Sinz E.J., Böhnlein M., Ulbrich-vom Ende A.: Konzeption eines Data Warehouse-Systems für Hochschulen. Angenommen für: Workshop „Unternehmen Hochschule“ im Rahmen der 29. Jahrestagung der Gesellschaft für Informatik, Paderborn, 6. Oktober 1999

Nr. 53 (1999) Sinz E.J.: Konstruktion von Informationssystemen. Der Beitrag wurde in geringfügig modifizierter Fassung angenommen für: Rechenberg P., Pomberger G. (Hrsg.): Informatik-Handbuch. 2., aktualisierte und erweiterte Auflage, Hanser, München 1999

Nr. 54 (1999) Herda N., Janson A., Reif M., Schindler T., Augsburger W.: Entwicklung des Intranets SPICE: Erfahrungsbericht einer Praxiskooperation.

Nr. 55 (2000) Böhnlein M., Ulbrich-vom Ende A.: Grundlagen des Data Warehousing. Modellierung und Architektur

Nr. 56 (2000) Freitag B, Sinz E.J., Wismans B.: Die informationstechnische Infrastruktur der Virtuellen Hochschule Bayern (vhb). Angenommen für Workshop "Unternehmen Hochschule 2000" im Rahmen der Jahrestagung der Gesellschaft f. Informatik, Berlin 19. - 22. September 2000

Nr. 57 (2000) Böhnlein M., Ulbrich-vom Ende A.: Developing Data Warehouse Structures from Business Process Models.

Nr. 58 (2000) Knobloch B.: Der Data-Mining-Ansatz zur Analyse betriebswirtschaftlicher Daten.

Nr. 59 (2001) Sinz E.J., Böhnlein M., Plaha M., Ulbrich-vom Ende A.: Architekturkonzept eines verteilten Data-Warehouse-Systems für das Hochschulwesen. Angenommen für: WI-IF 2001, Augsburg, 19.-21. September 2001

Nr. 60 (2001) Sinz E.J., Wismans B.: Anforderungen an die IV-Infrastruktur von Hochschulen. Angenommen für: Workshop „Unternehmen Hochschule 2001“ im Rahmen der Jahrestagung der Gesellschaft für Informatik, Wien 25. – 28. September 2001

Änderung des Titels der Schriftenreihe *Bamberger Beiträge zur Wirtschaftsinformatik* in *Bamberger Beiträge zur Wirtschaftsinformatik und Angewandten Informatik* ab Nr. 61

Note: The title of our technical report series has been changed from *Bamberger Beiträge zur Wirtschaftsinformatik* to *Bamberger Beiträge zur Wirtschaftsinformatik und Angewandten Informatik* starting with TR No. 61

Bamberger Beiträge zur Wirtschaftsinformatik und Angewandten Informatik

Nr. 61 (2002) Goré R., Mendler M., de Paiva V. (Hrsg.): Proceedings of the International Workshop on Intuitionistic Modal Logic and Applications (IMLA 2002), Copenhagen, July 2002.

Nr. 62 (2002) Sinz E.J., Plaha M., Ulbrich-vom Ende A.: Datenschutz und Datensicherheit in einem landesweiten Data-Warehouse-System für das Hochschulwesen. Erscheint in: Beiträge zur Hochschulforschung, Heft 4-2002, Bayerisches Staatsinstitut für Hochschulforschung und Hochschulplanung, München 2002

Nr. 63 (2005) Aguado, J., Mendler, M.: Constructive Semantics for Instantaneous Reactions

Nr. 64 (2005) Ferstl, O.K.: Lebenslanges Lernen und virtuelle Lehre: globale und lokale Verbesserungspotenziale. Erschienen in: Kerres, Michael; Keil-Slawik, Reinhard (Hrsg.); Hochschulen im digitalen Zeitalter: Innovationspotenziale und Strukturwandel, S. 247 – 263; Reihe education quality forum, herausgegeben durch das Centrum für eCompetence in Hochschulen NRW, Band 2, Münster/New York/München/Berlin: Waxmann 2005

Nr. 65 (2006) Schönberger, Andreas: Modelling and Validating Business Collaborations: A Case Study on RosettaNet

Nr. 66 (2006) Markus Dorsch, Martin Grote, Knut Hildebrandt, Maximilian Röglinger, Matthias Sehr, Christian Wilms, Karsten Loesing, and Guido Wirtz: Concealing Presence Information in Instant Messaging Systems, April 2006

Nr. 67 (2006) Marco Fischer, Andreas Grünert, Sebastian Hudert, Stefan König, Kira Lenskaya, Gregor Scheithauer, Sven Kaffille, and Guido Wirtz: Decentralized Reputation Management for Cooperating Software Agents in Open Multi-Agent Systems, April 2006

Nr. 68 (2006) Michael Mendler, Thomas R. Shiple, Gérard Berry: Constructive Circuits and the Exactness of Ternary Simulation

Nr. 69 (2007) Sebastian Hudert: A Proposal for a Web Services Agreement Negotiation Protocol Framework . February 2007

Nr. 70 (2007) Thomas Meins: Integration eines allgemeinen Service-Centers für PC-und Medientechnik an der Universität Bamberg – Analyse und Realisierungs-Szenarien. February 2007 (out of print)

Nr. 71 (2007) Andreas Grünert: Life-cycle assistance capabilities of cooperating Software Agents for Virtual Enterprises. März 2007

Nr. 72 (2007) Michael Mendler, Gerald Lüttgen: Is Observational Congruence on μ-Expressions Axiomatisable in Equational Horn Logic?

Nr. 73 (2007) Martin Schissler: out of print

Nr. 74 (2007) Sven Kaffille, Karsten Loesing: Open chord version 1.0.4 User's Manual. Bamberger Beiträge zur Wirtschaftsinformatik und Angewandten Informatik Nr. 74, Bamberg University, October 2007. ISSN 0937-3349.

Nr. 75 (2008) Karsten Loesing (Hrsg.): Extended Abstracts of the Second *Privacy Enhancing Technologies Convention* (PET-CON 2008.1). Bamberger Beiträge zur Wirtschaftsinformatik und Angewandten Informatik Nr. 75, Bamberg University, April 2008. ISSN 0937-3349.

Nr. 76 (2008) Gregor Scheithauer, Guido Wirtz: Applying Business Process Management Systems – A Case Study. Bamberger Beiträge zur Wirtschaftsinformatik und Angewandten Informatik Nr. 76, Bamberg University, May 2008. ISSN 0937-3349.

Nr. 77 (2008) Michael Mendler, Stephan Scheele: Towards Constructive Description Logics for Abstraction and Refinement. Bamberger Beiträge zur Wirtschaftsinformatik und Angewandten Informatik Nr. 77, Bamberg University, September 2008. ISSN 0937-3349.

Nr. 78 (2008) Gregor Scheithauer, Matthias Winkler: A Service Description Framework for Service Ecosystems. Bamberger Beiträge zur Wirtschaftsinformatik und Angewandten Informatik Nr. 78, Bamberg University, October 2008. ISSN 0937-3349.

Nr. 79 (2008) Christian Wilms: Improving the Tor Hidden Service Protocol Aiming at Better Performances. Bamberger Beiträge zur Wirtschaftsinformatik und Angewandten Informatik Nr. 79, Bamberg University, November 2008. ISSN 0937-3349.

Nr. 80 (2009) Thomas Benker, Stefan Fritzemeier, Matthias Geiger, Simon Harrer, Tristan Kessner, Johannes Schwalb, Andreas Schönberger, Guido Wirtz: QoS Enabled B2B Integration. Bamberger Beiträge zur Wirtschaftsinformatik und Angewandten Informatik Nr. 80, Bamberg University, May 2009. ISSN 0937-3349.

Nr. 81 (2009) Ute Schmid, Emanuel Kitzelmann, Rinus Plasmeijer (Eds.): Proceedings of the ACM SIGPLAN Workshop on *Approaches and Applications of Inductive Programming* (AAIP'09), affiliated with ICFP 2009, Edinburgh, Scotland, September 2009. Bamberger Beiträge zur Wirtschaftsinformatik und Angewandten Informatik Nr. 81, Bamberg University, September 2009. ISSN 0937-3349.

Nr. 82 (2009) Ute Schmid, Marco Ragni, Markus Knauff (Eds.): Proceedings of the KI 2009 Workshop *Complex Cognition*, Paderborn, Germany, September 15, 2009. Bamberger Beiträge zur Wirtschaftsinformatik und Angewandten Informatik Nr. 82, Bamberg University, October 2009. ISSN 0937-3349.

Nr. 83 (2009) Andreas Schönberger, Christian Wilms and Guido Wirtz: A Requirements Analysis of Business-to-Business Integration. Bamberger Beiträge zur Wirtschaftsinformatik und Angewandten Informatik Nr. 83, Bamberg University, December 2009. ISSN 0937-3349.

Nr. 84 (2010) Werner Zirkel, Guido Wirtz: A Process for Identifying Predictive Correlation Patterns in Service Management Systems. Bamberger Beiträge zur Wirtschaftsinformatik und Angewandten Informatik Nr. 84, Bamberg University, February 2010. ISSN 0937-3349.

Nr. 85 (2010) Jan Tobias Mühlberg und Gerald Lüttgen: Symbolic Object Code Analysis. Bamberger Beiträge zur Wirtschaftsinformatik und Angewandten Informatik Nr. 85, Bamberg University, February 2010. ISSN 0937-3349.

Nr. 86 (2010) Werner Zirkel, Guido Wirtz: Proaktives Problem Management durch Eventkorrelation – ein *Best Practice* Ansatz. Bamberger Beiträge zur Wirtschaftsinformatik und Angewandten Informatik Nr. 86, Bamberg University, August 2010. ISSN 0937-3349.

Nr. 87 (2010) Johannes Schwalb, Andreas Schönberger: Analyzing the Interoperability of WS-Security and WS-ReliableMessaging Implementations. Bamberger Beiträge zur Wirtschaftsinformatik und Angewandten Informatik Nr. 87, Bamberg University, September 2010. ISSN 0937-3349.

Nr. 88 (2011) Jörg Lenhard: A Pattern-based Analysis of WS-BPEL and Windows Workflow. Bamberger Beiträge zur Wirtschaftsinformatik und Angewandten Informatik Nr. 88, Bamberg University, March 2011. ISSN 0937-3349.

Nr. 89 (2011) Andreas Henrich, Christoph Schlieder, Ute Schmid [eds.]: Visibility in Information Spaces and in Geographic Environments – Post-Proceedings of the KI'11 Workshop. Bamberger Beiträge zur Wirtschaftsinformatik und Angewandten Informatik Nr. 89, Bamberg University, December 2011. ISSN 0937-3349.

Nr. 90 (2012) Simon Harrer, Jörg Lenhard: Betsy - A BPEL Engine Test System. Bamberger Beiträge zur Wirtschaftsinformatik und Angewandten Informatik Nr. 90, Bamberg University, July 2012. ISSN 0937-3349.

Nr. 91 (2013) Michael Mendler, Stephan Scheele: On the Computational Interpretation of CKn for Contextual Information Processing - Ancillary Material. Bamberger Beiträge zur Wirtschaftsinformatik und Angewandten Informatik Nr. 91, Bamberg University, May 2013. ISSN 0937-3349.

Nr. 92 (2013) Matthias Geiger: BPMN 2.0 Process Model Serialization Constraints. Bamberger Beiträge zur Wirtschaftsinformatik und Angewandten Informatik Nr. 92, Bamberg University, May 2013. ISSN 0937-3349.

Nr. 93 (2014) Cedric Röck, Simon Harrer: Literature Survey of Performance Benchmarking Approaches of BPEL Engines. Bamberger Beiträge zur Wirtschaftsinformatik und Angewandten Informatik Nr. 93, Bamberg University, May 2014. ISSN 0937-3349.

Nr. 94 (2014) Joaquin Aguado, Michael Mendler, Reinhard von Hanxleden, Insa Fuhrmann: Grounding Synchronous Deterministic Concurrency in Sequential Programming. Bamberger Beiträge zur Wirtschaftsinformatik und Angewandten Informatik Nr. 94, Bamberg University, August 2014. ISSN 0937-3349.

Nr. 95 (2014) Michael Mendler, Bruno Bodin, Partha S Roop, Jia Jie Wang: WCRT for Synchronous Programs: Studying the Tick Alignment Problem. Bamberger Beiträge zur Wirtschaftsinformatik und Angewandten Informatik Nr. 95, Bamberg University, August 2014. ISSN 0937-3349.

Nr. 96 (2015) Joaquin Aguado, Michael Mendler, Reinhard von Hanxleden, Insa Fuhrmann: Denotational Fixed-Point Semantics for Constructive Scheduling of Synchronous Concurrency. Bamberger Beiträge zur Wirtschaftsinformatik und Angewandten Informatik Nr. 96, Bamberg University, April 2015. ISSN 0937-3349.

Nr. 97 (2015) Thomas Benker: Konzeption einer Komponentenarchitektur für prozessorientierte OLTP- & OLAP-Anwendungssysteme. Bamberger Beiträge zur Wirtschaftsinformatik und Angewandten Informatik Nr. 97, Bamberg University, Oktober 2015. ISSN 0937-3349.

Nr. 98 (2016) Sascha Fendrich, Gerald Lüttgen: A Generalised Theory of Interface Automata, Component Compatibility and Error. Bamberger Beiträge zur Wirtschaftsinformatik und Angewandten Informatik Nr. 98, Bamberg University, March 2016. ISSN 0937-3349.