Component Interfaces with Contracts on Ports

Rolf Hennicker

Ludwig-Maximilians-Universität München, Germany

Joint work with Sebastian Bauer, Axel Legay

Introduction

- *Reactive software components* interact with their environment; they have a significant dynamic behavior depending on states.
- *Interface specifications* are important for the **correct usage** of a component ("black box") and also for the **correct implementation** of a component.
- Crucial aspects:
  - Compatibility of interfaces of interacting components (no communication errors!)
  - Implementation of interface specifications (correct refinement!)
- Dimensions of system development:
  - *Compatibility* ("horizontal" dimension)
  - *Refinement* ("vertical" dimension)
  - *Composition* ("horizontal" dimension, hierarchical development)

Requirement 1: Preservation of Compatibility by Refinement

$$
\begin{array}{cccc}
 & & \text{Compatibility} & \\
 & S & \leftrightarrows & T \\
\text{Refinement} & \rotatebox{90}{$\leq$} & & \rotatebox{90}{$\leq$} \\
 & S' & \leftrightarrows & T'
\end{array}
$$

Requirement 2: Preservation of Refinement by Composition

if $S \leftrightarrows T$, then

Composition

$$\begin{array}{ccccc}
 & S & & & T \\
 & & & S \otimes T & \\
\text{Refinement} \dashrightarrow & \rotatebox{90}{$\leq$} & & & \rotatebox{90}{$\leq$} \\
 & & & \rotatebox{90}{$\leq$} & \\
 & S' & & & T' \\
 & & & S' \otimes T' &
\end{array}$$

Definition (inspired by De Alfaro, Henzinger)

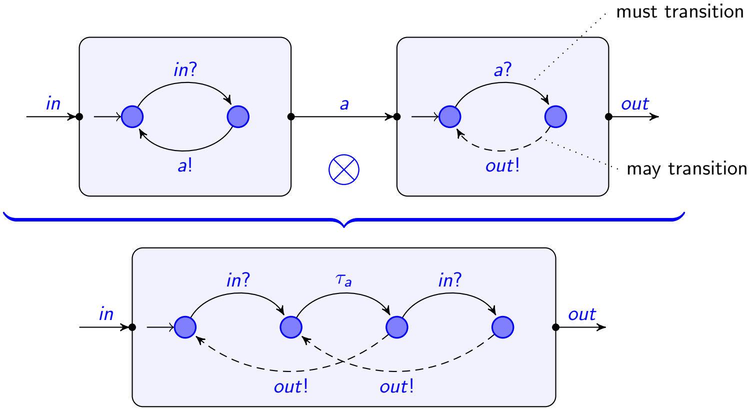
An **interface theory** is a tuple $(\mathfrak{S}, \leq, \leftrightarrows, \otimes)$ consisting of

- a class $\mathfrak{S}$ of **specifications**
- a reflexive and transitive **refinement relation** $\leq \; \subseteq \mathfrak{S} \times \mathfrak{S}$
- a symmetric **compatibility relation** $\leftrightarrows \; \subseteq \mathfrak{S} \times \mathfrak{S}$
- a partial, commutative **composition operator** $\otimes : \mathfrak{S} \times \mathfrak{S} \to \mathfrak{S}$

satisfying

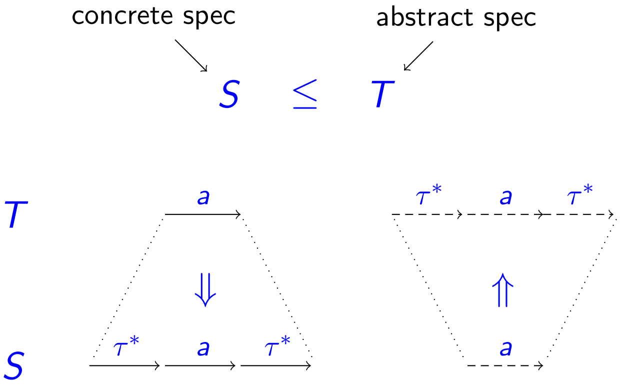
1. **Preservation of compatibility**
2. **Compositional refinement**

Example: Modal Input/Output Automata (MIOs) [Larsen, Nyman, Wasowski 2007]



"must $\otimes$ must = must"

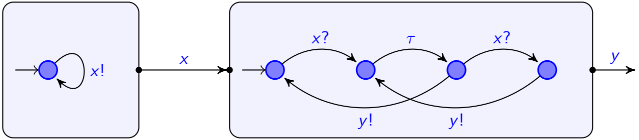
Weak Modal Refinement [Hüttel, Larsen 1989]



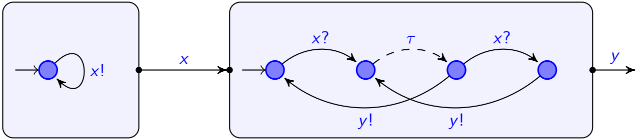
- If all transitions are "may", then $\leq$ is weak trace inclusion.
- If all transitions are "must", then $\leq$ is weak bisimulation.

Weak Compatibility [Bauer et al. 2010]

Weakly compatible MIOs:



Incompatible MIOs:



Theorem: MIOs with weak modal refinement, weak compatibility and synchronous composition form an interface theory.

We need more ...

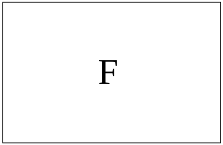
Interface Theories provide

- a nice abstract framework focusing on rudimentary requirements for component-based design.

But

- there is a lack of structure; they do not provide any mechanism to identify communication points.

Interface specification (no structure)



Labeled Interface Theory

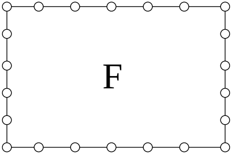
Definition

A *labeled interface theory* is a quadruple $(\mathfrak{S}, \mathcal{L}, \ell, \leq, \leftrightarrows, \otimes)$ consisting of

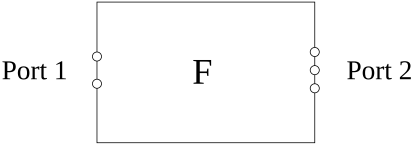
- an interface theory $(\mathfrak{S}, \leq, \leftrightarrows, \otimes)$,
- a set $\mathcal{L}$ of labels,
- a function $\ell : \mathfrak{S} \to \wp_{\mathrm{fin}}(\mathcal{L})$ assigning a finite set of labels, such that
  - if $\ell(S) \cap \ell(T) = \emptyset$, then $S \otimes T$ is defined,
  - If $S \otimes T$ is defined, then $\ell(S \otimes T) = (\ell(S) \cup \ell(T)) \setminus (\ell(S) \cap \ell(T))$,
  - ...

From Labeled Interfaces to Component Interfaces

(1) Interface specification **with labels**

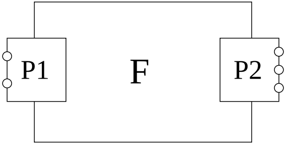


(2) Interface specification **with ports**

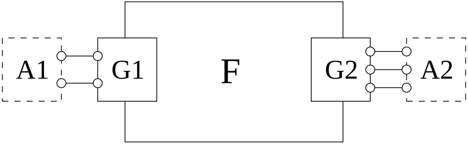


From Labeled Interfaces to Component Interfaces

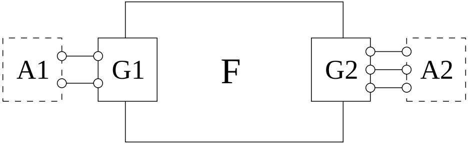
(3) Interface specification with **port specifications (protocols)**



(4) Interface specification with **port contracts**



Semantic Requirements



1. **Reliability:**
   The frame specification F should satisfy each guarantee (on one port) under the given assumptions (on the other ports), i.e.

   $$A1 \otimes F \leq G2 \quad \text{and} \quad A2 \otimes F \leq G1.$$

2. **Compatibility on ports:**
   Each port contract should have compatible assumptions and guarantees, i.e.

   $$A1 \leftrightarrows G1 \quad \text{and} \quad A2 \leftrightarrows G2.$$

Port Contracts and Component Interfaces (formally)

Given a labeled interface theory $(\mathfrak{S}, \mathcal{L}, \ell, \leq, \leftrightarrows, \otimes)$.

Definition

A *port contract* is a pair (A, G) with $A, G \in \mathfrak{S}$ such that $\ell(A) = \ell(G)$ and $G \leftrightarrows A$.

Definition

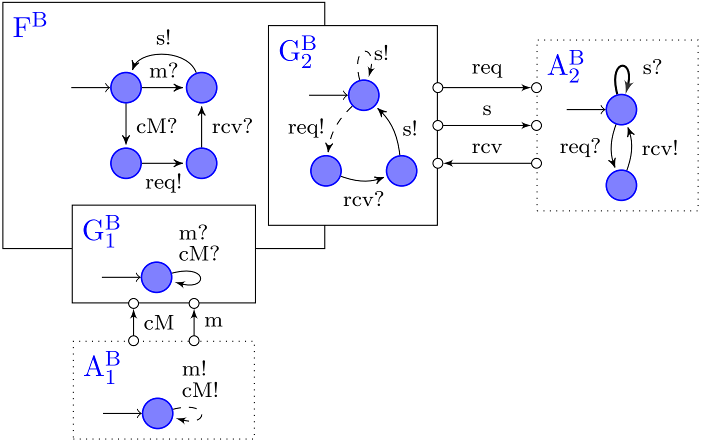
A *component interface* is a pair $C = (F, \{P_1, \dots P_n\})$ such that

- $F \in \mathfrak{S}$ is an interface specification, called *component frame*,
- $\{P_1, \dots P_n\}$ is a set of port contracts $P_i = (A_i, G_i)$.

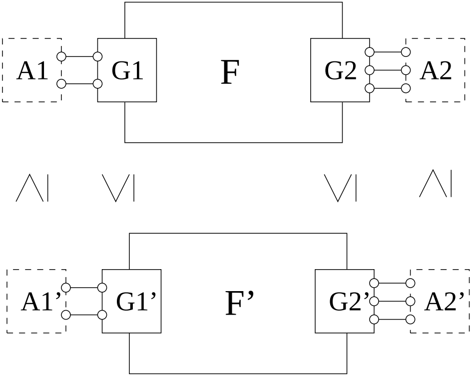
such that:

1. $\ell(F) = \ell(P_1) \cup \dots \cup \ell(P_n)$,
2. $\ell(P_i) \cap \ell(P_j) = \emptyset$ for all $i \neq j$,
3. $(A_1 \otimes \dots \otimes A_{i-1} \otimes A_{i+1} \dots \otimes A_n \otimes F) \leq G_i$ for $i = 1, \dots, n$.

Example: Broker with Port Contracts

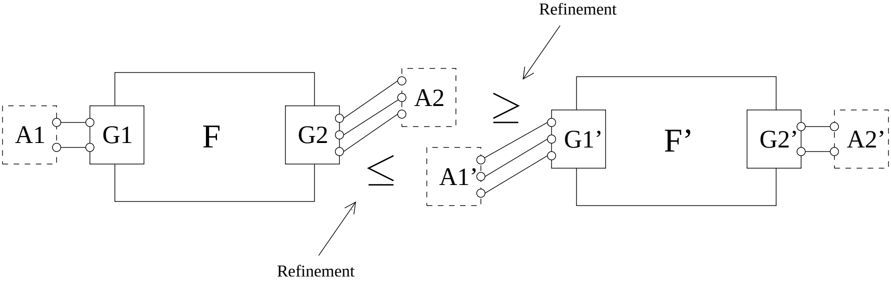


Refinement of Component Interfaces



Notation: $C' \sqsubseteq C$

Compatibility of Component Interfaces

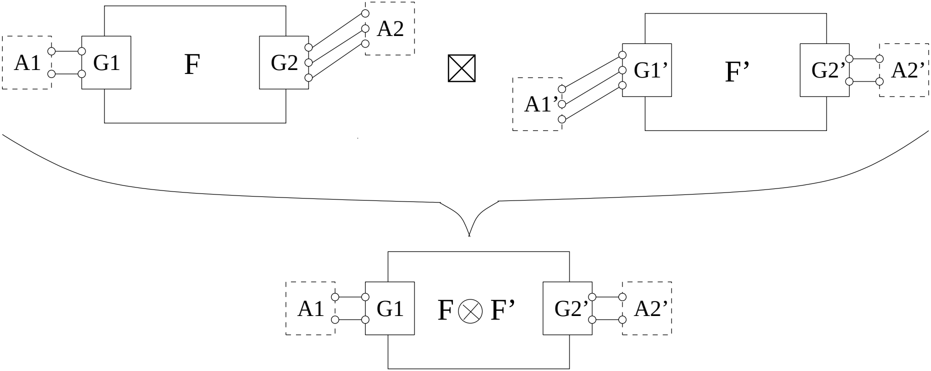


Notation: $C \leftleftarrows\!\!\!\!\rightrightarrows C'$

Facts: If $C \leftleftarrows\!\!\!\!\rightrightarrows C'$ then

- $G2 \leftrightarrows G1'$
- $A1 \otimes F \leftrightarrows A2' \otimes F'$
- if $E1 \leq A1, I \leq F$ and $E2' \leq A2', I' \leq F'$, then $E1 \otimes I \leftrightarrows E2' \otimes I'$
- if $E1 \leq A1$, $A1 \otimes I \leq G2$ and $E2' \leq A2', A2' \otimes I' \leq G1'$, then $E1 \otimes I \leftrightarrows E2' \otimes I'$

Composition of Compatible Component Interfaces



Composition preserves reliability:

$(A1 \otimes F \otimes F') \leq G2'$ and $(A2' \otimes F' \otimes F) \leq G1$.

Proof: $A1 \otimes F \leq G2 \leq A1'$ and $A1' \otimes F' \leq G2'$.

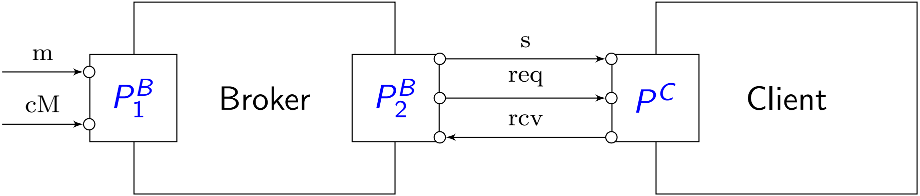
Hence, $(A1 \otimes F \otimes F') \leq A1' \otimes F' \leq G2'$.

Results

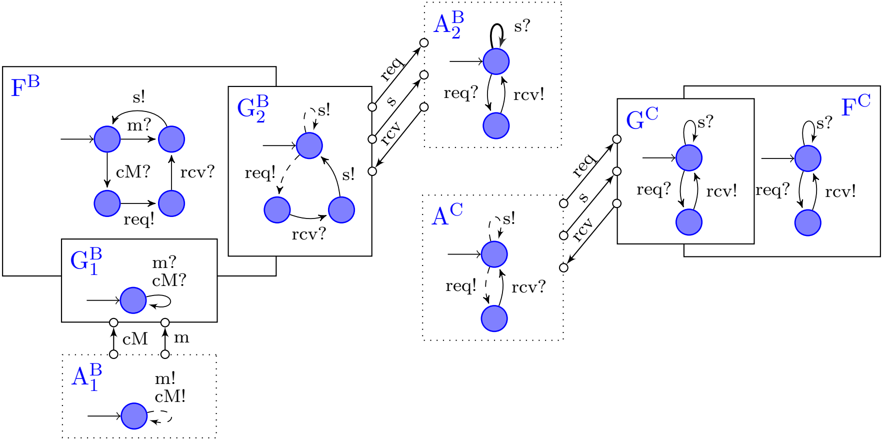
- Preservation of component compatibility by component refinement: $C \leftrightarrows D, C' \sqsubseteq C$ and $D' \sqsubseteq D$ implies $C' \leftrightarrows D'$.
- Preservation of component refinement by component composition: $C' \sqsubseteq C, D' \sqsubseteq D$ and $C \leftrightarrows D$ implies $C' \boxtimes D' \sqsubseteq C \boxtimes D$.

Theorem:
Let $LTh = (\mathfrak{S}, \mathcal{L}, \ell, \leq, \leftrightarrows, \otimes)$ be an arbitrary labeled interface theory. The class of component interfaces over LTh is itself a labeled interface theory with $\sqsubseteq, \leftrightarrows$ and $\boxtimes$.

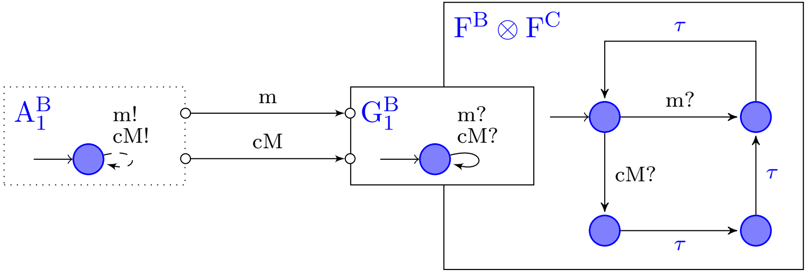
Example: Broker and Client Components



Example: Broker and Client Component Interfaces



Example: Composition of Broker and Client Interfaces



Conclusion

- Interface theories are a nice abstract framework but they lack structure for proper component-based design.
- Just by introducing labels for interfaces one can do a lot more.
- One can construct a generic, contract-based framework for component interfaces with ports *on top of any labeled interface theory*.
- Instantiation by modal I/O-transition systems.
- Further instantiations should be studied, e.g. integrating data constraints, asnychronous communication, ...
- Application to established design languages (like Wright, UML).
- Tool support by extending the MIO-Workbench.