Modal Interface Automata

Gerald Lüttgen
Lehrstuhl Softwaretechnik und Programmiersprachen
Universität Bamberg

(Joint work with Walter Vogler, Universität Augsburg)
Interface Theories

• **Traditional interfaces** (from 1960s)
  – Deal with *types of data and operations*
  – Are implemented in most compilers

• **Sequential, object-oriented interfaces** (from mid 1980s)
  – *Behavioural types*, as supported by automated verification tools
  – E.g., pre-/post-conditions and invariants of classes and methods
  – Prominent example: *Meyer’s contracts in Eiffel*

• **Concurrent, behavioural interfaces** (from early 2000s)
  – *Take concurrency into account*
  – Describe the interaction between system components
  – Prominent prototype: *de Alfaro/Henzinger’s Interface Automata (IA)*
Semantic Framework

- **Labelled transition system** representing component behaviour
  - *States* $p$ connected via *transitions* $p' \rightarrow a \rightarrow p''$
  - Transition labelled with *input* ($i?$), *output* ($o!$) or *internal* ($\tau$) *action*, taken from some alphabet

- **Desiderata**
  - **Parallel composition**
    - $p|q$ with a sensible notion of *component compatibility*
  - **Compositional refinement preorder**
    - $p \leq q$ implies $p|r \leq q|r$ (monotonicity of parallel composition)
  - **Conjunction**
    - Supporting perspective-based specification
    - $r \leq p \land q$ if and only if $r \leq p$ and $r \leq q$ (greatest lower bound)
  - **Alphabet extension**
    - Permitting the addition of new `features` when refining interfaces
Parallel Composition on IA

- Basic idea adopted from process algebra (cf. Milner’s CCS)
  - Synchronize on shared actions ($a!$, $a?$), resulting in the internal action $\tau$
  - Interleave on non-shared actions

- Example

- Optimistic notion of compatibility
  - Inputs can be delayed, outputs are immediate
  - If output of $p$ is not expected by $q$, then $p|q$ is an error state
  - Remove states that can reach an error locally (i.e., via outputs or $\tau$s)
IA-Refinement

• An alternating simulation: the largest relation $\leq$ on states s. t. for all $p \leq q$
  
  – $q - a? \rightarrow q'$ implies $\exists p'. p - a? \rightarrow p'$ and $p' \leq q'$
  
  – $p - a! \rightarrow p'$ implies $\exists q'. q (\tau \rightarrow)^* - a! \rightarrow q'$ and $p' \leq q'$
    
    (and similarly for $\tau$-actions)

• Intuition
  
  – *Specified inputs* (outputs) *must be implemented* (are not required)
  
  – *Unspecified inputs* (outputs) *are always allowed* (are forbidden)

• Compositionality result for input-deterministic IA
  
  – $p \leq q$ implies $p|r \leq q|r$ for any input-deterministic $r$
  
  – Direct matching of inputs (i.e., without leading $\tau$s) is important
Modal Transition Systems (MTS)

• Interface automata cannot enforce outputs

  BlackHole $\leq p$ (for any interface automaton $p$)

• Adopt idea of modal transition systems [Larsen, early 1990s]
  
  – Add a must-/may-modality to each transition
    
    • Must- (may-)transitions must (may) be implemented
    
    • Unspecified transitions are prohibited in any implementation/refinement
  
  – Syntactic consistency, i.e., every must-transition is also a may-transition

• Modal-refinement is again an alternating simulation $\leq$ s. t.
  
  – $q \rightarrow_a q'$ implies $\exists p'. p \rightarrow_a p'$ and $p' \leq q'$
  
  – $p \rightarrow a \rightarrow p'$ implies $\exists q'. q (\rightarrow \tau \rightarrow)^* \rightarrow a \rightarrow q'$ and $p' \leq q'$

  (and similarly for $\tau$-actions; $\tau$ is never must)
I/O Modal Transition Systems (IOMTS)

- Devised by Larsen, Nyman and Wasowski [ESOP‘07]
  - MTS with input and output, as well as classic modal-refinement
  - Parallel composition and compatibility as in IA
  - Embedding of IA in MTS: inputs (outputs) are must- (may-)transitions

- Compositionality flaw (alphabet of p, q is \{i?, o!\}; alphabet of r is \{o?\})

\[
p \xrightarrow{i?} q \xrightarrow{i?} o! \xrightarrow{} r
\]

\[
p|r \xrightarrow{i?} q|r \not\xrightarrow{} (potential error after i?)
\]

- No consideration of \(\tau\)-must-transitions on specification side

\[
\xrightarrow{} \xrightarrow{\tau} o! \xrightarrow{}
\]
Conjunction on IA

• Example, using interfaces that have the same alphabets

\[\text{p} \quad \text{i?} \quad \rightarrow \quad \text{p'} \quad \text{o!} \quad \rightarrow \quad \bullet \]

\[\text{q} \quad \text{i?} \quad \rightarrow \quad \text{q'} \quad \text{o!} \quad \rightarrow \quad \bullet \]

\[\text{j?} \quad \rightarrow \quad \text{q''} \quad \text{o!} \quad \rightarrow \quad \bullet \]

\[\text{p} \land \text{q} \quad \text{i?} \quad \rightarrow \quad \text{p'} \land \text{q'} \quad \text{o!} \quad \rightarrow \quad \bullet \]

\[\text{j?} \quad \rightarrow \quad \text{q''} \quad \text{o!} \quad \rightarrow \quad \bullet \]

• Any set of interfaces can always be implemented

\[\text{BlackHole} \quad \bullet \quad \leq \quad \text{p} \quad (\text{for any interface automaton p})\]

• No inconsistencies arise under IA-conjunction
Conjunction on MTS

- **Example**
  (cf. Benes, Cerna and Kretinsky [ATVA‘11])

- After action c, something impossible must be done
- **Inconsistency propagates backwards**
- Pruning similar to the one in parallel composition for IA

- **Our contribution here relates to dealing with τ-transitions**
  - E.g., if \( p ( -- \tau --> )^+ p' \) and \( q ( -- \tau --> )^+ q' \) then \( p \land q -- \tau --> p' \land q' \)
  - **Conjunction is no parallel composition!**
Conjunction: Disjunctive Must-Transitions

• Another example

\[ p \xrightarrow{a} \bullet \xrightarrow{b} \bullet \xrightarrow{c} \bullet \]

\[ q \xleftarrow{a} \bullet \xrightarrow{b} \bullet \xrightarrow{c} \bullet \]

• Common refinement must have at least one of

\[ a \xrightarrow{b} \bullet \]

\[ a \xrightarrow{c} \bullet \]

• Disjunctive must-transitions

\[ p \land q \xrightarrow{a} \bullet \xrightarrow{b} \bullet \xrightarrow{c} \bullet \]

• Disjunctive must-transitions are easier than syntactic inconsistency
Modal Interface Automata (MIA)

• Inherits from IA
  – Parallel composition, including the notion of compatibility
  – Input-determinism, but relaxed due to disjunctive input-must-transitions

• Inherits from MTS
  – Disjunctive must-transitions incl. τ-must-transitions, unlike in IOMTS

• Merges ideas from IA and MTS
  – Refinement as in MTS, but inputs are always allowed as in IA
    • All may-inputs are must-inputs
    • Parallel composition will be monotonic, unlike in IOMTS
    • Must-τs on specification side will be considered, unlike in IOMTS
  – Conjunction, where inputs (outputs) are treated as in IA (MTS)

• Investigates/considers alphabet extension
  – As advocated by Raclet, Caillaud et al. [Fund. Inform., 2011]
Weak Transition Relation

- **Not done before in the presence of disjunctive transitions**
  - To the best of our knowledge

- **Weak may-transition relation \( \Rightarrow \) as usual**
  - Leading and trailing \( \tau \)-may-transitions

- **Weak output-must-transition relation as the least relation s.t.**
  - \( p = \varepsilon \Rightarrow \{p\} \)
  - \( p = \varepsilon \Rightarrow P', p' \in P' \) and \( p' - \tau \rightarrow P'' \) implies \( p = \varepsilon \Rightarrow (P' \setminus \{p'\}) \cup P'' \)
  - \( p = \varepsilon \Rightarrow P' = \{p_1, ..., p_n\} \) and \( \forall j. p_j - o \rightarrow P_j \) implies \( p = o \Rightarrow \bigcup_j P_j \)
  - \( p = o \Rightarrow P', p' \in P' \) and \( p' - \tau \rightarrow P'' \) implies \( p = o \Rightarrow (P' \setminus \{p'\}) \cup P'' \)
MIA-Refinement

• Refinement preorder
  – **Considers** $\tau$-must-transitions on the specification side

• MIA-refinement $\leq$

Given MIAs $P$, $Q$ with the same input and output alphabets and $p \leq q$:

1. $q \xrightarrow{i} Q'$ implies $\exists P'. p \xrightarrow{i} P'$ and $\forall p' \in P' \exists q' \in Q'. p' \leq q'$
2. $q \xrightarrow{o} Q'$ implies $\exists P'. p = o \Rightarrow P'$ and $\forall p' \in P' \exists q' \in Q'. p' \leq q'$
3. $q \xrightarrow{\tau} Q'$ implies $\exists P'. p = \varepsilon \Rightarrow P'$ and $\forall p' \in P' \exists q' \in Q'. p' \leq q'$

4. **No requirement for input transitions, i.e., inputs are always allowed**

5. $p \xrightarrow{o} p'$ implies $\exists q'. q :: o ::= q'$ and $p' \leq q'$
6. $p \xrightarrow{\tau} p'$ implies $\exists q'. q :: \varepsilon ::= q'$ and $p' \leq q'$
Parallel Composition on MIA

• Compatibility as before, but a new transition rule is needed
  – Let action a be an input of p and an output of q, or vice versa
  – \( p|q - \tau \rightarrow P'|Q' \) if \( p - a \rightarrow P' \) and \( q - a \rightarrow Q' \) (and symmetrically)

• Compositionality result still holds, but its proof requires a subtle lemma for weak must-transitions
  – If \( p = a \Rightarrow P' \) and \( q - a \rightarrow Q' \) then \( p|q = \varepsilon \Rightarrow R \) for some \( R \subseteq P'|Q' \)
  – Observe that \( R = P'|Q' \) does not hold in general
Disjunction & Conjunction on MIA

• Disjunction
  – *Intuitive encoding using disjunctive τ-must- and input-must-transitions*
    • \( p \lor q \rightarrow \tau \rightarrow \{p, q\} \)
    • \( p \lor q \rightarrow i \rightarrow P' \cup Q' \) if \( p \rightarrow_i P' \) and \( q \rightarrow_i Q' \)
    • Plus the underlying may-transitions
  – \( \lor \) is the least upper bound wrt. \( \leq \), which implies compositionality

• Conjunction
  – Defined – for the moment – on interfaces with the same alphabets
  – *Treatment of inputs as for IA and of outputs and τs as for MTS*
  – \( \land \) is the greatest lower bound wrt. \( \leq \), which implies compositionality
Conjunction Problem for Dissimilar Alphabets

Alphabet \{o!, o'!\}: \quad \begin{array}{c}
\text{p} \quad \begin{array}{c}
\text{i?} \\
\circlearrowright \quad \text{i?}
\end{array} \\
\begin{array}{c}
o! \\
\circlearrowleft
\end{array}
\end{array} \\
\quad \begin{array}{c}
\text{q} \\
\begin{array}{c}
o!, o'!
\end{array}
\end{array}
\end{array}

Alphabet \{i?\}: \quad \begin{array}{c}
\text{q} \\
\begin{array}{c}
o!, o'!
\end{array}
\end{array}
\end{array}

Alphabets \{i?, o!, o'!\}:

\begin{array}{c}
\text{u} \\
\begin{array}{c}
o! \\
\circlearrowleft
\end{array}
\end{array} \\
\quad \begin{array}{c}
v \\
\begin{array}{c}
o! \\
\circlearrowleft
\end{array}
\end{array} \\
\quad \begin{array}{c}
w \\
\begin{array}{c}
o! \\
\circlearrowleft
\end{array}
\end{array}
\end{array}

- **Extending alphabets morally means adding `neutral´ may-loops**
  - Here, \(p \land q\) should intuitively be implementable by \(u\) and \(v\) but not \(w\)
  - **No MIA** \(r\) has \(u\) and \(v\) but not \(w\) as implementations, since if \(v\) refines \(r\) then also \(w\) must refine \(r\)
  - In MIA one may always add some input-transition with arbitrary subsequent behaviour
Alphabet Extension Problem

Alphabet \{o!, o'!\}: \quad p \xrightarrow{o!} \bullet \quad \text{Alphabet \{i?\}:} \quad q \bullet

Alphabets \{i?, o!, o'!\}:

u \quad \text{v} \quad \text{w}

• Generalizing refinement \leq so that the refining interface may have new inputs i is problematic
  
  – Consider demanding for new i: \quad w \xleftarrow{i} W' \quad \text{implies} \quad \forall w' \in W'. \quad w' \leq p
  
  – Then, refinement would not be transitive since \quad w \leq v \leq p \quad \text{but not} \quad w \leq p
The Pessimistic Approach

• Restrictive, pessimistic notion of compatibility
  – Advocated by Bauer, Hennicker et al. in their MIO theory [TACAS’10]
  – If any error state in a parallel composition can be reached via any actions (including input actions), then the composition is not defined
  – This approach distinguishes between \[ ?i \rightarrow E \] and \[ \_i \rightarrow \_ \]

• Trade-off
  – Classic view of may-inputs (i.e., input allowed or prohibited)
  – Much fewer parallel compositions are permitted

• Investigating MIA for pessimistic compatibility
  – Here, one can do away with input-determinism and with the requirement that may-inputs are also must-inputs
  – `Standard´ modal-refinement, with compositional parallel operator
Conjunction with Alphabet Extension

- **Conjuncts with dissimilar alphabets**

  \[ p \quad \quad b! \quad \quad q \]

- **Alphabet extension** \([p]_{\{c!}\}}\) of conjunct \(p\)

  \[
  \begin{align*}
  [p]_{\{c!\}} & \quad \quad a? \quad b! \quad c!
  \end{align*}
  \]

- **Conjunctive composition**

  \[
  p \land q = [p]_{\{c!\}} \land [q]_{\{\}}
  \]
### Summary & Comparison

<table>
<thead>
<tr>
<th>Optimistic Approach</th>
<th>Pessimistic Approach</th>
</tr>
</thead>
<tbody>
<tr>
<td>• IA, IOMTS, MIA\textsubscript{opt}</td>
<td>• MIO, MIA\textsubscript{pes}</td>
</tr>
<tr>
<td>• <strong>True open-systems view</strong></td>
<td>• <strong>Blurred open-systems view</strong></td>
</tr>
<tr>
<td>– Generous compatibility</td>
<td>– Restrictive compatibility</td>
</tr>
<tr>
<td>• <strong>IA-style modal refinement</strong></td>
<td>• <strong>Standard modal refinement</strong></td>
</tr>
<tr>
<td>– Inputs: required, allowed</td>
<td>– Inputs: required, allowed, <strong>forbidden</strong></td>
</tr>
<tr>
<td>– Input-determinism</td>
<td>– No input-determinism needed</td>
</tr>
<tr>
<td>• <strong>No alphabet extension capturing perspective-based specification</strong></td>
<td>• <strong>Alphabet extension capturing perspective-based specification</strong></td>
</tr>
<tr>
<td>– <strong>Extension is problematic</strong></td>
<td>– Alphabet extension operator</td>
</tr>
<tr>
<td>– Logic operators restricted to MIAs with same alphabets</td>
<td></td>
</tr>
</tbody>
</table>
Conclusions & Future Work

• **MIA is a compositional interface theory that allows one to**
  - Enforce outputs
  - Express disjunctive must-transitions
  - Specify non-deterministic behaviour
  - Abstract from internal computation
  - Compose interfaces in parallel, conjunctively and disjunctively
  - Interpret compatibility optimistically or pessimistically
  - Extend alphabets while refining (in the pessimistic version)

• **Questions left for future work**
  - Are there ‘clean´ interface theories in-between the optimistic and pessimistic approaches?
  - *Is there a possibility to allow new may-inputs while maintaining a true open-systems view?*
Selected Literature


