Refinement

Luís S. Barbosa

DI - UNIV. MINHO

UNIVERSITY OF TARTU

December 2003

Component Refinement

- Interface Refinement: concerned with plugging compatibility and extension of functionality
- Behaviour Refinement:
  - p behaviouraly refines q if the behaviour patterns observed for p are a structural restriction, wrt B, of those of q
  - preserves interfaces — it is internal to Cp hom categories
- Static Refinement:
  - data refinement of the component's state space
  - preserves interfaces and entails bisimilarity
  - cf. SETS calculus [Oliveira92, Oliveira98]

Interface Refinement

Can a component can be transformed, by suitable wiring, to replace another component with a different interface?

- plugging (or type) compatibility
- as interface types may encode operations signatures, may capture extension of the functionality or design sophistication

Motivation:

$$p \boxplus q \;\sim\; (q \boxplus p)[\mathsf{s}_+, \mathsf{s}_+]$$

i.e., $p \boxplus q$ and $q \boxplus p$ are bisimilar up to isomorphic wiring

- ... leading to a notion of replaceability

Interface Refinement

Definition

$\boxed{p \lessdot q}$

Component $p : I \longrightarrow O$ is replaceable by $q : I' \longrightarrow O'$, or q is a replacement of p if there exist witnessing functions $w_1 : I \longrightarrow I'$ and $w_2 : O' \longrightarrow O$ such that

$$p \;\sim\; q[w_1, w_2]$$

$\boxed{p \doteqdot q}$

Components p and q are called *interchangeable* if w_1 and w_2 are isomorphisms

Example: $p \boxplus q \doteqdot q \boxplus p$

using isomorphism s_+ as a wire in both cases

Interface Refinement

Lemma

$$p \doteqdot q \quad \text{iff} \quad p \lessdot q \;\wedge\; q \lessdot p$$

Proof

- ($\Rightarrow$): $p \lessdot q$ is witnessed by iso w_1 and w_2. Then also $p[w_1^\circ, w_2^\circ] \sim q$, *i.e.*, $q \lessdot p$.
- ($\Leftarrow$): Let $p \lessdot q$ and $q \lessdot p$ hold, *i.e.*, there exist w_1 and w_2 such that $p \sim q[w_1, w_2]$ and $q \sim p[w_3, w_4]$. This implies $q[w_1, w_2][w_3, w_4] \sim q$ which equivales $q[w_1 \cdot w_3, w_4 \cdot w_2] \sim q$. Then $w_1 \cdot w_3 = \mathsf{id}_I$ and $w_4 \cdot w_2 = \mathsf{id}_{O'}$. Similarly, using $p[w_3, w_4][w_1, w_2] \sim p$, one gets $w_3 \cdot w_1 = \mathsf{id}_{I'}$ and $w_2 \cdot w_4 = \mathsf{id}_O$. Then, w_1, w_2, w_3 and w_4 are iso and $p \doteqdot q$.

Interface Refinement

Lemma Replaceability ($\lessdot$) is a partial order up to interchangeablity ($\doteqdot$), which is an equivalence on components.

Proof

- Anti-symmetry up to $\doteqdot$ just proved
- Reflexibility holds because $p \lessdot p$ is witnessed by $p \sim p[\mathsf{id}, \mathsf{id}]$.
- If $p \lessdot q$ and $q \lessdot r$ hold, there exist w_1, w_2, w_3 and w_4 such that $p \sim q[w_1, w_2]$ and $q \sim r[w_3, w_4]$. Thus wrapping composition and transitivity of $\sim$ entail $p \sim r[w_1 \cdot w_3, w_4 \cdot w_2]$, *i.e.*, $p \lessdot r$.
- Reflexivity and transitivity of $\doteqdot$ are proved similarly and $q \doteqdot p$ is witnessed by the converse of isomorphism witnessing $p \doteqdot q$.

Interface Refinement

Examples

- $\ulcorner \iota_1 \urcorner ; (p \circledast q) \lessdot p \boxplus q$ and $(p \boxplus q) ; \ulcorner \iota_1 \urcorner \lessdot p \circledast q$
- any p may replace an *inert* component: $\mathsf{inert}_1 \lessdot p$
- $\lessdot$ fails to be a pre-congruence with respect to the component algebra:
  - ok for most operators:

$$p \lessdot p' \;\Rightarrow\; p[f,g] \lessdot p'[f,g] \quad \text{and} \quad p \boxplus q \lessdot p' \boxplus q$$

  - but problems with sequential composition and feedback

Behaviour Refinement

p behaviouraly refines q if the behaviour patterns observed from p are a structural restriction, with respect to the behavioural model B, of those of q.

- The classical recipe: look for a retrieve morphism to witness
- ... but component morphisms — as pointed coalgebra morphisms — entail bisimilarity

Look for a weaker notion of a morphism between coalgebras

Behaviour Refinement

Coalgebras as T-shaped transition structures

$$u \longrightarrow_p u' \quad \text{iff} \quad u' \in_{\mathsf{T}} p\, u$$

where $\in_{\mathsf{T}} \subseteq U \times \mathsf{T}\, U$ is defined by induction of the structure of T:

$$\begin{array}{rcl}
x \in_{\mathsf{Id}} y & \text{iff} & x = y \\
x \in_{\underline{K}} y & \text{iff} & \mathsf{ff} \\
x \in_{\mathsf{T}_1 \times \mathsf{T}_2} y & \text{iff} & x \in_{\mathsf{T}_1} \pi_1\, y \ \vee \ x \in_{\mathsf{T}_2} \pi_2\, y \\
x \in_{\mathsf{T}_1 + \mathsf{T}_2} y & \text{iff} & \begin{cases} y = \iota_1\, y' & \Rightarrow x \in_{\mathsf{T}_1} y' \\ y = \iota_2\, y' & \Rightarrow x \in_{\mathsf{T}_2} y' \end{cases} \\
x \in_{\mathsf{T}^{\underline{K}}} y & \text{iff} & \exists_{k \in K} .\ x \in_{\mathsf{T}} y\, k \\
x \in_{\mathcal{P}\mathsf{T}} y & \text{iff} & \exists_{y' \in y} .\ x \in_{\mathsf{T}} y'
\end{array}$$

Behaviour Refinement

For $p : I \longrightarrow O$, based on $\mathsf{T}^{\mathsf{B}} = \mathsf{B}(\mathsf{Id} \times O)^I$:

$$u \xrightarrow{\langle i,o \rangle}_p u' \quad \text{iff} \quad \langle u', o \rangle \in_{\mathsf{B}} (p u)\ i$$

Now

$$\mathsf{B}(h \times \mathsf{id}) \cdot a_p \ = \ a_q \cdot (h \times id)$$

gives rise to:

$$\begin{array}{ll}
\text{Preservation} & u \xrightarrow{\langle i,o \rangle}_p u' \ \Rightarrow \ h\, u \xrightarrow{\langle i,o \rangle}_q h\, u' \\
\text{Reflection} & h\, u \xrightarrow{\langle i,o \rangle}_q v' \ \Rightarrow \ \exists_{u' \in U} .\ u \xrightarrow{\langle i,o \rangle}_p u' \ \wedge \ u' = h\, v'
\end{array}$$

Behaviour Refinement

Refinement Predorder

[JH03] An order $\leq$ on T is a functor $\leq$ (concretely, mapping every set U into a collection of preorders $\leq_{\mathsf{T}U}$) which makes the following diagram to commute:

$$\begin{tikzcd} & \mathsf{PreOrd} \arrow[d] \\ \mathsf{Set} \arrow[ur, "\leq"] \arrow[r, "\mathsf{T}"'] & \mathsf{Set} \end{tikzcd} \qquad \text{concretely} \qquad \begin{tikzcd} & (\mathsf{T}U, \leq_{\mathsf{T}U}) \arrow[d] \\ U \arrow[ur] \arrow[r] & \mathsf{T}U \end{tikzcd}$$

This means that for any function $h : X \longrightarrow Y$, $\mathsf{T}h$ preserves $\leq$, *i.e.*

$$x_1 \leq x_2 \quad \Rightarrow \quad (\mathsf{T}h)\, x_1 \leq (\mathsf{T}h)\, x_2$$

Behaviour Refinement

Also impose $\leq_{\mathsf{T}U}$ compatibility with $\in_{\mathsf{T}}$:

$$x \in_{\mathsf{T}} x_1 \;\wedge\; x_1 \leq_{\mathsf{T}} x_2 \quad \Rightarrow \quad x \in_{\mathsf{T}} x_2$$

Definition

A forward morphism $h : \alpha \longrightarrow \beta$ with respect to a refinement preorder $\leq$, is a function from U to V such that

$$\mathsf{T}\, h \cdot \alpha \;\leq\; \beta \cdot h$$

Dually, h is called a backwards morphism if

$$\beta \cdot h \;\leq\; \mathsf{T}\, h \cdot \alpha$$

Behaviour Refinement

Lemma

T-coalgebras and forward (respectively, backward) morphisms define a category

Proof In both cases, identities are the identities on the carrier and composition is inherited from Set. What remains to be shown is that the composition of forward (respectively, backward) morphisms yields again a forward (respectively, backward) morphism. So, let $h : \alpha \longrightarrow \beta$ and $k : \beta \longrightarrow \gamma$ be two forward (respectively, backward) morphisms. Then

Behaviour Refinement

$\mathsf{T}(k \cdot h) \cdot \alpha$	$\gamma \cdot (k \cdot h)$
$=$ { T functor }	$=$ { $\cdot$ associative }
$\mathsf{T}k \cdot (\mathsf{T}h \cdot \alpha)$	$(\gamma \cdot k) \cdot h$
$\leq$ { h forward, $\leq$ }	$\leq$ { k backward }
$\mathsf{T}k \cdot (\beta \cdot h)$	$(\mathsf{T}k \cdot \beta) \cdot h$
$=$ { $\cdot$ associative }	$=$ { $\cdot$ associative }
$(\mathsf{T}k \cdot \beta) \cdot h$	$\mathsf{T}k \cdot (\beta \cdot h)$
$\leq$ { k forward }	$\leq$ { h backward, $\leq$ }
$(\gamma \cdot k) \cdot h$	$\mathsf{T}k \cdot \mathsf{T}h \cdot \alpha$
$=$ { $\cdot$ associative }	$=$ { T functor }
$\gamma \cdot (k \cdot h)$	$\mathsf{T}(k \cdot h) \cdot \alpha$

Behaviour Refinement

$\boxed{\text{Lemma}}$ Let T be an extended polynomial functor in Set, and α and β two T-coalgebras, with $\longrightarrow_\alpha$ and $\longrightarrow_\beta$ denoting the corresponding transition relations. A forward (respectively, backward) morphism $h : \alpha \longrightarrow \beta$ preserves (respectively, reflects) such transition relations.

Proof ...

Behaviour Refinement

Preservation follows from

$$
\begin{aligned}
& u \longrightarrow_\alpha u' \\
\equiv \quad & \{ \longrightarrow \text{ definition } \} \\
& u' \in_{\mathsf{T}} \alpha\, u \\
\Rightarrow \quad & \{ \in_{\mathsf{T}} \text{ definition } \} \\
& h\, u' \in_{\mathsf{T}} (\mathsf{T}h \cdot \alpha)\, u \\
\equiv \quad & \{ h \text{ forward and compatibility with } \in_{\mathsf{T}} \} \\
& h\, u' \in_{\mathsf{T}} (\beta \cdot h)\, u \\
\equiv \quad & \{ \cdot \text{ associative and } \longrightarrow \text{ definition } \} \\
& h\, u \longrightarrow_\beta h\, u'
\end{aligned}
$$

Behaviour Refinement

To establish reflection suppose that $h\ u \longrightarrow_\beta v'$, *i.e.*, $v' \in_\mathsf{T} (\beta \cdot h)\ u$. As h is a backward morphism we have $\beta \cdot h \leq \mathsf{T}\ h \cdot \alpha$, which, together with the compatibility requirement, entails $v' \in_\mathsf{T} (\mathsf{T}\ h \cdot \alpha)\ u$. This implies the existence of a state $u' \in U$ such that $v' = h\ u'$ and $u' \in_\mathsf{T} \alpha\ u$, *i.e.*, $u \longrightarrow_\alpha u'$.

Behaviour Refinement

... as existence of a forward morphism up to bisimulation

$q \trianglelefteq p \quad \equiv \quad$ exist r, s st $p \sim r \ \wedge\ q \sim s \ \wedge$ there is forward from r to s

The meaning of a refinement assertion $q \trianglelefteq p$ depends on the refinement preorder $\leq$ adopted

Behaviour Refinement

- T-*structural* inclusion: $x \leq y$ iff $\forall_{e \in_{\mathsf{T}} x} .\ e \in_{\mathsf{T}} y$
-

$$
\begin{array}{rcl}
x \subseteq_{\mathsf{Id}} y & \text{iff} & x = y \\
x \subseteq_{\underline{K}} y & \text{iff} & x =_K y \\
x \subseteq_{\mathsf{T}_1 \times \mathsf{T}_2} y & \text{iff} & \pi_1\ x \subseteq_{\mathsf{T}_1} \pi_1\ y\ \wedge\ \pi_2\ x \subseteq_{\mathsf{T}_2} \pi_2\ y \\
x \subseteq_{\mathsf{T}_1 + \mathsf{T}_2} y & \text{iff} & \begin{cases} x = \iota_1\ x' \wedge y = \iota_1\ y' & \Rightarrow x' \subseteq_{\mathsf{T}_1} y' \\ x = \iota_2\ x' \wedge y = \iota_2\ y' & \Rightarrow x' \subseteq_{\mathsf{T}_2} y' \end{cases} \\
x \subseteq_{\mathsf{T}\underline{K}} y & \text{iff} & \forall_{k \in K} .\ x\ k \subseteq_{\mathsf{T}} y\ k \\
x \subseteq_{\mathcal{P}\mathsf{T}} y & \text{iff} & \forall_{e \in x} \exists_{e' \in y} .\ e \subseteq_{\mathsf{T}} e'
\end{array}
$$

Behaviour Refinement

- $\subseteq^E_{\mathsf{T}}$: failure forcing variant defined as $\subseteq_{\mathsf{T}}$ by replacing the clause for powerset by

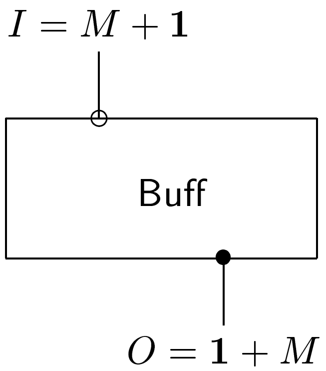
$$x \subseteq^E_{\mathcal{P}\mathsf{T}} y \quad \text{iff} \quad (x = \emptyset \Rightarrow y = \emptyset)\ \wedge\ \forall_{e \in x} \exists_{e' \in y} .\ e \subseteq_{\mathsf{T}} e'$$

- $\subseteq^F_{\mathsf{T}}$: for partial components defined as $\subseteq_{\mathsf{T}}$ by replacing the clause for sum by

$$x \subseteq^F_{\mathsf{T}_1 + \mathsf{T}_2} y \quad \text{iff} \quad \begin{cases} x = \iota_1\ x' \wedge y = \iota_1\ y' & \Rightarrow x' \subseteq_{\mathsf{T}} y' \\ x = \iota_2\ * & \Rightarrow y = \iota_2\ * \\ y = \iota_2\ * & \Rightarrow \mathsf{tt} \end{cases}$$

Behaviour Refinement

$$\begin{cases} \mathsf{put}: & M \longrightarrow \mathbf{1} \\ \mathsf{pick}: & \mathbf{1} \longrightarrow M \end{cases}$$



- LBuff (Lossy Buffer): for $\mathsf{T} = \mathcal{P}(\mathsf{Id} \times O)^I$ and dynamics over state space $U = M^*$ given by

$$\begin{aligned} a_{\mathsf{LBuff}}\langle u, \mathsf{put}\ m\rangle &= \{\langle u, \iota_1 *\rangle, \langle m : u, \iota_1 *\rangle\} \\ a_{\mathsf{LBuff}}\langle u, \mathsf{pick}\rangle &= \{\langle \mathsf{tail}\ u, \iota_2\ (\mathsf{head}\ u)\rangle\} \end{aligned}$$

Behaviour Refinement

- Buff (Deterministic Buffer): for $\mathsf{T} = (\mathsf{Id} \times O)^I$ and dynamics over state space $U = M^*$ given by

$$\begin{aligned} a_{\mathsf{Buff}}\langle u, \mathsf{put}\ m\rangle &= \langle m : u, \iota_1 *\rangle \\ a_{\mathsf{Buff}}\langle u, \mathsf{pick}\rangle &= \langle \mathsf{tail}\ u, \iota_2\ (\mathsf{head}\ u)\rangle \end{aligned}$$

To check LBuff $\trianglelefteq$ Buff:

- embedd the behaviour models: via nt from $(\mathsf{Id} \times O)^I$ to $\mathcal{P}(\mathsf{Id} \times O)^I$ canonnically extending monad morphism $\mathsf{sing}\ x = \{x\}$
- then identity function on M^* is a *forward*morphism, wrt $\subseteq_{\mathsf{T}}$:

$$(\mathsf{id} \times O) \cdot a_{\mathsf{Buff}} \subseteq a_{\mathsf{LBuff}}$$