Putting interactors together ...

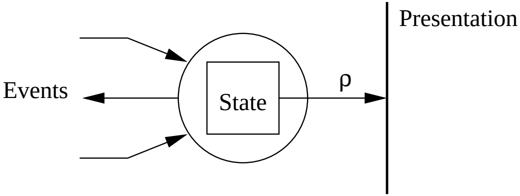
Luís Barbosa[1] José Campos[1] Marco Barbosa[2]

[1]FAST Group
DI - CCTC, Univ. Minho, Braga
Portugal
[2]Univiversidade de Cruz Alta
Brasil

CIC'07
October, 22 - 23, 2007
CWI, Amsterdam

Interactors

- Framework for structuring user interface specifications: [Faconti & Paternò, 90] [Duke & Harrison, 95].
- Objects with a rendering relation that maps their internal state into some presentation medium:



- Behaviour specifications in MAL [Campos 2001] or LOTOS [Paternò 95]

Interactors

- Hierarchical ('tree-like' aggregation) composition does not promote a clear separation of concerns between modelling interactors and the specification of how they are organized into an architecture and how they interact with each other

Aims

- Adopt an exogenous coordination approach, close to Arbab's Reo
- Given that both interactors and connectors exhibit *reactive* behaviour, look for a common modelling language.

Plan

- A modal language, with transitions indexed by (sub-)sets of actions
- Modelling interactors
- Modelling the coordination layer
- Configurations (putting everything together)
- Future work

A logic for behaviour

$$\langle a \rangle \phi$$

where interpretation 'indexed by *a*' is replaced by

'indexed by sth of which *a* is part of'

- modalities are relative to sets of actions regarded as factors of a (eventually larger) compound action.
- positive or negative action *factors*: K and $\sim K$, for $K \subseteq Act$

A logic for behaviour

$$\phi \ ::= \Psi \ | \ \text{true} \ | \ \text{false} \ | \ \phi_1 \wedge \phi_2 \ | \ \phi_1 \vee \phi_2 \ | \ \phi_1 \rightarrow \phi_2 \ | \ \langle W \rangle \phi \ | \ [W]\phi$$

Satisfiability

$$s \models \langle W \rangle \phi \ \equiv \ \langle \exists\, s' \ : \ \langle \exists\, \theta \ : \ s \xrightarrow{\theta} s' \ : \ W \prec \theta \rangle \ : \ s' \models \phi \rangle$$

$$s \models [W]\phi \ \equiv \ \langle \forall\, s' \ : \ \langle \exists\, \theta \ : \ s \xrightarrow{\theta} s' \ : \ W \prec \theta \rangle \ : \ s' \models \phi \rangle$$

where

$$W \prec X \ \triangleq \ \begin{cases} W = K, \text{for } K \subseteq Act & \Rightarrow \quad K \subseteq X \\ W = \sim K, \text{for } K \subseteq Act & \Rightarrow \quad K \nsubseteq X \end{cases}$$

A logic for behaviour

Extension to families

$$s \models \langle F \rangle \phi \;\equiv\; \langle \exists\, W \;:\; W \in F : \; \langle W \rangle \phi \rangle$$
$$s \models [F] \phi \;\equiv\; \langle \forall\, W \;:\; W \in F : \; [W] \phi \rangle$$

where $F \subseteq (\mathcal{P}(Act) \cup \sim\mathcal{P}(Act))$.

- $\langle abc \rangle$ vs $\langle J, K, L \rangle$

A logic for behaviour

Lemma

For all $a, a' \in Act$, $K, K' \subseteq Act$,

$$[a]\phi \Rightarrow [aa']\phi \quad \text{and} \quad \langle a\rangle\phi \Leftarrow \langle aa'\rangle\phi \tag{1}$$

$$[K]\phi \Leftarrow [K, K']\phi \quad \text{and} \quad \langle K\rangle\phi \Rightarrow \langle K, K'\rangle\phi \tag{2}$$

$$[K]\phi \wedge [K']\phi \equiv [K, K']\phi \tag{3}$$

$$\langle K\rangle\phi \vee \langle K'\rangle\phi \equiv \langle K, K'\rangle\phi \tag{4}$$

for K and K' action factors both positive or both negative,

$$[K, K']\phi \Rightarrow [K \cup K']\phi \tag{5}$$

$$\langle K, K'\rangle\phi \Leftarrow \langle K \cup K'\rangle\phi \tag{6}$$

A logic for behaviour

Proof

$$
\begin{array}{ll}
 & s \models [a]\phi \\
\equiv & \quad \{ \text{ definition } \} \\
 & \langle \forall\, s' \;:\; \langle \exists\, \theta \;:\; s \xrightarrow{\theta} s' \;:\; \{a\} \subseteq \theta \rangle \;:\; s' \models \phi \rangle \\
\Leftarrow & \quad \{ \text{ set inclusion } \} \\
 & \langle \forall\, s' \;:\; \langle \exists\, \theta \;:\; s \xrightarrow{\theta} s' \;:\; \{a, a'\} \subseteq \theta \rangle \;:\; s' \models \phi \rangle \\
\equiv & \quad \{ \text{ definition } \} \\
 & s \models [aa']\phi
\end{array}
$$

$\square$

Typical properties

- to preclude interactions in which some action factor K is absent:

$$\text{only}\, K \triangleq [\sim K]\text{false} \tag{7}$$

$$\text{forbid}\, K \triangleq \text{only}\ \sim K \tag{8}$$

- to assert the existence of at least a transition of which a particular action pattern is a factor:

$$\text{perm}\, K \triangleq \langle K \rangle \text{true} \tag{9}$$

or

$$\text{mandatory}\, K \triangleq \langle - \rangle \text{true} \wedge \text{only}\, K \tag{10}$$

- nested modalities: $[K]\langle L \rangle \phi$

Interactors

interactor *window*

attributes

$\boxed{\text{vis}}$ *visible*, *newinfo* : *bool*

actions

hide show update invalidate

axioms

$[hide]\ \neg visible$

$[show]\ visible$

$[update]\ newinfo$

$[invalidate]\ \neg newinfo$

Figure: A window interactor

Example

interactor *space*

attributes

$\boxed{\text{vis}}$ $state : \{open, closed\}$

actions

open close

axioms

perm $open \rightarrow state = closed$

$[open]\ state = open$

perm $close \rightarrow state = open$

$[close]\ state = closed$

Figure: The space interactor

Compositing interactors

The 'classical', tree-like,recipe:

- import two instances of the *window* interactor into the *space* interactor.
- add axioms:

CR1 the open indicator must be made visible and have its information updated whenever the system is opened.

CR2 the close indicator must be made visible and have its information updated whenever the system is closed.

Compositing interactors

interactor *spaceSign*
aggregates
window **via** *ol*
window **via** *cl*
attributes
$\boxed{\text{vis}}$ $state : \{open, closed\}$
actions
$\boxed{\text{vis}}$ *open close*
axioms
$\text{perm } open \rightarrow state = closed$
$[open]\ state' = open$
$\text{perm } close \rightarrow state = open$
$[close]\ state' = closed$
$\text{only}\,\{S.open, ol.update, ol.show\}$
$\vee \text{ only}\,\{S.close, cl.update, cl.show\}$

Connectors

Connectors are specified at two levels:

- the data level, which records the flow of data, through a relation

$$\text{data}.[\![\mathbb{C}]\!] : \mathbb{D}^n \longleftarrow \mathbb{D}^m$$

- the behavioural level which prescribes the activation patterns for their ports, through an assertion

$$\text{port}.[\![\mathbb{C}]\!]$$

over its sort $\text{sort}.[\![\mathbb{C}]\!]$.

Elementary connectors

Synchronous channel

$$\text{data}.[\![\, a \longrightarrow a' \,]\!] = \text{Id}_{\mathbb{D}} \qquad \text{sort}.[\![\, a \longrightarrow a' \,]\!] = \{aa'\}$$

$$\text{port}.[\![\, a \longrightarrow a' \,]\!] = \text{only } aa'$$

unreliable channel

$$\text{data}.[\![\, a \xrightarrow{\diamond} a' \,]\!] \subseteq \text{Id}_{\mathbb{D}} \qquad \text{sort}.[\![\, a \xrightarrow{\diamond} a' \,]\!] = \{a, aa'\}$$

$$\text{port}.[\![\, a \xrightarrow{\diamond} a' \,]\!] = \text{only } a$$

Elementary connectors

Filter channel

$$\text{data.}[\![\ a \xrightarrow{\ \phi\ } a'\ ]\!] = \mathrm{R}_\phi \qquad \text{sort.}[\![\ a \xrightarrow{\ \phi\ } a'\ ]\!] \ = \ \{aa'\}$$

$$\text{port.}[\![\ a \xrightarrow{\ \phi\ } a't\ ]\!] \ = \ \text{only}\, aa'$$

Fifo1 channel

$$\text{data.}[\![\ a \longrightarrow\!\square\!\!\rightarrow a'\ ]\!] \ = \ \mathrm{Id}_{\mathbb{D}} \qquad \text{sort.}[\![\ a \longrightarrow\!\square\!\!\rightarrow a'\ ]\!] \ = \ \{a, a'\}$$

$$\text{port.}[\![\ a \longrightarrow\!\square\!\!\rightarrow a'\ ]\!] \ = \ [a]\text{only}\, a', \sim a$$

This port specification equivales to $[a](\text{only}\, a' \wedge \text{forbid}\, a)$, formalising the intuition of a strict alternation between the activation of ports a and a'.

Elementary connectors

Sync drain

$$\text{data}.[\![\ a \overset{\blacktriangledown}{\longmapsto\!\!\!\dashv} a'\ ]\!] = \mathbb{D} \times \mathbb{D} \qquad \text{sort}.[\![\ a \overset{\blacktriangledown}{\longmapsto\!\!\!\dashv} a'\ ]\!] \ = \ \{aa'\}$$

$$\text{port}.[\![\ a \overset{\blacktriangledown}{\longmapsto\!\!\!\dashv} a'\ ]\!] = \text{only}\ aa'$$

Assync drain

$$\text{data}.[\![\ a \overset{\triangledown}{\longmapsto\!\!\!\dashv} a'\ ]\!] = \mathbb{D} \times \mathbb{D} \qquad \text{sort}.[\![\ a \overset{\triangledown}{\longmapsto\!\!\!\dashv} a'\ ]\!] \ = \ \{a, a'\}$$

$$\text{port}.[\![\ a \overset{\triangledown}{\longmapsto\!\!\!\dashv} a'\ ]\!] = \text{only}\ a, a' \ \wedge\ \text{forbid}\ aa'$$

New connectors from old

Terminology

- 'nomal' forms:

$$\text{port}.[\![\mathbb{C}]\!] = \phi_1 \vee \phi_2 \vee \cdots \vee \phi_n$$

  where each ϕ_i is a conjunction of

$$\underbrace{[K]\cdots[K]}_{n}\ \text{only}\ F$$

  each ϕ_i is either a non modal proposition, a only F assertion, for a family F of both positive or negative action factors prefixed by zero or more henceforth connectives $[x]$ for the same action x, or else a conjunction thereof.
- $t_{\#a}$, for $t \in \mathbb{D}^n$ and $a \in \mathcal{P}$, as the component of data tuple t corresponding to port a.
-

New connectors from old

Join

Purpose: to place two connectors side-by-side

$$\begin{aligned}
\text{data}.[\![\mathbb{C}_1 \boxtimes \mathbb{C}_2]\!] &= \text{data}.[\![\mathbb{C}_1]\!] \times \text{data}.[\![\mathbb{C}_2]\!] \\
\text{sort}.[\![\mathbb{C}_1 \boxtimes \mathbb{C}_2]\!] &= \text{sort}.[\![\mathbb{C}_1]\!] \cup \text{sort}.[\![\mathbb{C}_2]\!] \\
\text{port}.[\![\mathbb{C}_1 \boxtimes \mathbb{C}_2]\!] &= \text{port}.[\![\mathbb{C}_1]\!] \vee \text{port}.[\![\mathbb{C}_2]\!]
\end{aligned}$$

New connectors from old

Why sorts?

Example:

$$\text{port.}[\![(\ a \longrightarrow a' \ \boxtimes \ c \longrightarrow c'\ )]\!] \ = \ \text{only}\, aa' \ \vee \ \text{only}\, cc'$$

If, say, assertion only aa' is not interpreted wrt the sort of the new connector, but to the set of its ports instead, a transition labelled by $aa'c$ would be valid.

New connectors from old

Right share $(\mathbb{C}\ {}^{i}_{j} > z)$

Purpose: share input ports

$$r\,(\mathrm{data}.[\![\mathbb{C}\ {}^{i}_{j} > z]\!])\,t \quad \text{iff}$$
$$t'\,(\mathrm{data}.[\![\mathbb{C}]\!])\,t \ \wedge\ r_{|z} = t'_{|i,j} \ \wedge\ (r_{\#z} = t'_{\#i} \ \vee\ r_{\#z} = t'_{\#j})$$

$$\mathrm{port}.[\![(\mathbb{C}\ {}^{i}_{j} > z)]\!] \ = \ \{z \leftarrow i, z \leftarrow j\}\ \mathrm{port}.[\![\mathbb{C}]\!]$$

over

$$\mathrm{sort}.[\![(\mathbb{C}\ {}^{i}_{j} > z)]\!] \ = \ \{z \leftarrow i, z \leftarrow j\}\ \mathrm{sort}.[\![\mathbb{C}]\!]$$

New connectors from old

Example

$$\begin{pmatrix} a \longrightarrow a' \\ b \overset{\Box}{\longrightarrow} b' \end{pmatrix} \begin{smallmatrix} a' \\ b' \end{smallmatrix} > w \quad = \quad \begin{tikzcd} a \arrow[dr, bend left] & \\ & w \\ b \arrow[ur, bend right, "\Box"'] & \end{tikzcd}$$

Figure: A *merger*: only $aw \vee [b]$only w, $\sim b$.

New connectors from old

Left share $(z <^i_j \mathbb{C})$

Purpose: share output ports

$$\text{port}.[\![(z <^i_j \mathbb{C})]\!] = \sigma(\langle \bigwedge \; \phi_\theta \; : \; i \in \theta \vee j \in \theta : \; \phi_\theta \rangle) \\ \vee \; \langle \bigvee \; \phi_{\theta'} \; : \; i \notin \theta' \wedge j \notin \theta' \; : \; \phi_{\theta'} \rangle$$

and

$$\text{sort}.[\![(z <^i_j \mathbb{C})]\!] \; = \; \{z \leftarrow i, z \leftarrow j\} \, \text{sort}.[\![\mathbb{C}]\!]$$

On the other hand, relation $\text{data}.[\![z <^i_j \mathbb{C}]\!] : \mathbb{D}^n \longleftarrow \mathbb{D}^{m-1}$ is given by

$$t' \, (\text{data}.[\![z <^i_j \mathbb{C}]\!]) \, r \quad \text{iff} \\ t' \, (\text{data}.[\![\mathbb{C}]\!]) \, t \; \wedge \; r_{|z} = t_{|i,j} \; \wedge \; r_{\# z} = t_{\# i} = t_{\# j}$$

New connectors from old

Example

Sharing input ports a and b in a connector composed by three, otherwise no interfering, synchronous channels,

$$
\begin{array}{ll}
 & \mathrm{port}.[\![z <^{a}_{b} (\mathbb{C}_{a,a'} \boxtimes \mathbb{C}_{b,b'} \boxtimes \mathbb{C}_{c,c'})]\!] \\
\equiv & \quad \{ \text{ definition } \} \\
 & \{z \leftarrow a, z \leftarrow b\}(\mathrm{only}\, aa' \wedge \mathrm{only}\, bb') \vee \mathrm{only}\, cc' \\
\equiv & \quad \{ \text{ renaming and (3) } \} \\
 & \mathrm{only}\, za', zb' \vee \mathrm{only}\, cc' \\
\Rightarrow & \quad \{ \text{ by (5) } \} \\
 & \mathrm{only}\, za'b' \vee \mathrm{only}\, cc'
\end{array}
$$

which asserts that input on z co-occurs with output at both a' and b'.

New connectors from old

Example

$$z <^{a}_{b} \begin{pmatrix} a \longrightarrow a' \\ b \longrightarrow\!\square\!\!\rightarrow b' \\ c \longrightarrow c' \end{pmatrix} \quad = \quad \begin{tikzcd} & a' \\ z \arrow[ur, bend left] \arrow[dr, "\square"] & \\ & b' \\ c \arrow[r] & c' \end{tikzcd}$$

$$
\begin{array}{ll}
 & \text{port}.[\![z <^{a}_{b} (\mathbb{C}_{a,a'} \boxtimes \mathbb{C}_{b,b'} \boxtimes \mathbb{C}_{c,c'})]\!] \\
\equiv & \qquad \{ \text{ definition } \} \\
 & \{z \leftarrow a, z \leftarrow b\}(\text{only } aa' \wedge [b]\text{only } b', \sim b) \vee \text{only } cc' \\
\equiv & \qquad \{ \text{ renaming } \} \\
 & (\text{only } za' \wedge [z]\text{only } b', \sim b) \vee \text{only } cc'
\end{array}
$$

New connectors from old

Hook $\mathbb{C} \looparrowleft_i^j$

Purpose: connect an input to an output port

Let $\Phi = \text{port}.[\![\mathbb{C}]\!]$. $\Phi \looparrowleft_i^j = \Gamma \wedge \Psi$, where

- Remove from Φ all assertions μ_i and μ_j which involve at least an occurrence of action i or j, respectively. Let Ψ be the remaining formula, *i.e.*, the original Φ where all removed μ are replaced by the relevant identity (either true or false).
- For all μ involving simultaneously actions i and j, compute $\gamma = \{\emptyset \leftarrow i, \emptyset \leftarrow j\}\,\mu$.
- For all pairs μ_i and μ_j, involving i and j, respectively, compute $\gamma = \overline{\mu_i \mu_j}$ by

New connectors from old

Hook $\mathbb{C} \Lsh_i^j$

$$
\begin{cases}
\mu_i = \text{only } iK \wedge \mu_j = \text{only } jL \;\rightsquigarrow\; \text{only } K, L \\
\mu_i = \text{only } iK \wedge \mu_j = \underbrace{[j]\cdots[j]}_{n} \text{ only } L, {\sim}j \;\rightsquigarrow \\
\qquad \underbrace{[K]\cdots[K]}_{n} \text{ only } L, {\sim}K \\
\mu_i = \underbrace{[K]\cdots[K]}_{m} \text{ only } i \wedge \mu_j = \text{only } jL \;\rightsquigarrow\; \underbrace{[K]\cdots[K]}_{m} \text{ only } L \\
\mu_i = \underbrace{[K]\cdots[K]}_{m} \text{ only } i \wedge \mu_j = \underbrace{[j]\cdots[j]}_{n} \text{ only } L, {\sim}j \;\rightsquigarrow \\
\qquad \underbrace{[K]\cdots[K]}_{m+n} \text{ only } L, {\sim}K
\end{cases}
$$

New connectors from old

Hook $\mathbb{C} \Lsh^j_i$

- The sort of $\mathbb{C} \Lsh^j_i$ is obtained from that of $\mathbb{C}$ by consistently removing from each elementary interaction port identifiers i and j.
- The effect of *hook* on data, assuming data.$[\![\mathbb{C}]\!] : \mathbb{D}^n \longleftarrow \mathbb{D}^m$, is modelled by relation

$$\text{data}.[\![\mathbb{C} \Lsh^j_i]\!] : \mathbb{D}^{n-1} \longleftarrow \mathbb{D}^{m-1}$$

specified by

$$t'_{|j} \, (\text{data}.[\![\mathbb{C} \Lsh^j_i]\!]) \, t_{|i} \quad \text{iff} \quad t' \, (\text{data}.[\![\mathbb{C}]\!]) \, t \;\wedge\; t'_{\#j} = t_{\#i}$$

New connectors from old

Examples

Hooking a synchronous channel:

$$(\text{only}\, aa') \Lsh^{a}_{a'} \;=\; \text{only}\, \emptyset \;=\; [\sim\emptyset]\text{false} \;=\; \text{true}$$

Hooking a false a 1-place buffer:

$$[a](\text{only}\, a', \sim a) \Lsh^{a}_{a'} \;=\; [\emptyset](\text{true} \wedge \text{false}) \;=\; \text{false}$$

New connectors from old

Examples

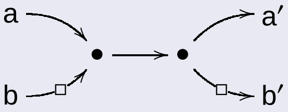


Figure: An example of *hook* usage.

New connectors from old

Examples

$$\text{port}.[\![(\mathbb{M} \boxtimes \mathbb{B}) \Lsh^{w}_{z}]\!]$$

$$\equiv \quad \{ \text{ definition } \}$$

$$((\text{only } aw \vee [b]\text{only } w, \sim b) \ \vee \ (\text{only } za' \wedge [z]\text{only } b', \sim z)) \Lsh^{w}_{z}$$

$$\equiv \quad \{ \ \textit{hook} \text{ definition } \}$$

$$\text{only } aa' \wedge [a]\text{only } b', \sim a \wedge [b]\text{only } a', \sim b \wedge [b][b]\text{only } b', \sim b$$

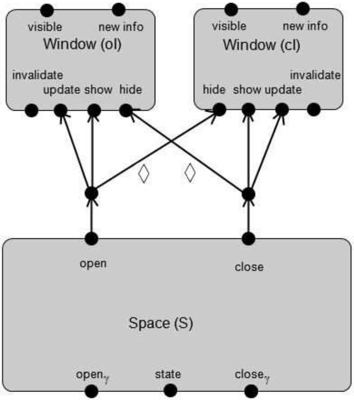
Configuration

$$\langle I, \mathbb{C}, \sigma \rangle$$

where $I = \{I_i \mid i \in n\}$ is a collection of interactors, $\mathbb{C}$ is a connector and σ a mapping of ports in I to ports in $\mathbb{C}$.

- its behaviour is given by the conjunction of the modal theories in each $I_n \in I$, as specified by their axioms, and the port specification port.$[\![\mathbb{C}]\!]$ of connector $\mathbb{C}$, after renaming by σ.

Example I


visible
new info
Window (ol)
invalidate
update show hide
visible
new info
Window (cl)
invalidate
hide show update
◊
◊
open
close
Space (S)
open$_\gamma$
state
close$_\gamma$

Example I

The connector

$$\mathbb{BC} \triangleq \mathbb{B} \boxtimes \mathbb{B}$$

$$\mathbb{B} \triangleq z <^{w}_{c} (w <^{a}_{b} (a \longrightarrow a' \boxtimes b \longrightarrow b') \boxtimes c \xrightarrow{\diamond} c')$$

An easy calculation yields

$$\text{port}.[\![\mathbb{B}]\!] = \text{only } za', zb', z$$

which, by (5), entails only $za'b'$.

Example I

Some properties

A default axiom perm $S.open$ and σ only $za'b'$ entails

$$\text{perm}\,\{S.open, oI.update, oI.show\}$$

i.e., there are transitions in which all the three ports are activated at the same time.

Example I

Some properties

A default axiom perm $S.open$ and σ only $za'b'$ entails

$$\text{perm}\,\{S.open, oI.update, oI.show\}$$

i.e., there are transitions in which all the three ports are activated at the same time. Moreover

$$\begin{aligned} &\text{perm}\,\{S.open, oI.update, oI.show\} \\ &\quad\wedge\ \text{only}\,\{S.open, oI.update, oI.show\} \end{aligned}$$

Example I

Some properties

Because action zc' is in sort.$[\![\mathbb{B}]\!]$, one also has

$$\mathrm{perm}\,\{S.open, cl.hide\}$$

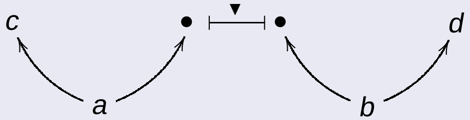
but, now only as a possibility, because a lossy channel was used to connect these ports.
From this property and only $\{S.open, ol.update, ol.show\}$ conclude

$$\mathrm{forbid}\,\{\sim S.open, cl.hide\}$$

Example II

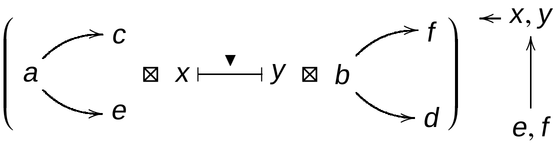
Sync Barrier

An interactor is expected to receive a password and an identity confirmation sent by two different interactors but received simultaneously.



Formally, connector $\mathbb{SB}$ is implemented through the composition of two broadcasters with two of their output ports connected by a synchronous drain.

Example II



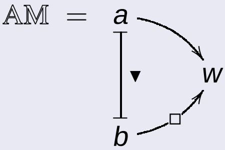
Now:

$$
\begin{array}{ll}
 & \text{port}.[\![\mathbb{SB}]\!] \\
\equiv & \quad \{ \text{ hook definition } \} \\
 & \text{only } ac, a, b, bd \\
\Rightarrow & \quad \{ \text{ by (5) } \} \\
 & \text{only } abcd
\end{array}
$$

Example III

Alternate merger

An interactor which has to receive the location coordinates supplied by two different input devices but in strict alternation



Formally,

$$b <^{d'}_{f} a <^{d}_{c} \left(c \longrightarrow c' \boxtimes d \xrightarrow{\blacktriangledown} d' \boxtimes f \longrightarrow\!\square\!\!\rightarrow f' \right)^{c'}_{f'} > w$$

whose behavioural pattern is

$$\mathrm{port}.[\![\mathbb{AM}]\!] = \text{only } aw, ab \wedge [b]\text{only } w, \sim b$$

Current research on Ivy

- Extension of $\mathbb{M}$ to express temporal properties through fix points.
- Tool support for $\mathbb{M}$-interactors in the Ivy workbench, and automated model analysis.
- Dynamic reconfiguration.