

# Towards a Theory of Glue

Simon Bliudze

Rigorous System Design Laboratory  
EPFL IC, Station 14, CH-1015, Lausanne, Switzerland  
Simon.Bliudze@epfl.ch



ÉCOLE POLYTECHNIQUE  
FÉDÉRALE DE LAUSANNE

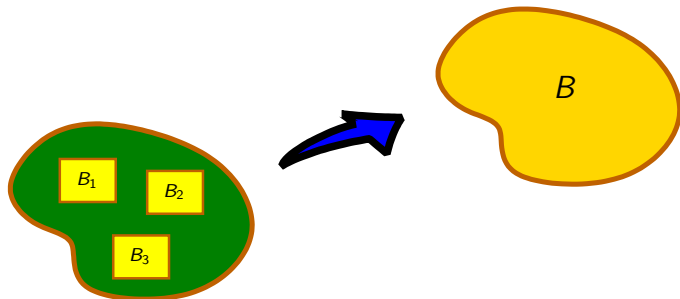
Interaction and Concurrency Experience 2012  
Stockholm, June 16<sup>th</sup>, 2012

# Outline

- Why behaviour types?
- Definition
- Co-algebraic example
- Some speculation about glue

- Why behaviour types?
- Definition
- Co-algebraic example
- Some speculation about glue

# Component-based design



**Components:** semantic behaviour domain  $\mathcal{B}$  (objects, actors, machines, CORBA components, JavaBeans, threads, etc.)

**Composition:** operators  $\mathcal{B}^n \rightarrow \mathcal{B}$  or  $2^{\mathcal{B}} \rightarrow \mathcal{B}$

# Comparing component-based frameworks

How do we compare two component-based frameworks  $\mathcal{C}_1, \mathcal{C}_2$ ?

## Common approach:

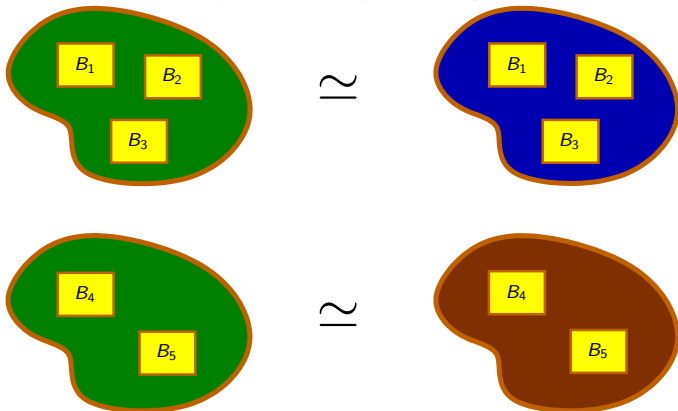
- What behaviours can be obtained in a given model?

$$\mathcal{C}_1(2^{\mathcal{B}}) \stackrel{?}{=} \mathcal{C}_2(2^{\mathcal{B}})$$

- Not satisfactory:
  - ▶  $id \in \mathcal{C}$  implies  $\mathcal{B} \subseteq \mathcal{C}(2^{\mathcal{B}})$
  - ▶  $\mathcal{C}(2^{\mathcal{B}}) \subseteq \mathcal{B}$
  - ▶ Hence,  $\mathcal{C}(2^{\mathcal{B}}) = \mathcal{B}$
  - ▶ Most “*behaviour*” models are Turing complete.

# Example: Strong Expressiveness Preorder

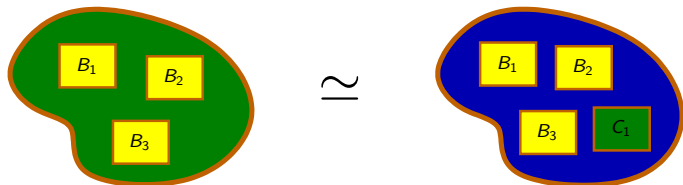
From Bludze & Sifakis, "A Notion of Glue Expressiveness for Component-Based Systems". In CONCUR 2008.



$$C_1 \leq C_2 \stackrel{\text{def}}{\iff} \forall g_1 \in C_1, \forall \mathbf{B} \subset \mathcal{B}, \exists g_2 \in C_2 : g_1(\mathbf{B}) \simeq g_2(\mathbf{B})$$

What if behaviour domains are not the same? ( $\mathcal{B}_1 \neq \mathcal{B}_2$ )

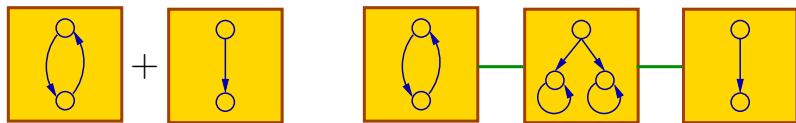
# Separation of concerns



- Encapsulating features into different entities
- Layers (e.g. network protocol stack, hardware abstraction)
- Base functionality vs.
  - ▶ component life-cycle management
  - ▶ introspection
  - ▶ non-functional behaviour
- Sequential computation vs. (memoryless) coordination

# Memoryless coordination intuitively

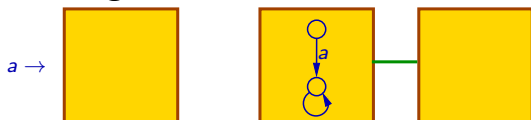
## Non-deterministic choice:



It is important to distinguish from interleaving

$$\frac{P \rightarrow P'}{P|Q \rightarrow P'|Q} \quad \frac{Q \rightarrow Q'}{P|Q \rightarrow P|Q'} \quad \Longrightarrow \quad \frac{P \rightarrow P' \quad Q \rightarrow Q'}{P|Q \rightarrow P'|Q \rightarrow P'|Q'}$$

## Prefixing:





# Outline

- Why behaviour types?
- Definition
- Co-algebraic example
- Some speculation about glue

# Understanding the types of behaviour

- What is the behaviour of several parallel components without any coordination constraints?
- How does one compare two behaviours?
- How does one model the simultaneous application of several composition operators?

# Parallel composition

$(\mathcal{B}, \parallel, \mathbf{0})$  — a monoid (associative, not necessarily commutative).

## Traces

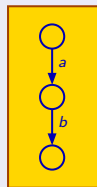
$B = (A, T)$ , where  $A \subseteq \mathbf{A}$  and  $T \subseteq A^*$

$\mathbf{0} = (\emptyset, \{\varepsilon\})$

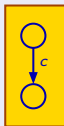
$B_1 \parallel B_2 = (A_1 \cup A_2, T_{12})$ : 

## Labelled Transition Systems

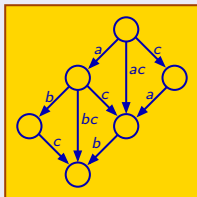
$B = (Q, A, \rightarrow)$ , where  $Q$  is finite,  $A \subseteq \mathbf{A}$ ,  $\rightarrow \subseteq Q \times 2^A \times Q$



$\parallel$



$=$



$\mathbf{0} = (\{*\}, \emptyset, \emptyset)$

$\mathbf{A}$  is a universal set of actions

# Comparing behaviours

Property satisfaction, specification conformance

**Simulation preorder:**  $\sqsubseteq \subseteq \mathcal{B} \times \mathcal{B}$ , s.t.

- it is preserved by  $\parallel$  ( $B_2 \sqsubseteq B_3 \Rightarrow B_1 \parallel B_2 \sqsubseteq B_1 \parallel B_3$ ),
- $\mathbf{0} \sqsubseteq B$ .

(Trace containment, simulation — not necessarily pre-congruences.)

Traces:  $B = (A, T)$ ,  $T \subseteq A^*$

$$B_1 \sqsubseteq B_2 \stackrel{\text{def}}{\iff} A_1 \subseteq A_2 \wedge T_1 \subseteq T_2 \lambda_{A_1} \quad (T \lambda_A \text{ — subsequences in } A^*)$$

Labelled Transition Systems:  $B = (Q, A, \rightarrow)$ ,  $\rightarrow \subseteq Q \times 2^A \times Q$

Consider the maximal relation  $R \subseteq Q_1 \times Q_2$  s.t.

$$q_1 R q_2 \Rightarrow \forall q_1 \xrightarrow{a}_1 q'_1, \exists q'_2 \in Q_2, b \subseteq A_2 : (q_2 \xrightarrow{b} q'_2 \wedge a \subseteq b \wedge q'_1 R q'_2)$$

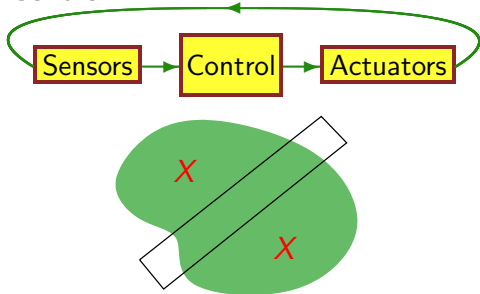
$$B_1 \sqsubseteq B_2 \stackrel{\text{def}}{\iff} (A_1 \subseteq A_2) \wedge (R \text{ is total on } Q_1)$$

# Composition as constraints

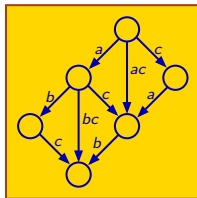
## Dataflow:



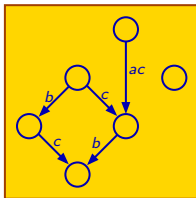
## Control:



## Concurrency coordination:



$$\begin{aligned} a &= c \\ b &\neq c \\ &\rightsquigarrow \end{aligned}$$



# Comparing behaviours & Composition operators

## Component substitutability

**Composition operators:**  $f : \mathcal{B}^n \rightarrow \mathcal{B}$

- $f(B_1, \dots, B_n) \sqsubseteq B_1 \parallel \dots \parallel B_n$ ,
- $B_i \preceq \tilde{B} \Rightarrow f(B_1, \dots, B_i, \dots, B_n) \preceq f(B_1, \dots, \tilde{B}, \dots, B_n)$ .

**Semantic preorder:**  $\preceq \subseteq \mathcal{B} \times \mathcal{B}$  s.t.

- $\preceq$  is preserved by  $\parallel$  (i.e.  $B_2 \preceq B_3 \Rightarrow B_1 \parallel B_2 \preceq B_1 \parallel B_3$ )

(Ready simulation, bisimulation relations — precongruences.)

**Meet operator:**  $\otimes : \mathcal{B} \times \mathcal{B} \rightarrow \mathcal{B}$  s.t.

- $(\mathcal{B}/\simeq, \preceq, \otimes)$ , with  $\simeq \stackrel{\text{def}}{=} \preceq \cap \preceq^{-1}$ , is a meet-semilattice.

Traces, LTS

$\preceq = \sqsubseteq$ . (Another example below.)

# Software architectures

Components:

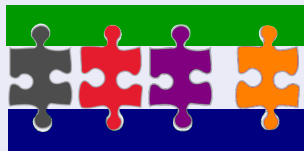


$\{B_i\}_{i=1}^4$

$f_1(B_1, B_2, B_3, B_4)$



$f_2(B_1, B_2, B_3, B_4)$



$(f_1 \otimes f_2)(B_1, B_2, B_3, B_4)$

# Behaviour type

## Summary

### Parallel composition

$(\mathcal{B}, \parallel, \mathbf{0})$  — a monoid (associative, not necessarily commutative).

### Simulation preorder

$\sqsubseteq \subseteq \mathcal{B} \times \mathcal{B}$ , s.t.

- it is preserved by  $\parallel$  ( $B_2 \sqsubseteq B_3 \Rightarrow B_1 \parallel B_2 \sqsubseteq B_1 \parallel B_3$ ),
- $\mathbf{0} \sqsubseteq B$ .

### Semantic preorder & meet operator

$\preceq \subseteq \mathcal{B} \times \mathcal{B}$  and  $\otimes : \mathcal{B} \times \mathcal{B} \rightarrow \mathcal{B}$ , s.t.

- $\preceq$  is preserved by  $\parallel$  ( $B_2 \preceq B_3 \Rightarrow B_1 \parallel B_2 \preceq B_1 \parallel B_3$ )
- $(\mathcal{B}/\simeq, \preceq, \otimes)$ , with  $\simeq \stackrel{\text{def}}{=} \preceq \cap \preceq^{-1}$ , is a meet-semilattice.



# Outline

- Why behaviour types?
- Definition
- Co-algebraic example
- Some speculation about glue

# Deterministic functor coalgebras

## $F$ -coalgebras:

$(S, f : S \rightarrow F(S))$ , where  $S$  — state-space,  $f$  — transition structure

## Some examples:

$M = (\mathbf{B} \times \mathbf{Id})^{\mathbf{A}}$  input-enabled Mealy machines

$D = \mathbb{B} \times (1 + \mathbf{Id})^{\mathbf{A}}$  deterministic automata (no initial state)

$N = \mathbb{B}^2 \times \mathcal{P}_{\omega}(\mathbf{Id})^{\mathbf{A}}$  non-deterministic automata

## Coalgebra homomorphism:

$$\begin{array}{ccc} S_1 & \xrightarrow{h} & S_2 \\ f_1 \downarrow & & \downarrow f_2 \\ F(S_1) & \xrightarrow{F(h)} & F(S_2) \end{array}$$

## Coalgebra bisimulation:

$$\begin{array}{ccccc} S_1 & \xleftarrow{\pi_1} & R & \xrightarrow{\pi_2} & S_2 \\ f_1 \downarrow & & g \downarrow & & \downarrow f_2 \\ F(S_1) & \xleftarrow{F(\pi_1)} & F(R) & \xrightarrow{F(\pi_2)} & F(S_2) \end{array}$$

## Non-Deterministic functors:

$F ::= \mathbf{Id} \mid \mathbf{B} \mid F \times F \mid F^{\mathbf{A}} \mid F + F \mid \mathcal{P}_{\omega}^*(F)$  ( $\mathbf{B}$  is a join-semilattice with  $\perp$ )

## Example: DF-coalgebra-based behaviour type (1/3)

### Parallel composition & zero behaviour

$$(S_1, f_1) \parallel (S_2, f_2) \stackrel{\text{def}}{=} (S_1 \times S_2, (s_1, s_2) \mapsto f_1(s_1) \boxtimes_{S_1, S_2}^F f_2(s_2))$$

$$x \boxtimes_{X, Y}^F y \stackrel{\text{def}}{=}$$

$$\begin{cases} (x, y), & F = \mathbf{Id}, \\ x \vee y, & F = \mathbf{B}, \\ \left( x_1 \boxtimes_{X, Y}^{F_1} y_1, x_2 \boxtimes_{X, Y}^{F_2} y_2 \right), & F = F_1 \times F_2, x = (x_1, x_2), y = (y_1, y_2), \\ \lambda a. \left( x(a) \boxtimes_{X, Y}^G y(a) \right), & F = G^{\mathbf{A}}. \end{cases}$$

$$\mathbf{0}_F = (1, f_F^0) \quad f_F^0(*) \stackrel{\text{def}}{=} \begin{cases} *, & F = \mathbf{Id}, \\ \perp, & F = \mathbf{B}, \\ (f_{F_1}^0(*), f_{F_2}^0(*)), & F = F_1 \times F_2, \\ \lambda a. f_G^0(*), & F = G^{\mathbf{A}}, \end{cases}$$

## Example: DF-coalgebra-based behaviour type (2/3)

### Simulation preorder

$$x \leq_R^F y \stackrel{\text{def}}{\iff} \begin{array}{l} R \subseteq S_1 \times S_2, x \in F(S_1), y \in F(S_2) \\ \left\{ \begin{array}{l} (x, y) \in R, \\ x \vee y = y, \\ x_1 \leq_R^{F_1} y_1 \wedge x_2 \leq_R^{F_2} y_2, \\ \forall a \in \mathbf{A}, x(a) \leq_R^G y(a), \end{array} \right. \begin{array}{l} F = \mathbf{Id}, \\ F = \mathbf{B}, \\ F = F_1 \times F_2, x = (x_1, x_2), y = (y_1, y_2), \\ F = G^{\mathbf{A}}, \end{array} \end{array}$$

$$(S_1, f_1) \sqsubseteq (S_2, f_2) \stackrel{\text{def}}{\iff} \exists R \subseteq S_1 \times S_2 : \text{total on } S_1 \text{ and s.t. } \forall (s_1, s_2) \in R, f_1(s_1) \leq_R^F f_2(s_2)$$

### Semantic preorder & meet operator

$$(S_1, f_1) \preceq (S_2, f_2) \stackrel{\text{def}}{\iff} \exists R \subseteq S_1 \times S_2 : \text{bisimulation total on } S_1$$
$$(S_1, f_1) \otimes (S_2, f_2) \stackrel{\text{def}}{=} (R, g) \text{ — maximal bisimulation (} g \text{ unique for DF)}$$

## Example: DF-coalgebra-based behaviour type (3/3)

**Proposition:**  $(DF/\simeq, \preceq, \otimes)$  is a meet-semilattice.

*Proof* — Let  $(R, g) = (S_1, f_1) \otimes (S_2, f_2)$  and  $(S, f) \preceq (S_1, f_1), (S_2, f_2)$

$$\begin{array}{ccccc} (T, h) & \longrightarrow & (R_1, g_1) & \longrightarrow & (S_1, f_1) \\ \downarrow & & \downarrow & & \uparrow \\ (R_2, g_2) & \longrightarrow & (S, f) & & \uparrow \\ \downarrow & & & & \uparrow \\ (S_2, f_2) & \longleftarrow & \longleftarrow & \longleftarrow & (R, g) \end{array}$$

- 1  $T = \{(s', s_1, s'', s_2) \mid (s', s_1) \in R_1, (s'', s_2) \in R_2 \text{ and } s' = s''\}$ .
- 2 Projection  $T_{2,4} \subseteq R$ .
- 3  $T \cong \{(s, s_1, s_2) \mid (s, s_1) \in R_1, (s, s_2) \in R_2\}$  — bisimulation on  $(S, f)$  and  $(R, g)$  total on  $S$ . □

# Outline

- Why behaviour types?
- Definition
- Co-algebraic example
- Some speculation about glue

# Memoryless glue operators

**Separation of concerns:** separating the computation of a system from the application of *glue operators* coordinating its atomic components.

- 1 A glue operator should not add *state* to the coordinated system.
- 2 The actions possible in a global state of the composed system should be completely determined by the properties of the corresponding states of the constituent subsystems.

**Glue operators:** Let  $(\mathcal{B}, \dots)$  be a behaviour type over  $F$ -coalgebras. Composition operator  $gl : \mathcal{B}^n \rightarrow \mathcal{B}$  is a *glue operator* iff there exists a natural transformation  $sync : F^n \rightarrow F$ , such that, for any  $\{B_i = (S_i, f_i)\}_{i=1}^n$ ,  $gl(B_1, \dots, B_n) = (S, f)$  with

- 1 
$$S = \prod_{i=1}^n S_i,$$

- 2 
$$f(s) = sync(f_1(s_1), \dots, f_n(s_n)), \text{ for all } s = (s_1, \dots, s_n) \in S.$$

# Some speculation

## Choice hypothesis

**Choice operator:**  $(S_1, f_1) + (S_2, f_2) \stackrel{\text{def}}{=} (S_1 + S_2, f_1 + f_2)$ .

**Hypothesis:** There is no glue operator  $g!$ , such that  $g! \simeq +$ .

/ Consider something like  $(\{q_1\}, \{q_1 \xrightarrow{a} q_1\})$  and  $(\{q_2\}, \{q_2 \xrightarrow{b} q_2\})$ . /



# Some speculation

## Choice hypothesis

**Choice operator:**  $(S_1, f_1) + (S_2, f_2) \stackrel{\text{def}}{=} (S_1 + S_2, f_1 + f_2)$ .

**Hypothesis:** There is no glue operator  $gl$ , such that  $gl \simeq +$ .

/ Consider something like  $(\{q_1\}, \{q_1 \xrightarrow{a} q_1\})$  and  $(\{q_2\}, \{q_1 \xrightarrow{b} q_2\})$ . /



# Some speculation

## Choice hypothesis

**Choice operator:**  $(S_1, f_1) + (S_2, f_2) \stackrel{\text{def}}{=} (S_1 + S_2, f_1 + f_2)$ .

**Hypothesis:** There is no glue operator  $gl$ , such that  $gl \simeq +$ .

/ Consider something like  $(\{q_1\}, \{q_1 \xrightarrow{a} q_1\})$  and  $(\{q_2\}, \{q_1 \xrightarrow{b} q_2\})$ . /



?



# Some speculation

## Choice hypothesis

**Choice operator:**  $(S_1, f_1) + (S_2, f_2) \stackrel{\text{def}}{=} (S_1 + S_2, f_1 + f_2)$ .

**Hypothesis:** There is no glue operator  $gl$ , such that  $gl \simeq +$ .

/ Consider something like  $(\{q_1\}, \{q_1 \xrightarrow{a} q_1\})$  and  $(\{q_2\}, \{q_1 \xrightarrow{b} q_2\})$ . /



?



$\simeq$



# Some speculation

## Choice hypothesis

**Choice operator:**  $(S_1, f_1) + (S_2, f_2) \stackrel{\text{def}}{=} (S_1 + S_2, f_1 + f_2)$ .

**Hypothesis:** There is no glue operator  $gl$ , such that  $gl \simeq +$ .

/ Consider something like  $(\{q_1\}, \{q_1 \xrightarrow{a} q_1\})$  and  $(\{q_2\}, \{q_1 \xrightarrow{b} q_2\})$ . /



?



$\simeq$



**Hypothesis 2:** For any glue operator  $gl$ ,

$$gl(B_1, B_2) \simeq B_1 + B_2 \implies B_1 \simeq B_2.$$