# A model of Asymmetric Replicated State Machines

Roland Kuhn@ Actyx
Daniela Marottoli @ UBA    Hernán Melgratti @ UBA
Emilio Tuosto @ GSSI

Dagstuhl seminar 21372
Behavioural Types: Bridging Theory and Practice

September 12–17, 2021

Research partly supported by the EU H2020 RISE programme under the Marie Skłodowska-Curie grant agreement No 778233.

BEHAPI Behavioural Application Program Interfaces

# Take-away message

We define behavioural specs that

- feature
  - pub-subscribe (instead point-to-point)
  - (generalised) choices
  - arbitrary (and variable) number of instances
- trade coordination for availability
- trade “old” properties (eg. session fidelity) for new ones (eventual-consistency)

# Types: Syntax

## LoGal types

$$\mathtt{G} \overset{\text{co}}{::=} \sum_{i \in I} \mathtt{c}_i @ \mathtt{R}_i \langle \mathtt{l}_i \rangle \,.\, \mathtt{G}_i$$

## Machines

$$\mathtt{M} \overset{\text{co}}{::=} \kappa \cdot [\mathtt{t}_1 ? \mathtt{M}_1 \ \& \ \cdots \ \& \ \mathtt{t}_n ? \mathtt{M}_n]$$

# Types: Syntax

## LoGal types

$$G \overset{co}{::=} \sum_{i \in I} c_i @ R_i \langle l_i \rangle \,.\, G_i$$

$$G = \text{publish}@A\langle p \rangle \,.\, G'$$
$$G' = \text{bid}@B\langle b \rangle \,.\, G'$$
$$+$$
$$\text{select}@A\langle s \rangle \,.\, \text{finish}@A\langle f \rangle \,.\, 0$$

## Machines

$$M \overset{co}{::=} \kappa \cdot [t_1 ? M_1 \;\&\; \cdots \;\&\; t_n ? M_n]$$

$$M_A = \{\text{publish} / p\} \cdot [p ? M_A{}']$$
$$M_A{}' = \{\text{select} / s\} \cdot [b ? M_A{}' \;\&\; s ? \{\text{finish} / f\} \cdot f ? 0]$$
$$M_B = p ? M_B{}'$$
$$M_B{}' = \{\text{bid} / b\} \cdot [b ? M_B{}' \;\&\; s ? f ? 0]$$

# Types: Syntax

## LoGal types

$$\mathtt{G} \overset{\text{co}}{::=} \sum_{i \in I} \mathsf{c}_i @ \mathtt{R}_i \langle \mathtt{l}_i \rangle . \mathtt{G}_i$$

$$\mathtt{G} = \mathsf{publish} @ \mathtt{A} \langle \mathtt{p} \rangle . \mathtt{G}'$$

$$\mathtt{G}' = \mathsf{bid} @ \mathtt{B} \langle \mathtt{b} \rangle . \mathtt{G}' + \mathsf{select} @ \mathtt{A} \langle \mathtt{s} \rangle . \mathsf{finish} @ \mathtt{A} \langle \mathtt{f} \rangle . 0$$

## Machines

$$\mathtt{M} \overset{\text{co}}{::=} \kappa \cdot [\mathtt{t}_1 ? \mathtt{M}_1 \ \& \ \cdots \ \& \ \mathtt{t}_n ? \mathtt{M}_n]$$

$$\mathtt{M}_A = \{\mathsf{publish} / \mathtt{p}\} \cdot [\mathtt{p} ? \mathtt{M}_A{}']$$

$$\mathtt{M}_A{}' = \{\mathsf{select} / \mathtt{s}\} \cdot [\mathtt{b} ? \mathtt{M}_A{}' \ \& \ \mathtt{s} ? \{\mathsf{finish} / \mathtt{f}\} \cdot \mathtt{f} ? 0]$$

$$\mathtt{M}_B = \mathtt{p} ? \mathtt{M}_B{}'$$

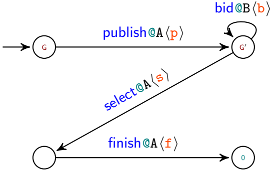
$$\mathtt{M}_B{}' = \{\mathsf{bid} / \mathtt{b}\} \cdot [\mathtt{b} ? \mathtt{M}_B{}' \ \& \ \mathtt{s} ? \mathtt{f} ? 0]$$

# Types: Syntax

## LoGal types

$$G \overset{co}{::=} \sum_{i\in I} c_i @ R_i \langle l_i \rangle \,.\, G_i$$

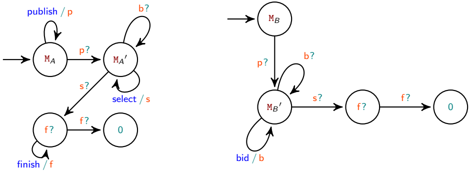
$$G = \text{publish}@A\langle p\rangle \,.\, G'$$
$$G' = \text{bid}@B\langle b\rangle \,.\, G' + \text{select}@A\langle s\rangle \,.\, \text{finish}@A\langle f\rangle \,.\, 0$$




## Machines

$$M \overset{co}{::=} \kappa \cdot [t_1 ? M_1 \;\&\; \cdots \;\&\; t_n ? M_n]$$

$$M_A = \{\text{publish} / p\} \cdot [p ? {M_A}']$$
$${M_A}' = \{\text{select} / s\} \cdot [b ? {M_A}' \;\&\; s ? \{\text{finish} / f\} \cdot f ? 0]$$
$$M_B = p ? {M_B}'$$
$${M_B}' = \{\text{bid} / b\} \cdot [b ? {M_B}' \;\&\; s ? f ? 0]$$



# Types: Semantics...intuitively

- Types "produce/consume" events

  LoGal types: how/when roles produce events

  Machines: how/when instances consume events "skipping" those irrelevant events to them

- Deterministic types only

  LoGal types: log types of branches have no common prefixes

  Machines: event types of branches are pairwise distinct

- Non-deterministic events' propagation

# Types: Semantics...formally

## LoGal types

$$\frac{\delta(\mathsf{G}, l) \xrightarrow{\mathtt{c}\,/\,\mathtt{1}} \mathsf{G}' \qquad \vdash l' : \mathtt{1} \qquad l' \textit{ fresh}}{(\mathsf{G}, l) \xrightarrow{\mathtt{c}\,/\,\mathtt{1}} (\mathsf{G}, l \cdot l')}$$

## Machines

$$\frac{\delta(\mathtt{M}, l) = \mathtt{M}' \qquad \mathtt{M}' \downarrow_{\mathtt{c}\,/\,\mathtt{1}} \qquad \vdash l' : \mathtt{1} \qquad l' \text{ fresh}}{(\mathtt{M}, l) \xrightarrow{\mathtt{c}\,/\,\mathtt{1}} (\mathtt{M}, l \cdot l')}$$

# Types: Semantics...formally

## LoGal types

$$\frac{\delta(\mathsf{G}, l) \xrightarrow{\mathsf{c}\,/\,\mathtt{l}} \mathsf{G}' \qquad \vdash l' : \mathtt{l} \qquad l' \textit{ fresh}}{(\mathsf{G}, l) \xrightarrow{\mathsf{c}\,/\,\mathtt{l}} (\mathsf{G}, l \cdot l')}$$

## where

$l$ is an (idealised) global/shared log

$$\sum_{i \in I} \mathsf{c}_i @ \mathrm{R}_i \langle \mathtt{l}_i \rangle \,.\, \mathsf{G}_i \xrightarrow{\mathsf{c}_i\,/\,\mathtt{l}_i} \mathsf{G}_i \qquad i \in I$$

$$\delta(\mathsf{G}, \epsilon) = \mathsf{G}$$

$$\delta(\mathsf{G}, l) = \begin{cases} \delta(\mathsf{G}', l \cdot l') & \text{if } \mathsf{G} \xrightarrow{\mathsf{c}\,/\,\mathtt{l}} \mathsf{G}', l \neq \epsilon, \vdash l' : \mathtt{l} \\ \bot & \text{otherwise} \end{cases}$$

## Machines

$$\frac{\delta(\mathtt{M}, l) = \mathtt{M}' \qquad \mathtt{M}' \downarrow_{\mathsf{c}\,/\,\mathtt{l}} \qquad \vdash l' : \mathtt{l} \qquad l' \text{ fresh}}{(\mathtt{M}, l) \xrightarrow{\mathsf{c}\,/\,\mathtt{l}} (\mathtt{M}, l \cdot l')}$$

## where

$l$ is the local log accessible to $\mathtt{M}$

$\mathtt{M}' \downarrow_{\mathsf{c}\,/\,\mathtt{l}} \iff \mathsf{c}\,/\,\mathtt{l}$ enabled at $\mathtt{M}'$

$$\delta(\mathtt{M}, \epsilon) = \mathtt{M}$$

$$\delta(\mathtt{M}, e \cdot l) = \begin{cases} \delta(\mathtt{M}_j, l) & \text{if } \vdash e : \mathtt{t_j}, \\ & \mathtt{M} = \kappa \cdot [\ldots \;\&\; \mathtt{t}_j ?\, \mathtt{M}_j \;\&\; \ldots] \\ \delta(\mathtt{M}, l) & \text{otherwise} \end{cases}$$

# Systems

## Systems: finitely many machines with local logs + global log

$$(\mathtt{S}, l) = (\mathtt{M}_1, l_1) \mid \ldots \mid (\mathtt{M}_n, l_n) \mid l$$

(BTW: the global log is an optical illusion)

Events univocally associated to the machines generating them: $l_1 \sqsubseteq l_2 \iff$ there is an order-preserving and downward-total morphism from $l_1$ into $l_2$ on events of a same machine

## Well-formedness

A system $(\mathtt{M}_1, l_1) \mid \ldots \mid (\mathtt{M}_n, l_n) \mid l$ is well-formed if

$$\text{for all } i,\ l_i \sqsubseteq l \qquad \text{and} \qquad l = \bigcup_{i \in \underline{n}} l_i$$

# Systems' semantics...intuitively

- # Events' generation

  The local log of a machine is extended with the fresh events generated by the machine

- # Events' propagation

  Emitted events propagate asynchronously & non-deterministically

# Systems' semantics: formally

**[Local]**

$$\frac{i \in \mathrm{dom}\, \mathtt{S} \qquad \mathtt{S}(i) = (\mathtt{M}, l_i) \qquad (\mathtt{M}, l_i) \xrightarrow{\mathtt{c}\,/\,\mathtt{l}} (\mathtt{M}, l'_i) \qquad l' \in l \bowtie l'_i}{(\mathtt{S}, l) \xrightarrow{\mathtt{c}\,/\,\mathtt{l}} (\mathtt{S}[i \mapsto (\mathtt{M}, l'_i)], l')}$$

where

$$l_1 \bowtie l_2 = \{ l \mid l \subseteq l_1 \cup l_2 \;\wedge\; l_1 \sqsubseteq l \;\wedge\; l_2 \sqsubseteq l \}$$

**[Prop]**

$$\frac{i \in \mathrm{dom}\, \mathtt{S} \qquad \mathtt{S}(i) = (\mathtt{M}, l_i) \qquad l_i \sqsubseteq l' \sqsubseteq l \qquad l_i \subset l'}{(\mathtt{S}, l) \xrightarrow{\tau} (\mathtt{S}[i \mapsto (\mathtt{M}, l')], l)}$$

# Semantics at work (I)

If

$$(\mathtt{B}, b_1) \xrightarrow{\mathtt{c}\,/\,\mathtt{l}} (\mathtt{B}, b_1 \cdot b_2 \cdot b_3) \qquad \text{with} \qquad \vdash b_2 \cdot b_3 : \mathtt{l}$$

then, by [LOCAL],

$$(\mathtt{A}, a) \mid (\mathtt{B}, b_1) \mid (\mathtt{C}, c) \mid a \cdot b_1 \cdot c \xrightarrow{\mathtt{c}\,/\,\mathtt{l}} (\mathtt{A}, a) \mid (\mathtt{B}, b_1 \cdot b_2 \cdot b_3) \mid (\mathtt{C}, c) \mid l'$$

for all

$$\begin{aligned} l' &\in (a \cdot b_1 \cdot c) \bowtie (b_1 \cdot b_2 \cdot b_3) \\ &= \{a \cdot b_1 \cdot c \cdot b_2 \cdot b_3,\ a \cdot b_1 \cdot b_2 \cdot c \cdot b_3,\ a \cdot b_1 \cdot b_2 \cdot b_3 \cdot c\} \end{aligned}$$

# Semantics at work (II)

Consider the (well-formed) system

$$\mathtt{S} = (\mathtt{A}, a) \mid (\mathtt{B}, b_1 \cdot b_2 \cdot b_3) \mid (\mathtt{C}, c) \mid a \cdot b_1 \cdot b_2 \cdot c \cdot b_3$$

Then, by rule [Prop],

$$\mathtt{S} \xrightarrow{\tau} (\mathtt{A}, a) \mid (\mathtt{B}, b_1 \cdot b_2 \cdot b_3) \mid (\mathtt{C}, a \cdot c) \mid a \cdot b_1 \cdot b_2 \cdot c \cdot b_3$$

or

$$\mathtt{S} \xrightarrow{\tau} (\mathtt{A}, a) \mid (\mathtt{B}, b_1 \cdot b_2 \cdot b_3) \mid (\mathtt{C}, c \cdot b_1) \mid a \cdot b_1 \cdot b_2 \cdot c \cdot b_3$$

but

$$\mathtt{S} \xrightarrow{\tau} (\mathtt{A}, a) \mid (\mathtt{B}, b_1 \cdot b_2 \cdot b_3) \mid (\mathtt{C}, c \cdot b_2) \mid a \cdot b_1 \cdot b_2 \cdot c \cdot b_3$$

# Properties of our semantics

**Well-Formedness preservation**

[LOCAL] & [PROP] preserve well-formedness

**Eventual Consistency**

If

$$\mathtt{S} = (\mathtt{M}_1, I_1) \mid \ldots \mid (\mathtt{M}_n, I_n) \mid I \text{ is well-formed}$$

then

$$\mathtt{S} \xrightarrow{\tau}^{\star} (\mathtt{M}_1, I) \mid \ldots \mid (\mathtt{M}_n, I) \mid I$$

# On realisation (I)

It is hard to get it right (even without multi-instances or choices!)

**A trivial protocol**

Take

$$G = c_1@R_1\langle t_1\rangle \,.\, c_2@R_2\langle t_2\rangle \,.\, 0$$

Do

$$M_1 = \{c_1 \,/\, t_1\}\cdot 0 \qquad \text{and} \qquad M_2 = t_1?\,\{c_2 \,/\, t_2\}\cdot 0$$

realise $G$?

# On realisation (I)

It is hard to get it right (even without multi-instances or choices!)

**A trivial protocol**

Take

$$G = c_1@R_1\langle t_1\rangle \,.\, c_2@R_2\langle t_2\rangle \,.\, 0$$

Do

$$M_1 = \{c_1 \,/\, t_1\}\cdot 0 \qquad \text{and} \qquad M_2 = t_1?\,\{c_2 \,/\, t_2\}\cdot 0$$

(correctly) realise $G$?

# On realisation (II)

### Well-formedness of loGal types

Each guard, say $\mathtt{l}_i$, should be

- causal consistent
  - each selector in (the continuation of) $\mathtt{l}_i$ reacts to $\mathtt{l}_i$
  - each role involved in the continuation of $\mathtt{l}_i$ cannot react to more events on $\mathtt{l}_i$ than selectors on the branch
- determined
  - each role in the continuation of $\mathtt{l}_i$ reacts to $\mathtt{l}_i[0]$
  - selectors in the continuation of $\mathtt{l}_i$ react to the same set of event types in $\mathtt{l}_i$
- confusion-free
  - guards of different branches start with distinct event types
  - an event type cannot occur in more than one guard

# Conclusions

- reference documentation for Actyx's developers
- possibly useful to derive "minimal" subscriptions
- projectable global specs
- tools / develop typing
- compensations (hence causality tracking) / active monitoring?
- failures

Thank you!