Behavioural, Functional, and Non-Functional Contracts for Dynamic Selection of Services

Carlos G. Lopez Pombo
@UNRN&CONICET

Hernán Melgratti
@UBA&CONICET

Agustín E. Martinez-Suñé
@ University of Oxford

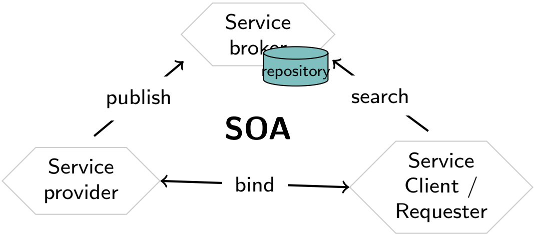
Diego Senarruzza Anabia
@UBA

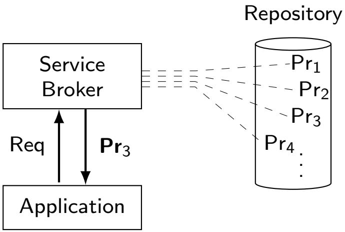
Emilio Tuosto
@ GSSI

Work partly supported by the PRIN 2022 PNRR project DeLiCE (F53D23009130001)

In the next few minutes...

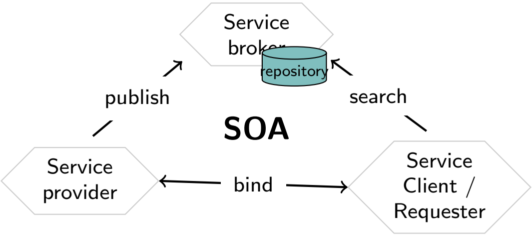
We'll talk about dynamic discovery&binding of service

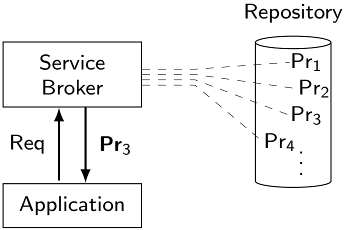




In the next few minutes...

We'll talk about dynamic discovery&binding of service

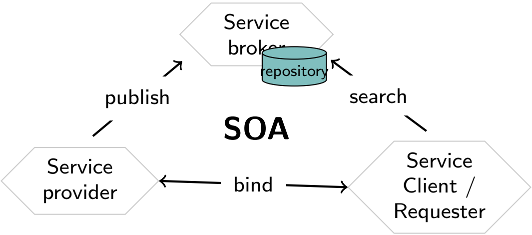


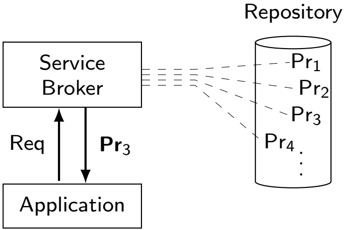


Prologue . . . . A bird-eye view

In the next few minutes...

We'll talk about dynamic discovery&binding of service



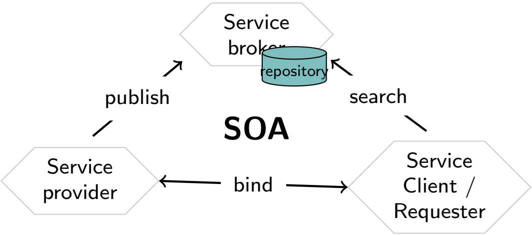


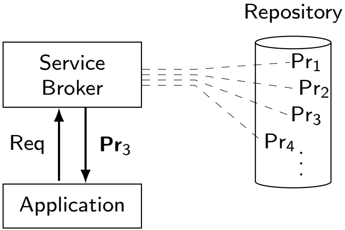
Prologue . . . . A bird-eye view

Act I . . . . A proposal

In the next few minutes...

We'll talk about dynamic discovery&binding of service





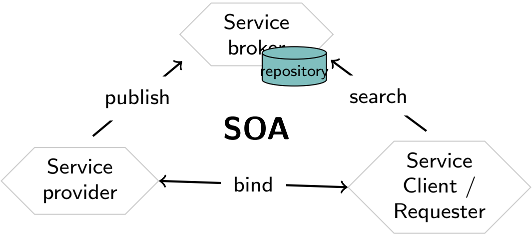
Prologue . . . . A bird-eye view

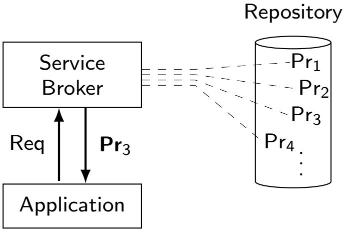
Act I . . . . A proposal

Act III . . . . Some conclusions

In the next few minutes...

We'll talk about dynamic discovery&binding of service



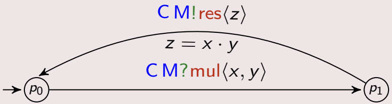


– Prologue –

[Contracts & SOAs]

This talk in 1 slide

The problem

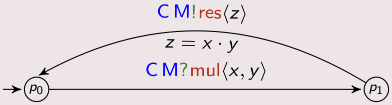


This talk in 1 slide

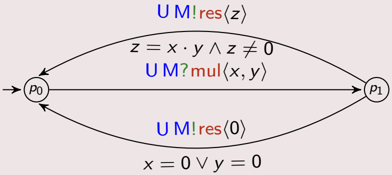
The problem

This talk in 1 slide

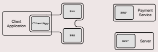
The problem



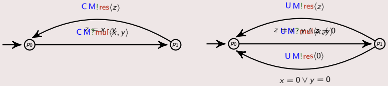
$\stackrel{?}{\simeq}$



Our proposal: [1, 5] + [2] + [3, 4]



+



+ QoS-CFSMs

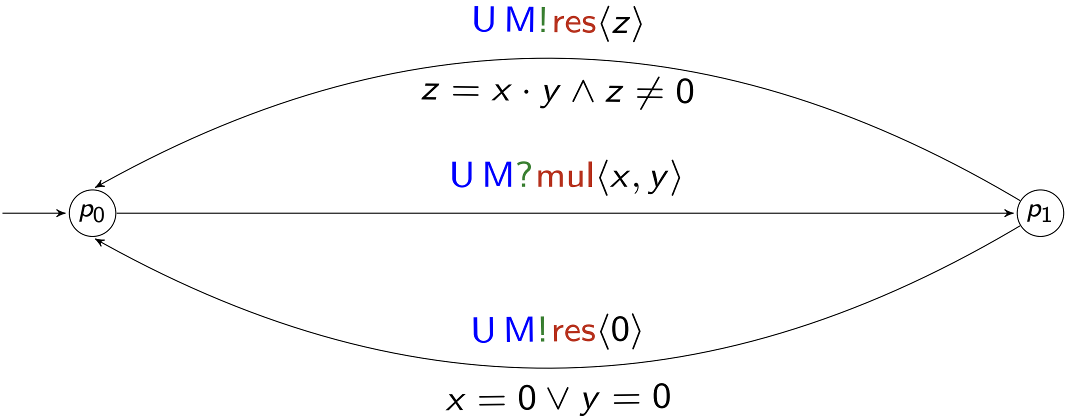
which yields

- A uniform model for discovery based on behavioural and (non-)functional contracts
- Bisimulation-based compliance
- A bisimulation algorithm of service compliance modulo name matching

Preliminaries

Our behavioural and functional contracts

We essentially borrow (with some adaptation) asserted CFSM from [2]



Our non-functional contracts

A variant of CFSMs yields our behavioural contracts; our non-functional contracts are

$$\underline{\text{QoS constraints}} = \text{FOL}_{=} + \text{Real Close Fields}$$

Our non-functional contracts

A variant of CFSMs yields our behavioural contracts; our non-functional contracts are

$$\underline{\text{QoS constraints}} = \text{FOL}_{=} + \text{Real Close Fields}$$

where RCF are totally ordered fields such that

- positive elements have square roots
- polynomial of odd degrees have zeros

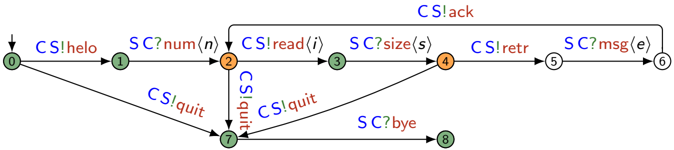
Our non-functional contracts

A variant of CFSMs yields our behavioural contracts; our non-functional contracts are

$$\text{QoS constraints} = \text{FOL}_{=} + \text{Real Close Fields}$$

where RCF are totally ordered fields such that

- positive elements have square roots
- polynomial of odd degrees have zeros



$$\Gamma_{\text{Low}} = \{\mathsf{t} \leq 0.01,\ \mathsf{c} \leq 0.01,\ \mathsf{m} \leq 0.01\}$$

$$\Gamma_{\text{DB}} = \{\mathsf{t} \leq 3 \implies (\exists x)(0.5 \leq x \leq 1 \wedge \mathsf{c} = \mathsf{t} \cdot x),\ \mathsf{t} > 3 \implies \mathsf{c} = 10,\ \mathsf{m} \leq 5\}$$

$\dots$

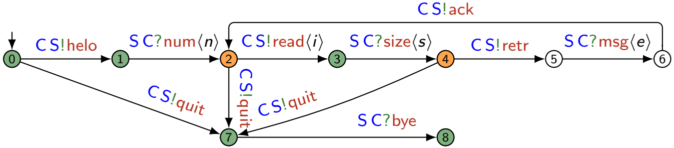
Our non-functional contracts

A variant of CFSMs yields our behavioural contracts; our non-functional contracts are

$$\underline{\text{QoS constraints}} = \text{FOL}_{=} + \text{Real Close Fields}$$

where RCF are totally ordered fields such that

- positive elements have square roots
- polynomial of odd degrees have zeros



$$\Gamma_{\text{Low}} = \{\mathsf{t} \leq 0.01,\ \mathsf{c} \leq 0.01,\ \mathsf{m} \leq 0.01\}$$

$$\Gamma_{\text{DB}} = \{\mathsf{t} \leq 3 \implies (\exists x)(0.5 \leq x \leq 1 \land \mathsf{c} = \mathsf{t} \cdot x),\ \mathsf{t} > 3 \implies \mathsf{c} = 10,\ \mathsf{m} \leq 5\}$$

$\dots$

RCFs allow us to formalise QoS aggregation operators [3, 4]

– Act I –

[A proposal]

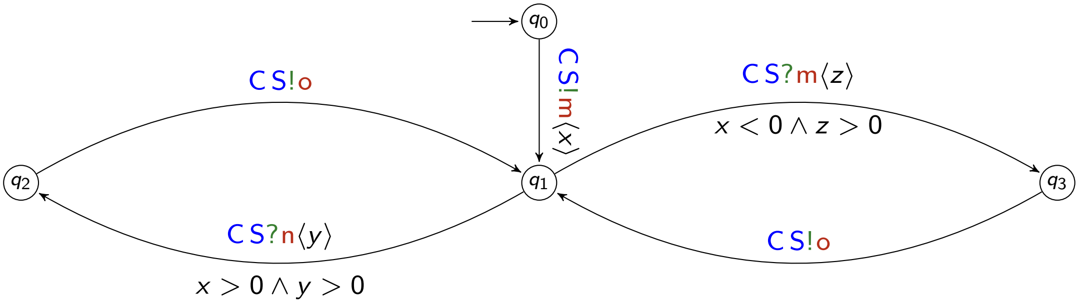
Extended CFSMs

An extended CFSM (e-CFSM for short) is a tuple $\langle M, F, \text{qos}, \text{asrt}\rangle$ where:

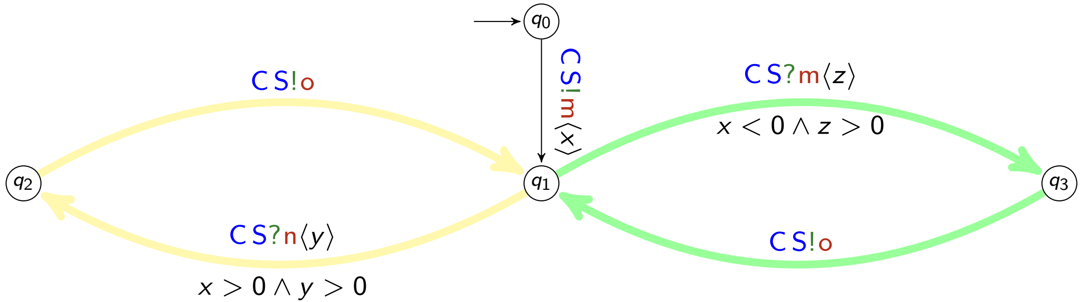
- $M = \langle Q, q_0, \rightarrow\rangle$ is a CFSM with $F \subseteq Q$ the set of final states ,
- asrt maps transitions of M to first-order formulae in $\mathcal{F}(\Sigma)$, and
- $\text{qos} : Q \rightarrow \mathcal{C}$ maps states of M to QoS specifications.

An extended communicating system is a map $(M_{\mathsf{A}})_{\mathsf{A}\in\mathcal{P}}$ assigning an A-local e-CFSM M_{A} to each $\mathsf{A} \in \mathcal{P}$.

Oddities

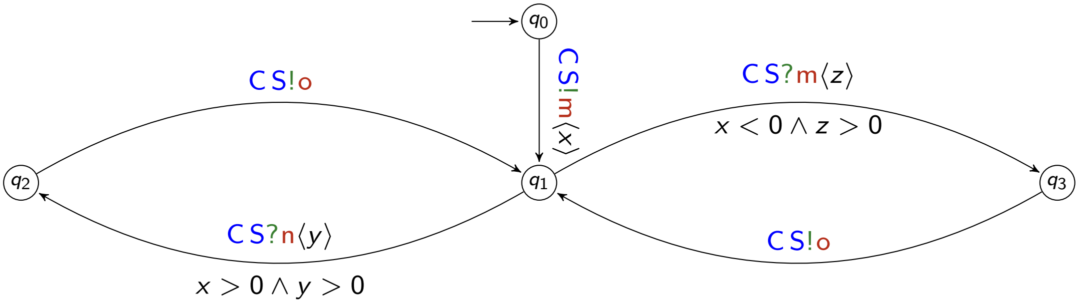


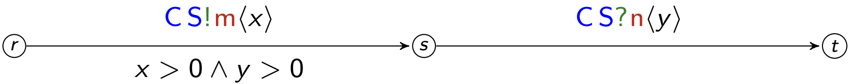
Oddities



q_0
$C\,S!m\langle x\rangle$
$C\,S!o$
$C\,S?m\langle z\rangle$
$x < 0 \wedge z > 0$
q_2
q_1
q_3
$C\,S?n\langle y\rangle$
$C\,S!o$
$x > 0 \wedge y > 0$

Oddities


q_0
q_1
q_2
q_3
C S!m$\langle x \rangle$
C S!o
C S?m$\langle z \rangle$
$x < 0 \land z > 0$
C S?n$\langle y \rangle$
$x > 0 \land y > 0$
C S!o


r
s
t
C S!m$\langle x \rangle$
$x > 0 \land y > 0$
C S?n$\langle y \rangle$

Knowledge

The residual of an assertion ϕ after l, written $\phi \mathbin{\overline{\wedge}} l$, is $\bot$ unless

$$
\begin{aligned}
p(x_1, \ldots, x_n) \mathbin{\overline{\wedge}} l &= p(x_1, \ldots, x_n) && \text{if } var(l) \cap \{x_1, \ldots, x_n\} = \emptyset \\
(\neg\phi) \mathbin{\overline{\wedge}} l &= \neg(\phi \mathbin{\overline{\wedge}} l) && \text{if } \phi \mathbin{\overline{\wedge}} l \neq \bot \\
(\phi_1 \vee \phi_2) \mathbin{\overline{\wedge}} l &= (\phi_1 \mathbin{\overline{\wedge}} l) \vee (\phi_2 \mathbin{\overline{\wedge}} l) && \text{if } \phi_1 \mathbin{\overline{\wedge}} l \neq \bot \text{ and } \phi_2 \mathbin{\overline{\wedge}} l \neq \bot \\
((\exists x)\phi) \mathbin{\overline{\wedge}} l &= (\exists x)(\phi \mathbin{\overline{\wedge}} l) && \text{if } x \notin var(l) \text{ and } \phi \mathbin{\overline{\wedge}} l \neq \bot \\
((\exists x)\phi) \mathbin{\overline{\wedge}} l &= ((\exists x)\phi) \mathbin{\overline{\wedge}} l((\exists y)(\phi[y/x])) \mathbin{\overline{\wedge}} l && \text{if } x \in var(l),\ y \text{ fresh, and } \phi \mathbin{\overline{\wedge}} l \neq \bot
\end{aligned}
$$

Knowledge

The residual of an assertion ϕ after l, written $\phi \bar{\wedge} l$, is $\bot$ unless

$$
\begin{aligned}
p(x_1, \ldots, x_n) \bar{\wedge} l &= p(x_1, \ldots, x_n) && \text{if } var(l) \cap \{x_1, \ldots, x_n\} = \emptyset \\
(\neg\phi) \bar{\wedge} l &= \neg(\phi \bar{\wedge} l) && \text{if } \phi \bar{\wedge} l \neq \bot \\
(\phi_1 \vee \phi_2) \bar{\wedge} l &= (\phi_1 \bar{\wedge} l) \vee (\phi_2 \bar{\wedge} l) && \text{if } \phi_1 \bar{\wedge} l \neq \bot \text{ and } \phi_2 \bar{\wedge} l \neq \bot \\
((\exists x)\phi) \bar{\wedge} l &= (\exists x)(\phi \bar{\wedge} l) && \text{if } x \notin var(l) \text{ and } \phi \bar{\wedge} l \neq \bot \\
((\exists x)\phi) \bar{\wedge} l &= ((\exists x)\phi) \bar{\wedge} l((\exists y)(\phi[y/x])) \bar{\wedge} l && \text{if } x \in var(l),\ y \text{ fresh, and } \phi \bar{\wedge} l \neq \bot
\end{aligned}
$$

The knowledge $\mathcal{K}(\pi)$ on π is $K(\pi, \{True\})$ where

$$
K(\pi, X) = \begin{cases} \bigwedge_{\psi \in X} \psi, & \pi \text{ is the empty path} \\ K(\pi', \{\psi \mid \psi \in X \text{ and } \psi \bar{\wedge} l \neq \bot\} \cup \{\phi\}), & \pi = q \xrightarrow[\phi]{l} \pi' \end{cases}
$$

Bisimulating e-CFSMs

Let M_1 and M_2 be two e-CFSMs respectively with states Q_1 and Q_2 and initial states $p_0 \in Q_1$ and $q_0 \in Q_2$.

Bisimulating e-CFSMs

Let M_1 and M_2 be two e-CFSMs respectively with states Q_1 and Q_2 and initial states $p_0 \in Q_1$ and $q_0 \in Q_2$.
A relation $\mathcal{R} \subseteq (Q_1 \times \mathrm{FOL}_=) \times (Q_2 \times \mathrm{FOL}_=)$ is a simulation if $\big(p, K\big)\mathcal{R}\big(q, K'\big)$ and $p \xrightarrow[\phi]{l} p'$ in M_1 imply that there is $T = \{q \xrightarrow[\psi_1]{l} q_1, \ldots, q \xrightarrow[\psi_k]{l} q_k\} \neq \emptyset$ in M_2 and

Bisimulating e-CFSMs

Let M_1 and M_2 be two e-CFSMs respectively with states Q_1 and Q_2 and initial states $p_0 \in Q_1$ and $q_0 \in Q_2$.

A relation $\mathcal{R} \subseteq (Q_1 \times \mathrm{FOL}_=) \times (Q_2 \times \mathrm{FOL}_=)$ is a simulation if $\big(p, K\big)\mathcal{R}\big(q, K'\big)$ and $p \xrightarrow[\phi]{l} p'$ in M_1 imply that there is $T = \{q \xrightarrow[\psi_1]{l} q_1, \ldots, q \xrightarrow[\psi_k]{l} q_k\} \neq \emptyset$ in M_2 and

1. $\neg\big(((K \,\overline{\wedge}\, l) \wedge \phi) \implies \bigvee_{q \xrightarrow[\psi]{l} q' \in T} ((K' \,\overline{\wedge}\, l) \wedge \psi)\big)$ unsat

Bisimulating e-CFSMs

Let M_1 and M_2 be two e-CFSMs respectively with states Q_1 and Q_2 and initial states $p_0 \in Q_1$ and $q_0 \in Q_2$.
A relation $\mathcal{R} \subseteq (Q_1 \times \mathrm{FOL}_=) \times (Q_2 \times \mathrm{FOL}_=)$ is a simulation if $(p, K)\mathcal{R}(q, K')$ and $p \xrightarrow[\phi]{l} p'$ in M_1 imply that there is $T = \{q \xrightarrow[\psi_1]{l} q_1, \ldots, q \xrightarrow[\psi_k]{l} q_k\} \neq \emptyset$ in M_2 and

1. $\neg(((K \,\overline{\wedge}\, l) \wedge \phi) \implies \bigvee_{q \xrightarrow[\psi]{l} q' \in T}((K' \,\overline{\wedge}\, l) \wedge \psi))$ unsat
2. for all $q \xrightarrow[\psi]{l} q' \in T$, $(p', \overline{(K \,\overline{\wedge}\, l) \wedge \phi \wedge \psi})\mathcal{R}(q', \overline{(K' \,\overline{\wedge}\, l) \wedge \psi})$

Bisimulating e-CFSMs

Let M_1 and M_2 be two e-CFSMs respectively with states Q_1 and Q_2 and initial states $p_0 \in Q_1$ and $q_0 \in Q_2$.
A relation $\mathcal{R} \subseteq (Q_1 \times \mathrm{FOL}_=) \times (Q_2 \times \mathrm{FOL}_=)$ is a simulation if $\big(p, K\big)\mathcal{R}\big(q, K'\big)$ and $p \xrightarrow[\phi]{l} p'$ in M_1 imply that there is $T = \{q \xrightarrow[\psi_1]{l} q_1, \ldots, q \xrightarrow[\psi_k]{l} q_k\} \neq \emptyset$ in M_2 and

1. $\neg\big(((K \,\overline{\wedge}\, l) \wedge \phi) \implies \bigvee_{q \xrightarrow[\psi]{l} q' \in T}((K' \,\overline{\wedge}\, l) \wedge \psi)\big)$ unsat
2. for all $q \xrightarrow[\psi]{l} q' \in T$, $\big(p', \overline{(K \,\overline{\wedge}\, l) \wedge \phi \wedge \psi}\big)\mathcal{R}\big(q', \overline{(K' \,\overline{\wedge}\, l) \wedge \psi}\big)$
3. if $p \in F_1$, then $q \in F_2$ and $\mathrm{qos}(p) = \langle \Sigma, \Gamma_1 \rangle$ and $\mathrm{qos}(q) = \langle \Sigma, \Gamma_2 \rangle$ then $\neg\big(\bigwedge_{\phi \in \Gamma_1} \phi \implies \bigwedge_{\phi \in \Gamma_2} \phi\big)$ unsat

Bisimulating e-CFSMs

Let M_1 and M_2 be two e-CFSMs respectively with states Q_1 and Q_2 and initial states $p_0 \in Q_1$ and $q_0 \in Q_2$.

A relation $\mathcal{R} \subseteq (Q_1 \times \text{FOL}_=) \times (Q_2 \times \text{FOL}_=)$ is a simulation if $\big(p, K\big)\mathcal{R}\big(q, K'\big)$ and $p \xrightarrow[\phi]{l} p'$ in M_1 imply that there is $T = \{q \xrightarrow[\psi_1]{l} q_1, \dots, q \xrightarrow[\psi_k]{l} q_k\} \neq \emptyset$ in M_2 and

1. $\neg\big(((K \,\overline{\wedge}\, l) \wedge \phi) \implies \bigvee_{q \xrightarrow[\psi]{l} q' \in T} ((K' \,\overline{\wedge}\, l) \wedge \psi)\big)$ unsat
2. for all $q \xrightarrow[\psi]{l} q' \in T$, $\big(p', \overline{(K \,\overline{\wedge}\, l) \wedge \phi \wedge \psi}\big)\mathcal{R}\big(q', \overline{(K' \,\overline{\wedge}\, l) \wedge \psi}\big)$
3. if $p \in F_1$, then $q \in F_2$ and $\text{qos}(p) = \langle \Sigma, \Gamma_1 \rangle$ and $\text{qos}(q) = \langle \Sigma, \Gamma_2 \rangle$ then $\neg\big(\bigwedge_{\phi \in \Gamma_1} \phi \implies \bigwedge_{\phi \in \Gamma_2} \phi\big)$ unsat

M_2 simulates M_1 if there is a simulation $\mathcal{R}$ such that $\big(p_0, \mathit{True}\big)\mathcal{R}\big(q_0, \mathit{True}\big)$.

– Act II –

[Some conclusions]

In the paper

- an example on the POP protocol
- Bisimulation checking algorithm module name matching

In the paper

- an example on the POP protocol
- Bisimulation checking algorithm module name matching

We are planning to

- extend **SEArch** to support
- generalise to name embeddings

In the paper

an example on the POP protocol

Bisimulation checking algorithm module name matching

We are planning to

extend **SEArch** to support

generalise to name embeddings

Some doubts...

is the approach feasible? (From one reviewer)

isn't bisimulation to strong a notion for compliance?

Thank you

References I

[1] and. An interface theory for service-oriented design.
TCS, 503:1–30, 2013.

[2] L. Gheri, I. Lanese, N. Sayers, E. Tuosto, and N. Yoshida. Design-By-Contract for Flexible Multiparty Session Protocols.
In K. Ali and J. Vitek, editors, *36th European Conference on Object-Oriented Programming, ECOOP 2022, June 6-10, 2022, Berlin, Germany*, volume 222 of *LIPIcs*, pages 8:1–8:28. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022.

[3] C. Pombo, A. Suñé, and E. Tuosto. A dynamic temporal logic for quality of service in choreographic models.
In E. Ábrahám, C. Dubslaff, and S. Tarifa, editors, *Theoretical Aspects of Computing – ICTAC 2023*, pages 119–138. Springer, 2023.

References II

[4] C. L. Pombo, A. E. M. Suñé, and E. Tuosto. A dynamic temporal logic for quality of service in choreographic models.
Theor. Comput. Sci., 1043:115247, 2025.

[5] I. Vissani, C. G. L. Pombo, and E. Tuosto. Communicating machines as a dynamic binding mechanism of services.
In S. Gay and J. Alglave, editors, *Proceedings Eighth International Workshop on Programming Language Approaches to Concurrency- and Communication-cEntric Software, PLACES 2015, London, UK, 18th April 2015*, volume 203 of *EPTCS*, pages 85–98, 2015.