Towards Refinable Choreographies

Ugo de'Liguoro[a], Hernán Melgratti[b], Emilio Tuosto[c]

a - University of Turin, Italy
b - Universidad de Buenos Aires, Argentina
c - Gran Sasso Science Institute, Italy

ICE 2020 - 19th June

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

W3C **service choreography**:

$$\text{Global view} \xrightarrow{\text{project}} \text{Local view} \xleftarrow{\text{comply}} \text{Local systems}$$

A major issue with choreographies is **lack of modularity**

> *"The basic pattern of my approach will be to compose the program in minute steps, deciding each time as little as possible. As the problem analysis proceeds, so does the further refinement of my program"*
>
> E. W. Dijkstra: Notes on Structured Programming

We propose a framework of step-by-step **refinement** of abstract choreographies into concrete ones

Global choreographies

Syntax of **global choreographies** (**g-choreographies** for short), $\mathcal{G}$:

$$\mathsf{G} ::= \mathbf{0} \mid \mathsf{A} \xrightarrow{\mathsf{m}} \mathsf{B} \mid \mathsf{G}; \mathsf{G}' \mid \mathsf{G} \mid \mathsf{G}' \mid \mathsf{G} + \mathsf{G}'$$

Example:

$$\mathsf{C} \xrightarrow{\mathsf{md}} \mathsf{S} + \mathsf{C} \xrightarrow{\mathsf{req}} \mathsf{S}; \mathsf{S} \xrightarrow{\mathsf{done}} \mathsf{C}$$

Adding **refinable** (and multiple) interaction:

$$\mathsf{G} \; ::= \; \cdots \mid \mathsf{A} \overset{\mathsf{m}_1 \ldots \mathsf{m}_n}{\dashrightarrow} \mathsf{B}_1 \ldots \mathsf{B}_n$$

Which are legal refinements of the following?

$$\mathsf{C} \overset{\mathsf{md}}{\dashrightarrow} \mathsf{S} + \mathsf{C} \overset{\mathsf{req}}{\dashrightarrow} \mathsf{S}; \mathsf{S} \overset{\mathsf{done}}{\dashrightarrow} \mathsf{C}$$

Sound and wrong refinements:

$$\mathsf{C} \xrightarrow{\mathsf{md}} \mathsf{S} + \mathsf{C} \xrightarrow{\mathsf{req}} \mathsf{S}; (\mathsf{S} \xrightarrow{\mathsf{stats}} \mathsf{C}; \mathsf{S} \xrightarrow{\mathsf{done}} \mathsf{C}) \qquad \checkmark$$

$$(\mathsf{C} \xrightarrow{\mathsf{md}} \mathsf{B}; \mathsf{B} \xrightarrow{\mathsf{md}} \mathsf{S}) + \mathsf{C} \xrightarrow{\mathsf{req}} \mathsf{S}; (\mathsf{S} \xrightarrow{\mathsf{stats}} \mathsf{C}; \mathsf{S} \xrightarrow{\mathsf{done}} \mathsf{C}) \qquad ✗$$

$$(\mathsf{C} \xrightarrow{\mathsf{md}} \mathsf{B}; \mathsf{B} \xrightarrow{\mathsf{md}} \mathsf{S}) + (\mathsf{C} \xrightarrow{\mathsf{start}} \mathsf{B}; \mathsf{B} \xrightarrow{\mathsf{req}} \mathsf{S}); (\mathsf{S} \xrightarrow{\mathsf{stats}} \mathsf{C}; \mathsf{S} \xrightarrow{\mathsf{done}} \mathsf{C}) \qquad \checkmark$$

Well-formed choreographies

$$[\![\mathsf{G}]\!] = \begin{cases} \mathcal{E} & \text{if certain conditions are satisfied} \\ \bot & \text{otherwise} \end{cases}$$

where $\mathcal{E} = (E, \leq, \#, \lambda)$ is a *labelled (prime) event structure*, namely $(E, \leq)$ is a poset, $\# \subseteq E^2$ s.t. for all $e, e', e'' \in E$:

$$\{e' \in E \mid e' \leq e\} \text{ is finite} \qquad e \# e' \ \& \ e' \leq e'' \implies e \# e''$$

$\lambda : E \to \mathcal{M}$ with

$$\lambda(e) = \mathsf{A\,B!m} \quad \text{“A sends m to B” (whose subject is A)}$$
$$\lambda(e) = \mathsf{A\,B?m} \quad \text{“B receives m from A” (whose subject is B)}$$

We say that G is **well-formed** if $[\![\mathsf{G}]\!] \neq \bot$.

Well-branched choice

A **branch** of $\mathcal{E} = (E, \leq, \#, \lambda)$ is a maximal subest $x \subseteq E$ of conflict free events (also called a *maximal configuration*)

$[\![\mathsf{G}_1]\!] = \mathcal{E}_1$ and $[\![\mathsf{G}_2]\!] = \mathcal{E}_2$ are **well-branched** if

- there is a unique **active** A that locally and unambiguously decides which branch to take in a choice
- all $\mathsf{B} \neq \mathsf{A}$ either behaves the same in all branches, or its behaviour functionally depends on the messages it receives on each branch A opted for: these are **passive**

where the actives and passives are participants of G_1, G_2 (and so subjects of labels of $\mathcal{E}_1$, $\mathcal{E}_2$)

A well-formed choreography

Consider $G = C \xrightarrow{md} S + C \xrightarrow{req} S; S \xrightarrow{stats} C; S \xrightarrow{done} C$

$$[\![C \xrightarrow{md} S]\!] = \begin{tikzcd} C\,S!md \arrow[d] \\ C\,S?md \end{tikzcd} \quad \text{and} \quad [\![C \xrightarrow{req} S; S \xrightarrow{stats} C; S \xrightarrow{done} C]\!] = \begin{tikzcd} C\,S!req \arrow[d] & \\ C\,S?req \arrow[d, red] & \\ S\,C!stats \arrow[r, red] \arrow[d] & S\,C!done \arrow[d] \\ S\,C?stats \arrow[r, red] & S\,C?done \end{tikzcd}$$

The sum operation on event structures introduces conflicts between the events in $[\![C \xrightarrow{md} S]\!]$ and those in $[\![C \xrightarrow{req} S; S \xrightarrow{stats} C; S \xrightarrow{done} C]\!]$, hence:

$$[\![G]\!] = [\![C \xrightarrow{md} S]\!] + [\![C \xrightarrow{req} S; S \xrightarrow{stats} C; S \xrightarrow{done} C]\!] = \begin{tikzcd} C\,S!md \arrow[d] \arrow[r, dash, dashed, "\#"] & C\,S!req \arrow[d] & \\ C\,S?md & C\,S?req \arrow[d] & \\ & S\,C!stats \arrow[r] \arrow[d] & S\,C!done \arrow[d] \\ & S\,C?stats \arrow[r] & S\,C?done \end{tikzcd}$$

Breaking well-branchedness

On the contrary $G' = C \xrightarrow{md} B; B \xrightarrow{md} S + C \xrightarrow{req} S; S \xrightarrow{done} C$

$$
\llbracket C \xrightarrow{md} B; B \xrightarrow{md} S \rrbracket =
\begin{tikzcd}
C\,B!md \arrow[d] \\
C\,B?md \arrow[d] \\
B\,S!md \arrow[d] \\
B\,S?done
\end{tikzcd}
\quad \text{and} \quad
\llbracket C \xrightarrow{req} S; S \xrightarrow{stats} C; S \xrightarrow{done} C \rrbracket =
\begin{tikzcd}
C\,S!req \arrow[d] & \\
C\,S?req \arrow[d] & \\
S\,C!stats \arrow[d] \arrow[r] & S\,C!done \arrow[d] \\
S\,C?stats \arrow[r] & S\,C?done
\end{tikzcd}
$$

but

$$
\begin{tikzcd}
C\,B!md \arrow[d] \arrow[r, dash, dashed, "\#"] & C\,S!req \arrow[d] & \\
C\,B?md \arrow[d] & C\,S?req \arrow[d] & \\
B\,S!md \arrow[d] & S\,C!stats \arrow[d] \arrow[r] & S\,C!done \arrow[d] \\
B\,S?done & S\,C?stats \arrow[r] & S\,C?done
\end{tikzcd}
$$

is not well-branched because of B which is not passive in the right branch

Abstracting properties of well-formed choreographies

To determine when $[\![G_1 \odot G_2]\!] \neq \bot$ it suffices to know:

- the set Π_i of participants of G_i
- the set $\phi_i = \min([\![G_i]\!] \upharpoonright A)$ of the (labels of) the minimal events in the projection of $[\![G_i]\!]$ to A, for all $A \in \Pi_i$
- the set $\Lambda_i = \max([\![G_i]\!] \upharpoonright A)$ of the (labels of) the maximal events in the projection of $[\![G_i]\!]$ to A, for all $A \in \Pi_i$

Idea

We introduce a typing judgement

$$\Pi \vdash G : \langle \phi, \Lambda \rangle$$

meaning that $\Pi = \mathcal{P}(G)$, ϕ and Λ are the minimal and maximal actions of all participants in G respectively, and define typing rules that are sound w.r.t. well-formedness

Type rules for interaction and sequencing

$$\frac{\phi = \Lambda = \{\mathsf{A\,B!m}, \mathsf{A\,B?m}\}}{\{\mathsf{A}, \mathsf{B}\} \vdash \mathsf{A} \xrightarrow{\mathsf{m}} \mathsf{B} : \langle \phi, \Lambda \rangle}\ \text{T-INT}$$

$$\frac{\Pi_1 \vdash \mathsf{G}_1 : \langle \phi_1, \Lambda_1 \rangle \qquad \Pi_2 \vdash \mathsf{G}_2 : \langle \phi_2, \Lambda_2 \rangle}{\Pi_1 \cup \Pi_2 \vdash \mathsf{G}_1 ; \mathsf{G}_2 : \langle \phi_1 \cup (\phi_2 - \Pi_1), \Lambda_2 \cup (\Lambda_1 - \Pi_2) \rangle}\ \text{T-SEQ}$$

where for $L \subseteq \mathcal{L}$ and $\Pi \subseteq \mathcal{P}$ we set $L - \Pi = \{l \in L \mid \mathsf{sbj}\ l \notin \Pi\}$

Example:

$$\dfrac{\dfrac{\phi_1 = \Lambda_1 = \{\mathsf{C\,S!req}, \mathsf{C\,S?req}\}}{\{\mathsf{C}, \mathsf{S}\} \vdash \mathsf{C} \xrightarrow{\mathsf{req}} \mathsf{S} : \langle \phi_1, \Lambda_1 \rangle} \qquad \dfrac{\phi_2 = \Lambda_2 = \{\mathsf{S\,C!done}, \mathsf{S\,C?done}\}}{\{\mathsf{C}, \mathsf{S}\} \vdash \mathsf{S} \xrightarrow{\mathsf{done}} \mathsf{C} : \langle \phi_2, \Lambda_2 \rangle}}{\{\mathsf{C}, \mathsf{S}\} \vdash \mathsf{C} \xrightarrow{\mathsf{req}} \mathsf{S}; \mathsf{S} \xrightarrow{\mathsf{done}} \mathsf{C} : \langle \phi_1, \Lambda_2 \rangle}$$

Type rule for choice

Let $\widehat{L}(\mathsf{A}) = \{l \in L \mid \mathsf{sbj}\ l = \mathsf{A}\}$ for $L \subseteq \mathcal{L}$

$$\frac{\Pi \vdash \mathsf{G}_1 : \langle \phi_1, \Lambda_1 \rangle \qquad \Pi \vdash \mathsf{G}_2 : \langle \phi_2, \Lambda_2 \rangle \qquad \phi_1 \bowtie_\Pi \phi_2}{\Pi \vdash \mathsf{G}_1 + \mathsf{G}_2 : \langle \phi_1 \cup \phi_2, \Lambda_1 \cup \Lambda_2 \rangle}\ \text{T-CH}$$

where the condition $\phi_1 \bowtie_\Pi \phi_2$ is defined by the clauses:

there is a unique $\mathsf{A} \in \Pi$ such that $\widehat{\phi_1}(\mathsf{A})$ and $\widehat{\phi_2}(\mathsf{A})$ are disjoint sets of output actions and both non-empty;

for all $\mathsf{B} \neq \mathsf{A} \in \Pi$, $\widehat{\phi_1}(\mathsf{B})$ and $\widehat{\phi_2}(\mathsf{B})$ are disjoint sets of input actions and $\widehat{\phi_1}(\mathsf{B}) = \emptyset$ if and only if $\widehat{\phi_2}(\mathsf{B}) = \emptyset$

Example:

$$\frac{\dfrac{\phi_1 = \Lambda_1 = \{\mathsf{C\,S!md}, \mathsf{C\,S?md}\}}{\{\mathsf{C}, \mathsf{S}\} \vdash \mathsf{C} \xrightarrow{\mathsf{md}} \mathsf{S} : \langle \phi_1, \Lambda_1 \rangle} \qquad \{\mathsf{C}, \mathsf{S}\} \vdash \mathsf{C} \xrightarrow{\mathsf{req}} \mathsf{S}; \mathsf{S} \xrightarrow{\mathsf{done}} \mathsf{C} : \langle \phi_2, \Lambda_3 \rangle}{\{\mathsf{C}, \mathsf{S}\} \vdash \mathsf{C} \xrightarrow{\mathsf{md}} \mathsf{S} + \mathsf{C} \xrightarrow{\mathsf{req}} \mathsf{S}; \mathsf{S} \xrightarrow{\mathsf{done}} \mathsf{C} : \langle \phi_1 \cup \phi_2, \Lambda_1 \cup \Lambda_3 \rangle}$$

where $\phi_2 = \Lambda_2 = \{\mathsf{C\,S!req}, \mathsf{C\,S?req}\}$ and $\phi_3 = \Lambda_3 = \{\mathsf{C\,B!md}, \mathsf{C\,B?md}\}$

The type system

$$\frac{}{\emptyset \vdash \mathbf{0} : \langle \emptyset, \emptyset \rangle}\ \textsc{T-Emp} \qquad \frac{\phi = \Lambda = \{\mathsf{A\,B!m}, \mathsf{A\,B?m}\}}{\{\mathsf{A}, \mathsf{B}\} \vdash \mathsf{A} \xrightarrow{\mathsf{m}} \mathsf{B} : \langle \phi, \Lambda \rangle}\ \textsc{T-Int}$$

$$\frac{\Pi_1 \vdash \mathsf{G}_1 : \langle \phi_1, \Lambda_1 \rangle \qquad \Pi_2 \vdash \mathsf{G}_2 : \langle \phi_2, \Lambda_2 \rangle}{\Pi_1 \cup \Pi_2 \vdash \mathsf{G}_1 ; \mathsf{G}_2 : \langle \phi_1 \cup (\phi_2 - \Pi_1), \Lambda_2 \cup (\Lambda_1 - \Pi_2) \rangle}\ \textsc{T-Seq}$$

$$\frac{\Pi_1 \vdash \mathsf{G}_1 : \langle \phi_1, \Lambda_1 \rangle \qquad \Pi_2 \vdash \mathsf{G}_2 : \langle \phi_2, \Lambda_2 \rangle \qquad \Pi_1 \cap \Pi_2 = \emptyset}{\Pi_1 \cup \Pi_2 \vdash \mathsf{G}_1 \mid \mathsf{G}_2 : \langle \phi_1 \cup \phi_2, \Lambda_1 \cup \Lambda_2 \rangle}\ \textsc{T-Par}$$

$$\frac{\Pi \vdash \mathsf{G}_1 : \langle \phi_1, \Lambda_1 \rangle \qquad \Pi \vdash \mathsf{G}_2 : \langle \phi_2, \Lambda_2 \rangle \qquad \phi_1 \bowtie_{\Pi} \phi_2}{\Pi \vdash \mathsf{G}_1 + \mathsf{G}_2 : \langle \phi_1 \cup \phi_2, \Lambda_1 \cup \Lambda_2 \rangle}\ \textsc{T-Ch}$$

Theorem (Soundness)

If $\Pi \vdash \mathsf{G} : \langle \phi, \Lambda \rangle$ *is derivable then* $[\![\mathsf{G}]\!] \neq \bot$, $\Pi = \mathcal{P}(\mathsf{G})$, *and*

$$\widehat{\phi}(\mathsf{A}) = \min([\![\mathsf{G}]\!] \upharpoonright \mathsf{A}) \qquad \textit{and} \qquad \widehat{\Lambda}(\mathsf{A}) = \max([\![\mathsf{G}]\!] \upharpoonright \mathsf{A})$$

holds for all $\mathsf{A} \in \Pi$.

Remark: a choreography G has at most one typing $\Pi \vdash \mathsf{G} : \langle \phi, \Lambda \rangle$ and it is computable

The refinement relation

Let $\mathsf{A} \overset{\overline{\mathsf{m}}}{\dashrightarrow} \overline{\mathsf{B}} \equiv \mathsf{A} \overset{\mathsf{m}_1 \ldots \mathsf{m}_n}{\dashrightarrow} \mathsf{B}_1 \ldots \mathsf{B}_n$, then a ground g-choreography G **refines** $\mathsf{A} \overset{\overline{\mathsf{m}}}{\dashrightarrow} \overline{\mathsf{B}}$, written $\mathsf{G}\ \mathit{ref}\ \mathsf{A} \overset{\overline{\mathsf{m}}}{\dashrightarrow} \overline{\mathsf{B}}$, if

$[\![\mathsf{G}]\!] = \mathcal{E} \neq \bot$;

sbj $\min(\mathcal{E}) = \{\mathsf{A}\}$, by which we say that A is the (unique) **initiator** of G;

for all branch x of $\mathcal{E}$ and $1 \leq h \leq n$ there exists $\mathsf{C} \in \mathcal{P}(\mathsf{G})$ such that $\mathsf{C}\,\mathsf{B}_\mathsf{h}?\mathsf{m}_\mathsf{h} \in \max(x \restriction \mathsf{B}_h)$

Axioms for refinable interactions

Lemma

Let $\mathsf{A} \xrightarrow{\overline{\mathsf{m}}} \overline{\mathsf{B}} \equiv \mathsf{A} \xrightarrow{\mathsf{m}_1 \ldots \mathsf{m}_n} \mathsf{B}_1 \ldots \mathsf{B}_n$ *If* $\Pi \subseteq \mathcal{P}$ *and* $\phi, \Lambda \subseteq \mathcal{L}$ *are such that*

- $\mathit{sbj}\ \phi = \mathit{sbj}\ \Lambda = \Pi$,
- $\mathit{sbj}\ (\phi \cap \mathcal{L}^!) = \{\mathsf{A}\}$, *and*
- *for all* $1 \leq h \leq n$ *there exists* C *such that* $\widehat{\Lambda}(\mathsf{B}_h) = \{\mathsf{C}\,\mathsf{B}_h?\mathsf{m}_h\}$

then $\Pi \vdash \mathsf{G} : \langle \phi, \Lambda \rangle$ *implies* $\mathsf{G}\ \mathrm{ref}\ \mathsf{A} \xrightarrow{\overline{\mathsf{m}}} \overline{\mathsf{B}}$.

Axiom schema for refinable interactions:

$$\frac{\mathsf{sbj}\ \phi = \mathsf{sbj}\ \Lambda = \Pi \quad \mathsf{sbj}\ (\phi \cap \mathcal{L}^!) = \{\mathsf{A}\} \quad \forall \mathrm{h}\, \exists \mathsf{C} \in \Pi.\ \widehat{\Lambda}(\mathsf{B}_h) = \{\mathsf{C}\,\mathsf{B}_h?\mathsf{m}_h\}}{\Pi \vdash \mathsf{A} \xrightarrow{\mathsf{m}_1 \ldots \mathsf{m}_n} \mathsf{B}_1 \ldots \mathsf{B}_n : \langle \phi, \Lambda \rangle}\ \textsc{t-ref}$$

Let $\Pi = \{C, S\}$:

$$
\dfrac{\dfrac{\phi_1 = \Lambda_1 = \{C\,S!md, C\,S?md\}}{\Pi \vdash C \overset{md}{\dashrightarrow} S : \langle \phi_1, \Lambda_1 \rangle} \qquad \dfrac{\dfrac{\phi_2 = \Lambda_2 = \{C\,S!req, C\,S?req\}}{\Pi \vdash C \overset{req}{\dashrightarrow} S : \langle \phi_2, \Lambda_2 \rangle} \quad \dfrac{\phi_3 = \Lambda_3 = \{S\,C!done, S\,C?done\}}{\Pi \vdash S \overset{done}{\dashrightarrow} C : \langle \phi_3, \Lambda_3 \rangle}}{\Pi \vdash C \overset{req}{\dashrightarrow} S; S \overset{done}{\dashrightarrow} C : \langle \phi_2, \Lambda_3 \rangle}}{\Pi \vdash C \overset{md}{\dashrightarrow} S + C \overset{req}{\dashrightarrow} S; S \overset{done}{\dashrightarrow} C : \langle \phi_1 \cup \phi_2, \Lambda_1 \cup \Lambda_3 \rangle} \ \text{T-CH}
$$

Consider $G_1 \equiv C \xrightarrow{md} B; B \xrightarrow{md} S$ s.t. $G_1 \text{ ref } C \overset{md}{\dashrightarrow} S$, then compute

$$
\Pi \cup \{B\} \vdash C \xrightarrow{md} B; B \xrightarrow{md} S : \langle \phi_1 \cup \{C\,B?md\}, \Lambda_1 \cup \{B\,S!md\} \rangle
$$

Let $\Pi' = \Pi \cup \{B\}$, $\phi_1' = \phi_1 \cup \{C\,B?md\}$, $\Lambda_1' = \Lambda_1 \cup \{B\,S!md\}$; then

$$
\dfrac{}{\Pi' \vdash C \overset{md}{\dashrightarrow} S : \langle \phi_1', \Lambda_1' \rangle} \ \text{T-REF}
$$

but now rule T-CH doesn't apply since $\Pi' \neq \Pi$

Let $\Pi = \{\mathsf{C}, \mathsf{S}\}$:

$$\dfrac{\dfrac{\phi_1 = \Lambda_1 = \{\mathsf{C}\,\mathsf{S}!\mathsf{md}, \mathsf{C}\,\mathsf{S}?\mathsf{md}\}}{\Pi \vdash \mathsf{C} \overset{\mathsf{md}}{\dashrightarrow} \mathsf{S} : \langle \phi_1, \Lambda_1 \rangle} \qquad \dfrac{\dfrac{\phi_2 = \Lambda_2 = \{\mathsf{C}\,\mathsf{S}!\mathsf{req}, \mathsf{C}\,\mathsf{S}?\mathsf{req}\}}{\Pi \vdash \mathsf{C} \overset{\mathsf{req}}{\dashrightarrow} \mathsf{S} : \langle \phi_2, \Lambda_2 \rangle} \quad \dfrac{\phi_3 = \Lambda_3 = \{\mathsf{S}\,\mathsf{C}!\mathsf{done}, \mathsf{S}\,\mathsf{C}?\mathsf{done}\}}{\Pi \vdash \mathsf{S} \overset{\mathsf{done}}{\dashrightarrow} \mathsf{C} : \langle \phi_3, \Lambda_3 \rangle}}{\Pi \vdash \mathsf{C} \overset{\mathsf{req}}{\dashrightarrow} \mathsf{S}; \mathsf{S} \overset{\mathsf{done}}{\dashrightarrow} \mathsf{C} : \langle \phi_2, \Lambda_3 \rangle}}{\Pi \vdash \mathsf{C} \overset{\mathsf{md}}{\dashrightarrow} \mathsf{S} + \mathsf{C} \overset{\mathsf{req}}{\dashrightarrow} \mathsf{S}; \mathsf{S} \overset{\mathsf{done}}{\dashrightarrow} \mathsf{C} : \langle \phi_1 \cup \phi_2, \Lambda_1 \cup \Lambda_3 \rangle} \ \text{T-CH}$$

Consider $\mathsf{G}_1 \equiv \mathsf{C} \xrightarrow{\mathsf{md}} \mathsf{B}; \mathsf{B} \xrightarrow{\mathsf{md}} \mathsf{S}$ which is s.t. $\mathsf{G}_1 \text{ ref } \mathsf{C} \overset{\mathsf{md}}{\dashrightarrow} \mathsf{S}$, then compute

$$\Pi \cup \{\mathsf{B}\} \vdash \mathsf{C} \xrightarrow{\mathsf{md}} \mathsf{B}; \mathsf{B} \xrightarrow{\mathsf{md}} \mathsf{S} : \langle \phi_1 \cup \{\mathsf{C}\,\mathsf{B}?\mathsf{md}\}, \Lambda_1 \cup \{\mathsf{B}\,\mathsf{S}!\mathsf{md}\} \rangle$$

Consider $\mathsf{G}_2 \equiv \mathsf{C} \xrightarrow{\mathsf{x}} \mathsf{B}; \mathsf{B} \xrightarrow{\mathsf{req}} \mathsf{S}$ which is s.t. $\mathsf{G}_2 \text{ ref } \mathsf{C} \overset{\mathsf{req}}{\dashrightarrow} \mathsf{S}$, and compute

$$\Pi' \vdash \mathsf{C} \xrightarrow{\mathsf{x}} \mathsf{B}; \mathsf{B} \xrightarrow{\mathsf{req}} \mathsf{S} : \langle \phi_2', \Lambda_2' \rangle$$

where $\Pi' = \Pi \cup \{\mathsf{B}\}$, $\phi_2' = \{\mathsf{C}\,\mathsf{B}!\mathsf{x}, \mathsf{C}\,\mathsf{B}?\mathsf{x}, \mathsf{B}\,\mathsf{S}!\mathsf{rep}\}$ and $\Lambda_2' = \{\mathsf{C}\,\mathsf{B}?\mathsf{x}, \mathsf{B}\,\mathsf{S}!\mathsf{rep}, \mathsf{B}\,\mathsf{S}?\mathsf{rep}\}$, and take $\mathsf{G}_3 \equiv \mathsf{S} \xrightarrow{\mathsf{done}} \mathsf{C} \text{ ref } \mathsf{S} \overset{\mathsf{done}}{\dashrightarrow} \mathsf{C}$ s.t.

$$\Pi \vdash \mathsf{S} \xrightarrow{\mathsf{done}} \mathsf{C} \langle \phi_3, \Lambda_3 \rangle$$

We end up with:

$$\dfrac{\dfrac{}{\Pi' \vdash \mathsf{G}_1 : \langle \phi'_1, \Lambda'_1 \rangle} \qquad \dfrac{\dfrac{}{\Pi' \vdash \mathsf{G}_2 : \langle \phi'_2, \Lambda'_2 \rangle} \qquad \dfrac{}{\Pi \vdash \mathsf{G}_3 : \langle \phi_3, \Lambda_3 \rangle}}{\Pi' \vdash \mathsf{G}_2 ; \mathsf{G}_3 : \langle \phi'_2, \Lambda'_3 \rangle}}{\Pi' \vdash \mathsf{G}_1 + \mathsf{G}_2 ; \mathsf{G}_3 : \langle \phi'_1 \cup \phi'_2, \Lambda'_1 \cup \Lambda'_3 \rangle} \text{ T-CH}$$

where

$$\mathsf{G}_1 \equiv \mathsf{C} \xrightarrow{\mathsf{md}} \mathsf{B} ; \mathsf{B} \xrightarrow{\mathsf{md}} \mathsf{S} \qquad \mathsf{G}_2 \equiv \mathsf{C} \xrightarrow{\mathsf{x}} \mathsf{B} ; \mathsf{B} \xrightarrow{\mathsf{req}} \mathsf{S} \qquad \mathsf{G}_3 \equiv \mathsf{S} \xrightarrow{\mathsf{done}} \mathsf{C}$$

and

$$\begin{array}{rcl} \Pi & = & \{\mathsf{C}, \mathsf{S}\} \\ \phi'_1 & = & \{\mathsf{C\,S!md}, \mathsf{C\,S?md}, \mathsf{C\,B?md}\} \\ \phi'_2 & = & \{\mathsf{C\,B!x}, \mathsf{C\,B?x}, \mathsf{B\,S!rep}\} \\ \phi_3 = \Lambda_3 & = & \{\mathsf{S\,C!done}, \mathsf{S\,C?done}\} \end{array} \qquad \begin{array}{rcl} \Pi' & = & \Pi \cup \{\mathsf{B}\} \\ \Lambda'_1 & = & \{\mathsf{C\,S!md}, \mathsf{C\,S?md}, \mathsf{B\,S!md}\} \\ \Lambda'_2 & = & \{\mathsf{C\,B?x}, \mathsf{B\,S!rep}, \mathsf{B\,S?rep}\} \\ \Lambda'_3 & = & \Lambda_3 \cup \{\mathsf{B\,S!rep}\} \end{array}$$

Achievements and future work

the type system provides a means to establish which concrete choreographies refine which abstract ones

the mechanism for choosing how to type refinable interactions needs more investigation

we will addres the study of properties of abstract protocols that curry over to concrete ones

Thank You