
Hasso Plattner Institut

IT Systems Engineering | Universität Potsdam

# Service selection by choreography-driven matching

Emerging Web Service Technology

*Christoph Thiele*

*10.12.2009*

# Agenda



- From reuse & selection
- Aspirin
- Math preparation
- Goal-preserving match
  - What doesn‘t work
  - What works
- Conclusion

# From reuse & selection



- Service reuse
  - Retrieve service according to needs
  - No exact match and flexibility (relaxed match)
  - Reuse outside original context
- Semantic annotation & IOPE
- Hierarchy
  - Single operations
  - Sequence → global point of view
  - Choreography
- Web service selection
  - Conformance to a specification
  - Use of service allows achievement of a goal

# Aspirin




painkiller
blood-thinner
ASPIRIN®
conformance
Tablette mit 500 mg Acetylsalicylsäure
painkiller
global goal
20 Tabletten
patient in pain
patient in pain
with a clotting problem

# Math prep.

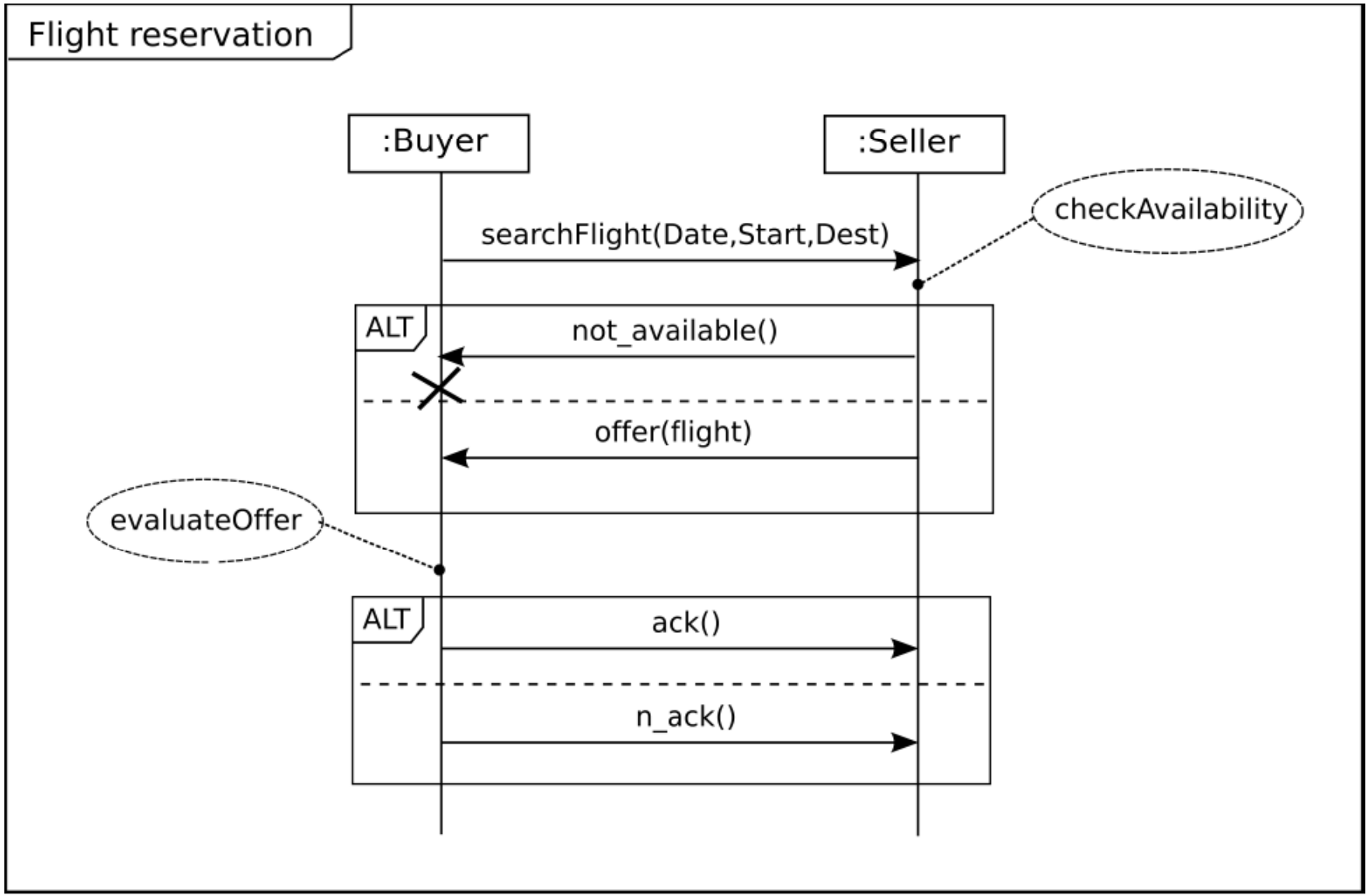


- Fluent: properties whose truth value can change over time due to the application of actions
- State: set of fluents

- One could not assume that the value of a fluent is known.
- $\mathcal{B}$: Beliefs of an entity about the world
  - $\mathcal{B}f$ – f is known to be true
  - $\mathcal{B}\neg f$ – f is known to be false
  - $\neg\mathcal{B}f \wedge \neg\mathcal{B}\neg f$ – f is undefined
  - A fluent could be: true, false or unknown

# Math prep. – service representation



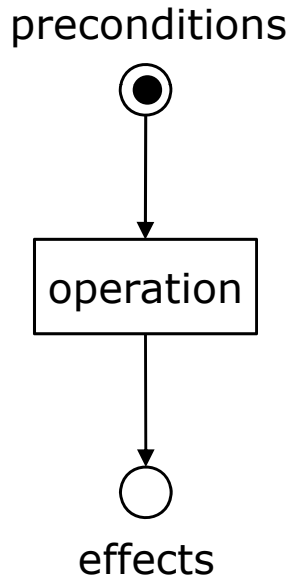


# Math prep. – service representation



- Service description: $\langle \mathcal{O}, \mathcal{G}, \mathcal{P} \rangle$
  - $\mathcal{O}$ – set of operations
  - $\mathcal{G}$ – set of actions that allow to receive messages
  - $\mathcal{P}$ – description of interactive behavior

- $\mathcal{O}$ – set of operations (atomic action)
  - Description in terms of:
    - Preconditions
    - Effects
    - Both sets of fluents
    - Trigger revision process on beliefs



# Math prep. – service representation

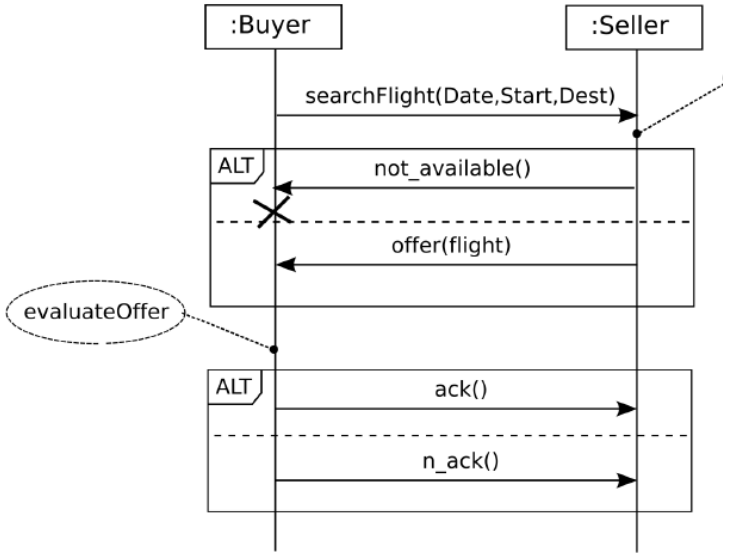


- Service
  - Initiator – $operation^{\gg}$
  - Servant – $operation^{\ll}$

- $operation^{d}(interlocutor, content)$ **possible if** $\{P_1 \dots P_t\}$
- $operation^{d}(interlocutor, content)$ **causes** $\{E_1 \dots E_n\}$

- Example:

$$searchflight^{\gg}(seller, Date, Start, Dest)$$
**possible if** $\{\mathcal{B}Date, \mathcal{B}Start, \mathcal{B}Dest\}$
$$searchflight^{\gg}(seller, Date, Start, Dest)$$
**causes** $\{\mathcal{B}will\_get\_offer\}$

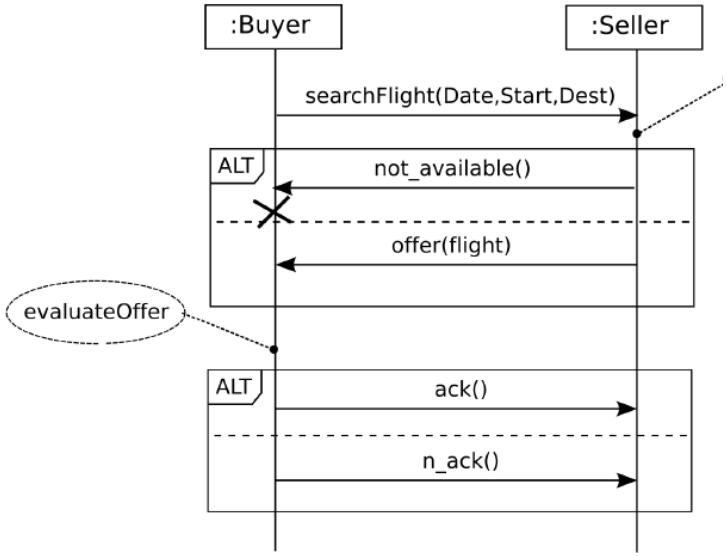


# Math prep. – service representation



- $\mathcal{G}$ – get_answer actions
  - $receive\_act(interlocutor, content)$ **receives** $\mathcal{I}$
  - Finite set of possibilities
  - Example:

$get\_answer(Seller)$ **receives** $[not\_available^{\ll}(Seller)$ **or** $offer^{\ll}(Seller)]$



# Math prep. – service representation



- $\mathcal{P}$ – encodes the behavior of a service
  - Collection of kind:

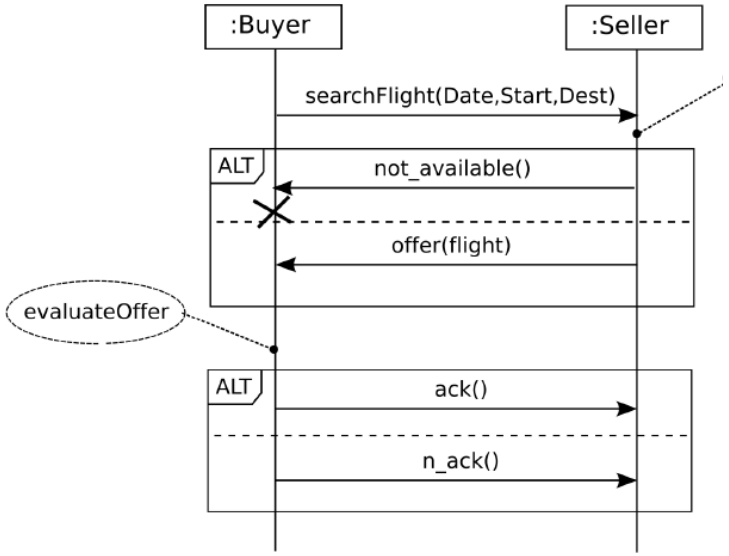
    $p_0$ **is** $p_1 \ldots p_n$

    $p_0$ – procedure

    $p_1$ – atomic operation, get_answer action, testing action, procedure call
  - Example:

    $booking(Seller, Date, Start, Dest)$ **is**
    $search\_flight^{\gg}(Seller, Date, Start, Dest)$,
    $get\_answer(Seller), \mathcal{B}offer(not\_avail)?$

    $booking(Seller, Date, Start, Dest)$ **is**
    $search\_flight^{\gg}(Seller, Date, Start, Dest)$,
    $get\_answer(Seller), \mathcal{B}offer(flight)?$,
    $eval\_offer, finalize(Seller)$



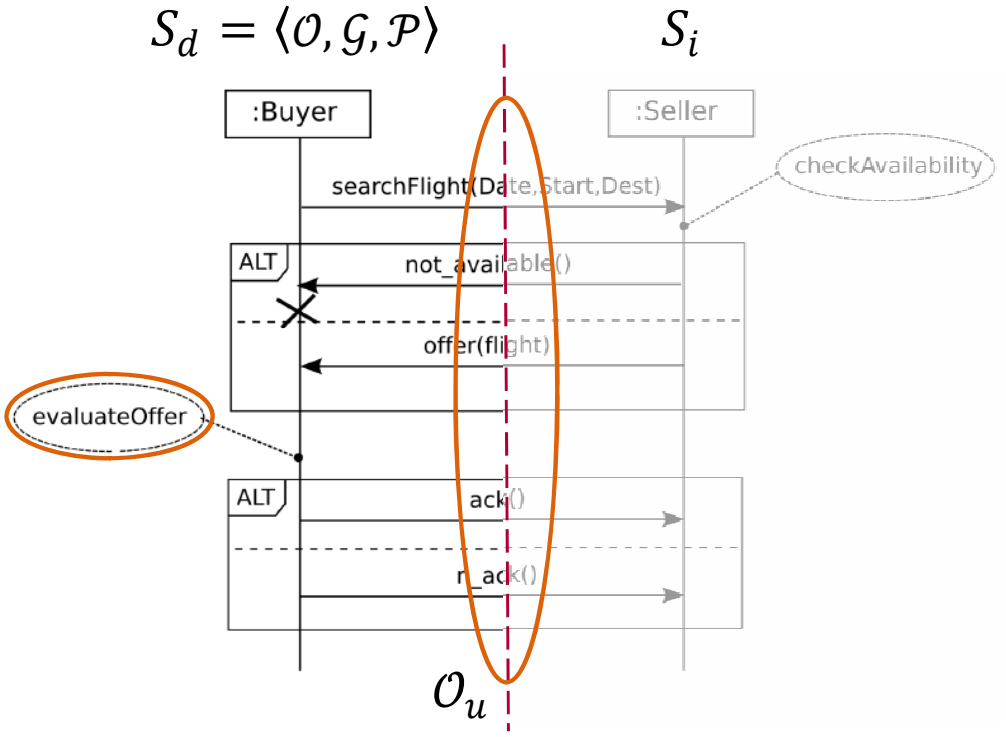
# Math prep. – substitution



- Choreography
  - = set of interacting roles
  - $\mathcal{O}$ devided in
    - Bound operations
    - Unbound operations
- Binding by substitution $\theta$
-

$$\theta = [\mathcal{O}_{S_i}/\mathcal{O}_u]$$

$$S_d\theta = \langle \mathcal{O}\theta, \mathcal{G}\theta, \mathcal{P}\theta \rangle$$



# Math prep. – substitution



$$\theta = \left[\mathcal{O}_{S_{Aspirin}} \ / \mathcal{O}_u\right]$$
$$S_d\theta = \langle \mathcal{O}\theta, \mathcal{G}\theta, \mathcal{P}\theta \rangle$$

$$release\_pain^{\gg}(Pharmaceutical) \textbf{ possible if } \{\mathcal{B}in\_pain\}$$
$$release\_pain^{\gg}(Pharmaceutical) \textbf{ causes } \{\mathcal{B}\neg in\_pain\}$$
$$\in \mathcal{O}, \mathcal{O}_u$$

$$release\_pain^{\gg}(Pharmaceutical) \textbf{ possible if } \{\mathcal{B}in\_pain\}$$
$$release\_pain^{\gg}(Pharmaceutical) \textbf{ causes } \{\mathcal{B}\neg in\_pain, \mathcal{B}thin\_blood\}$$
$$\in \mathcal{O}_{S_{Aspirin}}, \mathcal{O}\theta$$

# Math prep. – reasoning

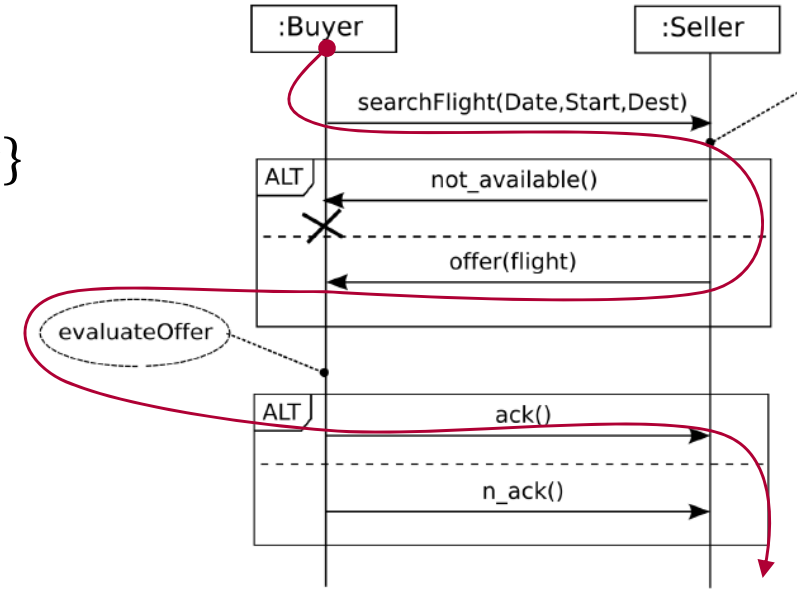


- „Is it possible to execute p in such a way, that the condition Fs is true in the final state?“
- $Fs$ **after** $p$
- If true → sequence of actions $\sigma$
- $(\langle \mathcal{O}, \mathcal{G}, \mathcal{P} \rangle, S_0) \vdash G \;\; w.a. \;\; \sigma$

- Example:

$$S_0 = \{\mathcal{B}date, \mathcal{B}start, \mathcal{B}dest, \mathcal{B}smoking\_flight\}$$

$$G = \{\mathcal{B}booked(flight), \mathcal{B}smoking\_flight\} \textbf{ after } booking(\ldots)$$



# Agenda



- From reuse & selection
- Aspirin
- Math preparation
- Goal-preserving match
  - What doesn‘t work
  - What works
- Conclusion

# Goal-preserving match



**Definition 1 (Conservative substitution).** *Let us consider a service* $S_i = \langle \mathcal{O}, \mathcal{G}, \mathcal{P} \rangle$ *which plays a role* $R_i$ *in a given choreography, and a query* $G$ *such that, given an initial state* $S_0$*,*

$$(\langle \mathcal{O}, \mathcal{G}, \mathcal{P} \rangle, S_0) \vdash G \ w.a. \ \sigma$$

*Consider a substitution* $\theta = [\mathcal{O}_{S_j} / \mathcal{O}^{\sigma}_{u(R_j)}]$*, where* $\mathcal{O}^{\sigma}_{u(R_j)} = \{o_u \in \mathcal{O} \mid o \text{ occurs in } \sigma\}$ *is the set of all unbound operations that refer to another role* $R_j$*,* $j \neq i$*, of the same choreography, that are used in the execution trace* $\sigma$*.* $\theta$ *is conservative when the following holds:*

$$(\langle \mathcal{O}\theta, \mathcal{G}\theta, \mathcal{P}\theta \rangle, S_0) \vdash G \ w.a. \ \sigma\theta$$

# Matching – what doesn‘t work



- EM – Exact Pre/Post Match
  - $Precs(r) = Precs(s) \wedge Effs(r) = Effs(s)$
- PIM – Plugin Match
  - Strongest of the flexible
  - $Precs(r) \supseteq Precs(s) \wedge Effs(s) \supseteq Effs(r)$
  - Allow at least all old conditions
  - Provide a guarantee at least as strong

# What doesn't work



Example using PIM:

$$S_0 = \{\mathcal{B}bp, \mathcal{B}in\_pain, \mathcal{B}\neg thin\_blood\}$$

$$G = \{\mathcal{B}\neg in\_pain, \mathcal{B}\neg thin\_blood\} \ \textbf{after}\ medication(\dots)$$

$$(\langle \mathcal{O}, \mathcal{G}, \mathcal{P}\rangle, S_0) \vdash G \ \ w.a.\ \sigma$$

$$release\_pain^{\gg}(Pharm)\ possible\ if\ \{\mathcal{B}in\_pain\}$$

$$release\_pain^{\gg}(Pharm)\ causes\ \{\mathcal{B}\neg in\_pain\}$$

$Precs(r) \supseteq Precs(s)$

$Effs(s) \supseteq Effs(r)$

$$release\_pain^{\gg}(Pharm)\ possible\ if\ \{\mathcal{B}in\_pain\}$$

$$release\_pain^{\gg}(Pharm)\ causes\ \{\mathcal{B}\neg in\_pain, \mathcal{B}thin\_blood\}$$

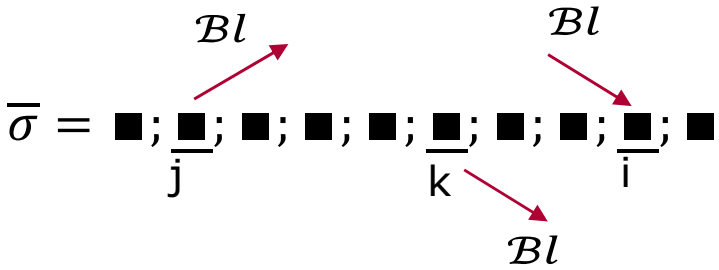
# Goal-preserving match – what works



- Dependency
  - $\sigma = a_1; a_2; \ldots; a_n$
  - 2 fictitious actions
    - $a_0 \rightarrow Effs(a_0) = S_0$
    - $a_{n+1} \rightarrow Precs(a_{n+1}) = Fs$
  - $\overline{\sigma} = a_0; a_1; a_2; \ldots; a_n; a_{n+1}$
  - Indexes $i, j = 0, \ldots n+1$ with $j < i$

    $a_i$ *depends on* $a_j$ *for the fluent* $\mathcal{B}l$ *in* $\sigma$ : $a_j \rightsquigarrow_{(\mathcal{B}l, \overline{\sigma})} a_i$

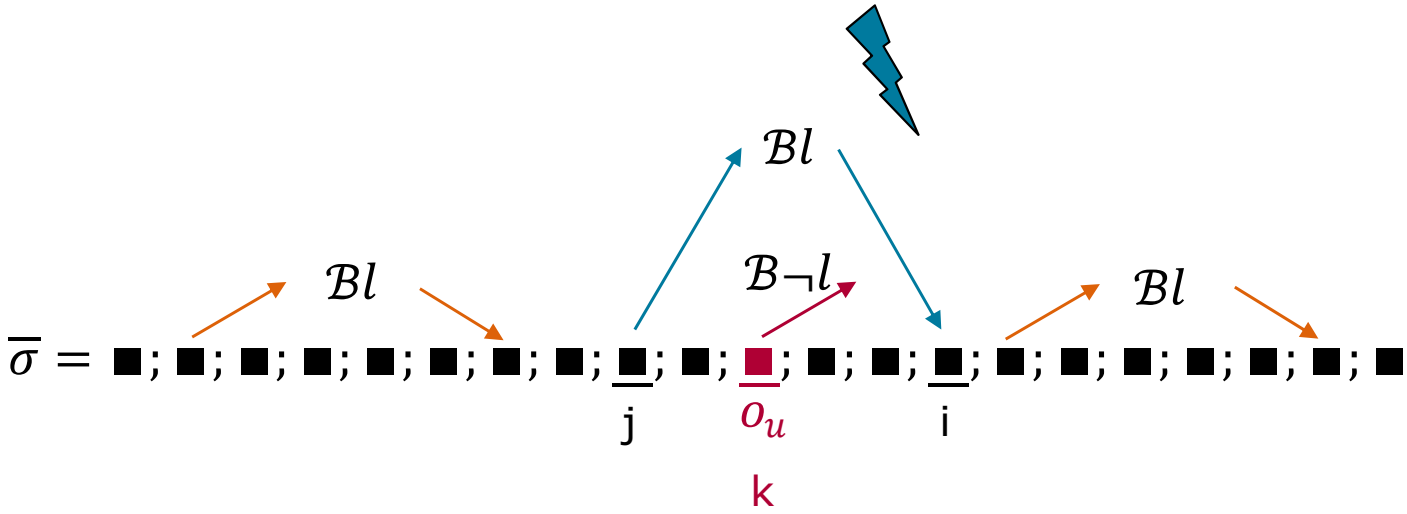
    if $\mathcal{B}l \in Effs(a_j), \mathcal{B}l \in Precs(a_i), \neg\exists k(j < k < i, \mathcal{B}l \in Effs(a_k))$




- Dependency set: $Deps(\mathcal{B}l, \sigma) = \{(j, i) | a_j \rightsquigarrow_{(\mathcal{B}l, \overline{\sigma})} a_i\}$

# Goal-preserving match – what works






- Uninfluential fluent
  - $[s/o_u] \in \theta_{PIM}$
  - $\mathcal{B}\neg l \in Effs(s) - Effs(o_u)$ is uninfluential fluent iff all pairs $(j, i) \in Deps(\mathcal{B}l, \sigma)$ with $k$ identifying the position of $o_u$ in $\sigma$ and $k < j$ or $i \leq k$

# Goal-preserving match – what works



- A substitution $\theta_{PIM}$ is called uninfluential iff for any substitution $[s/o_u]$ in $\theta_{PIM}$, all beliefs in $Effs(s) - Effs(o_u)$ are uninfluential fluents w.r.t. $\sigma$

# Goal-preserving match – what works



**Theorem 2.** *Let us consider a service* $S_i = \langle \mathcal{O}, \mathcal{G}, \mathcal{P} \rangle$ *which plays a role* $R_i$ *in a given choreography, and a query* $G$ *such that, given an initial state* $S_0$,

$$(\langle \mathcal{O}, \mathcal{G}, \mathcal{P} \rangle, S_0) \vdash G \ w.a. \ \sigma$$

*Consider an uninfluential substitution* $\theta_{PIM} = [\mathcal{O}_{S_j} / \mathcal{O}^{\sigma}_{u(R_j)}]$, *where* $\mathcal{O}^{\sigma}_{u(R_j)} = \{o_u \in \mathcal{O} \mid o \text{ occurs in } \sigma\}$ *is the set of all unbound operations that refer to another role* $R_j$, $j \neq i$, *of the same choreography, that are used in the execution trace* $\sigma$. *Then, the following holds:*

$$(\langle \mathcal{O}\theta_{PIM}, \mathcal{G}\theta_{PIM}, \mathcal{P}\theta_{PIM} \rangle, S_0) \vdash G \ w.a. \ \sigma\theta_{PIM}$$

# Conclusion



- Achived
  - Formal representation of a service
  - Uninfluential Plugin Match
  - Definition of a goal-preserving match
- Semantical annotation
- Definition of unbound operations
- Feasibility