SAP RESEARCH

SYSTEMATIC THOUGHT LEADERSHIP FOR INNOVATIVE BUSINESS

Web Service Behavior Matchmaking Using DL

Jens Lemcke
Research Associate, SAP AG

Co-Authors: Markus Fronk

THE BEST-RUN BUSINESSES RUN SAP

Background

Objectives

Modeling

Conclusion

SAP RESEARCH

Background

Objectives

Modeling

Conclusion

THE BEST-RUN BUSINESSES RUN SAP

Background – Context & Shortcomings & Contribution

Industry Needs

- Dynamic markets → Demand business **flexibility** & **interoperability**

Solution: Web Services

- Define software components using **standardized** interface language
- WSDL: **Inputs** & **outputs** – Business processes: Demand **behavior**

Semantic Web Service Approaches

- Integrate many aspects (eg. inputs, outputs, behavior)
- Rich constructs (eg. subclasses, subproperties, quantifiers, restrictions)
- Powerful reasoner (consistency, subsumption, instance classification)
- Do not leverage standard reasoning for main tasks (eg. matchmaking)

Contribution

- Describe Web services behavior by *description logic* (DL) expressions
- Formulate DL expressions in a way that reasoner can do matchmaking

SAP RESEARCH

Background

Objectives

Modeling

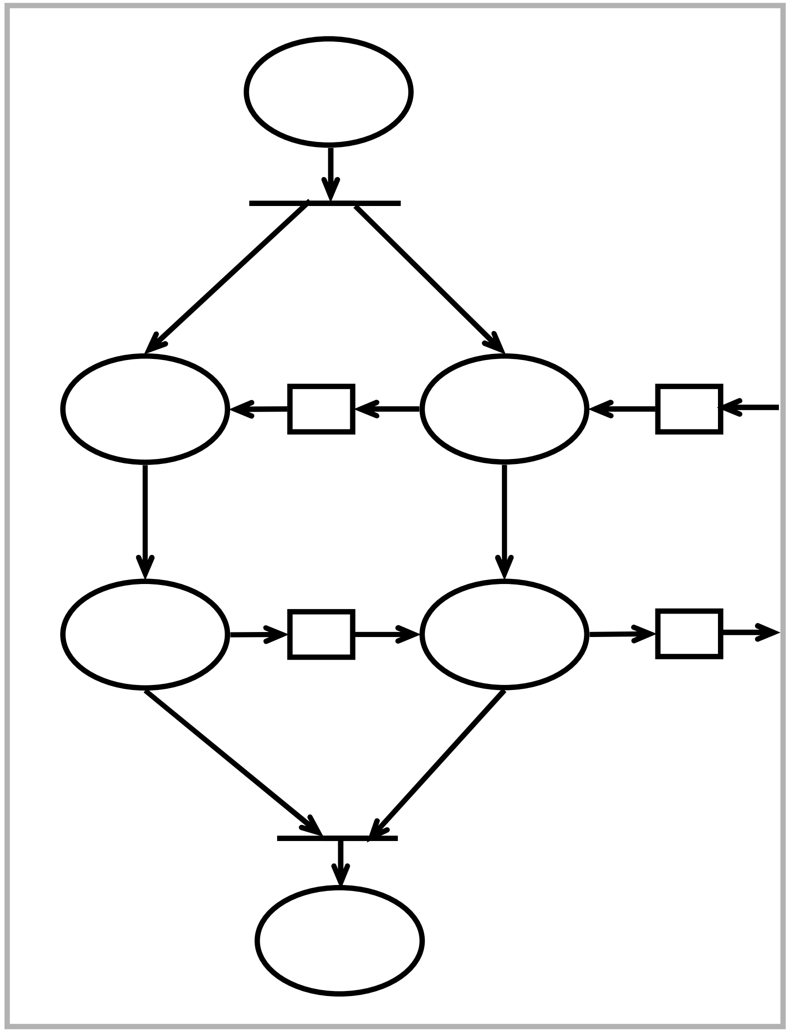
Conclusion

THE BEST-RUN BUSINESSES RUN SAP

Objectives – Aspects of Web Service Behavior

Web Service Behavior

- **Explanation**
  - Causal constraints between operations
  - Required for successful invocation
- **Constituents**
  - Inputs
  - Outputs
  - Sequence
  - Fork & Join
  - Decision & Merge
  - Other complex patterns



➔ **Web service behavior as workflow**
  - Cp. YAWL
  - Cp. UML v2.0 activity diagrams

Objectives – Requests vs. Services

Use Case

- Existing business process
- Outsourcing of one step
- Find a new service fitting the hole the removed service left behind

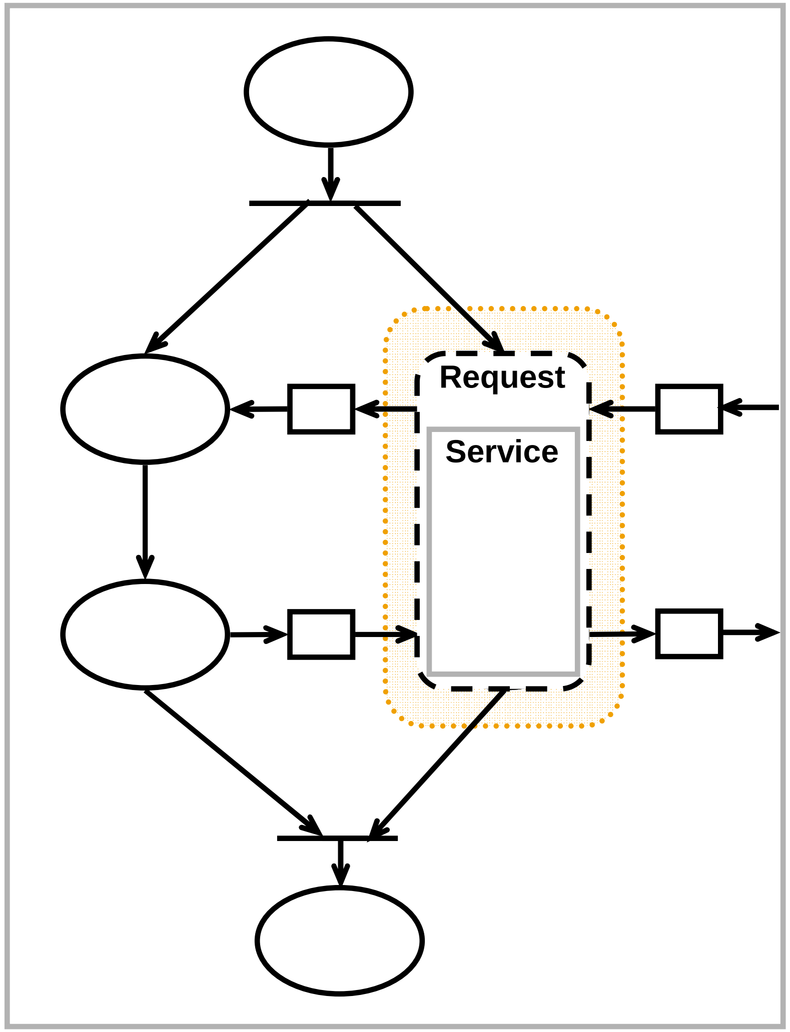
Request

- Defines indispensable constraints
- All possibilities must be followed

Service

- Defines all possibile interactions
- Must be able to follow all possible interactions offered

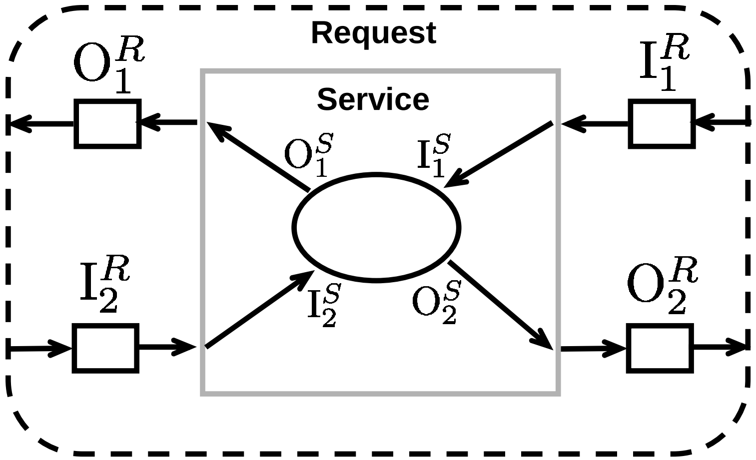
➔ Examples to follow…



Objectives – Inputs vs. Outputs

Use Case

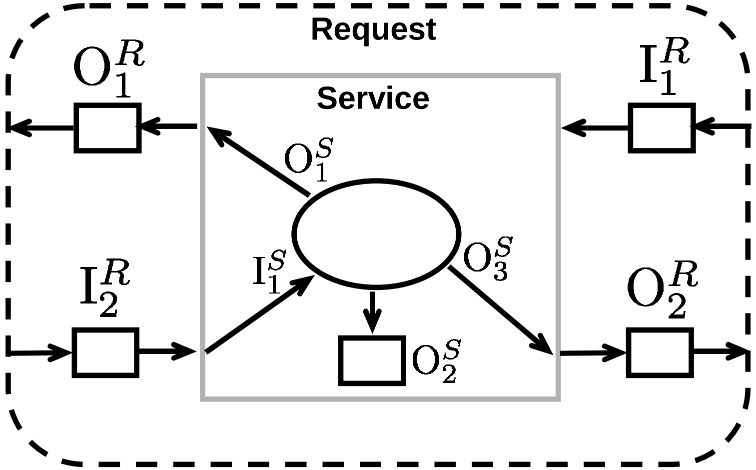
- Outsourcing of one process step
- Find services that
  only take provided inputs while
  serving **all** required outputs



Inputs

$$|I_R| \geq |I_S|$$

- Request I_R
  - Inputs that can be used
- Service I_S
  - Inputs that are required



Outputs

$$|O_R| \leq |O_S|$$

- Request O_R
  - Outputs that are required
- Service O_S
  - Outputs that can be used

SAP RESEARCH

Background

Objectives

Modeling

Conclusion

Modeling – Inputs

SAP RESEARCH

Inputs (repeated)

$$|\mathrm{I}_R| \geq |\mathrm{I}_S|$$

- **Request** I_R
  - **Inputs that can be used**
- **Service** I_S
  - **Inputs that are required**

Implementation

- $\mathrm{I}_R = \{\, \mathrm{I}_1^R, \mathrm{I}_2^R, \ldots, \mathrm{I}_m^R \,\}$

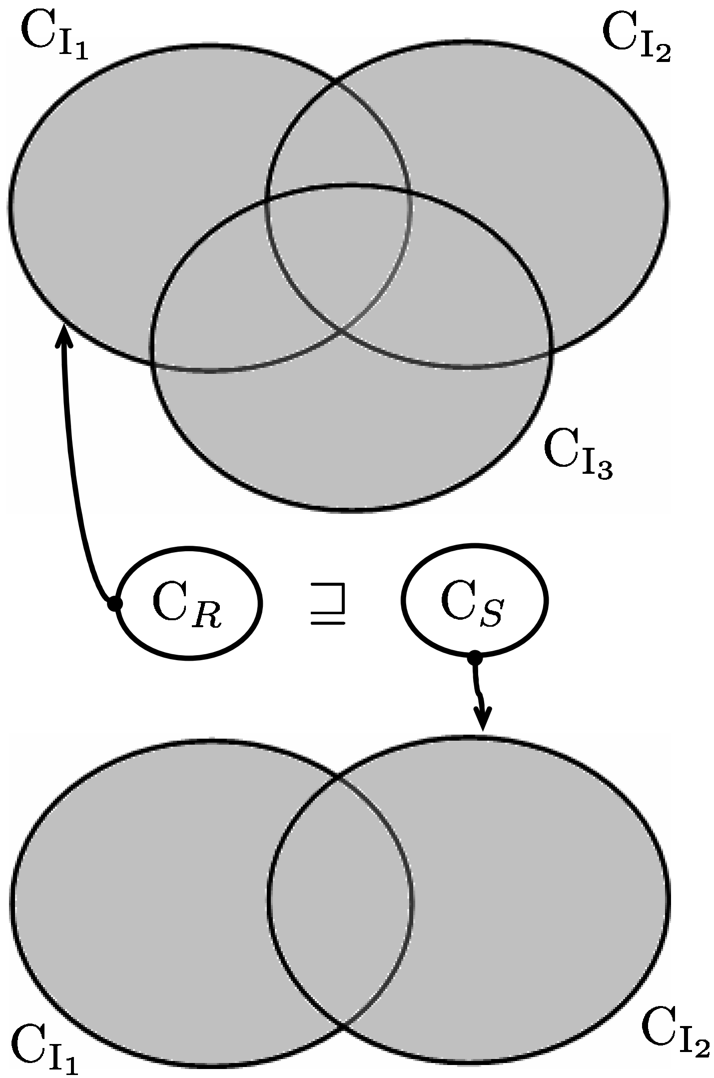
➔ $\mathrm{C}_R \equiv \exists\, \mathrm{has}\,.\left(\mathrm{C}_{\mathrm{I}_1^R} \sqcup \mathrm{C}_{\mathrm{I}_2^R} \sqcup \ldots \sqcup \mathrm{C}_{\mathrm{I}_k^R}\right)$

- $\mathrm{I}_S = \{\, \mathrm{I}_1^S, \mathrm{I}_2^S, \ldots, \mathrm{I}_n^S \,\}$

➔ $\mathrm{C}_S \equiv \exists\, \mathrm{has}\,.\left(\mathrm{C}_{\mathrm{I}_1^S} \sqcup \mathrm{C}_{\mathrm{I}_2^S} \sqcup \ldots \sqcup \mathrm{C}_{\mathrm{I}_l^S}\right)$

Analysis

- **Check for** $\mathrm{C}_R \sqsupseteq \mathrm{C}_S$

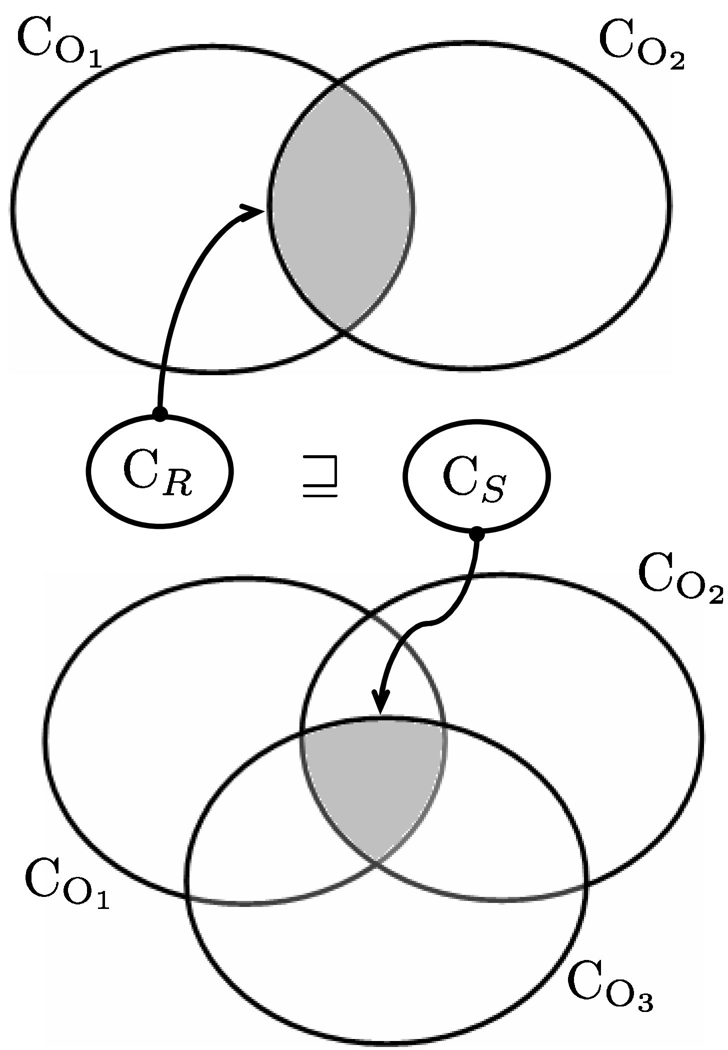


Modeling – Outputs

Outputs (repeated)

$$\boxed{|\mathrm{O}_R| \leq |\mathrm{O}_S|}$$

- Request O_R
  - Outputs that are required
- Service O_S
  - Outputs that can be used



Implementation

- $\mathrm{O}_R = \{\, \mathrm{O}_1^R, \mathrm{O}_2^R, \ldots, \mathrm{O}_m^R \,\}$

➔ $\mathrm{C}_R \equiv \exists\, \mathrm{has}\,.\, \left(\mathrm{C}_{\mathrm{O}_1^R} \sqcap \mathrm{C}_{\mathrm{O}_2^R} \sqcap \ldots \sqcap \mathrm{C}_{\mathrm{O}_m^R}\right)$

- $\mathrm{O}_S = \{\, \mathrm{O}_1^S, \mathrm{O}_2^S, \ldots, \mathrm{O}_n^S \,\}$

➔ $\mathrm{C}_S \equiv \exists\, \mathrm{has}\,.\, \left(\mathrm{C}_{\mathrm{O}_1^S} \sqcap \mathrm{C}_{\mathrm{O}_2^S} \sqcap \ldots \sqcap \mathrm{C}_{\mathrm{O}_n^S}\right)$

Analysis

- Check for $\mathrm{C}_R \sqsupseteq \mathrm{C}_S$

Modeling – Inputs & Outputs

Implementation

$$|\mathrm{I}_R| \geq |\mathrm{I}_S|$$

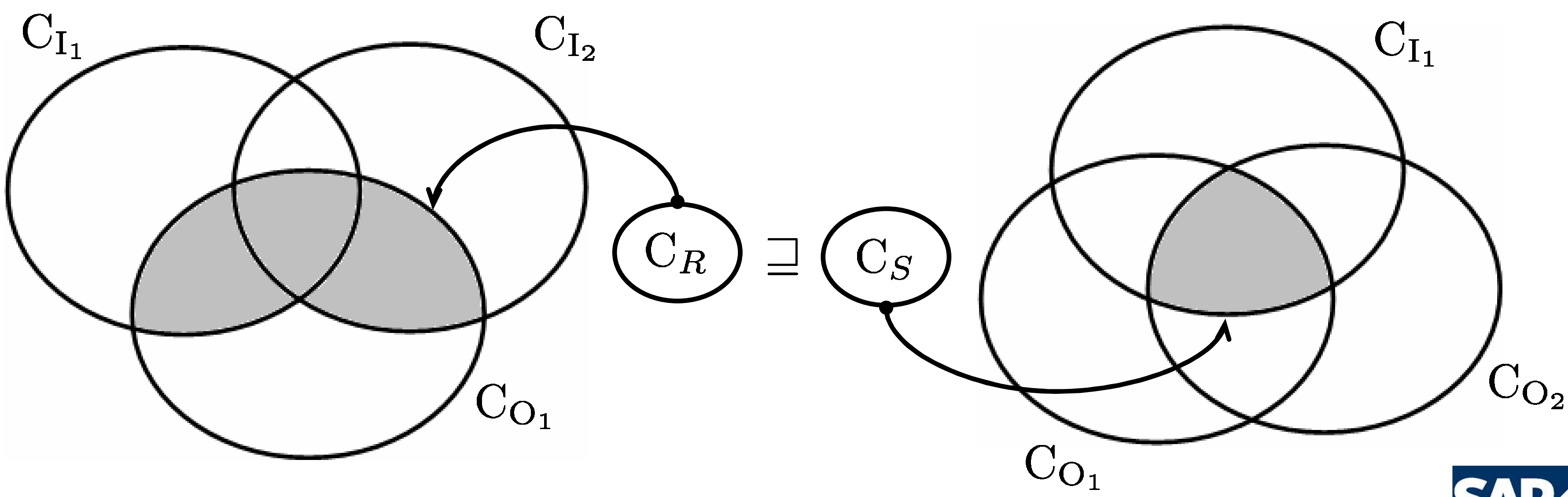
$$|\mathrm{O}_R| \leq |\mathrm{O}_S|$$

- $\mathrm{IO}_R = \mathrm{I}_R \cup \mathrm{O}_R$

➔ $\mathrm{C}_R \equiv \exists\,\mathrm{has}\,.\left[\left(\mathrm{C}_{\mathrm{I}_1^R} \sqcup \mathrm{C}_{\mathrm{I}_2^R} \sqcup \ldots \sqcup \mathrm{C}_{\mathrm{I}_k^R}\right) \sqcap \left(\mathrm{C}_{\mathrm{O}_1^R} \sqcap \mathrm{C}_{\mathrm{O}_2^R} \sqcap \ldots \sqcap \mathrm{C}_{\mathrm{O}_m^R}\right)\right]$

- $\mathrm{IO}_S = \mathrm{I}_S \cup \mathrm{O}_S$

➔ $\mathrm{C}_S \equiv \exists\,\mathrm{has}\,.\left[\left(\mathrm{C}_{\mathrm{I}_1^S} \sqcup \mathrm{C}_{\mathrm{I}_2^S} \sqcup \ldots \sqcup \mathrm{C}_{\mathrm{I}_l^S}\right) \sqcap \left(\mathrm{C}_{\mathrm{O}_1^S} \sqcap \mathrm{C}_{\mathrm{O}_2^S} \sqcap \ldots \sqcap \mathrm{C}_{\mathrm{O}_n^S}\right)\right]$

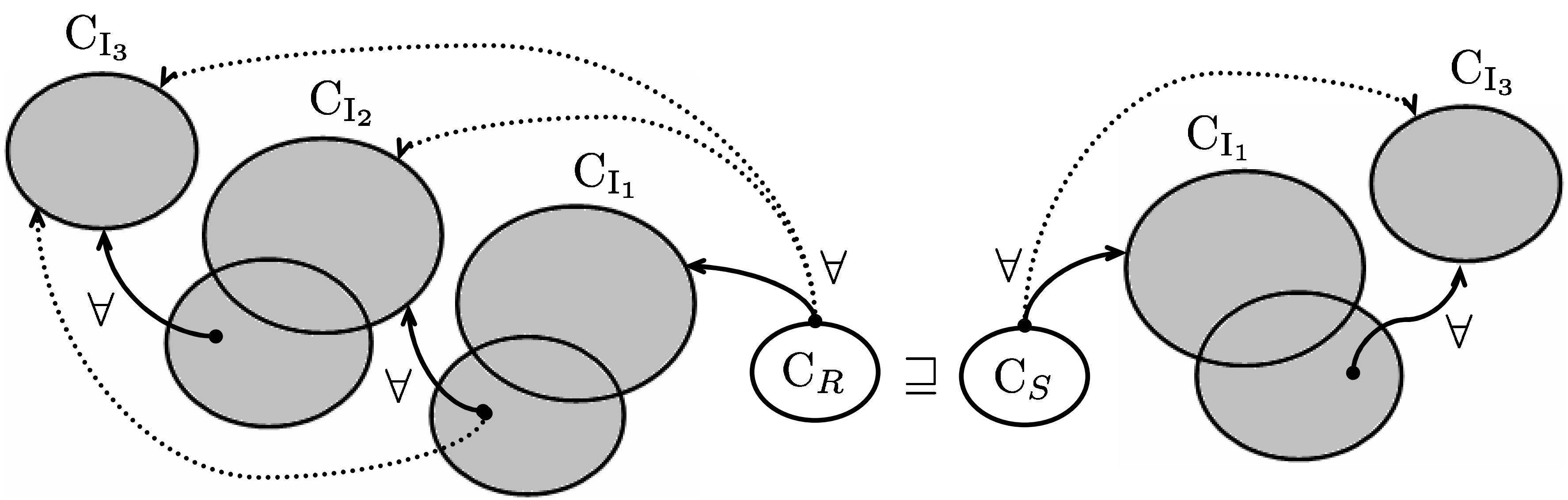


Implementation

$$|\mathrm{I}_R| \geq |\mathrm{I}_S|$$

- $\mathrm{transitive\,(then)}$
- $\mathrm{Seq}^{\mathrm{I}} = \mathrm{I}_1 \mapsto \mathrm{I}_2 \mapsto \ldots \mapsto \mathrm{I}_k$

➔ $\mathrm{C} \equiv \exists\,\mathrm{has}\,.\,[(\mathrm{C}_{\mathrm{I}_1} \sqcup \mathrm{C}_{\mathrm{I}_2} \sqcup \ldots \sqcup \mathrm{C}_{\mathrm{I}_k})$
$\sqcap\,(\forall\,\mathrm{then}\,.\,(\mathrm{C}_{\mathrm{I}_1} \sqcup \forall\,\mathrm{then}\,.\,(\mathrm{C}_{\mathrm{I}_2} \sqcup \forall\,\mathrm{then}\,.\,(\ldots \sqcup \forall\,\mathrm{then}\,.\,\mathrm{C}_{\mathrm{I}_k}))))]$



Modeling – Sequence of Outputs

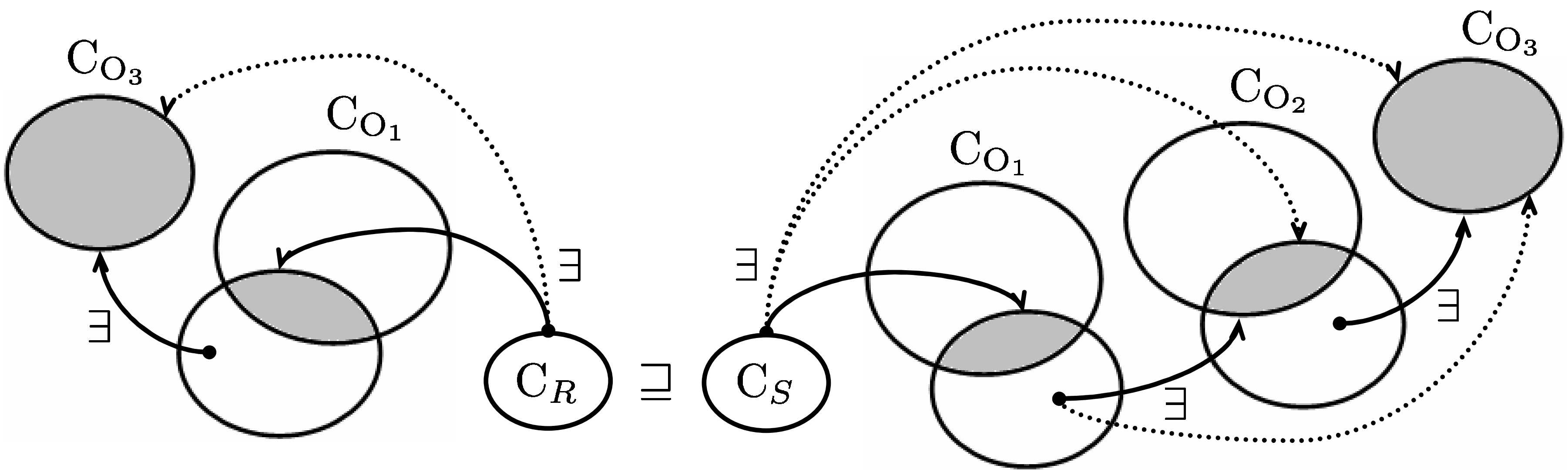
SAP RESEARCH

Implementation

$$|\mathrm{O}_R| \leq |\mathrm{O}_S|$$

- transitive (then)
- $\mathrm{Seq}^{\mathrm{O}} = \mathrm{O}_1 \mapsto \mathrm{O}_2 \mapsto \ldots \mapsto \mathrm{O}_m$

➔ $\mathrm{C} \equiv \exists\,\mathrm{has}\,.\,[(\mathrm{C}_{\mathrm{O}_1} \sqcap \mathrm{C}_{\mathrm{O}_2} \sqcap \ldots \sqcap \mathrm{C}_{\mathrm{O}_m}) \sqcap (\exists\,\mathrm{then}\,.\,(\mathrm{C}_{\mathrm{O}_1} \sqcap \exists\,\mathrm{then}\,.\,(\mathrm{C}_{\mathrm{O}_2} \sqcap \exists\,\mathrm{then}\,.\,(\ldots \sqcap \exists\,\mathrm{then}\,.\,\mathrm{C}_{\mathrm{O}_m}))))]$



Modeling – Sequence of Inputs & Outputs

SAP RESEARCH

Implementation

$|\mathrm{I}_R| \geq |\mathrm{I}_S|$

$|\mathrm{O}_R| \leq |\mathrm{O}_S|$

- $\mathrm{transitive\,(then)}$
- $\mathrm{Seq}^{\mathrm{IO}} = \mathrm{X}_1 \mapsto \mathrm{X}_2 \mapsto \ldots \mapsto \mathrm{X}_m, \quad \text{where } X \in \{I, O\}$

➔ $$\begin{aligned} \mathrm{C} \quad \equiv \quad & \exists\, \mathrm{has}\,.\,[(\mathrm{C}_{\mathrm{I}_1} \sqcup \mathrm{C}_{\mathrm{I}_2} \sqcup \ldots \sqcup \mathrm{C}_{\mathrm{I}_k}) \sqcap (\mathrm{C}_{\mathrm{O}_1} \sqcap \mathrm{C}_{\mathrm{O}_2} \sqcap \ldots \sqcap \mathrm{C}_{\mathrm{O}_l}) \\ & \sqcap (\Omega_1\, \mathrm{then}\,.\,(\mathrm{C}_{\mathrm{X}_1} \;\; \Delta_1 \;\; \Omega_2\, \mathrm{then}\,.\,(\mathrm{C}_{\mathrm{X}_2} \;\; \Delta_2 \;\; \Omega_3\, \mathrm{then} \\ & .\,(\ldots \;\; \Delta_{m-1} \;\; \Omega_m\, \mathrm{then}\,.\,\mathrm{C}_{\mathrm{X}_m}))))] \end{aligned}$$

$$\text{where} \qquad \Omega_i = \begin{cases} \forall, & \mathrm{C}_{\mathrm{X}_i} = \mathrm{C}_{\mathrm{I}_i} \\ \exists, & \mathrm{C}_{\mathrm{X}_i} = \mathrm{C}_{\mathrm{O}_i} \end{cases} \qquad k + l = m$$

$$\text{and} \qquad \Delta_i = \begin{cases} \sqcup, & \mathrm{C}_{\mathrm{X}_i} = \mathrm{C}_{\mathrm{I}_i} \\ \sqcap, & \mathrm{C}_{\mathrm{X}_i} = \mathrm{C}_{\mathrm{O}_i} \end{cases} \qquad 1 \leq i \leq m$$

➔ **Example to follow…**

Modeling – Sequence of Inputs & Outputs

Example

- **Request** $\mathrm{Ordering}^{\mathrm{IO}} = \mathrm{login}^{\mathrm{I}} \mapsto \mathrm{order}^{\mathrm{I}} \mapsto \mathrm{userData}^{\mathrm{I}} \mapsto \mathrm{conf}^{\mathrm{O}}$
- **Services** $\mathrm{Store}^{\mathrm{IO}} = \mathrm{login}^{\mathrm{I}} \mapsto \mathrm{wMsg}^{\mathrm{O}} \mapsto \mathrm{order}^{\mathrm{I}} \mapsto \mathrm{conf}^{\mathrm{O}}$

  $\mathrm{Fraud}^{\mathrm{IO}} = \mathrm{userData}^{\mathrm{I}} \mapsto \mathrm{login}^{\mathrm{I}} \mapsto \mathrm{order}^{\mathrm{I}} \mapsto \mathrm{conf}^{\mathrm{O}}$

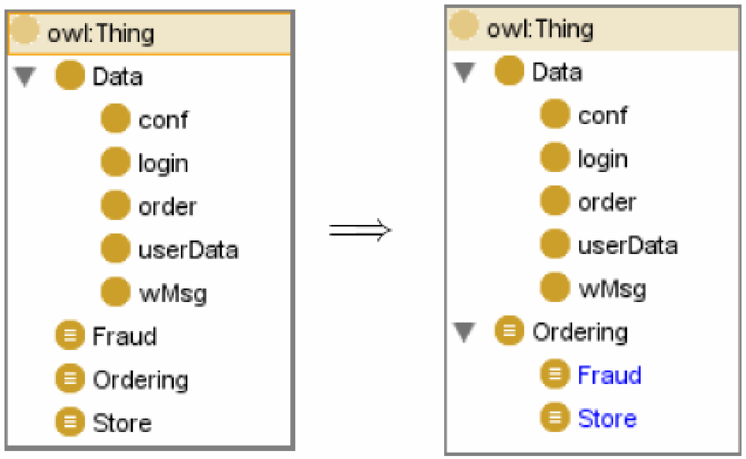


Fig. 1. Inference considering Interactions

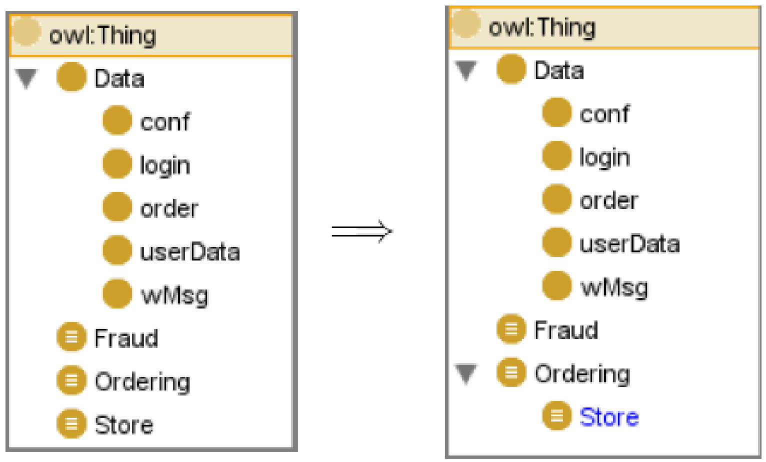


Fig. 2. Inference considering Behavior

SAP RESEARCH

Background

Objectives

Modeling

Conclusion

THE BEST-RUN BUSINESSES RUN SAP

Conclusion – Wrap-up

Summary

- DL models for service interactions
  - Correct and complete if not otherwise stated

- Restricting inputs
- Requiring outputs
- Restricting inputs and requiring outputs

- Sequence of inputs
- Sequence of outputs
- Sequence of mixed inputs and outputs
  - Correct (matches returned are never wrong)
  - Not complete (does not find all possible matches)

Conclusion – Features, Limitations & Open Questions

Features

- Robustness – Facilitation of existing reasoner component
- Integrability – Can reference other semantic Web resources
- Extendabiliy – Easily extendable by other facettes of Web services

Limitations

- At least one input and output, respectively
- No disjoint concepts allowed
- Expressivity

Open Questions

- Complex interaction patterns (eg. fork, decision, loop)
- Proof of properties (eg. correctness & completeness)
- Comparison with other “dynamic” approaches (eg. petri nets)

SAP RESEARCH

SYSTEMATIC THOUGHT LEADERSHIP FOR INNOVATIVE BUSINESS

Thank you

Questions and Answers

THE BEST-RUN BUSINESSES RUN SAP

Copyright 2005 SAP AG. Alle Rechte vorbehalten

SAP RESEARCH

THE BEST-RUN BUSINESSES RUN SAP