Workshop on Testing, Analysis and Verification
of Web Services and Applications

Portland July 17 2006

Towards
Self-Adaptive Service-Oriented Architectures

Giovanni Denaro – Mauro Pezzè – Davide Tosi

University of Milano–Bicocca
Milano, Italy

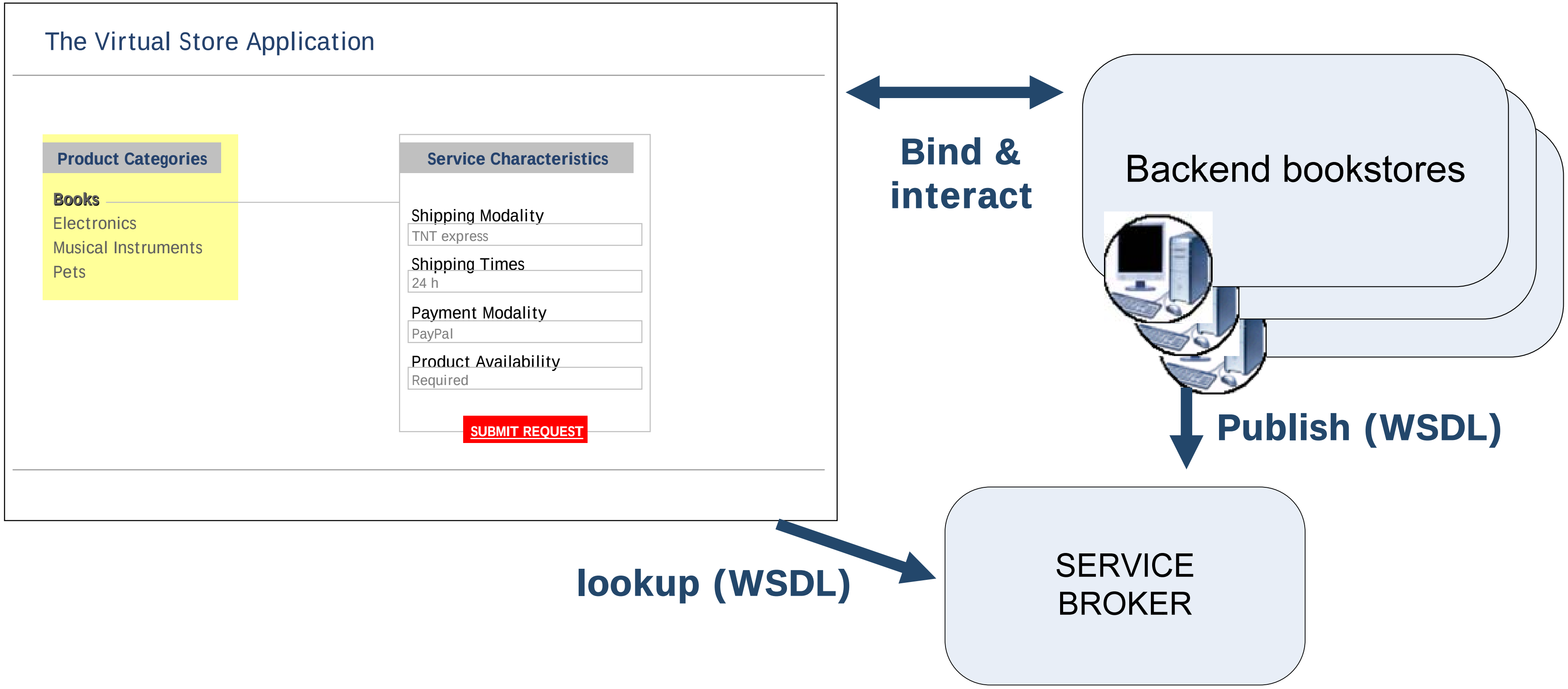
{denaro|pezze|tosi}@disco.unimib.it

Daniela Shilling

University of Paderborn
Paderborn, Germany

das@upb.de

Dynamic SOA Scenario



The Virtual Store Application

Product Categories

Books
Electronics
Musical Instruments
Pets

Service Characteristics

Shipping Modality: TNT express

Shipping Times: 24 h

Payment Modality: PayPal

Product Availability: Required

SUBMIT REQUEST

Bind & interact

Backend bookstores

Publish (WSDL)

lookup (WSDL)

SERVICE BROKER

[from TAV-WEB CfP]
[...] Interfaces play an important role in Web services coordination and interface violations can cause serious problems. [...]

Focus: Interaction Protocol Violations

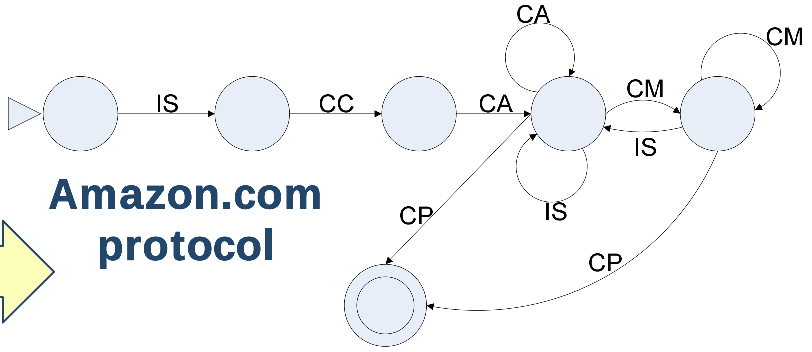
Web services offered by several providers can have:

Same operations (WSDL)...

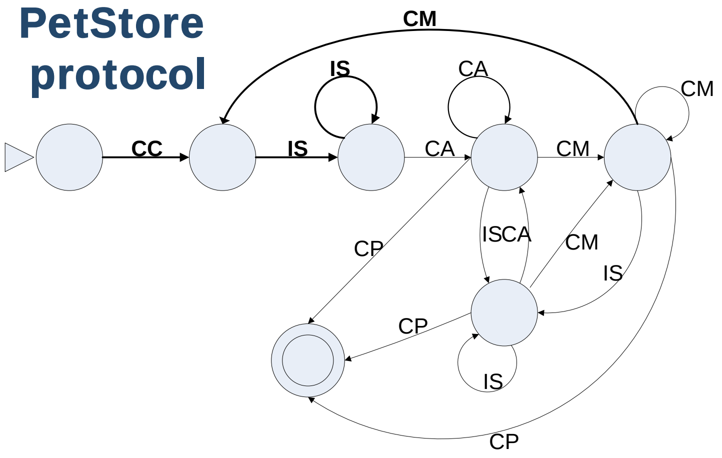
...but different interaction protocols

Shopping Cart Web Service (WSDL)

```
- <operation name="ItemSearch">
      <input message="tns:ItemSearchRequest"/>
      <output message="tns:ItemSearchResponse"/>
  </operation>
- <operation name="CartCreate"> ...
- <operation name="CartAdd"> ...
- <operation name="CartModify"> ...
- <operation name="CartPurchase"> ...
```



Amazon.com protocol



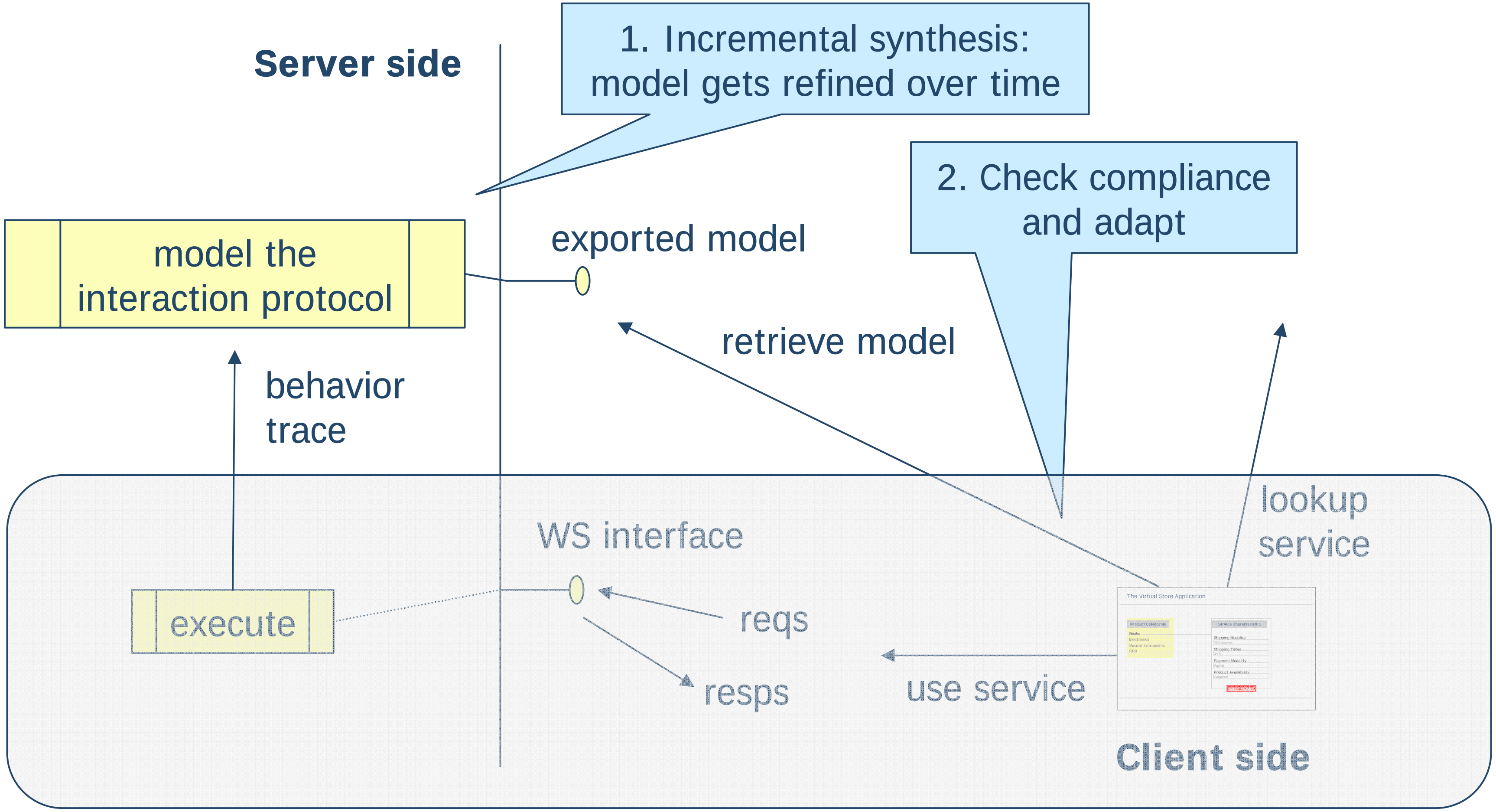
PetStore protocol

Related Work

- Ontologies, semantic web, semantic web services **[OWL-S, WSCI, ...]**
  - Difficulty of defining generally agreed domain ontologies
- Protocol automata **[Foster & al. ICWS'04, Beyer & al. WWW'05]**
  - Allow for checking service compatibility
- Assertions **[Baresi & al. ICSOC'04 and '05]**
  - Useful for specifying behavior of operations but not the interaction protocol

Enable reasoning about properties of WS, but..
Assume availability of specifications!!

Autonomic/Self-adaptive Approach



Server side

1. Incremental synthesis: model gets refined over time

2. Check compliance and adapt

model the interaction protocol

exported model

retrieve model

behavior trace

WS interface

lookup service

execute

reqs

resps

use service

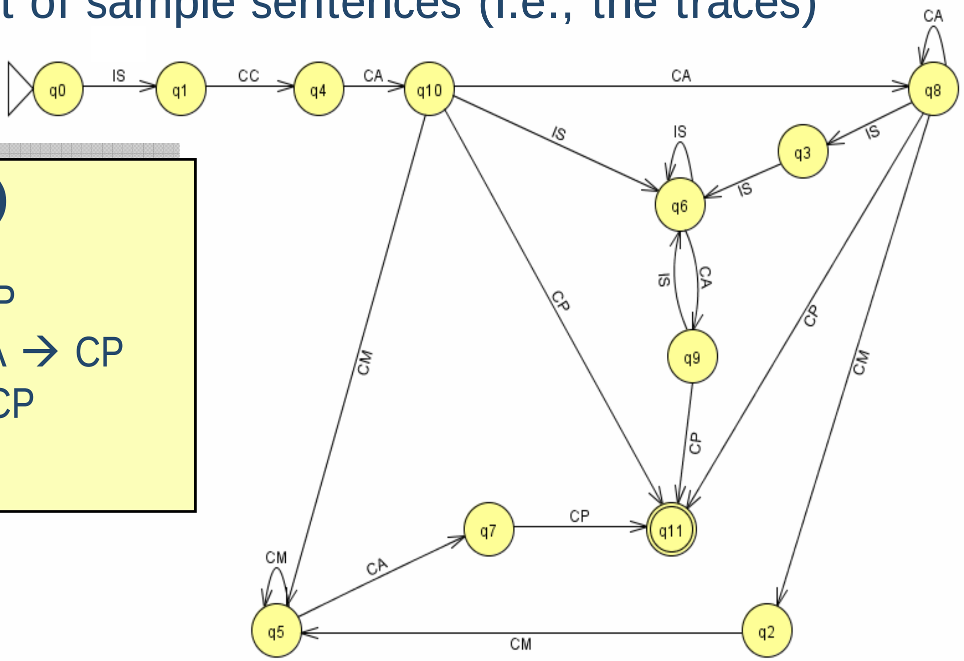
Client side

Synthesizing Interaction Protocols from Traces

- Can be formulated as a problem of inferring a language grammar (i.e., the protocol FSM) from a set of sample sentences (i.e., the traces)

Traces (traced behaviors of WS)

IS → CC → CA → CA → CP
IS → CC → CA → IS → CA → CP
IS → CC → CA → IS → IS → CA → CP
IS → CC → CA → CM → CA → CP
...



We experimented with a purely algorithmic method: **K-Tail**
Merge states based on **equivalence on future behaviors of length K**
(Other algorithms (K-inclusion, Reiss&Reineri) do not yield significant improvements)

K-Tail: [Biermann & Feldman ’72, Cook & Wolf ’96]

Experiments with K-Tail

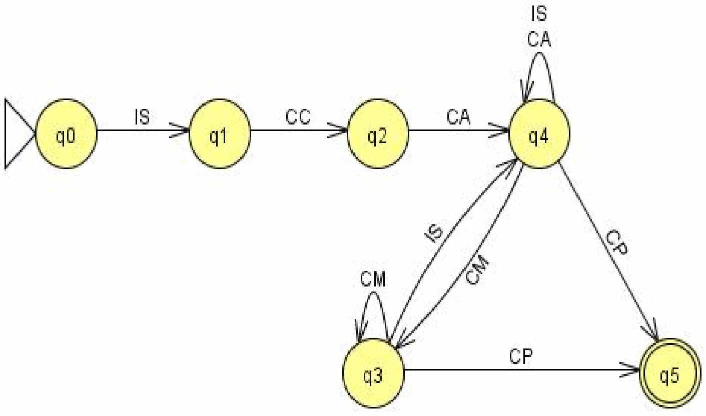
Amazon.com cart web service (K-Tail, k=2)

# Traces	States	Transitions	Analysis Time
3400	6	10	5 h
800	87	276	107 sec
64	24	58	3 sec

K=2 gave the cleanest results

PetStore cart web service (K-Tail, k=2)

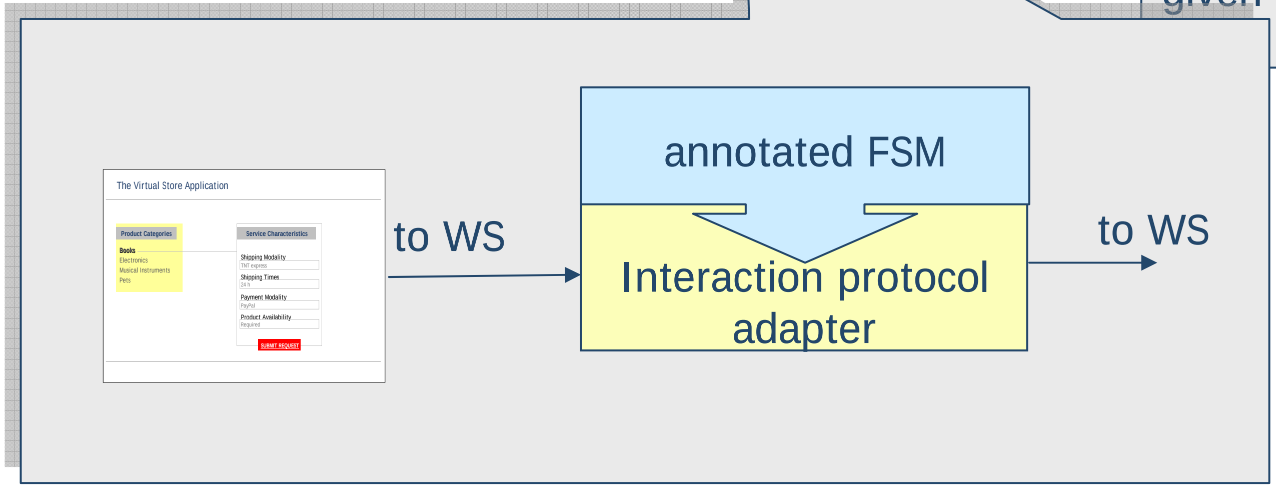
# Traces	States	Transitions	Analysis Time
5000	25	70	8 h
154	36	86	6 sec
105	44	42	3 sec



Client-side issues (work in progress)

- **Goal: check service compliance and adapt based on the retrieved interaction protocols**
- We are currently investigating a solution in which
  - Client-side constraints are given as **partial orders** of operations
  - Clients check FSMs against the partial orders and generate **annotated FSMs**, which point out satisfying paths
  - Uses the annotated FSM to **(re-)configure adapters**

Checking algorithm given in the paper



The Virtual Store Application

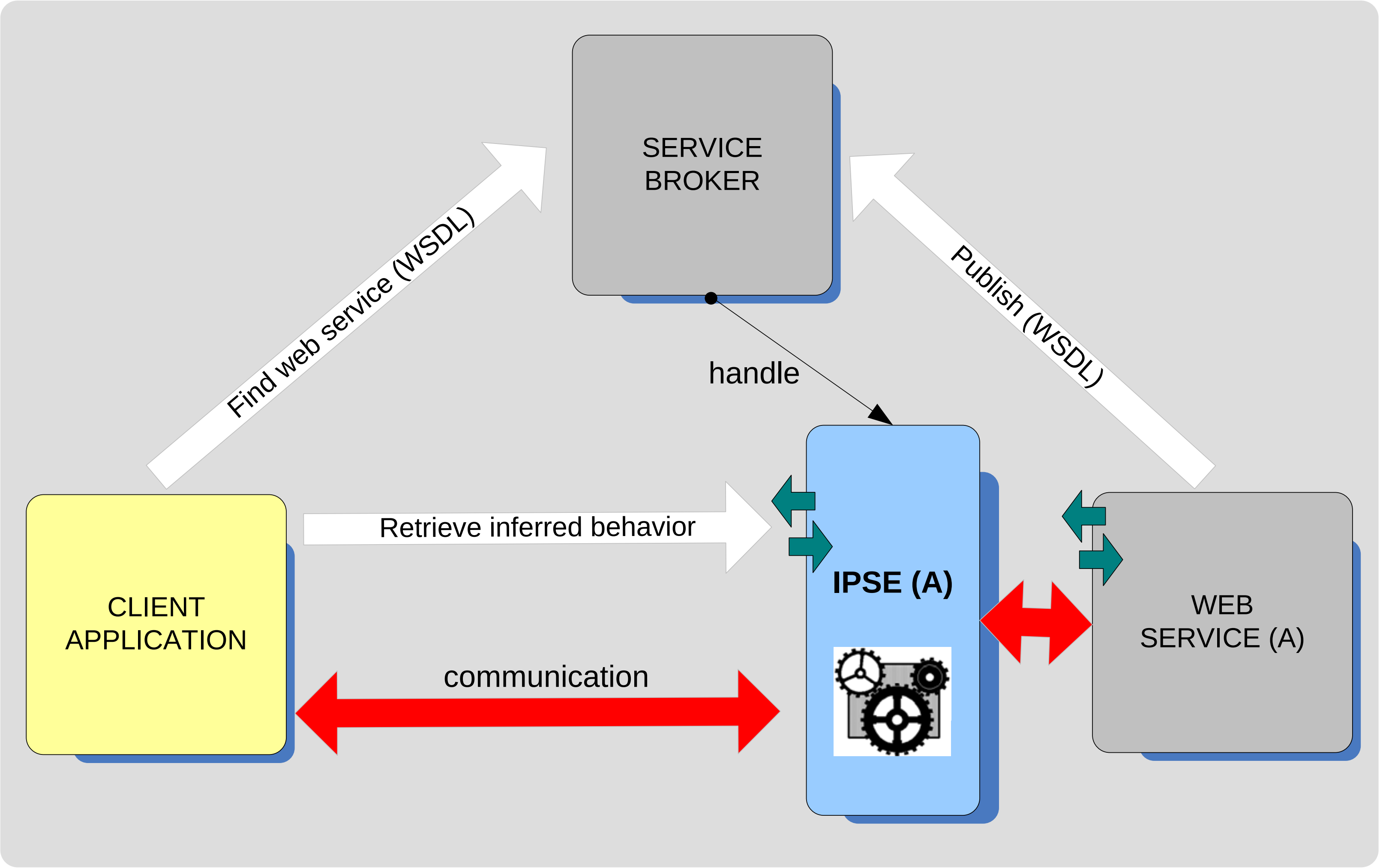
Product Categories

Books
Electronics
Musical Instruments
Pets

Service Characteristics

Shipping Modality: TNT express
Shipping Times: 24 h
Payment Modality: PayPal
Product Availability: Required

Deploying the Enhanced SOA



Conclusions and Research Agenda

- We are investigating a novel self-adaptive approach aimed at improving dependability of dynamic SOAs through:
  - Automatic and incremental discovery of WS interaction protocols
  - Client-side adaptation to “compatible” WS offered through different interaction protocols
- Current achievements
  - Definition and positioning of the approach
  - Experiments showing that synthesis of int. protocols with K-Tail is viable
  - Preliminary ideas on client side adaptation
- Plan
  - Implementation of the approach
  - Investigation of K-Tail using negative samples
  - Further investigation of client-side adaptation mechanisms
  - Empirical evaluation in the field