Capabilities: describing what services do

Phillipa Oaks
Arthur ter Hofstede
David Edmond
CITI, QUT, Brisbane

Outline

- Background
- A capability oriented model
- Previous work
- Approach
- Model
- Conclusion

Background

- 3 styles of web services
  - Toys
    - stock ticker
  - Replacement for EDI
    - WSDL operation signatures
    - XML documents
  - Dynamic, runtime problem solvers
    - not pre-programmed to WSDL interface
    - semantics are not agreed beforehand
    - little or no human involvement at runtime
    - specialization for competitive advantage

Capabilities

- The capability description is an alternative web service interface
  - well defined but not necessarily “standard”
  - structured, machine friendly definition
  - implementation independent
  - built-in facility for handling data mismatch problems

A capability oriented model provides

- Information hiding, separation of concerns
  - A capability description only exposes the relevant information.
  - WSDL exposes the entire interface.
- Loose coupling - what, not how
  - High level description of what is required.
  - Do not have to know implementation details
- Reuse
  - Capability descriptions can be reused, shared, specialized and extended.

Capability descriptions must …

1. Be able to operate in a heterogeneous environment.
2. Explicitly state the action performed or the effect produced.
3. Allow alternative sets of inputs.
4. Describe the objects used or affected by the capability that are not inputs or outputs.
5. Describe preconditions and effects in a named rule language.
6. Increase recall and precision by exploiting existing classifications and offering the full richness of well known cases.

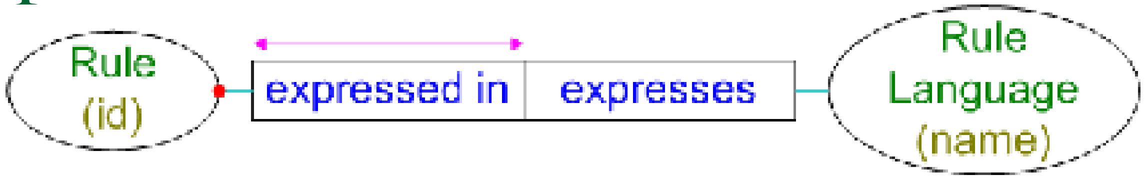
Previous work

- Case frames: who, what, when, where, why.
- Expect: frame structure, rule language.
- LARKS: refined frame. Input, output, preconditions and effects (IOPE's).
- DAML-S/OWL-S Profile: IOPE's in OWL/XML.
  - Capability tied to a specific service, and relies on further information available in the Process Model.
  - Work on discovery and mapping to UDDI.
- ROSA: "normalized" textual descriptions.

Our approach

- Graphical conceptual model
- Requirements based
- Frame structure
- Web aware

ORM conceptual model

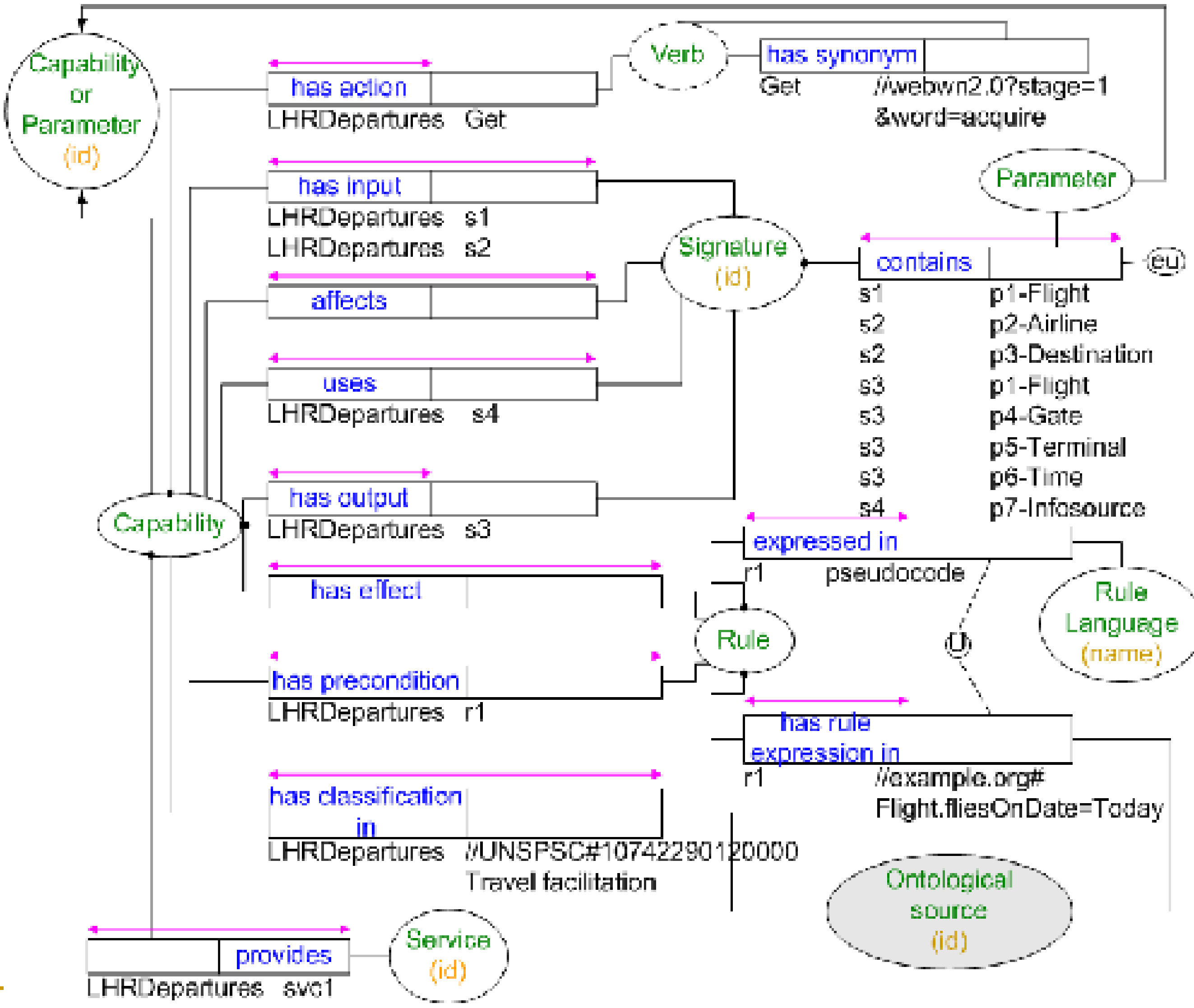


```
<owl:Class rdf:ID="Rule">
  <rdfs:subClassOf> <owl:Restriction> <owl:onProperty rdf:resource="#expressedIn"/>
      <owl:minCardinality rdf:datatype="xsd#nonNegativeInteger">1</owl:minCardinality>
  </owl:Restriction> </rdfs:subClassOf>
  <rdfs:subClassOf> <owl:Restriction> <owl:onProperty rdf:resource="#expressedIn"/>
      <owl:maxCardinality rdf:datatype="xsd#nonNegativeInteger">1</owl:maxCardinality>
  </owl:Restriction> </rdfs:subClassOf>
</owl:Class>
<owl:DatatypeProperty rdf:ID="id“ rdf:type="owl#FunctionalProperty">
  <rdfs:domain rdf:resource="#Rule"/>
</owl:DatatypeProperty>
<owl:ObjectProperty rdf:ID="expressedIn">
  <owl:inverseOf rdf:resource="#expresses"/>
  <rdfs:domain rdf:resource="#Rule"/>
  <rdfs:range rdf:resource="#RuleLanguage"/>
</owl:ObjectProperty>
<owl:Class rdf:ID="RuleLanguage"/>
<owl:DatatypeProperty rdf:ID="name“ rdf:type="owl#FunctionalProperty">
  <rdfs:domain rdf:resource="#RuleLanguage"/>
</owl:DatatypeProperty>
```

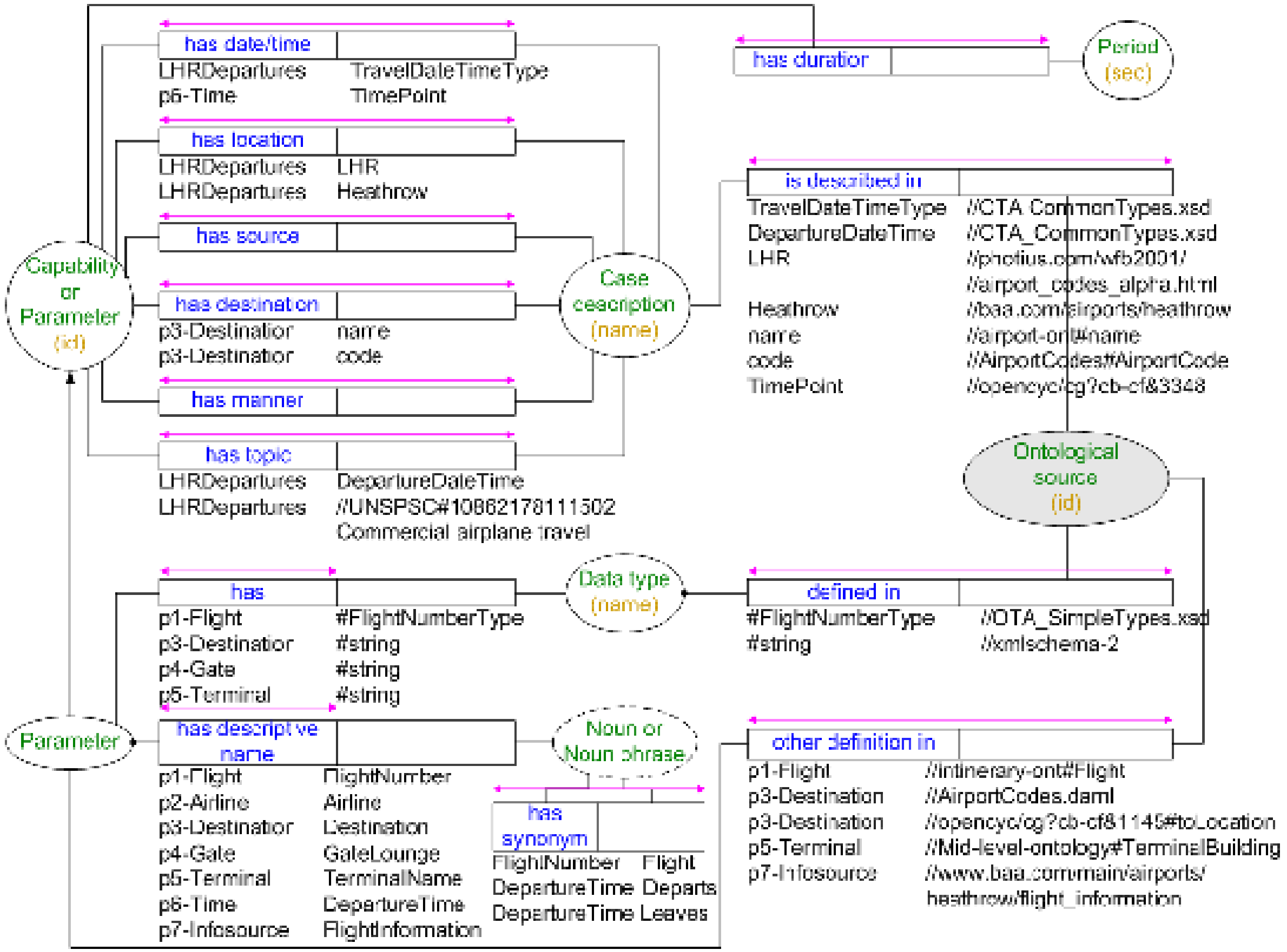
Example

- Consider a service that provides current information about flights departing from Heathrow.
- The user may know the flight number, or only the name of the airline and the destination.

The Model


Capability or Parameter (id)
has action
LHRDepartures Get
Verb
has synonym
Get //webwn2.0?stage=1 &word=acquire
has input
LHRDepartures s1
LHRDepartures s2
Parameter
Signature (id)
contains
eu
s1 p1-Flight
s2 p2-Airline
s2 p3-Destination
s3 p1-Flight
s3 p4-Gate
s3 p5-Terminal
s3 p6-Time
s4 p7-Infosource
affects
uses
LHRDepartures s4
Capability
has output
LHRDepartures s3
expressed in
r1 pseudocode
Rule Language (name)
has effect
Rule
U
has precondition
LHRDepartures r1
has rule expression in
r1 //example.org# Flight.fliesOnDate=Today
has classification in
LHRDepartures //UNSPSC#10742290120000 Travel facilitation
Ontological source (id)
provides
LHRDepartures svc1
Service (id)

model …



has date/time	
LHRDepartures	TravelDateTimeType
p6-Time	TimePoint

has location	
LHRDepartures	LHR
LHRDepartures	Heathrow

has source	

has destination	
p3-Destination	name
p3-Destination	code

has manner	

has topic	
LHRDepartures	DepartureDateTime
LHRDepartures	//UNSPSC#10862178111502 Commercial airplane travel

has duration	

is described in	
TravelDateTimeType	//CTA CommonTypes.xsd
DepartureDateTime	//CTA_CommonTypes.xsd
LHR	//photius.com/wfb2001/ //airport_codes_alpha.html
Heathrow	//baa.com/airports/heathrow
name	//airport-ont#name
code	//AirportCodes#AirportCode
TimePoint	//opencyc/cg?cb-cf&3348

has	
p1-Flight	#FlightNumberType
p3-Destination	#string
p4-Gate	#string
p5-Terminal	#string

defined in	
#FlightNumberType	//OTA_SimpleTypes.xsd
#string	//xmlschema-2

has descriptive name	
p1-Flight	FlightNumber
p2-Airline	Airline
p3-Destination	Destination
p4-Gate	GateLounge
p5-Terminal	TerminalName
p6-Time	DepartureTime
p7-Infosource	FlightInformation

has synonym	
FlightNumber	Flight
DepartureTime	Departs
DepartureTime	Leaves

other definition in	
p1-Flight	//intinerary-ont#Flight
p3-Destination	//AirportCodes.daml
p3-Destination	//opencyc/cg?cb-cf&1145#toLocation
p5-Terminal	//Mid-level-ontology#TerminalBuilding
p7-Infosource	//www.baa.com/main/airports/heathrow/flight_information

Querying the model

- Find a service that can tell when flight BA65 leaves Heathrow and from which terminal.

✓Service
+ provides Capability
  + has **location** Case description with name "**Heathrow**"
  + has **topic** Case description with name "**Departures**"
    or has topic Case description with name is synonym of name "Departures"
  + has **output** Signature
    + contains ✓ Parameter has descriptive name Noun or Noun Phrase "**Departs**"
      or has descriptive name is synonym of Noun or Noun Phrase "Departs"
    + contains ✓ Parameter has descriptive name Noun or Noun Phrase "**Terminal**"
      or has descriptive name is synonym of Noun or Noun Phrase "Terminal"
  + has **input** Signature
    + contains Parameter has descriptive name "**FlightNumber**"
      or has descriptive name is synonym of Noun or Noun Phrase "FlightNumber"

Conclusion: Capability orientation allows

- automated discovery and selection of services based on explicit requirements
- service composition plans independent of specific implementations
- loosely coupled interaction based on required and provided capabilities rather than WSDL interfaces
- access to external definitions and ontologies
- higher level interfaces

Current work

- Interaction model based on the provision of capabilities
- Alignment of the extensions to conceptual models for web-based applications (ER2003) with the capability description.

- Questions?