CoOL: A Context Ontology Language to enable Contextual Interoperability

Thomas Strang
thomas.strang@dlr.de

Korbinian Frank
frank@informatik.uni-muenchen.de

Claudia Linnhoff-Popien
linnhoff@informatik.uni-muenchen.de

Context-Aware Services

CoOL: A Context Ontology Language to enable Contextual Interoperability

Outline

- Motivation & Requirements
- New model to specify contextual knowledge
- Using the model in context-aware service frameworks
- Determination of contextual interoperability

Next step...

The Requirements

Ubiquitous Computing

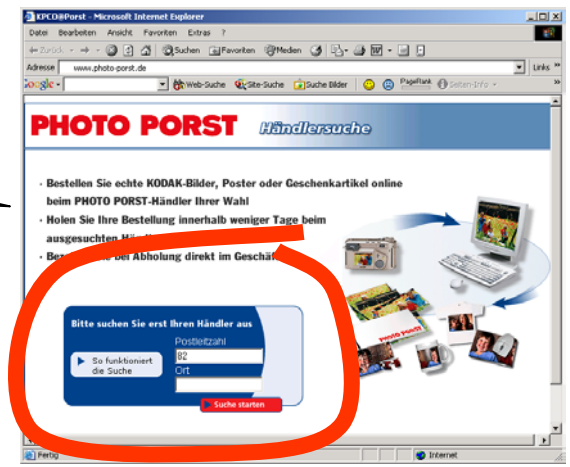
Distributed Computing → + **Mobile Computing** → + **Ubiquitous Computing** →

Mobile Networks
Mobile Information Access
Adaptive Applications

Context-Awareness
Ad-hoc Networks
Smart Sensors & Devices

GNSS
Mobile Radio Network
Inertial Navigation Sys
Short Range Network

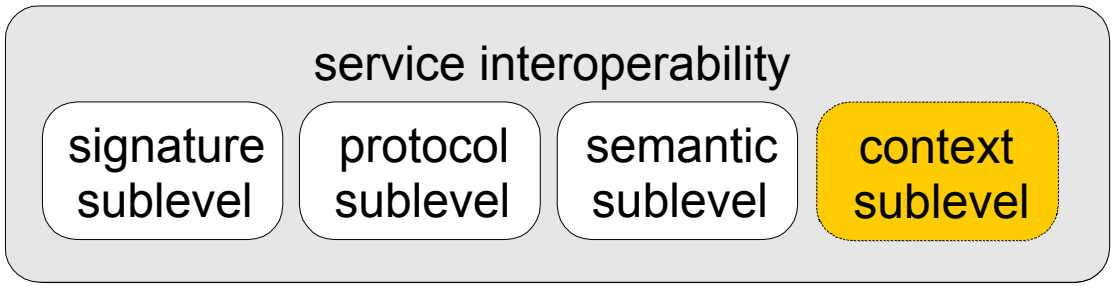
Interoperability Specification Gap



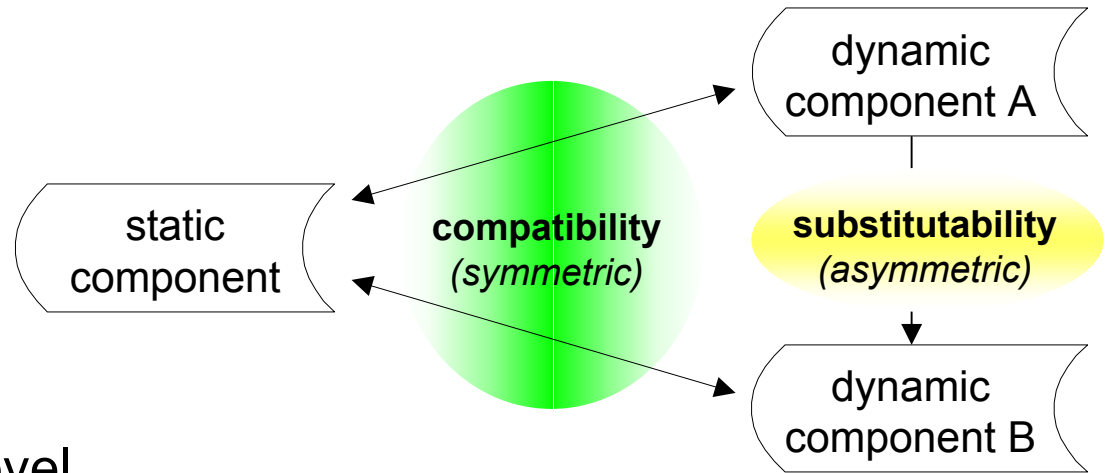
Everything which has to be evaluated by the computer must be specified!

Service Interoperability

Levels:



Perspectives:



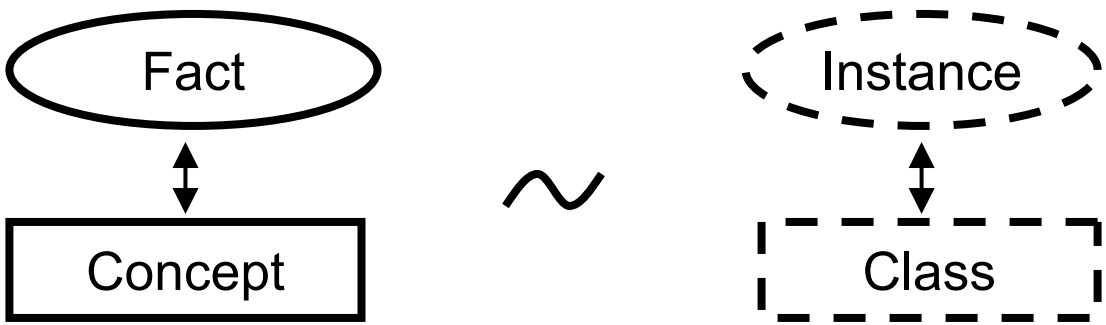
- on any (sub-)level
  - **compatibility** and **substitutability** defined individually
  - **specification of shared knowledge** enables interoperability

Requirements

Requirements on **specification methodology:**

- high level of formality √
- distributed composition √
- partial validation √
- incompleteness √
- quality of information
- equal use of scalar and non-scalar types
- applicability to existing service frameworks

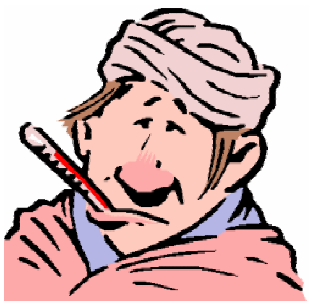
> ***"An ontology is a hierarchically structured set of terms for describing a domain that can be used as a skeletal foundation for a knowledge base."***
> by Swartout, Patil, Knight and Russ, 1996



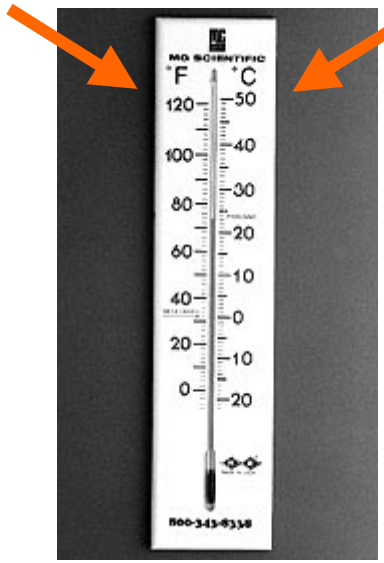
Next step...

The Model

New Model
to specify Contextual Knowledge



Temperature



5°C
35°C
3°F
-7°C
10°C
60°F
95°F
0°C
13°C
80°F

concept
Aspect

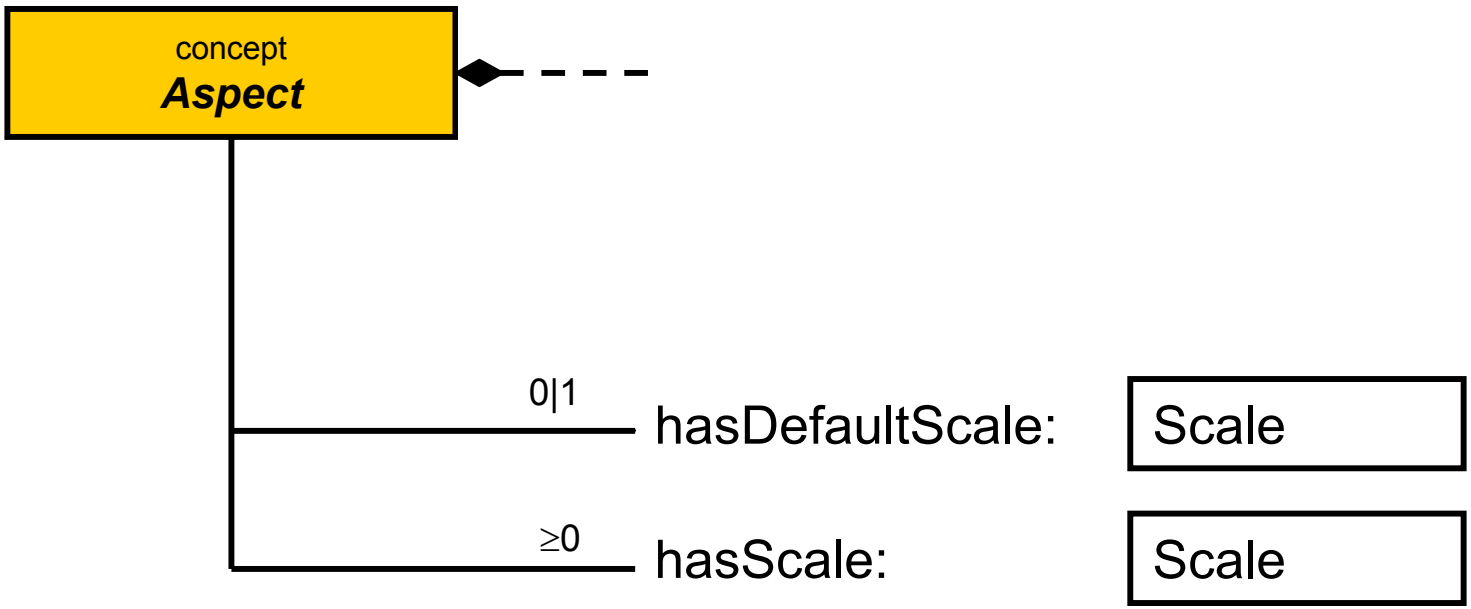
concept
Scale

concept
ContextInformation

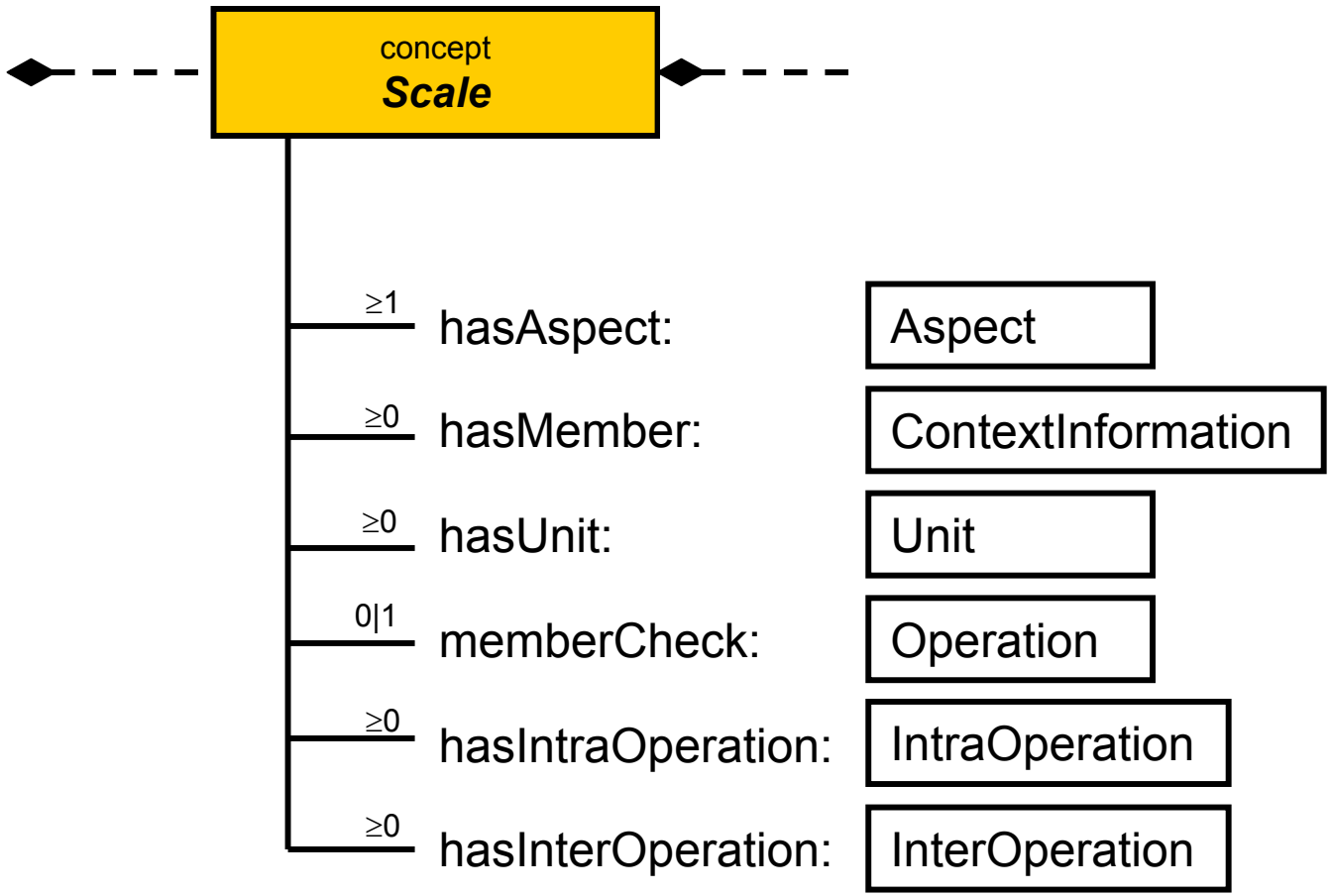
concept
Entity

characterized by

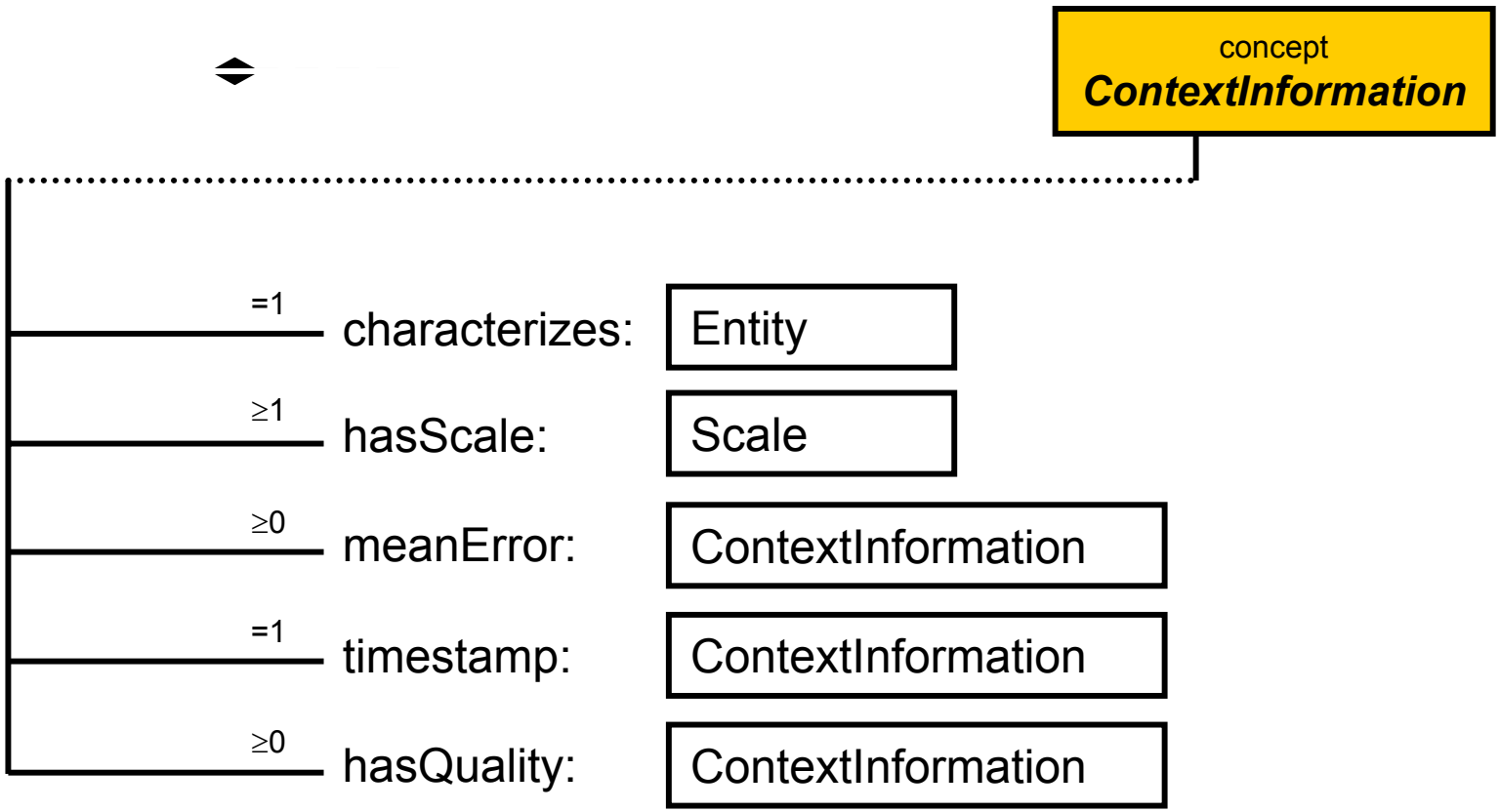
Frame oriented view to the ASC Model (I)



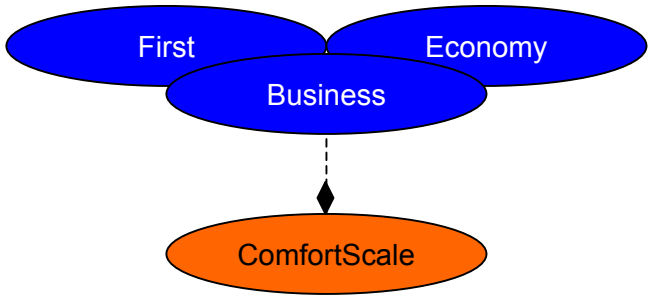
Frame oriented view to the ASC Model (II)



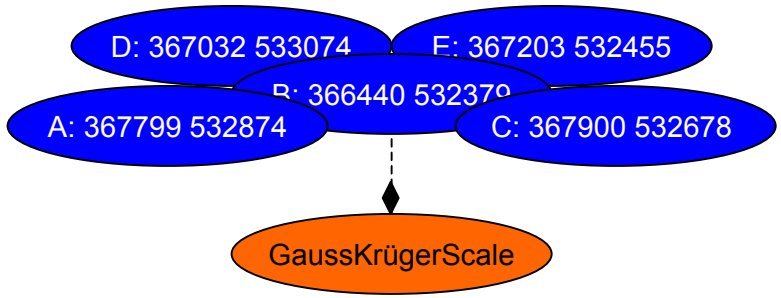
Frame oriented view to the ASC Model (III)

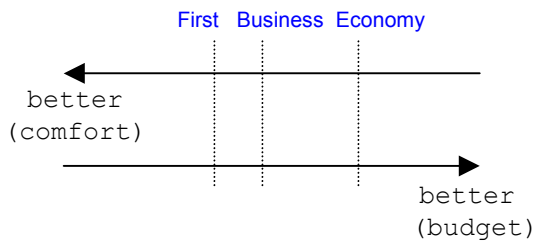


Scales and Metrics



Skales are
unsorted sets!

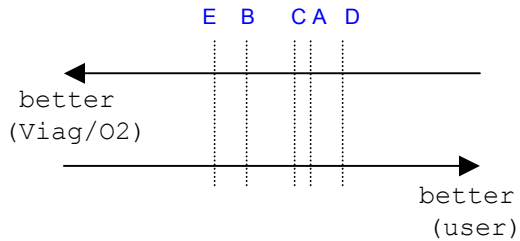




Application specific
order by use of
MetricOperation

$ci_1 | ci_2 = \text{better}(ci_1, ci_2, p_j)$

$c_i \in \text{Scale}$




```
ComfortScale comfortScale=
  new Vector( First,
              Business,
              Economy);
comfortScale.contains( ci );
```

Construction
and
Access

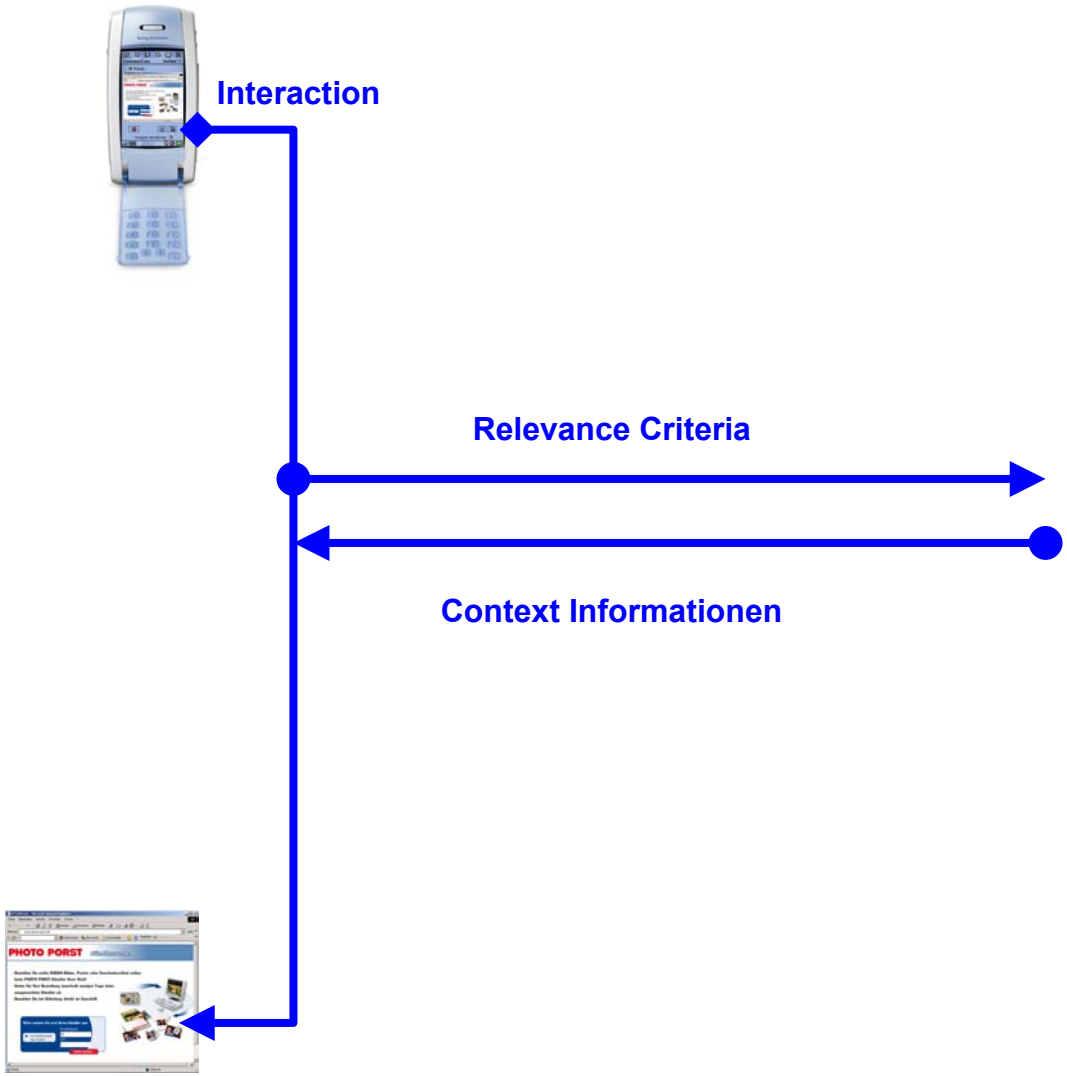
```
GKScale gkScale = new
  GKScale("Tour Eiffel", 10km);
gkScale.checkMember( ci );
```

Enables comparison and order of non-scalar types!

Next step...

The Architecture

Context-Aware Service Framework integrating the Model



Next step...

Determination of Interoperability

Context Bindings

Signature of an operation:

Signature Level

```
String orderPizza( int pizzaNumber,
                   String deliveryAddress,...);
```

What is it ? ? ?

Semantic Level

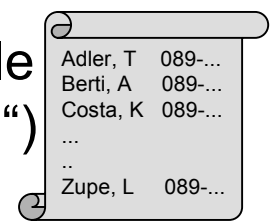
Which values ? ? ?

Context Level

Context Binding = Binding of **Parameter** to **Scale+Aspect**

Example: Parameter **`deliveryAddress`**

specific scale („all with 089...“)



Aspect **„Postal Address“**

Contextual Compatibility

Three types of parameter:

- input parameter
- output parameter
- implicit parameter

Which values **expects** the operation?

Which values **delivers** the operation?

For which values is the service itself **available**?

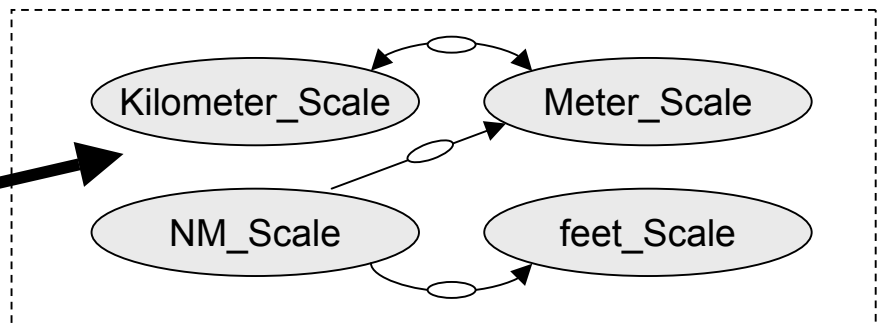
Determination of
Contextual Compatiblity

```
Example: Query

FORALL S <-
#isKompatible(#Kilometer_Scale,S).

Result: #Kilometer_Scale,
        #Meter_Scale,
        #NM_Scale
```

from Binding



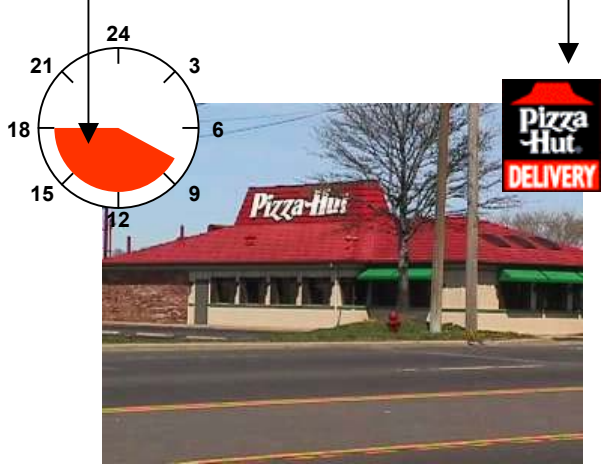
Contextual Substitutability

Three types of parameter:

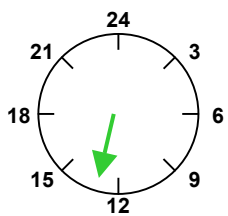
- input parameter → Which values **expects** the operation?
- output parameter → Which values **delivers** the operation?
- implicit parameter → For which values is the service itself **available**?

Scale is the **Scope**

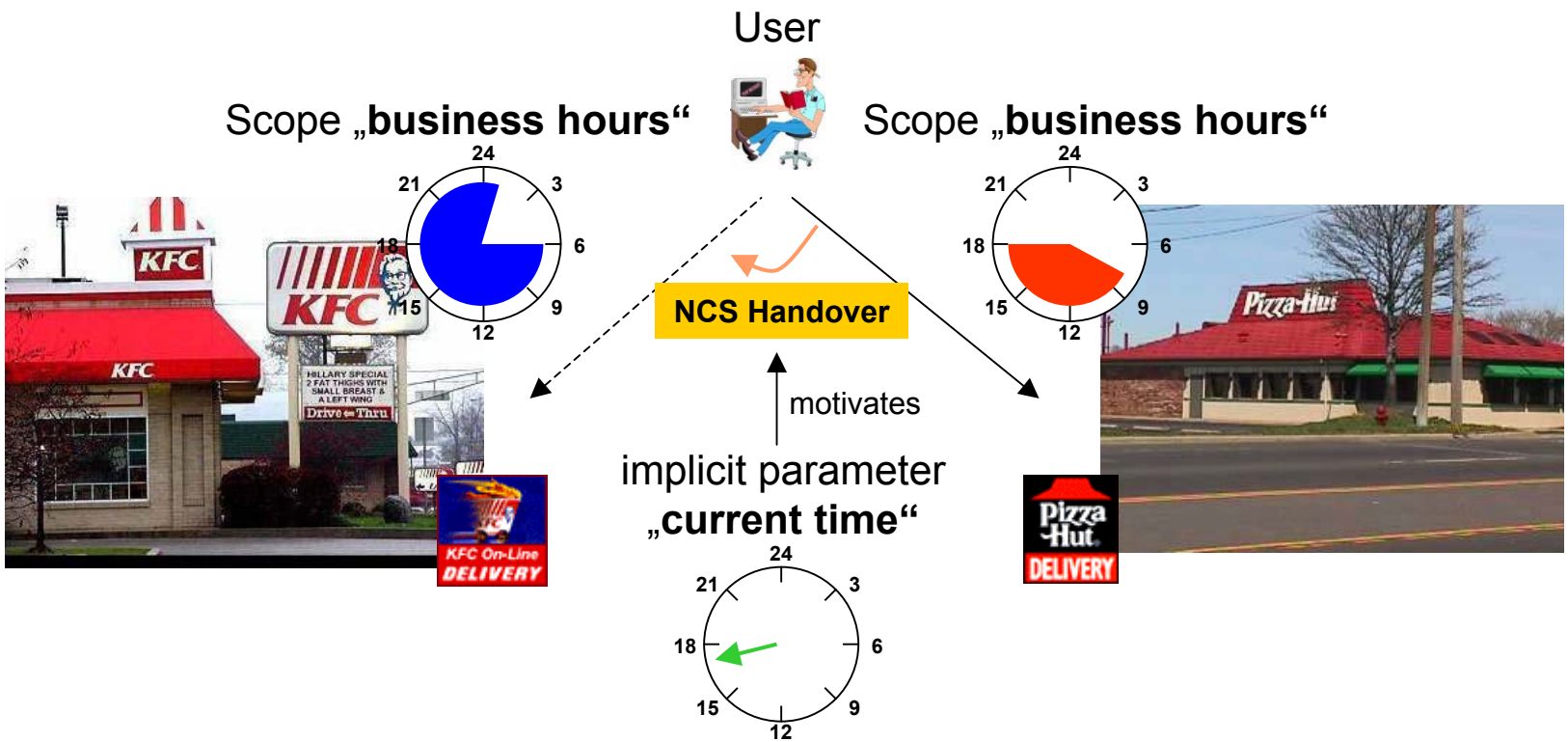
Scope „business hours“
for service **„food delivery“**



implicit parameter
„current time“

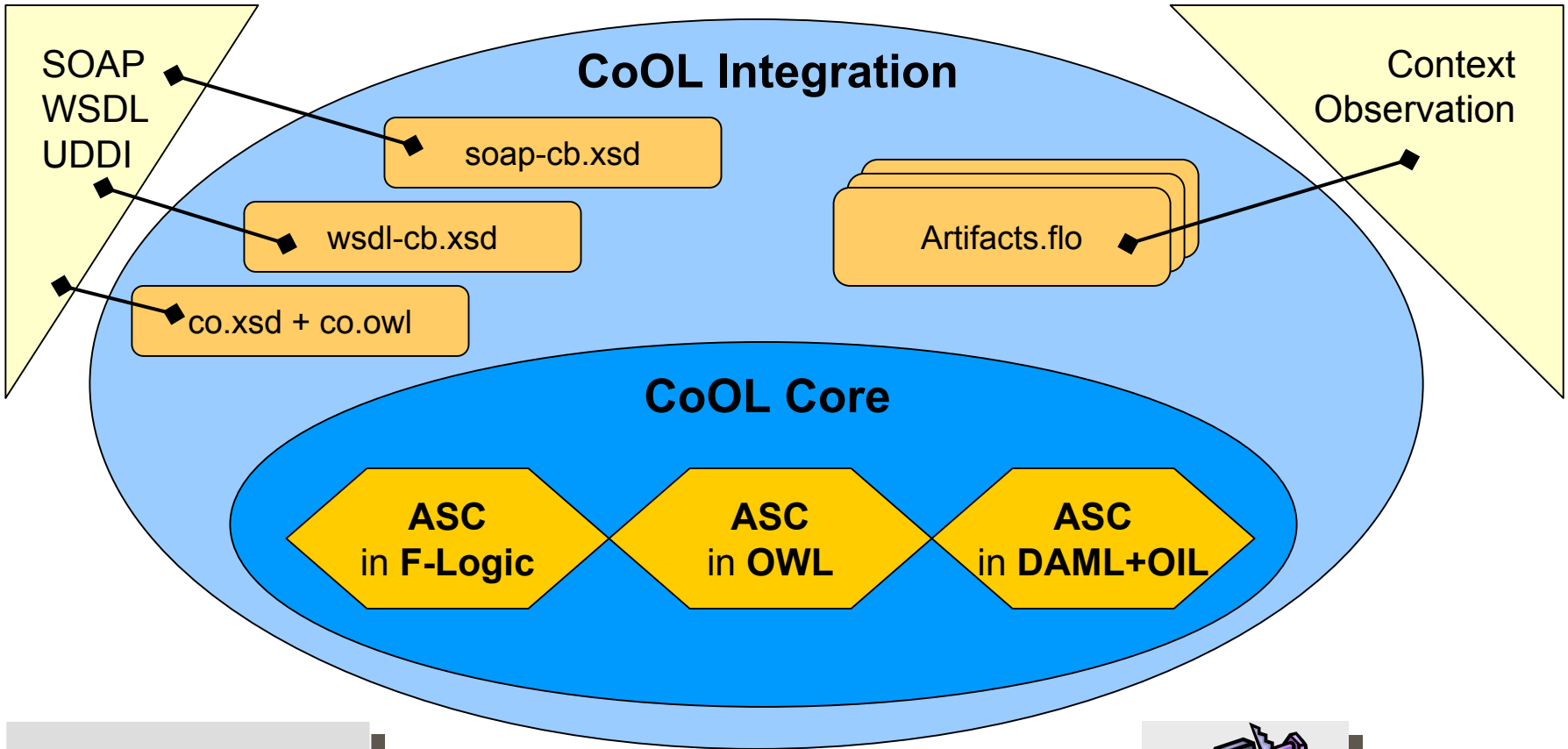


Contextual Substitutability



Any change in context can motivate/cause a handover!

Context Ontology Language (CoOL)



Reasoner
(Inference Engine)

individual use for
Inferencing + Validation

Tools
(e.g. from
Semantic Web)

Summary and Conclusion

- Specification of contextual knowledge based on the ASC model using ontologies
- Architecture with ontology reasoner in the context provider domain
- Interoperability determination via context bindings
- Conclusion: Context Ontology Language (CoOL) as a projection of the model into a specification language can be used to specify and determine interoperability from different perspectives.

Thank you!

For further information:

Thomas Strang <thomas.strang@dlr.de>

http://www.dlr.de/kn/kn-s/strang