



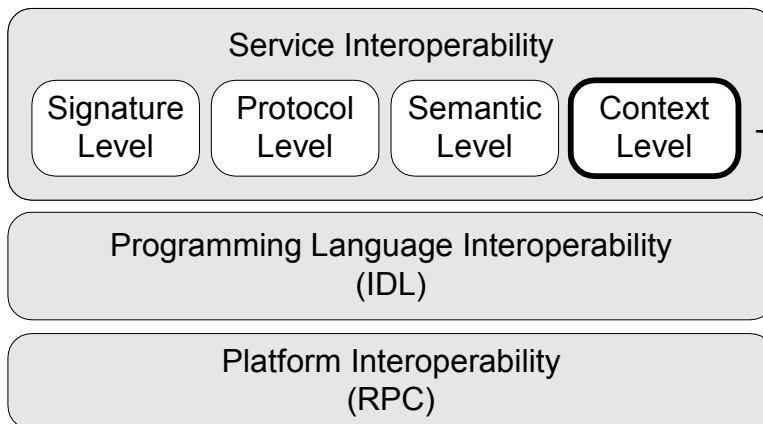
Contextual Service Interoperability

Thomas Strang <thomas.strang@dlr.de>

DLR Oberpfaffenhofen



Interoperability Levels



- **What is Context**
 - ▶ Definitions
 - ▶ Examples
- **Why own level?**
- **Contextual Interoperability**
 - ▶ Model View
 - ▶ Interface identification
- **Context Ontology Language**
 - ▶ Open Issues + Discussion



What is Context (1/2) ?

■ Definition of *Context Information*

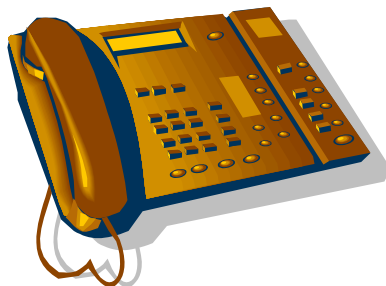
- ▶ “A *context information* is any information which can be used to characterize the state of an entity concerning a specific aspect”
- ▶ “An *entity* is a person a place or in general an object”
- ▶ “An *aspect* is a classification, symbol or value-range, whose subsets are a superset of all reachable states”

■ Example:

Entity: Telephone

Aspect: Call State

Context Information: specific state



{ { on-hook & idle },
 { on-hook & ringing },
 { off-hook } }

CI(entity=+4917912345,
 aspect=callState) =
 { off-hook }



What is Context (2/2) ?

■ Definition of *Context*

- ▶ “A *context* is the set of all context information characterizing the entities relevant for a specific task in their relevant aspects”
- ▶ “An *entity is relevant* for a specific task, if its state is characterized at least concerning one relevant aspect”
- ▶ “An *aspect is relevant*, if the state with respect to this aspect is accessed during a specific task or the state has any kind of influence on the task”

■ Example:

Task: traveller reaching plane using public transport



Relevant Entities

- traveller
- carrier (taxi, train, bus...)
- infrastructure (roads...)
- plane to reach

Context: all CI of relevant aspects

- willingness to spend money
- duration/delay, cost
- delay
- time of departure + buffer



Context Awareness

■ Definition of *Context Awareness*:

- ▶ “A system is *context aware* if it uses any kind of context information before or during service provisioning or service usage”

■ *Location Awareness* is special issue of **Context Awareness** (but by far not the only one!)

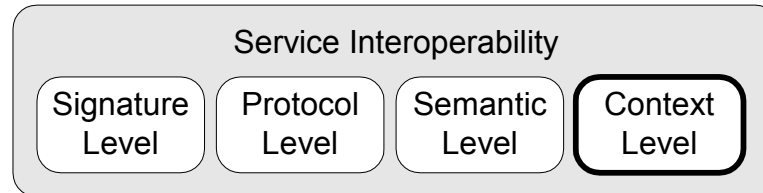
■ Two main benefits from **Context Awareness**:

- ▶ Adaptation of services to changes in environment reduces amount of interaction with user
- ▶ Improvement of UI (particularly on small mobile devices)





Context: Why own interoperability level?

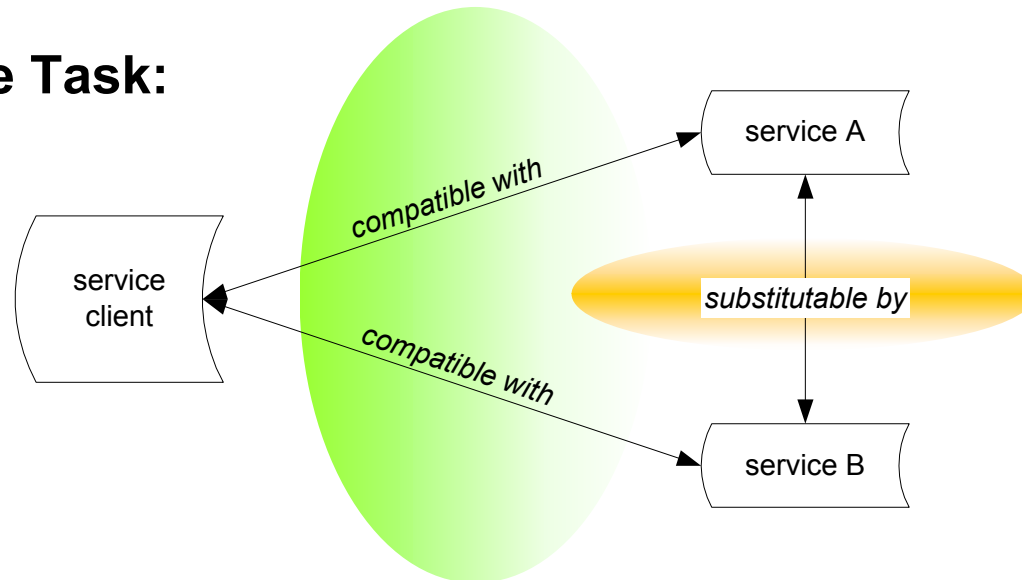


- eases **separation of services** which are interoperable on classic levels, but not on context level (and vice versa)
 - ▶ Example: electronic public transport timetable service
- context is **not covered very well** in classic levels
- enables **thematic concentration** for correctness and integrity



Interoperability in Middleware

■ Middleware Task:



■ Interoperability Evaluation during Service Discovery

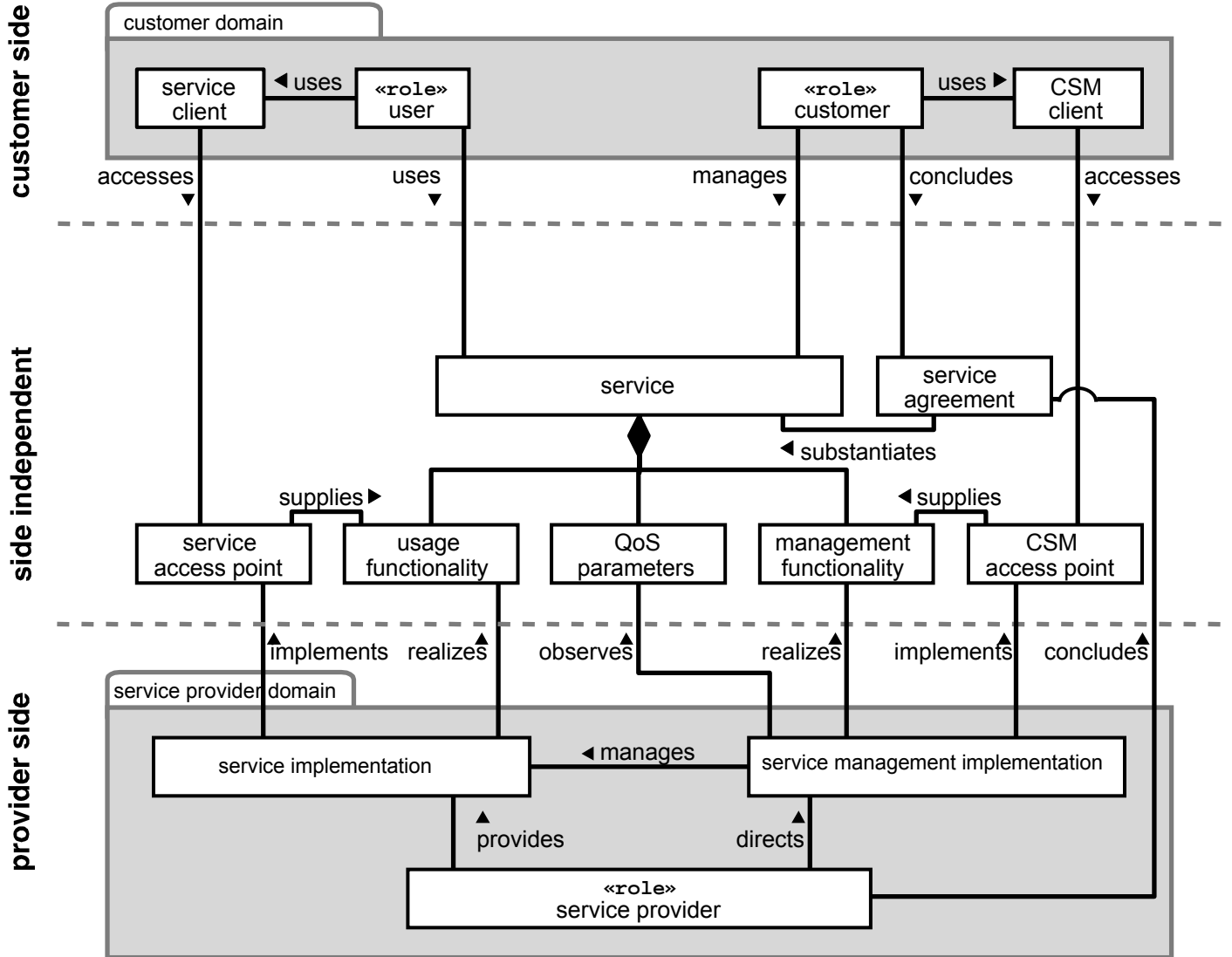
- ▶ service search (feed)
- ▶ service selection (feed back)

■ Interoperability Evaluation during Service Execution

- ▶ lifecycle monitoring and management
- ▶ notification about and adaptation to context changes
- ▶ inter-provider handover



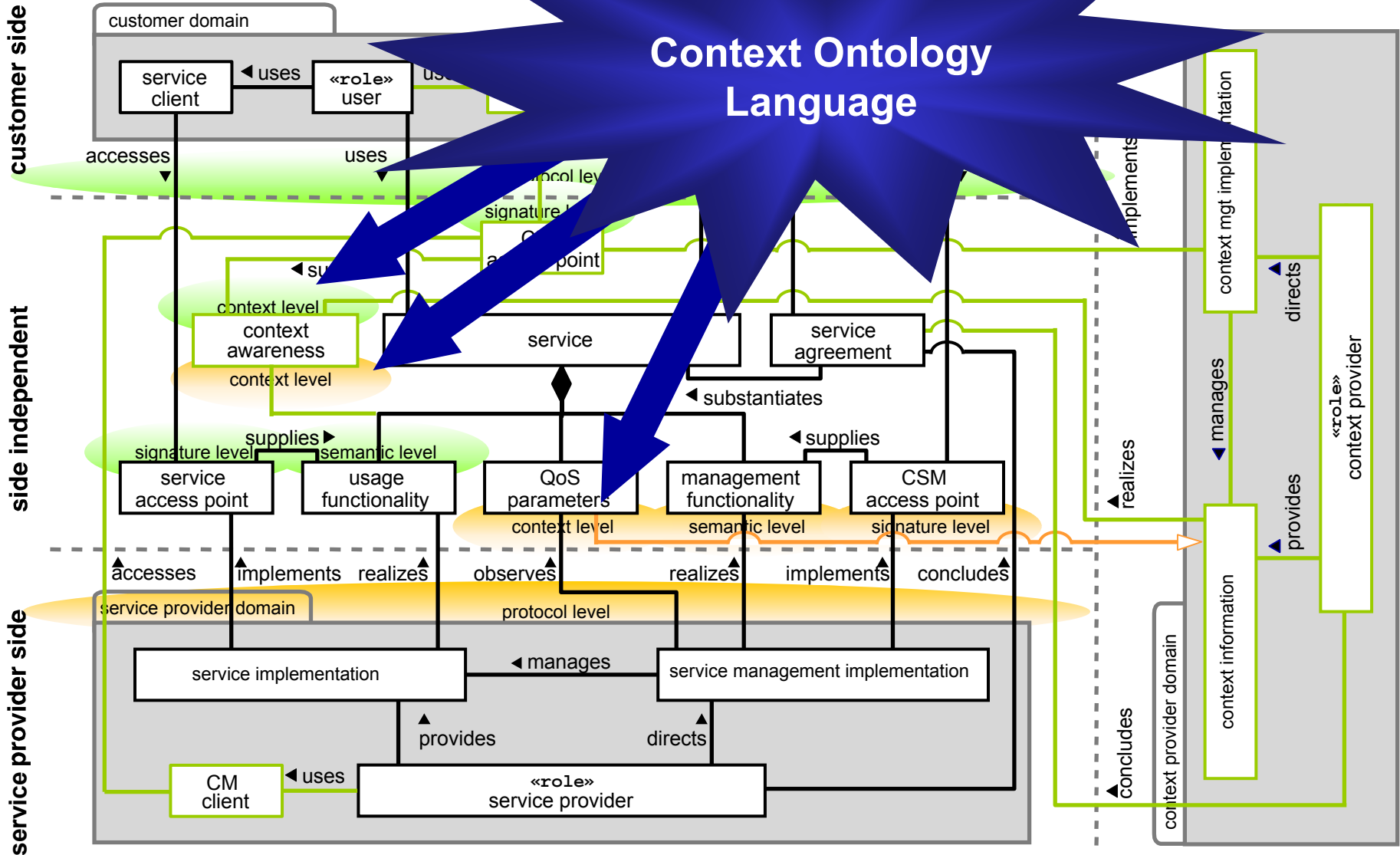
MNM Service Model: Service View



Source: Garschhammer, M., Hauck, R., Hegering, H.-G., Kempter, B., Radisic, I., Rölle, H., Schmidt, H., *A Case-Driven Methodology for Applying the MNM Service Model*, in Stadler, R., Ulema, M., editors, Proceedings of the 8th International IFIP/IEEE Network Operations and Management Symposium (NOMS 2002), pp. 697-710, IEEE Publishing, IFIP/IEEE, Florence, Italy, April, 2002.



Interoperability in Extended M2M S-Model





Design of Context Ontology Language (CoOL)

- XML & XML schema based
 - ▶ probably based also on RDF
- used to model
 - ▶ *aspects* defining valid context information
 - ▶ *relevance* conditions
 - ▶ *actors* and *dependencies*
- enables dynamic (at runtime) multi-party service interoperability checks at new *context level*
- **CoOL** is employed during service discovery **and** execution to negotiate and monitor context to enable context awareness of middleware and services
- Middleware uses CoOL statements to **link distributed object-associated context information** with **context un-aware objects and services**



Open issues: Current Investigations on CoOL

- **Pure XML schema vs. RDF / RDF Schema**
 - ▶ complexity of RDF/RDFS vs. resource limited devices
 - ▶ some drawbacks of RDF have already been identified e.g. in [Haustein2001] [Furche2001] [Indulska2003]
 - ▶ but significant body of work has been done with RDF

- **“Plugin Model” to DAML-S vs. Standalone Model**
 - ▶ DAML-S / DAML+OIL adds additional complexity

- **WebOnt (OWL) vs. minimum language**
 - ▶ OWL in very early state



Summary

- New **context level** of service interoperability
 - ▶ advantages particularly in Ubiquitous Computing Environments
- Clear **definitions** what context is and what not
 - ▶ Introduction of the **aspect**
 - ▶ affiliated context awareness
 - ▶ Examples
- **Context extension** of the MNM service model
 - ▶ Identification of interoperability requirements
- **Outline** of Context Ontology Language (CoOL)
 - ▶ still some major open issues



Thank You!

■ Any Questions?

Dipl.-Inform. Thomas Strang
<mailto:thomas.strang@dlr.de>
<http://www.dlr.de/kn/kn-s/strang>

German Aerospace Center (DLR)
Site Oberpfaffenhofen
Institute of Communications and Navigation (KN-S)
PO Box 1116
82230 Wessling-Oberpfaffenhofen
Tel: +49-8153-28-1354
Fax: +49-8153-28-1871

