Service modelling with ISO 19119
- with examples from ISO 19164 Registry –

ISO/TC211 Standards in Action Workshop
Berlin, Germany, June 4th, 2014

Dr. Arne-Jørgen Berre
Chief Scientist, SINTEF
Oslo, Norway
Phone: +47 9204 7452
E.mail: Arne.J.Berre@sintef.no
Outline

- ISO 19119
- ISO 19164 Registry – example (in progress …)
- ISO RM/ODP viewpoints
- Platform neutral and Platform specific service specifications
- Modeling of services in UML
- Service taxonomies
- Abstract Test Suites
- Next steps
ISO 19119 Scope

- The standard defines requirements for how platform neutral and platform specific specification of services shall be created, in order to allow for one service to be specified independently of one or more underlying distributed computing platforms.

- The standard defines requirements for a further mapping from platform neutral to platform specific service specifications, in order to enable conformant and interoperable service implementations.

- This International Standard addresses the Meta:Service foundation of the ISO geographic information reference model described in ISO 19101-1:2014, Clause 6 and 8 respectively.

- The standard defines how geographic services shall be categorised according to a service taxonomy based on architectural areas, and allows also for services to be categorised according to a usage life cycle perspective, as well as according to domain specific and user defined service taxonomies, providing support for easier publication and discovery of services.
ISO RM-ODP viewpoints

- **Enterprise**
  - Business Aspects
  - The purpose, scope and policies for the organizations that will own the systems

- **Information**
  - Information System Aspects
  - Information handled by the system and constraints on the use and interpretation of that information
  - What is it about?

- **Computational/Service**
  - Application Design Aspects
  - Functional decomposition of the system into objects suitable for distribution
  - How does each bit work?

- **Technology**
  - Implementation
  - System hardware & software and actual distribution
  - With what?

- **Engineering**
  - Solution Types & Distribution
  - Infrastructure required to support distribution
  - How do the bits work together?
ISO RM-ODP and Interoperability

- Political Context
- Organisational Interoperability
  - Computational Viewpoint
- Semantic Interoperability
  - Engineering Viewpoint
  - Implementation
- Technical Interoperability
  - Information Viewpoint
  - Technology Viewpoint
- Legal Interoperability
  - Enterprise Viewpoint

Semantic Interoperability
Technical Interoperability
## Viewpoint descriptions

<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>enterprise viewpoint</td>
<td>a viewpoint on an ODP system and its environment that focuses on the purpose, scope and policies for that system</td>
</tr>
<tr>
<td>computational viewpoint</td>
<td>a viewpoint on an ODP system and its environment that enables distribution through functional decomposition of the system into objects which interact at interfaces</td>
</tr>
<tr>
<td>information viewpoint</td>
<td>a viewpoint on an ODP system and its environment that focuses on the semantics of information and information processing</td>
</tr>
<tr>
<td>engineering viewpoint</td>
<td>a viewpoint on an ODP system and its environment that focuses on the mechanisms and functions required to support distributed interaction between objects in the system</td>
</tr>
</tbody>
</table>
From platform-neutral abstract specifications to multiple platform-specific specifications
## Enterprise Viewpoint

<table>
<thead>
<tr>
<th>Requirements class</th>
<th>/req/enterpriseviewpoint</th>
</tr>
</thead>
<tbody>
<tr>
<td>Target type</td>
<td>Service description</td>
</tr>
<tr>
<td>Requirement</td>
<td>/req/enterpriseviewpoint/servicename</td>
</tr>
<tr>
<td>Requirement</td>
<td>/req/enterpriseviewpoint/servicetypes</td>
</tr>
<tr>
<td>Requirement</td>
<td>/req/enterpriseviewpoint/purpose</td>
</tr>
<tr>
<td>Requirement</td>
<td>/req/enterpriseviewpoint/scope</td>
</tr>
<tr>
<td>Requirement</td>
<td>/req/enterpriseviewpoint/capabilities</td>
</tr>
<tr>
<td>Requirement</td>
<td>/req/enterpriseviewpoint/community</td>
</tr>
<tr>
<td>Recommendation</td>
<td>/rec/enterpriseviewpoint/scenarios</td>
</tr>
</tbody>
</table>
ISO 19164 Enterprise Viewpoint

- **Purpose**
  1. Support transactional registers of information artefacts. (e.g. Data sets, Services, CRS, .... )
  2. By 1. support catalogues of services & datasets.
  3. Provide a framework for describing semantics of geo-spatial features and related artefacts.
  4. Provide a framework for the realization of Master Data Management Hubs for data objects with location.

- **Community of users & kinds of information**
  - Geospatial dataset and services - owners and users
  - Aviation design-time artefacts. (i.e. SWIM) (Aviation)
  - Aviation run-time artefacts. (i.e. SWIM) (Aviation)
  - Geodetic codes & parameters (i.e. CRS) (Geophysics)
  - Pipeline Component types/instance register (Pipeline operation)
  - ....
ISO 19164 Enterprise Viewpoint

- Use Cases:
  - Create/Update/Delete Domain Information Model
  - Create/Update/Delete Domain Specific Artefact
  - Request (Domain Information Model; Request Parameters)
  - Request (Domain Information Model Fragment; Request Parameters)
  - Request (Domain Specific Artefact; Request Parameters)
  - Create/Update/Delete Domain Information Model Constraints
  - Register For Notification (Audible Event, Notification Method, User)
  - Notify User() - Need Event Object
  - Harvest (Source Data, Domain Information Model)
## Computational Viewpoint

<table>
<thead>
<tr>
<th>Requirements class</th>
<th>/req/computationalviewpoint</th>
</tr>
</thead>
<tbody>
<tr>
<td>Target type</td>
<td>UML service model</td>
</tr>
<tr>
<td>Dependency</td>
<td>ISO 19103:2014 (Conceptual Schema Language)</td>
</tr>
<tr>
<td></td>
<td>ISO 19115-1:2014 (Metadata)</td>
</tr>
<tr>
<td>Requirement</td>
<td>/req/computationalviewpoint/interfaces</td>
</tr>
<tr>
<td>Requirement</td>
<td>/req/computationalviewpoint/operations</td>
</tr>
<tr>
<td>Requirement</td>
<td>/req/computationalviewpoint/behaviour</td>
</tr>
<tr>
<td>Requirement</td>
<td>/req/computationalviewpoint/pre_and_post_conditions</td>
</tr>
<tr>
<td>Requirement</td>
<td>/req/computationalviewpoint/servicechaining</td>
</tr>
<tr>
<td>Recommendation</td>
<td>/req/computationalviewpoint/servicechaining</td>
</tr>
</tbody>
</table>
ISO 19164 Computational Viewpoint

- Service Interfaces
  - Capability
  - Query
  - Transaction
  - Harvest
Example of interfaces with operations

**Mandatory Interfaces**

```plaintext
<interface>
BasicSensorPlanner
<abstract>
+ describeResultAccess(DescribeResultAccess) : DescribeResultAccessResponse
+ describeTasking(DescribeTasking) : DescribeTaskingResponse
+ getCapabilities(GetCapabilities) : Capabilities
+ getStatus(GetStatus) : GetStatusResponse
+ getTask(GetTask) : GetTaskResponse
+ submit(Submit) : SubmitResponse
</interface>
```

```plaintext
<interface>
SensorProvider
<abstract>
+ describeSensor(DescribeSensor) : DescribeSensorResponse
</interface>
```

**Optional Interfaces**

```plaintext
<interface>
ReservationManager
<abstract>
+ confirm(Confirm) : ConfirmResponse
  + reserve(Reserve) : ReserveResponse
</interface>
```

```plaintext
<interface>
FeasibilityController
<abstract>
+ getFeasibility(GetFeasibility) : GetFeasibilityResponse
</interface>
```

```plaintext
<interface>
TaskUpdater
<abstract>
+ update(Update) : UpdateResponse
</interface>
```

```plaintext
<interface>
Task Canceller
<abstract>
+ cancel(Cancel) : CancelResponse
</interface>
```

```plaintext
<interface>
SensorDescriptionManager
<abstract>
+ updateSensorDescription(UpdateSensorDescription) : UpdateSensorDescriptionResponse
</interface>
```
Example of service behaviour
Information viewpoint

<table>
<thead>
<tr>
<th>Requirements class</th>
<th>/req/informationviewpoint</th>
</tr>
</thead>
<tbody>
<tr>
<td>Target type</td>
<td>UML service model</td>
</tr>
<tr>
<td>Dependency</td>
<td>ISO 19103:2014 (Conceptual Schema Language)</td>
</tr>
<tr>
<td>Requirement</td>
<td>/req/informationviewpoint/servicesmodel dependencies</td>
</tr>
<tr>
<td>Requirement</td>
<td>/req/informationviewpoint/operation input/output/exception parameters</td>
</tr>
</tbody>
</table>
ISO 19164 Information Viewpoint

- DomainInformationModel
- DomainInformationModelFragment
- AudibleEvent
- NotificationMethod
- SourceData
Example: MessageTypes
### Engineering Viewpoint

| req/engineeringviewpoint/architectural style mapping | The architectural style(s) supported for the service shall be described and reflected in refined interface and operation specifications that shows how the specified operations from the computational viewpoint will be realised in the selected architectural style(s) (i.e. RPC, OWS, REST, SOAP, etc.) . This includes a mapping for the requirements of the computational viewpoint, with interfaces, operations, behaviour, pre and post conditions and service chaining. |
ISO 1964 Engineering Viewpoint

- REST Style
  - Service Interfaces
    - Capability
      - GET Capabilities
    - Query
      - POST queryExpression (should this be PUT?)
      - GET queryExpression/Query
    - Transaction
      - POST transactionExpression (should this be PUT?)
      - GET transactionExpression/Transaction
    - Harvest
      - POST harvestExpression (should this be PUT?)
      - GET harvestExpression/Harvest
ISO 1964 Engineering Viewpoint

- **OWS (Architectural Style)**
  - **Service Interfaces**
  - **Capability**
    - **Operations**
      - GetCapabilities()
      - DescribeRecord(RecordID)
  - **Query**
    - **Operations**
      - GetRecords(RecordFilterExpression)
      - GetRecordByID(RecordID)
  - **Transaction**
    - **Operations**
      - Transaction(RecordFilterExpression)
  - **Harvest**
    - **Operations**

1  = Same as SOAP Style at this level
Technology Viewpoint

- Revealing the technical differences between the OWS profile Implementation details and the SOAP profile Implementation details.

/req/technologyviewpoint/technology mappings

The technology mappings shall be described based on mappings from the information, computational and engineering viewpoints. This includes a mapping for the elements of the information viewpoint with the operation input/output/exception parameters to their related technology representation and encoding.

Any needed further mapping from the computational viewpoint and the architectural style from the engineering viewpoint should be further mapped for interfaces, operations, behaviour, pre and post conditions and service chaining.
Example of mappings from UML information models to XML representation
GetCapabilities operation request with HTTP/URL parameters

<table>
<thead>
<tr>
<th>Name and example a</th>
<th>Optionality and use</th>
<th>Definition and format</th>
</tr>
</thead>
<tbody>
<tr>
<td>Service=WCS</td>
<td>Mandatory</td>
<td>Abbreviated service type identifier text b</td>
</tr>
<tr>
<td>Request=GetCapabilities</td>
<td>Mandatory</td>
<td>Operation name text</td>
</tr>
<tr>
<td>AcceptVersions=1.0.0.0.8.3</td>
<td>Optional</td>
<td>Prioritized sequence of one or more specification versions accepted by client, with preferred versions listed first</td>
</tr>
<tr>
<td>Sections=Contents</td>
<td>Optional</td>
<td>Comma-separated unordered list of zero or more names of sections of service metadata document to be returned in service metadata document</td>
</tr>
<tr>
<td>UpdateSequence=XXX (where XXX is character string previously provided by server)</td>
<td>Optional</td>
<td>Service metadata document version, value is “increased” whenever any change is made in complete service metadata document</td>
</tr>
<tr>
<td>AcceptFormats=text/xml</td>
<td>Optional</td>
<td>Prioritized sequence of zero or more response formats desired by client, with preferred formats listed first</td>
</tr>
<tr>
<td>AcceptLanguages=en-CA, fr-CA</td>
<td>Optional</td>
<td>List of languages desired by the client for all human readable text in the response, in order of preference. For every element, the first matching language available from the server shall be present in the response. See section 7.3.6</td>
</tr>
</tbody>
</table>

---

**a** All parameter names are listed here using mostly lower case letters. However, any parameter name capitalization shall be allowed in KVP encoding, see Subclause 11.5.2.

**b** A specific OWS specification shall define the abbreviated service type identifier to be used by all implementing services.
Support for generation of abstract test suites

- The availability of a service specification is a good foundation for semi-automatic generation of abstract test suites.

- With appropriate definitions of rules for the mappings from the logical service specification to various platform specific service specifications, it will also be possible to support semi-automatic generation of concrete test suites for various technology profiles.
## Service taxonomies

<table>
<thead>
<tr>
<th>Requirements class</th>
<th>/req/servicetaxonomies</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Target type</strong></td>
<td>Service description</td>
</tr>
<tr>
<td><strong>Dependency</strong></td>
<td>ISO 19103:2014 (Conceptual Schema Language)</td>
</tr>
<tr>
<td><strong>Requirement</strong></td>
<td>/req/servicetaxonomies/service type - architecture</td>
</tr>
<tr>
<td><strong>Requirement</strong></td>
<td>/req/servicetaxonomies/service type - lifecycle</td>
</tr>
<tr>
<td><strong>Recommendation</strong></td>
<td>/rec/ servicetaxonomies /service type - user-defined</td>
</tr>
</tbody>
</table>
6 Architectural service types

Legend
G - Geographic
IT - Information Technology

HS - Human Interaction Services
MS - Model Management Services
WS - Workflow/Task Services
SS - System Management Services
PS - Processing Services
CS - Communication Services

The approach is to define Geographic Information Services in each of the six groups, where general Information Technology services do not meet the requirements.
Geographic architecture services taxonomy

- Geographic boundary/human interaction services
- Geographic model/information management services
- Geographic workflow/task management services
- Geographic processing services
  - Geographic processing services — spatial
  - Geographic processing services — thematic
  - Geographic processing services — temporal
  - Geographic processing services — metadata
- Geographic communication services
- Geographic system management and security services
Spatial related decision support life cycle service taxonomy
Next steps

- ISO 19119 CD:2 finalised for July 2014
- Use of ISO 19119 in ISO 19164 is planned for fall 2014
- ISO 19119 should be applied also for other future service specifications
- Propose a new ISO 19119 part 2 – to specify transformation rules for often used technology profiles, i.e. OWS, REST, SOAP, …
- Synergy with other similar initiatives for service specifications: NIEM GRA-UML, SESAR ISRM/AIRM, …