# Experiences with data model conversions and use of SDF for Digital Twins

Petri Laari, Ericsson Research Finland

# Contents

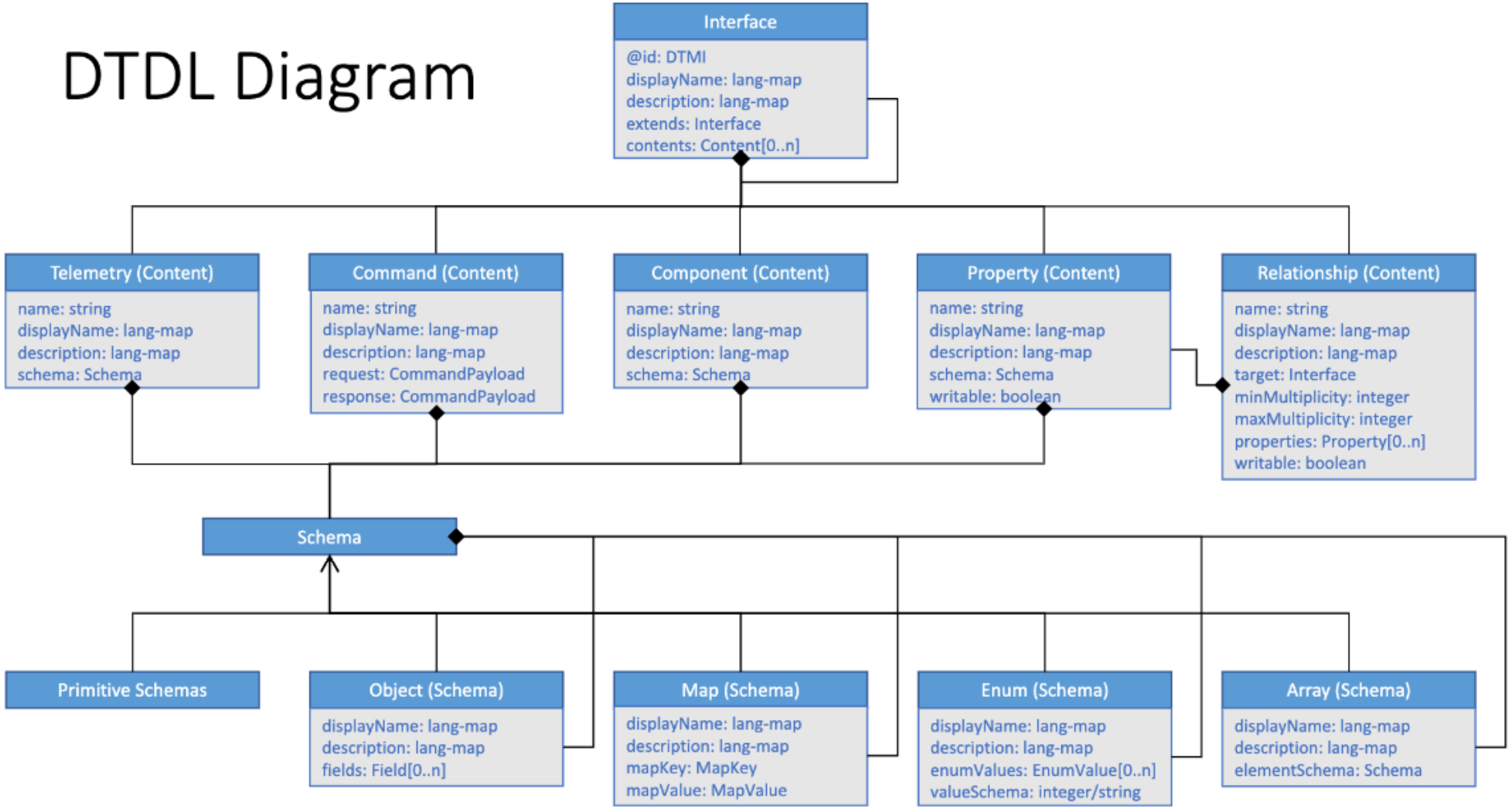
# Digital Twin Definition Language (DTDL) - intro

- By Microsoft, used with e.g. Azure Digital Twin definitions
- Open Source modeling language
  - Used to describe IoT devices, Digital Twins and systems of Digital Twins
  - Provides Relationships and Linking
    - Relationships describe this object's relations to other objects
    - JSON-LD to support linked data

From the presentation by Brian Crawford [1]:
https://github.com/t2trg/wishi/blob/master/slides/DTDL%20to%20WISHI%202020-07-30.pdf

# Digital Twin Definition Language (DTDL) – intro

(c) Brian Crawford



# Contents

# Modeling entities and matching different ecosystems

| | IPSO Smart Object | OneDM/SDF | Azure DTDL |
|---|---|---|---|
| Entity level | Object | sdfObject | Interface |
| | *Composite Object* | sdfThing | Interface *) |
| Affordances | Resource (RW) | sdfProperty | Property |
| | Resource (E) | sdfAction | Command |
| | *Implemented differently **)* | sdfEvent | Telemetry |
| | *Limited relations with Object Link* | sdfRelation (*proposed*) | Relationship |

*) An Interface can contain other Interfaces using the "Component" entity

**) Some capabilities can be implemented differently (Event/Telemetry), e.g. IPSO uses LwM2M SEND interface

# Modeling entities and matching different ecosystems

| | IPSO Smart Object | OneDM/SDF | Azure DTDL |
|---|---|---|---|
| Entity level | Object | sdfObject | Interface |
| | *Composite Object* | sdfThing | Interface *) |
| Affordances | Resource (RW) | sdfProperty | Property |
| | Resource (E) | sdfAction | Command |
| | *Implemented differently **)* | sdfEvent | Telemetry |
| | *Limited relations with Object Link* | sdfRelation (*proposed*) | Relationship |

*) An Interface can contain other Interfaces using the "Component" entity

**) Some capabilities can be implemented differently (Event/Telemetry), e.g. IPSO uses LwM2M SEND interface

# Modeling entities and matching different ecosystems

| | IPSO Smart Object | OneDM/SDF | Azure DTDL |
|---|---|---|---|
| Entity level | Object | sdfObject | Interface |
| | *Composite Object* | sdfThing | Interface *) |
| Affordances | Resource (RW) | sdfProperty | Property |
| | Resource (E) | sdfAction | Command |
| | *Implemented differently **)* | sdfEvent | Telemetry |
| | *Limited relations with Object Link* | sdfRelation (*proposed*) | Relationship |

*) An Interface can contain other Interfaces using the "Component" entity

**) Some capabilities can be implemented differently (Event/Telemetry), e.g. IPSO uses LwM2M SEND interface

# Temperature sensor: IPSO <-> SDF

## LwM2M / IPSO

```
<Resources>
    <Item ID="1">
        <Name>Sensor Value</Name>
        <Operations>R</Operations>
        <MultipleInstances>Single</MultipleInstances>
        <Mandatory>Mandatory</Mandatory>
        <Type>Float</Type>
        <RangeEnumeration/>
        <Units/>
        <Description>..measured value..</Description>
    </Item>
    <Item ID="5">
        <Name>Reset Min and Max Measured Values</Name>
        <Operations>E</Operations>
        <MultipleInstances>Single</MultipleInstances>
        <Mandatory>Optional</Mandatory>
        <RangeEnumeration></RangeEnumeration>
        <Units></Units>
        <Description>Reset the Min and Max...</Description>
    </Item>
</Resources>
```

## SDF

```
"sdfProperty": {
  "Sensor_Value": {
    "label": "Sensor Value",
    "description": ”..measured value..”
    "writable": false,
    "type": "number"
  }
}
"sdfAction": {
  "Reset_Min_and_Max_Measured_Values": {
    "label": "Reset Min and Max Measured Values",
    "description": "Reset the Min and Max..."
  }
}
```

- Most affordances have corresponding definitions in other modeling languages

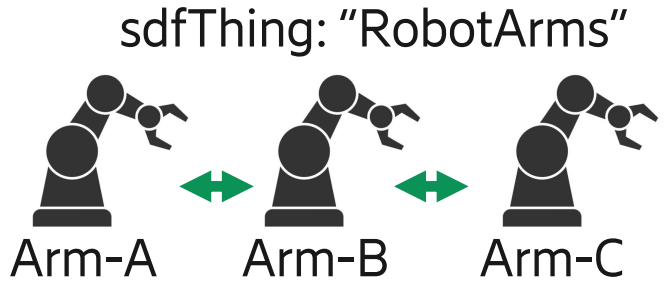
# **Challenges in conversions**: incompatible or missing affordances

- **Incompatible** affordances
  - Affordances may have been implemented differently, e.g. not modeled as explicit concepts in the data model
  - Converting from X to SDF may be trivial, but from SDF to Y can be challenging / impossible
  - Example:
    - DTDL Telemetry -> SDF sdfEvent -> IPSO
      - IPSO implements similar functionality with LwM2M SEND
- **Missing/incomplete** affordances
  - It may be some ecosystem specific definition that is not relevant elsewhere
  - How to handle these?
    - May require extension to SDF
    - Opportunity to influence the Data Model definitions in other ecosystems
      - We are actively participating in many SDOs, thus this is a way to influence the future direction

# Challenges in conversions example: Relationship

- **DTDL** defines arbitrary relations between entities with *Relationship*
  - This is an important feature in Digital Twins
  - E.g. IPSO SOs have a limited support using *Object Links*
- The first version of SDF didn't have a similar, flexible relation description
  - We are proposing *sdfRelation* extension to the next version of SDF
  - Tool has been implemented to support DTDL – SDF conversions using relations
  - Internet-Draft is being written
- Generally, relationship can be e.g.
  - Physical: inside, next-to
  - Functional: controls something
  - Semantic: what a certain affordance means for the other

# Current sdfRelation design




Ontology, where the relations are defined, e.g. Saref (ETSI)

sdfThing: consists of three robot arms (A, B, C)

the relationship type is defined in the corresponding namespace

target refers to the target object of this relation

```
...
 {
  "namespace": {
    "terms" : "https://example.com/relation-terms",
    "robots" : "https://robo-org.example.com/models"
  },
  "defaultNamespace": "robots",
  "sdfThing": {
    "RobotArms": {
      "sdfObject": {
        "Arm-A": {
          "sdfRelation": {
            "next-to": {
              "type": "terms:next-to",
              "target": "#/sdfThing/RobotArms/sdfObject/Arm-B"
            }
          }
        },
.....
```

# Contents

- Short intro to DTDL
- Data Model conversions and challenges
- Tool development
- Key takeaways and References

# Tools being developed

- Public releases: ER Github [2]
  - IPSO <-> SDF: conversions in both directions
  - DTDL <-> SDF: conversions in both directions
  - sdfThing creator: combine multiple unique sdfObjects into a sdfThing
  - Model validator (SDF Linter)
- Internally experimenting
  - OPC UA <-> SDF
  - NGSI-LD <-> SDF
  - OpenAPI & GraphQL API Converters
- Public web-based place to try the tools [3]: http://wishi.nomadiclab.com/sdf-converter/
  - IPSO & DTDL tools by Ericsson Research
  - WoT & YANG tools by University of Bremen

# Contents

# Key takeaways

- SDF can be used with Digital Twin languages such as DTDL
  - We can provide a system description on Digital Twin focused ecosystem
- We have tools to convert data models between different ecosystems and SDF
  - The SDF model can be converted into Digital Twin / Device description presented with DTDL
- We have the opportunity to suggest new and missing features to SDOs
  - This may provide better interoperability in the future for Digital Twin environments

# References

[1] DTDL Presentation by Brian Crawford, https://github.com/t2trg/wishi/blob/master/slides/DTDL%20to%20WISHI%202020-07-30.pdf

[2] http://wishi.nomadiclab.com/sdf-converter

[3] Ericsson Research, Github repository, published tools https://github.com/EricssonResearch/ipso-odm