Modelling Geometric Objects with ISO 15926

Geiza Hamazaki    Bruno Lopes

Departamento de Informática Aplicada
Universidade Federal do Estado do Rio de Janeiro

Instituto Tecgraf de Desenvolvimento de Software Técnico-Científico
Pontifícia Universidade Católica do Rio de Janeiro

Departamento de Informática
Pontifícia Universidade Católica do Rio de Janeiro

September 2014
Actual problems Oil and Gas Industry

- Scenarios are replaced in 3-4 years
- Projects lifecycle last more then 10 years

Owner, Manufactures, Engineering Contractors and Operators may use different Platforms
- Different project tools
- High-cost migration

NIST
US$15.8 billions for interoperability costs
Actual problems Oil and Gas Industry

- Scenarios are replaced in 3-4 years
- Projects lifecycle last more then 10 years

Owner, Manufactures, Engineering Contractors and Operators may use different Platforms
- Different project tools
- High-cost migration

NIST
US$15.8 billions for interoperability costs
Motivation

Actual problems Oil and Gas Industry

- Scenarios are replaced in 3-4 years
- Projects lifecycle last more then 10 years

Owner, Manufactures, Engineering Contractors and Operators may use different Platforms
- Different project tools
- High-cost migration

NIST
US$15.8 billions for interoperability costs
ISO 15926 - Industrial automation systems and integration—Integration of life-cycle data for process plants including oil and gas production facilities

Why use ISO 15926?

- Life-cycle description
- Flexibility and Extensibility
- Information context
- Validation
Motivation

ISO 15926

Data over Project Life Cycle (modified from Pawsey, 2012)
Nowadays...

**Part 1:** Overview and fundamental principles

**Part 2:** Data Model, it that represent information common to users and process plants. In Natural Language: Grammar

**Part 3:** Geometry and topology

**Part 4:** Reference Data Library (RDL). In Natural Language: The dictionary

**Part 7:** Templates. In Natural Language: It is equivalent to a phrase book
Nowadays...

Part 1: Overview and fundamental principles
Part 2: DataModel, it that represent information common to users and process plants. In Natural Language: Grammar
Part 3: Geometry and topology
Part 4: Reference Data Library (RDL). In Natural Language: The dictionary
Part 7: Templates. In Natural Language: It is equivalent to a phrase book
Nowadays...

Part 1: Overview and fundamental principles

Part 2: DataModel, it that represent information common to users and process plants. In Natural Language: Grammar

Part 3: Geometry and topology

Part 4: Reference Data Library (RDL). In Natural Language: The dictionary

Part 7: Templates. In Natural Language: It is equivalent to a phrase book
Nowadays...

**Part 1:** Overview and fundamental principles

**Part 2:** DataModel, it that represent information common to users and process plants. In Natural Language: Grammar

**Part 3:** Geometry and topology

**Part 4:** Reference Data Library (RDL). In Natural Language: The dictionary

**Part 7:** Templates. In Natural Language: It is equivalent to a phrase book
Nowadays...

**Part 1:** Overview and fundamental principles

**Part 2:** DataModel, it that represent information common to users and process plants. In Natural Language: Grammar

**Part 3:** Geometry and topology

**Part 4:** Reference Data Library (RDL). In Natural Language: The dictionary

**Part 7:** Templates. In Natural Language: It is equivalent to a phrase book
Nowadays...

**Part 8:** OWL implementation of the Templates. In Natural Language: It is like paper in a book, or a computer file

**Part 9:** Implementation standards, with the focus on Façades, standard web servers, web services, and security

**Part 10:** Formally named Implementation Methods for the Integration of Distributed Systems: Abstract Test Methods (draft)

**Part 11:** Industrial Usage Guidelines
Nowadays...

**Part 8:** OWL implementation of the Templates. In Natural Language: It is like paper in a book, or a computer file

**Part 9:** Implementation standards, with the focus on Façades, standard web servers, web services, and security

**Part 10:** Formally named Implementation Methods for the Integration of Distributed Systems: Abstract Test Methods (draft)

**Part 11:** Industrial Usage Guidelines
Nowadays...

**Part 8:** OWL implementation of the Templates. In Natural Language: It is like paper in a book, or a computer file

**Part 9:** Implementation standards, with the focus on Façades, standard web servers, web services, and security

**Part 10:** Formally named Implementation Methods for the Integration of Distributed Systems: Abstract Test Methods (draft)

**Part 11:** Industrial Usage Guidelines
Nowadays...

**Part 8:** OWL implementation of the Templates. In Natural Language: It is like paper in a book, or a computer file

**Part 9:** Implementation standards, with the focus on Façades, standard web servers, web services, and security

**Part 10:** Formally named Implementation Methods for the Integration of Distributed Systems: Abstract Test Methods (draft)

**Part 11:** Industrial Usage Guidelines
Defining objects

Template methodology

Complex objects must be defined as templates.

Part 3

A huge library of basic geometric terms.
Defining objects

Template methodology

Complex objects must be defined as templates.

Part 3

A huge library of basic geometric terms.
Identifying Part 3 elements

**Circle definition in Part 3**

An object is a **circle** if and only if: 1-it is **curve**; 2-it lies in a **plane**; 3- there is a centre point that is equi-distant from each point in the curve.

**NOTE 2** A **circle** has the geometric properties: radius; center and plane. These properties can be given for a **circle** by a **axial_reference_placement** and a **radius**. A **circle** has two alternative values for the **axial_reference_placement** corresponding to opposite directions for the normal.
Identifying classes

Which templates are necessary?
Identifying classes

Which templates are necessary?
### Modeling process steps

Definition of the signature, that describes the elements that compound the relationship;

<table>
<thead>
<tr>
<th><strong>Order</strong></th>
<th><strong>Rule</strong></th>
<th><strong>Type</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>hasProperty</td>
<td>Property</td>
</tr>
<tr>
<td>2</td>
<td>valPropertyValue</td>
<td>ExpressReal</td>
</tr>
<tr>
<td>3</td>
<td>hasScale</td>
<td>Scale</td>
</tr>
</tbody>
</table>
Modelling process steps

Definition of Axioms/Sentences in First Order Logic (FOL), that describes the semantics through the relations between the elements presented in the signature.

Axiom

\[
\text{RealMagnitudeOfProperty}(x_1, x_2, x_3) \iff \\
\text{property}(x_1) \land \text{ExpressReal}(x_2) \land \text{scale}(x_3) \land \\
\exists u \ (\text{MagnitudeOfProperty}(x_1, u, x_3) \land \\
\text{IdentificationByNumber}(x_2, u))
\]
RealMagnitudeOfProperty

The template **RealMagnitudeOfProperty** is used to connect a concept classified as a **property** with a numeric value and a **scale**.

**Signature**

<table>
<thead>
<tr>
<th>Order</th>
<th>Rule</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>hasProperty</td>
<td>Property</td>
</tr>
<tr>
<td>2</td>
<td>valPropertyValue</td>
<td>ExpressReal</td>
</tr>
<tr>
<td>3</td>
<td>hasScale</td>
<td>Scale</td>
</tr>
</tbody>
</table>

**Axiom**

\[
\text{RealMagnitudeOfProperty}(x_1, x_2, x_3) \leftrightarrow \\
\text{property}(x_1) \land \text{ExpressReal}(x_2) \land \text{scale}(x_3) \land \\
\exists u \ (\text{MagnitudeOfProperty}(x_1, u, x_3) \land \\
\text{IdentificationByNumber}(x_2, u))
\]
RealMagnitudeOfProperty

The template **RealMagnitudeOfProperty** is used to connect a concept classified as a *property* with a numeric value and a *scale*.

**Signature**

<table>
<thead>
<tr>
<th>Order</th>
<th>Rule</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>hasProperty</td>
<td>Property</td>
</tr>
<tr>
<td>2</td>
<td>valPropertyValue</td>
<td>ExpressReal</td>
</tr>
<tr>
<td>3</td>
<td>hasScale</td>
<td>Scale</td>
</tr>
</tbody>
</table>

**Axiom**

\[
\text{RealMagnitudeOfProperty}(x_1, x_2, x_3) \iff \\
\text{property}(x_1) \land \text{ExpressReal}(x_2) \land \text{scale}(x_3) \land \\
\exists u \ (\text{MagnitudeOfProperty}(x_1, u, x_3) \land \\
\text{IdentificationByNumber}(x_2, u))
\]
RealMagnitudeOfProperty

The template RealMagnitudeOfProperty is used to connect a concept classified as a property with a numeric value and a scale.

Signature

<table>
<thead>
<tr>
<th>Order</th>
<th>Rule</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>hasProperty</td>
<td>Property</td>
</tr>
<tr>
<td>2</td>
<td>valPropertyValue</td>
<td>ExpressReal</td>
</tr>
<tr>
<td>3</td>
<td>hasScale</td>
<td>Scale</td>
</tr>
</tbody>
</table>

Axiom

RealMagnitudeOfProperty(x₁, x₂, x₃) ↔

property(x₁) ∧ ExpressReal(x₂) ∧ scale(x₃) ∧

∃u (MagnitudeOfProperty(x₁, u, x₃) ∧ IdentificationByNumber(x₂, u))
Modelling a circle

ISO 15926

Modelling Geometry in ISO 15926

September 2014
GeometryRadiusTemplate

Property Range
- l
- z
- w

Scale
- j

GeometryMetric SpaceLength
- k

GeometryObjectWithRadius
- x

RealMagnitude OfProperty Range
- y

LowerUpper MagnitudeOf PropertyRange
- 1000
- 1000

mycircle
- 3,0
<table>
<thead>
<tr>
<th>Order</th>
<th>Rule</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>hasPossessor</td>
<td>ObjectWithRadius</td>
</tr>
<tr>
<td>2</td>
<td>hasRadius</td>
<td>RealNumber</td>
</tr>
<tr>
<td>3</td>
<td>hasLowerBound</td>
<td>RealNumber</td>
</tr>
<tr>
<td>4</td>
<td>hasUpperBound</td>
<td>RealNumber</td>
</tr>
</tbody>
</table>
RadiusTemplate\( (x, y, z, w) \leftrightarrow \)
\[
\text{ObjectWithRadius}(x) \land \text{RealNumber}(y) \land \text{RealNumber}(z) \land \text{RealNumber}(w) \land \exists m (\text{radius}(m) \land \text{hasEnd1}(m, x_1) \\
\land \text{hasEnd2}(m, k)) \land \\
\exists k (\text{metric_space_length}(k) \land \exists j (\text{Scale}(j) \land \exists \ell (\text{PropertyRange}(\ell) \land \\
\text{LowerUpperMagnitudeOfPropertyRange}(\ell, j, z, w) \land \\
\text{RealMagnitudeOfProperty}(k, y, j))) \land \\
\exists p (\text{MappingTriple}(m, x, k) \land \text{radius}(p)))
\]
AxialReferencePlacementTemplate\((q, p_x, p_y, p_z, d_x, d_y, d_z) \leftrightarrow \) 
ObjectWithAxialReferencePlacement\((q) \land \exists k (\) 
  \text{axis1\_placement}(k) \land 
  \text{ReferencePointTemplate}(k, p_x, p_y, p_z) \land 
  \text{ReferenceDirectionTemplate}(k, d_x, d_y, d_z) \land 
  \exists p (\text{MappingTriple}(p, q, k) \land 
  \text{axial\_reference\_placement}(p))) \)
GeometryAxialReference PlacementTemplate
Further work

- Implement tools:
  - for domain experts
  - for users
- Develop ISO 15926 research subjects
Tools

**Template Expander** expands FOL definitions to basic terms

**iRing Tools** interoperate data in a ISO 15926-like approach

**.15926** an environment to build and manipulate ISO 15926 compliant data

**15926:8** OWL visualization

**FOL2OWL** translates FOL template axioms to OWL
Modelling Geometry in ISO 15926

September 2014 22 / 22