A practical approach to ontology-enabled control systems for astronomical instrumentation.

Wim Pessemier
2013/10/08 – ICALEPCS 2013, San Francisco
Introduction

- MAIA: Mercator Advanced Imager for Asteroseismology
- Three-channel astronomical imager
Introduction

- PLC for controlling the instrument
- OPC UA for remote operation
Problem

- Interface to the rest of the control system: object-oriented

```
READ(MAIA.cryoU.ccdTempSensor.value)
```
Problem

- Interface to the rest of the control system: object-oriented

Diagram:

```
MAIA [Instrument]  
│                        │
│                        │
crioU [Cryostat]       
  │                        │
  │                        │
ccdTempSensor [Pt100]   
  │                        │
  │                        │
value = 160K
```

- Two changes:
  - Also model the detector
  - Change the sensor name

- The system has not changed, but the model has (twice!)

- Problem of expressivity:
  - OO model cannot express the meaning of the elements accurately (e.g., uses the name of the attributes)

Code:

```
READ(MAIA.cryoU.ccdTempSensor.value)
```
Problem

- Interface to the rest of the control system: object-oriented

- Two changes:
  - Also model the detector
  - Change the sensor name

- The system has not changed, but the model has (twice!)

- Problem of expressivity:
  - OO model cannot express the meaning of the elements accurately (e.g. uses the name of the attributes)

```plaintext
READ(MAIA.cryoU.ccdTempsensor.value)
```
Problem

- Interface to the rest of the control system: object-oriented

- Two changes:
  - Also model the detector
  - Change the sensor name

- The system has not changed, but the model has (twice!)

- Problem of expressivity:
  - OO model cannot express the meaning of the elements accurately (e.g., uses the name of the attributes)

```
READ(MAIA.cryoU.ccdTempSensor.value)
```
Problem

- Interface to the rest of the control system: object-oriented
- Two changes:
  - Also model the detector
  - Change the sensor name
- The system has not changed, but the model has (twice!)
- Problem of expressivity:
  - OO model cannot express the meaning of the elements accurately (e.g., uses the name of the attributes)

\[
\text{READ} (\text{MAIA.cryoU.ccdTempSensor.value})
\]
Problem

- Interface to the rest of the control system: object-oriented

- Two changes:
  - Also model the detector
  - Change the sensor name

- The system has not changed, but the model has (twice!)

- Problem of expressivity:
  - OO model cannot express the meaning of the elements accurately (e.g., uses the name of the attributes)

```
READ(MAIA.cryoU.ccdTempSensor.value)
```
Problem

- Interface to the rest of the control system: object-oriented

- Two changes:
  - Also model the detector
  - Change the sensor name

- The system has not changed, but the model has (twice!)

- Problem of expressivity:
  - OO model cannot express the meaning of the elements accurately (e.g. uses the name of the attributes)

```
READ(MAIA.cryoU.ccdTempsensor.value)
```
Problem

• Interface to the rest of the control system: object-oriented

- Two changes:
  ◦ Also model the detector
  ◦ Change the sensor name

- The system has not changed, but the model has (twice!)

- Problem of expressivity:
  ◦ OO model cannot express the meaning of the elements accurately (e.g., uses the name of the attributes)

```
READ(MAIA.cryoU.ccdTempsensor.value)
```
Problem

- Interface to the rest of the control system: object-oriented

Two changes:
- Also model the detector
- Change the sensor name

The system has not changed, but the model has (twice!)

Problem of expressivity:
- OO model cannot express the meaning of the elements accurately (e.g. uses the name of the attributes)
Ontologies
Ontologies

• Formal representation of knowledge
  ◦ … as a set of concepts within a domain
  ◦ … and the relationships between pairs of concepts

• Suppose we want to create an ontology about electronics:
  ◦ Namespace
    • **URI:**  [http://www.icalepcs2013.org/ontologies/electronics](http://www.icalepcs2013.org/ontologies/electronics)
    • **Prefix:** elec
  ◦ Concepts
    • **Classes:** Sensor, Pt100, Detector, Power, PowerSupply, …
    • **Instances:** THREE_PHASE_POWER
    • **Relations:** senses, isSensedBy, powers, isPoweredBy, …
  ◦ Facts
    • Pt100 is a subclass of Sensor
    • THREE_PHASE_POWER is an instance of Power
    • senses is a relation with Sensor as its domain
    • Any Sensor senses at least one Thing
Ontologies

• Formal representation of knowledge
  ◦ … as a set of **concepts within a domain**
  ◦ … and the **relationships** between pairs of concepts

• Suppose we want to create an ontology about electronics:
  ◦ Namespace
    • **URI:** [http://www.icalepcs2013.org/ontologies/electronics](http://www.icalepcs2013.org/ontologies/electronics)
    • **Prefix:** elec
  ◦ Concepts
    • **Classes:** Sensor, Pt100, Detector, Power, PowerSupply, …
    • **Instances:** THREE_PHASE_POWER
    • **Relations:** senses, isSensedBy, powers, isPoweredBy, …
  ◦ Facts
    • Pt100 is a subclass of Sensor
    • THREE_PHASE_POWER is an instance of Power
    • senses is a relation with Sensor as its domain
    • Any Sensor senses at least one Thing
Introducing MAIA revisited

<table>
<thead>
<tr>
<th>Prefix</th>
<th>URI</th>
</tr>
</thead>
<tbody>
<tr>
<td>phy</td>
<td><a href="http://www.icalepcs2013.org/ontologies/physics">http://www.icalepcs2013.org/ontologies/physics</a></td>
</tr>
<tr>
<td>astro</td>
<td><a href="http://www.icalepcs2013.org/ontologies/astronomy">http://www.icalepcs2013.org/ontologies/astronomy</a></td>
</tr>
<tr>
<td>mech</td>
<td><a href="http://www.icalepcs2013.org/ontologies/mechanics">http://www.icalepcs2013.org/ontologies/mechanics</a></td>
</tr>
<tr>
<td>elec</td>
<td><a href="http://www.icalepcs2013.org/ontologies/electronics">http://www.icalepcs2013.org/ontologies/electronics</a></td>
</tr>
<tr>
<td>sys</td>
<td><a href="http://www.icalepcs2013.org/ontologies/systems">http://www.icalepcs2013.org/ontologies/systems</a></td>
</tr>
<tr>
<td>maia</td>
<td><a href="http://www.icalepcs2013.org/ontologies/maia">http://www.icalepcs2013.org/ontologies/maia</a></td>
</tr>
</tbody>
</table>

**Semantic modeling**

**Prototype implementation**

**Conclusions**
MAIA revisited

<table>
<thead>
<tr>
<th>Prefix</th>
<th>URI</th>
</tr>
</thead>
<tbody>
<tr>
<td>phy</td>
<td><a href="http://www.icalpcs2013.org/ontologies/physics">http://www.icalpcs2013.org/ontologies/physics</a></td>
</tr>
<tr>
<td>astro</td>
<td><a href="http://www.icalpcs2013.org/ontologies/astronomy">http://www.icalpcs2013.org/ontologies/astronomy</a></td>
</tr>
<tr>
<td>mech</td>
<td><a href="http://www.icalpcs2013.org/ontologies/mechanics">http://www.icalpcs2013.org/ontologies/mechanics</a></td>
</tr>
<tr>
<td>elec</td>
<td><a href="http://www.icalpcs2013.org/ontologies/electronics">http://www.icalpcs2013.org/ontologies/electronics</a></td>
</tr>
<tr>
<td>sys</td>
<td><a href="http://www.icalpcs2013.org/ontologies/systems">http://www.icalpcs2013.org/ontologies/systems</a></td>
</tr>
<tr>
<td>maia</td>
<td><a href="http://www.icalpcs2013.org/ontologies/maia">http://www.icalpcs2013.org/ontologies/maia</a></td>
</tr>
</tbody>
</table>

General “engineering” ontologies
- classes
- relations
- some instances

Introduction
Problem
Semantic modeling
Prototype implementation
Conclusions
General “engineering” ontologies
- classes
- relations
- some instances

### Prefix URI

<table>
<thead>
<tr>
<th>Prefix</th>
<th>URI</th>
</tr>
</thead>
<tbody>
<tr>
<td>phy</td>
<td><a href="http://www.iclepcs2013.org/ontologies/physics">http://www.iclepcs2013.org/ontologies/physics</a></td>
</tr>
<tr>
<td>astro</td>
<td><a href="http://www.iclepcs2013.org/ontologies/astronomy">http://www.iclepcs2013.org/ontologies/astronomy</a></td>
</tr>
<tr>
<td>mech</td>
<td><a href="http://www.iclepcs2013.org/ontologies/mechanics">http://www.iclepcs2013.org/ontologies/mechanics</a></td>
</tr>
<tr>
<td>elec</td>
<td><a href="http://www.iclepcs2013.org/ontologies/electronics">http://www.iclepcs2013.org/ontologies/electronics</a></td>
</tr>
<tr>
<td>sys</td>
<td><a href="http://www.iclepcs2013.org/ontologies/systems">http://www.iclepcs2013.org/ontologies/systems</a></td>
</tr>
<tr>
<td>maia</td>
<td><a href="http://www.iclepcs2013.org/ontologies/maia">http://www.iclepcs2013.org/ontologies/maia</a></td>
</tr>
</tbody>
</table>

**MAIA revisited**

**Introduction**

**Problem**

**Semantic modeling**

**Prototype implementation**

**Conclusions**
MAIA revisited

### Prefix URI

<table>
<thead>
<tr>
<th>Prefix</th>
<th>URI</th>
</tr>
</thead>
<tbody>
<tr>
<td>phy</td>
<td><a href="http://www.iclepcs2013.org/ontologies/physics">http://www.iclepcs2013.org/ontologies/physics</a></td>
</tr>
<tr>
<td>astro</td>
<td><a href="http://www.iclepcs2013.org/ontologies/astronomy">http://www.iclepcs2013.org/ontologies/astronomy</a></td>
</tr>
<tr>
<td>mech</td>
<td><a href="http://www.iclepcs2013.org/ontologies/mechanics">http://www.iclepcs2013.org/ontologies/mechanics</a></td>
</tr>
<tr>
<td>elec</td>
<td><a href="http://www.iclepcs2013.org/ontologies/electronics">http://www.iclepcs2013.org/ontologies/electronics</a></td>
</tr>
<tr>
<td>sys</td>
<td><a href="http://www.iclepcs2013.org/ontologies/systems">http://www.iclepcs2013.org/ontologies/systems</a></td>
</tr>
<tr>
<td>maia</td>
<td><a href="http://www.iclepcs2013.org/ontologies/maia">http://www.iclepcs2013.org/ontologies/maia</a></td>
</tr>
</tbody>
</table>

- General “engineering” ontologies
  - classes
  - relations
  - some instances

---

**Introduction**

**Problem**

**Semantic modeling**

**Prototype implementation**

**Conclusions**
MAIA revisited

Prefix | URI
---|---
phy | http://www.icapecs2013.org/ontologies/physics
astro | http://www.icapecs2013.org/ontologies/astronomy
mech | http://www.icapecs2013.org/ontologies/mechanics
elec | http://www.icapecs2013.org/ontologies/electronics
sys | http://www.icapecs2013.org/ontologies/systems
maia | http://www.icapecs2013.org/ontologies/maia

General “engineering” ontologies
- classes
- relations
- some instances

Project specific ontology
- only instances

Introduction
Problem
Semantic modeling
Prototype implementation
Conclusions
MAIA revisited

<table>
<thead>
<tr>
<th>Prefix</th>
<th>URI</th>
</tr>
</thead>
<tbody>
<tr>
<td>phy</td>
<td><a href="http://www.icalepcs2013.org/ontologies/physics">http://www.icalepcs2013.org/ontologies/physics</a></td>
</tr>
<tr>
<td>astro</td>
<td><a href="http://www.icalepcs2013.org/ontologies/astronomy">http://www.icalepcs2013.org/ontologies/astronomy</a></td>
</tr>
<tr>
<td>mech</td>
<td><a href="http://www.icalepcs2013.org/ontologies/mechanics">http://www.icalepcs2013.org/ontologies/mechanics</a></td>
</tr>
<tr>
<td>elec</td>
<td><a href="http://www.icalepcs2013.org/ontologies/electronics">http://www.icalepcs2013.org/ontologies/electronics</a></td>
</tr>
<tr>
<td>sys</td>
<td><a href="http://www.icalepcs2013.org/ontologies/systems">http://www.icalepcs2013.org/ontologies/systems</a></td>
</tr>
<tr>
<td>maia</td>
<td><a href="http://www.icalepcs2013.org/ontologies/maia">http://www.icalepcs2013.org/ontologies/maia</a></td>
</tr>
</tbody>
</table>

General “engineering” ontologies
- classes
- relations
- some instances

Project specific ontology
- only instances

Introduction
Problem
Semantic modeling
Prototype implementation
Conclusions
MAIA revisited

**Prefix** | **URI**
---|---
phy | http://www.icapecs2013.org/ontologies/physics
astro | http://www.icapecs2013.org/ontologies/astronomy
mech | http://www.icapecs2013.org/ontologies/mechanics
elec | http://www.icapecs2013.org/ontologies/electronics
sys | http://www.icapecs2013.org/ontologies/systems
maia | http://www.icapecs2013.org/ontologies/maia

General “engineering” ontologies
- classes
- relations
- some instances

Project specific ontology
- only instances

provide context to

Introduction
Problem
Semantic modeling
Prototype implementation
Conclusions
Implementations
Implementations

- **Semantic Web standards**
  - Designed to add semantics to the huge amount of syntactic information on the WWW

- **Quick summary:**
  - **RDF** (Resource Description Framework)
    - Defines basic data model: subject – predicate – object “triples”
    - E.g. `elec:THREE_PHASE_POWER – rdf:type – elec:Power`
  - **RDF-S** (RDF-Schema)
    - Extends RDF so basic ontologies can be built
    - E.g. `elec:Pt100 – rdfs:subClassOf – elec:Sensor`
  - **OWL** (Web Ontology Language)
    - Extends RDF-S to build more advanced ontologies
    - E.g. `elec:senses – owl:inverseOf – elec:isSensedBy`
  - **SWRL** (Semantic Web Rule Language)
    - Even more expressive power
    - Not a standard
    - Need to be careful …
MAIA revisited

**Introduction**

**Problem**

**Semantic modeling**

**Prototype implementation**

**Conclusions**
MAIA revisited

**Introduction**

**Problem**

**Semantic modeling**

**Prototype implementation**

**Conclusions**
MAIA revisited

\[
\text{maia:MAIA} [\text{astro:Instrument}]
\]

\[
\text{maia:cryoU} [\text{mech:Cryostat}]
\]

\[
\text{maia:tempSensor} [\text{elec:Pt100}]
\]

\[
\text{160K}
\]
MAIA revisited

**Introduction**

**Problem**

**Semantic modeling**

**Prototype implementation**

**Conclusions**
MAIA revisited

**Introduction**

**Problem**

**Semantic modeling**

**Prototype implementation**

**Conclusions**
MAIA revisited

- \textit{maia:MAIA} \text{[\textit{astro:Instrument}]}
- \textit{maia:cryoU} \text{[\textit{mech:Cryostat}]}
- \textit{maia:tempSensor} \text{[\textit{elec:Pt100}]}
- \textit{phy:U} \text{[\textit{phy:SpectralBand}]}

\textit{maia:ccd} \text{[\textit{elec:Detector}]}

- \textit{phy:observes}
- \textit{phy:sensesTemperatureOf}
- \textit{phy:senses}

- \textit{160K}

- \textit{mech:encloses} \text{\textit{rdfs:subPropertyOf}} \text{\textit{sys:hasPart}}
MAIA revisited

MAIA [astro:Instrument]

cryoU [mech:Cryostat]

ccd [elec:Detector]

tempSensor [elec:Pt100]

phy:U [phy:SpectralBand]

- mech:encloses rdfs:subPropertyOf sys:hasPart

Introduction

Problem

Semantic modeling

Prototype implementation

Conclusions
MAIA revisited

- **maia:MAIA** [astro:Instrument]
  - sys:hasPart
    - **maia:cryoU** [mech:Cryostat]
      - mech:encloses
        - sys:hasPart
          - **maia:tempSensor** [elec:Pt100]
            - phy:senses
              - 160K

- **phy:U** [phy:SpectralBand]
  - phy:observes
    - phy:hasTemperature
      - phy:senses
        - **maia:ccd** [elec:Detector]
          - phy:sensesTemperatureOf
            - mech:encloses
              - sys:hasPart
                - rdfs:subPropertyOf
                  - sys:hasPart
MAIA revisited

- **MAIA** [astro:Instrument]
- **cryoU** [mech:Cryostat]
- **tempSensor** [elec:Pt100]
- **ccd** [elec:Detector]
- **phy:SpectralBand**

- **sys:hasPart**
- **rdfs:subPropertyOf**
- **sys:hasPart**
- **phy:observes**
- **phy:hasTemperature**
- **phy:senses**
- **phy:sensesTemperatureOf**
- **160K**

**Introduction**

**Problem**

**Semantic modeling**

**Prototype implementation**

**Conclusions**
MAIA revisited

- **maia:**MAIA [**astro:**Instrument]
  - **sys:**hasPart
    - **sys:**hasPart
      - **sys:**hasPart

- **maia:**cryoU [**mech:**Cryostat]
  - **mech:**encloses
    - **sys:**hasPart
      - **sys:**hasPart

- **maia:**tempSensor [**elec:**Pt100]

- **phy:**U [**phy:**SpectralBand]
  - **phy:**observes

- **phy:**sensesTemperatureOf

- **phy:**hasTemperature

- **phy:**senses

- 160K

- \( \text{• mech:encloses} \quad \text{rdfs:subPropertyOf} \quad \text{sys:hasPart} \)
MAIA revisited

Introduction

Problem

Semantic modeling

Prototype implementation

Conclusions

- mech:encloses
- sys:hasPart
- rdfs:subPropertyOf
- sys:hasPart
- rdf:type
- owl:TransitiveProperty

maia:MAIA [astro:Instrument]

maia:cryoU [mech:Cryostat]

maia:ccd [elec:Detector]

maia:tempSensor [elec:Pt100]

phy:U [phy:SpectralBand]

phy:senses

phy:sensesTemperatureOf

phy:hasTemperature

phy:observes

160K
MAIA revisited

\[ \text{maia:MAIA [astro:Instrument]} \]

\[ \text{maia:cryoU [mech:Cryostat]} \]

\[ \text{maia:tempSensor [elec:Pt100]} \]

\[ \text{phy:U [phy:SpectralBand]} \]

- \text{mech:encloses}
- \text{sys:hasPart}
- \text{rdfs:subPropertyOf}
- \text{sys:hasPart}
- \text{rdf:type}
- \text{owl:TransitiveProperty}

Introduction
Problem
Semantic modeling
Prototype implementation
Conclusions
MAIA revisited

Introduction
Problem
Semantic modeling
Prototype implementation
Conclusions

• mech:encloses
• sys:hasPart

rdfs:subPropertyOf
rdfs:subPropertyOf
sys:hasPart

owl:TransitiveProperty

160K

phy:hasTemperature

phy:senses

phy:senses

phy:observes

phy:U

[phy:SpectralBand]

phy:U

maia:tempSensor

[elec:Pt100]

maia:MAIA

[astro:Instrument]

maia:cryoU

[mech:Cryostat]

maia:ccd

[elec:Detector]
MAIA revisited

Introduction
Problem
Semantic modeling
Prototype implementation
Conclusions
MAIA revisited

**Introduction**

**Problem**

**Semantic modeling**

**Prototype implementation**

**Conclusions**
MAIA revisited

Introduction
Problem
Semantic modeling
Prototype implementation
Conclusions

- mech:encloses
- rdfs:subPropertyOf
- sys:hasPart
- rdf:type
- owl:TransitiveProperty
MAIA revisited

**Introduction**

**Problem**

**Semantic modeling**

**Prototype implementation**

**Conclusions**

**Diagram:**

- **maia:MAIA** [astro:Instrument]
  - sys:hasPart
  - sys:partOf
  - sys:hasPart
  - sys:partOf

- **maia:cryoU** [mech:Cryostat]
  - mech:encloses
  - sys:hasPart
  - sys:partOf
  - mech:encloses
  - sys:hasPart
  - sys:partOf

- **maia:tempSensor** [elec:Pt100]
  - mech:encloses
  - rdf:subPropertyOf
  - sys:hasPart
  - rdf:type
  - owl:TransitiveProperty

- **phy:U** [phy:SpectralBand]
  - phy:observes

- **maia:ccd** [elec:Detector]
  - phy:senses
  - phy:sensesTemperatureOf

- **160K**

- **sys:hasPart**
MAIA revisited

maia:MAIA [astro:Instrument]

maia:cryoU [mech:Cryostat]

maia:tempSensor [elec:Pt100]

sys:hasPart
sys:partOf

mech:encloses
sys:hasPart
sys:partOf

phy:U [phy:SpectralBand]

phy:observes

phy:sensesTemperatureOf

phy:senses

160K

• mech:encloses

• sys:hasPart

rdfs:subPropertyOf

sys:hasPart

rdf:type

owl:TransitiveProperty
MAIA revisited

**Introduction**

**Problem**

**Semantic modeling**

**Prototype implementation**

**Conclusions**

---

**MAIA** [astro:Instrument]

**maia:cryoU** [mech:Cryostat]

**maia:tempSensor** [elec:Pt100]

**maia:ccd** [elec:Detector]

**phy:U** [phy:SpectralBand]

- mech:encloses
- sys:hasPart
- rdf:type
- owl:inverseOf
- rdfs:subPropertyOf
- sys:hasPart
- sys:partOf
- phy:observes
- phy:senses
- phy:sensesTemperatureOf
- phy:hasTemperature

160K
MAIA revisited

Introduction

Problem

Semantic modeling

Prototype implementation

Conclusions

**maia:MAIA [astro:Instrument]**

- sys:hasPart
- sys:partOf

**maia:cryoU [mech:Cryostat]**

- mech:encloses
- sys:hasPart
- sys:partOf

**maia:tempSensor [elec:Pt100]**

- mech:encloses
- rdfs:subPropertyOf
- sys:hasPart

**maia:ccd [elec:Detector]**

- phy:sensesTemperatureOf

**phy:U [phy:SpectralBand]**

- phy:observes

phy:hasTemperature

phy:senses

160K
MAIA revisited

MAIA [astro:Instrument]

cryoU [mech:Cryostat]

ccd [elec:Detector]

tempSensor [elec:Pt100]

Problem

Semantic modeling

Prototype implementation

Conclusions
MAIA revisited

Introduction
Problem
Semantic modeling
Prototype implementation
Conclusions
MAIA revisited

Introduction

Problem

Semantic modeling

Prototype implementation

Conclusions

• mech:encloses rdfs:subPropertyOf sys:hasPart
• sys:hasPart rdf:type owl:TransitiveProperty
• sys:hasPart owl:inverseOf sys:partOf
MAIA revisited

**Introduction**

**Problem**

**Semantic modeling**

**Prototype implementation**

**Conclusions**
MAIA revisited

**Introduction**

- **Problem**
- **Semantic modeling**
- **Prototype implementation**
- **Conclusions**

**Diagram:**
- `maia:MAIA` [astro:Instrument]
  - `sys:hasPart`
  - `sys:partOf`
  - `sys:hasPart` `sys:partOf` `sys:hasPart` `sys:partOf`
  - `mech:encloses`
  - `sys:hasPart`
  - `sys:partOf`
- `maia:cryoU` [mech:Cryostat]
  - `mech:encloses`
  - `sys:hasPart`
  - `sys:partOf`
  - `mech:encloses`
  - `sys:hasPart`
  - `sys:partOf`
  - `mech:encloses`
  - `sys:hasPart`
  - `sys:partOf`
- `maia:tempSensor` [elec:Pt100]
  - `phy:senses`
  - `phy:isSensedBy`
  - `phy:senses`
  - `phy:hasTemperature`
  - `rdfs:subPropertyOf`
  - `sys:hasPart`
  - `rdf:type`
  - `owl:TransitiveProperty`
  - `sys:partOf`
  - `owl:inverseOf`
  - `sys:partOf`
  - `owl:inverseOf`
  - `sys:partOf`
  - `phy:isSensedBy`
  - `phy:Temperature`
  - `phy:hasTemperature`
  - `rdfs:range`
  - `phy:Temperature`

**SpectraBand:**
- `phy:U` [phy:SpectralBand]
  - `phy:observes`
  - `phy:hasTemperature`
  - `phy:senses`
  - `phy:isSensedBy`
MAIA revisited

Introduction
Problem
Semantic modeling
Prototype implementation
Conclusions
MAIA revisited

Introduction  
Problem  
Semantic modeling  
Prototype implementation  
Conclusions

MAIA [astro:Instrument]

maia:cryoU [mech:Cryostat]

maia:ccd [elec:Detector]

maia:tempSensor [elec:Pt100]

sys:hasPart
sys:partOf

sys:hasPart
sys:partOf

sys:hasPart
sys:partOf

mech:encloses
sys:hasPart
sys:partOf

phy:observes

phy:hasTemperature

phy:sensesTemperatureOf

phy:senses

phy:isSensedBy

phy:hasTemperature

sys:hasPart
sys:partOf

owl:TransitiveProperty

owl:inverseOf

sys:partOf

phy:isSensedBy

phy:Temperature

• mech:encloses
• sys:hasPart
• sys:hasPart
• phy:senses
• phy:hasTemperature

rdfs:subPropertyOf
rdf:type
owl:inverseOf
rdfs:range
MAIA revisited

Introduction
Problem
Semantic modeling
Prototype implementation
Conclusions

- **mech:encloses**
- **sys:hasPart**
- **sys:partOf**
- **rdfs:subPropertyOf**
- **sys:hasPart**
- **owl:inverseOf**
- **sys:partOf**
- **phy:senses**
- **owl:inverseOf**
- **phy:isSensedBy**
- **phy:hasTemperature**
- **rdfs:range**
- **phy:Temperature**
MAIA revisited

```
query = SELECT ?value
WHERE { ?det sys:partOf maia:Maia .
  ?det phy:hasTemperature ?t .
  ?t phy:hasValue ?value }
READ(query)
```
Prototype implementation

META-MODELS (RDF/XML)

MODELS (DSL)

RDF converter

import

export

Triplestore with SPARQL endpoint

SPARQL queries

Template engine

Templates

Docs (e.g. HTML)

Source code (e.g. IEC61131-3)

Confis (e.g. XML)

Run-time applications

Databases

Introduction

Problem

Semantic modeling

Prototype implementation

Conclusions
Prototype implementation

- “Engineering ontologies”
- Provide the context
- “Heavy-weight” ontologies
- Most appropriate tool: ontology editor

http://protege.stanford.edu
Prototype implementation

META-MODELS
(RDF/XML)

MODELS
(DSL)

RDF converter

import

export

Triplestore with
SPARQL endpoint

SPARQL queries

Template engine

Templates

Docs
(e.g. HTML)

Source code
(e.g. IEC61131-3)

Configs
(e.g. XML)

Run-time applications

Databases

Introduction

Problem

Semantic modeling

Prototype implementation

Conclusions
Prototype implementation

- Project specific ontologies
- Less heavy-weight (only instances)
- “Ontoscript” (internal DSL based on Coffeescript)
  - [http://github.com/WimPessemier/ontoscript](http://github.com/WimPessemier/ontoscript)
  - [http://github.com/WimPessemier/rdfconvert](http://github.com/WimPessemier/rdfconvert)

```
MODEL "http://www.icalepcs2013.org/maia" : "maia"
# imports
maia. IMPORT_METAMODEL "http://www.icalepcs2013.org/physics" : "phy"
maia. IMPORT_METAMODEL "http://www.icalepcs2013.org/electronics" : "elec"

maia. ADD phyx.CreateTemperature(phy.Kelvin) "ccdTemp" # macro call

maia. ADD elec.Pt100 "tempSensor" :
  phy.senses maia.ccdTemp
# add more knowledge...
```

Introduction
Problem
Semantic modeling
Prototype implementation
Conclusions
Prototype implementation

META-MODELS (RDF/XML) → RDF converter → Triplestore with SPARQL endpoint → Template engine → Templates

MODELS (DSL) → RDF converter → Triplestore with SPARQL endpoint → Template engine → Templates

Docs (e.g. HTML) → Template engine → Run-time applications

Source code (e.g. IEC61131-3) → Template engine → Run-time applications

Configs (e.g. XML) → Template engine → Run-time applications

Databases
Prototype implementation

- Database of RDF triples
- Off-the-shelf
- Comes with built-in reasoner and SPARQL endpoint
- E.g. Stardog (comes with Pellet reasoner)

→ http://stardog.com
Prototype implementation

**META-MODELS**
- (RDF/XML)

**MODELS**
- (DSL)

**Templates**
- (e.g. HTML)

**Source code**
- (e.g. IEC61131-3)

**Configs**
- (e.g. XML)

**Run-time applications**

**Databases**

**Triplestore with SPARQL endpoint**

**SPARQL queries**

**Template engine**

**RDF converter**

**Introduction**

**Problem**

**Semantic modeling**

**Prototype implementation**

**Conclusions**
Prototype implementation

- Off-the-shelf template engine
- E.g. Mako  
  → http://www.makotemplates.org
Prototype implementation

- Queries are performed by the template system → knowledge is used when the artifacts are generated.

```python
results = sparql.simpleQuery("""" SELECT ?svrUri ?nsIdx ?id WHERE {
    ?det    phy:hasTemperature    ?temp .
    ?temp    opcua:hasExpandedNodeId    ?nodeId .
    ?nodeId    opcua:hasServerUri    ?svrUri .
    ?nodeId    opcua:hasNamespaceIndex    ?nsIdx .
    ?nodeId    opcua:hasIdentifier    ?id .
} """")

def getUTemperatures():
    addresses = []
    for r in results:
        addresses.append(Address(NodeId(r.nsIdx, r.id), r.svrUri))
    return UAF_client.read(addresses)
```

* OPC UA Framework (UAF): http://github.com/uaf
Prototype implementation

- Queries can also be performed at run-time!
  - Semantic Web technology (http, slow)
  - OPC UA (binary, fast)

<table>
<thead>
<tr>
<th>Feature</th>
<th>Sem. web</th>
<th>OPC UA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Complex graphs</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>URI-qualified nodes and references</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Reading, writing, querying, ...</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Communication paradigm</td>
<td>Sync</td>
<td>Sync + Async</td>
</tr>
<tr>
<td>Communication protocol</td>
<td>Slow (http)</td>
<td>Fast (binary)</td>
</tr>
</tbody>
</table>

- Docs (e.g. HTML)
- Source code (e.g. IEC61131-3)
- Configs (e.g. XML)

Run-time applications
Databases
Conclusions

- Object-oriented models/interfaces are evil
  - They cannot express the rich context of multi-disciplinary distributed applications - such as control systems - accurately.

- Semantic models/interfaces are less evil
  - They can express this information much more accurately
  - Tools and languages (OWL, DSLs, OPC UA) are available!

- Prototype will be tested on MAIA soon!

→ Thanks for your attention!