Why semantics matter:

a demonstration on knowledge-based control system design
Why semantics matter?

- What are semantic models?
- Where to apply them?
- How to apply them?
- How to build them?
- How to use them?
- Conclusions

What are semantic models?
What are semantic models?

- Models that describe
  - pieces of information (data, descriptions)
  - their relations

Why semantics matter?
- What are semantic models?
- Where to apply them?
- How to apply them?
- How to build them?
- How to use them?
- Conclusions
What are semantic models?

- Models that describe
  - pieces of information (data, descriptions)
  - their relations \( \Rightarrow \) meaning (semantics)
Why semantics matter?
- What are semantic models?
- Where to apply them?
- How to apply them?
- How to build them?
- How to use them?
- Conclusions

What are semantic models?

- Models that describe
  - pieces of information (data, descriptions)
  - their relations ➔ meaning (semantics)

```
plug_X

socket_Y
```
What are semantic models?

- Models that describe
  - pieces of information (data, descriptions)
  - their relations → meaning (semantics)

Why semantics matter?
- What are semantic models?
- Where to apply them?
- How to apply them?
- How to build them?
- How to use them?
- Conclusions
What are semantic models?

- Models that describe
  - pieces of information (data, descriptions)
  - their relations → meaning (semantics)

Why semantics matter?

- What are semantic models?
- Where to apply them?
- How to apply them?
- How to build them?
- How to use them?
- Conclusions
What are semantic models?

- Models that describe
  - pieces of information (data, descriptions)
  - their relations → meaning (semantics)
What are semantic models?

- Models that describe
  - pieces of information (data, descriptions)
  - their relations → meaning (semantics)
What are semantic models?

- Models that describe
  - pieces of information (data, descriptions)
  - their relations ➔ meaning (semantics)
What are semantic models?

Models that describe

- pieces of information (data, descriptions)
- their relations \(\Rightarrow\) meaning (semantics)
What are semantic models?

- Models that describe
  - pieces of information (data, descriptions)
  - their relations → meaning (semantics)

Why semantics matter?
- What are semantic models?
- Where to apply them?
- How to apply them?
- How to build them?
- How to use them?
- Conclusions

![Semantic Model Diagram]
What are semantic models?

- Models that describe
  - pieces of information (data, descriptions)
  - their relations \(\rightarrow\) meaning (semantics)

Why semantics matter?

- What are semantic models?
- Where to apply them?
- How to apply them?
- How to build them?
- How to use them?
- Conclusions

IF {
  ?plugA isPluggedInto ?socketB .
  ?plugA hasContact ?contactA .
  ?socketB hasContact ?contactB .
  ?contactA hasNumber ?n .
  ?contactB hasNumber ?n }
THEN {
  ?contactA isConnectedTo ?contactB }

MECHANIC

isPluggedInto
What are semantic models?

- Models that describe
  - pieces of information (data, descriptions)
  - their relations → meaning (semantics)

Why semantics matter?

- What are semantic models?
- Where to apply them?
- How to apply them?
- How to build them?
- How to use them?
- Conclusions
What are semantic models?

- Models that describe
  - pieces of information (data, descriptions)
  - their relations \rightarrow \text{meaning (semantics)}

Why semantics matter?
- What are semantic models?
- Where to apply them?
- How to apply them?
- How to build them?
- How to use them?
- Conclusions
Why semantics matter?

• What are semantic models?
• Where to apply them?
• How to apply them?
• How to build them?
• How to use them?
• Conclusions

Where to apply them?
Why semantics matter?
- What are semantic models?
- Where to apply them?
- How to apply them?
- How to build them?
- How to use them?
- Conclusions

Where to apply them?
Why semantics matter?

• What are semantic models?
• Where to apply them?
• How to apply them?
• How to build them?
• How to use them?
• Conclusions
Why semantics matter?

• What are semantic models?
• Where to apply them?
• How to apply them?
• How to build them?
• How to use them?
• Conclusions

How to apply them?
How to apply them?

• Put them in a Knowledge Base and extract information!

Why semantics matter?

• What are semantic models?
• Where to apply them?
• How to apply them?
• How to build them?
• How to use them?
• Conclusions
Why semantics matter?

• What are semantic models?
• Where to apply them?
• How to apply them?
• How to build them?
• How to use them?
• Conclusions

How to build them?
Why semantics matter?

- What are semantic models?
- Where to apply them?
- How to apply them?
- How to build them?
- How to use them?
- Conclusions
• Using an existing modeling language?
  – UML, SysML, … : semantics not sufficiently formal
  – Modeling languages have no “programming” capabilities (loops, functions, if-then, …)
Why semantics matter?

• What are semantic models?
• Where to apply them?
• How to apply them?
• How to build them?
• How to use them?
• Conclusions

How to build them?

• Using an existing modeling language?
  – UML, SysML, … : semantics not sufficiently formal
  – Modeling languages have no “programming” capabilities (loops, functions, if-then, …)

• Using a Domain Specific Language (DSL)?
  – Internal DSL called Ontoscript
  – Based on coffeescript (~javascript)
  – Idea “adopted” from the Giant Magellan Telescope project [1]

Why semantics matter?
- What are semantic models?
- Where to apply them?
- How to apply them?
- How to build them?
- How to use them?
- Conclusions

Example: model of an I/O module type

How to build them?
### Why semantics matter?
- What are semantic models?
- Where to apply them?
- How to apply them?
- How to build them?
- How to use them?
- Conclusions

### Example: model of an I/O module type

```plaintext
Example: model of an I/O module type
```

### How to build them?

- What are semantic models?
- Where to apply them?
- How to apply them?
- How to build them?
- How to use them?
Why semantics matter?
- What are semantic models?
- Where to apply them?
- How to apply them?
- How to build them?
- How to use them?
- Conclusions

How to build them?

- Example: model of an I/O module type
Why semantics matter?
• What are semantic models?
• Where to apply them?
• How to apply them?
• How to build them?
• How to use them?
• Conclusions

Example: model of an I/O module instance
How to build them?

- Why semantics matter?
  - What are semantic models?
  - Where to apply them?
- How to apply them?
- How to build them?
- How to use them?
- Conclusions

Example: model of an I/O module instance

WEB3O05

29
How to build them?

- Example: model of an I/O module instance

```java
Model: http://mercator.iac.es/onto/models/mtc/cover/electronics

Models
- http://mercator.iac.es/onto/models/comfortable\nlives
- http://mercator.iac.es/onto/models/comfort/sofas\nmodels
- http://mercator.iac.es/onto/models/seasonal\nlives
- http://mercator.iac.es/onto/models/seasonal/summer
- http://mercator.iac.es/onto/models/seasonal/winter
- http://mercator.iac.es/onto/models/seasonal/spring
- http://mercator.iac.es/onto/models/seasonal/autumn
- http://mercator.iac.es/onto/models/seasonal/harvest
- http://mercator.iac.es/onto/models/seasonal/sun
- http://mercator.iac.es/onto/models/seasonal/moon
- http://mercator.iac.es/onto/models/seasonal/phoenix
- http://mercator.iac.es/onto/models/seasonal/tropical
- http://mercator.iac.es/onto/models/seasonal/atomic
- http://mercator.iac.es/onto/models/seasonal/software
- http://mercator.iac.es/onto/models/seasonal/hardware
- http://mercator.iac.es/onto/models/seasonal/converters
- http://mercator.iac.es/onto/models/seasonal/software
- http://mercator.iac.es/onto/models/seasonal/hardware
- http://mercator.iac.es/onto/models/seasonal/converters

Sources [Software]
- http://mercator.iac.es/onto/models/seasonal/software
- http://mercator.iac.es/onto/models/seasonal/hardware
- http://mercator.iac.es/onto/models/seasonal/converters

Model: http://mercator.iac.es/onto/models/mtc/cover/electronics

for slot, connector1, connector2, panel1, panel2 in ...
  for slot, connector1, connector2, panel1, panel2 in ...
    for slot, connector1, connector2, panel1, panel2 in ...
      for slot, connector1, connector2, panel1, panel2 in ...
        for slot, connector1, connector2, panel1, panel2 in ...
```

Why semantics matter?
- What are semantic models?
- Where to apply them?
- How to apply them?
- How to build them?
- How to use them?
- Conclusions
Why semantics matter?

• What are semantic models?
• Where to apply them?
• How to apply them?
• How to build them?
• How to use them?
• Conclusions

How to use them?
Why semantics matter?
- What are semantic models?
- Where to apply them?
- How to apply them?
- How to build them?
- How to use them?
- Conclusions

Electrical design

OntoManager @ Mercator Telescope

- Ontologies
- Dataset
- Problems
- Browse
- Query
- Systems
- Mechanics
- Electronics
- Software

- Cover
- M1
- M3
- Telemetry
- Timing
Electrical design

Why semantics matter?
- What are semantic models?
- Where to apply them?
- How to apply them?
- How to build them?
- How to use them?
- Conclusions
Why semantics matter?

- What are semantic models?
- Where to apply them?
- How to apply them?
- How to build them?
- How to use them?
- Conclusions

Electrical design
Why semantics matter?
- What are semantic models?
- Where to apply them?
- How to apply them?
- How to build them?
- How to use them?
- Conclusions

Electrical design

OntoManager @ Mercator Telescope

Cover

KO modules
- slot0
- slot1
- slot2
- slot3
- slot4
- slot5
- slot6
- slot7
- slot8
- slot9
- slot10
- slot11
- slot12

terminals

connections

M1
M3
Telemetry
Timing
Why semantics matter?
- What are semantic models?
- Where to apply them?
- How to apply them?
- How to build them?
- How to use them?
- Conclusions

Electrical design

OntoManager @ Mercator Telescope

Cover
- YO modules
  - slot0
  - slot1
  - slot2
  - slot3
  - slot4
  - slot5
  - slot6
  - slot7
  - slot8
  - slot9
  - slot10
  - slot11
  - slot12
  - slot13
- Terminals
  - PE
  - L
  - N
  - 24V
  - GND
- Connectors

Ontologies  Dataset  Problems  Browse  Query  Systems  Mechanics  Electronics  Software
Electrical design

Why semantics matter?
- What are semantic models?
- Where to apply them?
- How to apply them?
- How to build them?
- How to use them?
- Conclusions

OntoManager @ Mercator Telescope

Cover
- FO modules
  - slot0
  - slot1
  - slot2
  - slot3
  - slot4
  - slot5
  - slot6
  - slot7
  - slot8
  - slot9
  - slot10
  - slot11
  - slot12
  - slot13
- terminals
  - PE
  - L
  - N
  - 24V
  - GND
- connectors
  - ECAT
  - T1
  - T2
Why semantics matter?

- What are semantic models?
- Where to apply them?
- How to apply them?
- How to build them?
- How to use them?
- Conclusions
Why semantics matter?

- What are semantic models?
- Where to apply them?
- How to apply them?
- How to build them?
- How to use them?
- Conclusions
Electrical design

Why semantics matter?

• What are semantic models?
• Where to apply them?
• How to apply them?
• How to build them?
• How to use them?
• Conclusions

Connector instance T1

Connector type summary

<table>
<thead>
<tr>
<th>ID</th>
<th>D-sub 15 F</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gender</td>
<td>female</td>
</tr>
<tr>
<td>Manufacturer</td>
<td>ITT Corporation</td>
</tr>
<tr>
<td>Description</td>
<td>D-sub 15 female connector</td>
</tr>
<tr>
<td>Fits to</td>
<td>D-sub 15 M</td>
</tr>
<tr>
<td>Used in</td>
<td>Cover (9), M1 (1), M3 (2)</td>
</tr>
</tbody>
</table>

Connections

<table>
<thead>
<tr>
<th>Pin</th>
<th>Symbol</th>
<th>Description</th>
<th>Symbol</th>
<th>Description</th>
<th>Connected to</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>Pin 1</td>
<td>TC:T1:GND H</td>
<td>Top 1 GND of holding magnet</td>
<td>Cover : terminal GND</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>Pin 2</td>
<td>TC:T1:GND MOT</td>
<td>Top 1 GND of motor</td>
<td>Cover : terminal GND</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>Pin 3</td>
<td>TC:T1:MON</td>
<td>Top 1 motor monitor</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>Pin 4</td>
<td>TC:T1:DIR</td>
<td>Top 1 motor direction</td>
<td>I/O module slot1 : terminal 1</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>Pin 5</td>
<td>TC:T1:GND ENC</td>
<td>Top 1 GND of encoder</td>
<td>Cover : terminal GND</td>
</tr>
</tbody>
</table>
Why semantics matter?

• What are semantic models?
• Where to apply them?
• How to apply them?
• How to build them?
• How to use them?
• Conclusions
Electrical design

Why semantics matter?

• What are semantic models?
• Where to apply them?
• How to apply them?
• How to build them?
• How to use them?
• Conclusions

I/O Module instance slot3

Digital input terminal to read the status of the SSI encoders of all 8 cover panels

System properties

Satisfies cover_sys:panelDesign.requirements.absFeedbackStatus

Module type summary

<table>
<thead>
<tr>
<th>ID</th>
<th>EL1088</th>
</tr>
</thead>
<tbody>
<tr>
<td>Manufacturer</td>
<td>Beckhoff Automation</td>
</tr>
<tr>
<td>Description</td>
<td>8-channel digital input terminal 24V DC, negative switching</td>
</tr>
<tr>
<td>Used in</td>
<td>Cover (1), M3 (1)</td>
</tr>
</tbody>
</table>

[Diagram showing I/O module terminals and connectors]
Why semantics matter?

- What are semantic models?
- Where to apply them?
- How to apply them?
- How to build them?
- How to use them?
- Conclusions

Electrical design

- System properties
  - Satisfies cover_systemDesign.requirements.absFeedbackStatus

- Module type summary
  - ID: HJ0088
  - Manufacturer: Beckhoff Automation
  - Description: 8-channel digital input terminal 24V DC, negative switching
  - Used in: Cover (1), M3 (1)

- Conclusions
Why semantics matter?

- What are semantic models?
- Where to apply them?
- How to apply them?
- How to build them?
- How to use them?
- Conclusions
Why semantics matter?

• What are semantic models?
• Where to apply them?
• How to apply them?
• How to build them?
• How to use them?
• Conclusions
Why semantics matter?

- What are semantic models?
- Where to apply them?
- How to apply them?
- How to build them?
- How to use them?
- Conclusions
Systems design

Why semantics matter?

- What are semantic models?
- Where to apply them?
- How to apply them?
- How to build them?
- How to use them?
- Conclusions

Design panelDesign

The design of the telescope cover panels

Requirements derivation matrix

<table>
<thead>
<tr>
<th></th>
<th>panelDesign</th>
<th>concept</th>
</tr>
</thead>
<tbody>
<tr>
<td>open</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>closed</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>moveActuator</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>moveActuatorStatus</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>absFeedback</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>absFeedbackStatus</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

The status of absolute feedback shall be known

<table>
<thead>
<tr>
<th></th>
<th>aluminum</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>open</td>
<td></td>
<td></td>
</tr>
<tr>
<td>closed</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Why semantics matter?

- What are semantic models?
- Where to apply them?
- How to apply them?
- How to build them?
- How to use them?

- Conclusions

---

**Requirement absFeedbackStatus**

The status of the absolute feedback shall be known

**Properties**

<table>
<thead>
<tr>
<th>Derives</th>
<th>cover_sys:concept.requirements.monitorable</th>
</tr>
</thead>
<tbody>
<tr>
<td>Derived from</td>
<td>cover_sys:panelDesign.requirements.absFeedback</td>
</tr>
</tbody>
</table>

Satisfied by

- cover_sys:panelDesign.parts.encoder
- cover_sys:slot3

Declared by

- cover_sys:panelDesign
• Why semantics matter?
• What are semantic models?
• Where to apply them?
• How to apply them?
• How to build them?
• How to use them?
• Conclusions
Why semantics matter?
- What are semantic models?
- Where to apply them?
- How to apply them?
- How to build them?
- How to use them?
- Conclusions

Electrical design

OntoManager @ Mercator Telescope

I/O Module instance slot3

Digital input terminal to read the status of the SSI encoders of all 8 cover panels

System properties

Satisfies cover_board&panelDesign.requirements.absFeedbackStatus

Module type summary

<table>
<thead>
<tr>
<th>ID</th>
<th>EL1088</th>
</tr>
</thead>
<tbody>
<tr>
<td>Manufacturer</td>
<td>Beckhoff Automation</td>
</tr>
<tr>
<td>Description</td>
<td>8-channel digital input terminal 24V DC, negative switching</td>
</tr>
<tr>
<td>Used in</td>
<td>Cover (1), M3 (1)</td>
</tr>
</tbody>
</table>

WEB3O05 50
Why semantics matter?
- What are semantic models?
- Where to apply them?
- How to apply them?
- How to build them?
- How to use them?
- Conclusions
### Electrical design

#### Why semantics matter?

- **What are semantic models?**
- **Where to apply them?**
- **How to apply them?**
- **How to build them?**
- **How to use them?**
- **Conclusions**

#### Table of Components

<table>
<thead>
<tr>
<th>ID</th>
<th>Description</th>
<th>Used In</th>
</tr>
</thead>
<tbody>
<tr>
<td>FR1101</td>
<td>EtherCAT Coupler with IO switch</td>
<td>Cover (1), ML (1), Telemetry (1), Timing (1)</td>
</tr>
<tr>
<td>EL1088</td>
<td>8-channel digital input terminal, 24V DC, negative switching</td>
<td>Cover (1), ML (1)</td>
</tr>
<tr>
<td>EL2008</td>
<td>8-channel digital output terminal, 24V DC</td>
<td>Cover (1)</td>
</tr>
<tr>
<td>EL2024</td>
<td>4-channel digital output terminals, 24 V DC, 2 A</td>
<td>ML (2)</td>
</tr>
<tr>
<td>EL2124</td>
<td>4-channel digital output terminals, 5 V DC</td>
<td>ML (2)</td>
</tr>
<tr>
<td>EL2222</td>
<td>2-channel relay</td>
<td>Cover (5), ML (1)</td>
</tr>
<tr>
<td>EL3024</td>
<td>4-channel analog input terminals, 4...20mA, differential inputs, 12 bit</td>
<td>ML (2), Telemetry (1)</td>
</tr>
<tr>
<td>EL3102</td>
<td>2-channel analog input terminals, -10...+10 V, differential input, 16 bit</td>
<td>ML (1)</td>
</tr>
<tr>
<td>EL3164</td>
<td>4-channel analog input terminal, 0...10 V, single-ended, 16 bit</td>
<td>ML (1)</td>
</tr>
<tr>
<td>EL3202-0010</td>
<td>2-channel input terminals PT100 (RTD) for 4-wire connection, high-precision</td>
<td>Telemetry (?)</td>
</tr>
<tr>
<td>EL3351</td>
<td>1-channel resistor bridge terminal (strain gauge)</td>
<td>ML (3)</td>
</tr>
<tr>
<td>EL3581</td>
<td>Digital multimeter</td>
<td>Cover (1)</td>
</tr>
<tr>
<td>EL4008</td>
<td>8-channel analog output terminal 0...30V, 12 bit</td>
<td>Cover (1)</td>
</tr>
<tr>
<td>EL4022</td>
<td>2-channel analog output terminal 4...20 mA, 12 bit</td>
<td>ML (1)</td>
</tr>
<tr>
<td>EL5001</td>
<td>1-channel SSI encoder</td>
<td>ML (1), ML (1)</td>
</tr>
<tr>
<td>EL5002</td>
<td>2-channel SSI encoder</td>
<td>Cover (4)</td>
</tr>
<tr>
<td>EL5101</td>
<td>1-channel incremental encoder</td>
<td>ML (1), ML (1)</td>
</tr>
<tr>
<td>EL6001</td>
<td>RS-232 serial communication</td>
<td>Timing (1)</td>
</tr>
<tr>
<td>EL6688</td>
<td>IEEE 1588 external synchronisation interface</td>
<td>Timing (1)</td>
</tr>
<tr>
<td>EL6751</td>
<td>CANopen master/slave controller</td>
<td>ML (1)</td>
</tr>
<tr>
<td>EL9070</td>
<td>Shield terminal</td>
<td>ML (1)</td>
</tr>
<tr>
<td>EL9186</td>
<td>Potential distribution terminal, 8 x 24V</td>
<td>ML (1), Telemetry (1)</td>
</tr>
<tr>
<td>EL9187</td>
<td>Potential distribution terminal, 8 x 6V</td>
<td>ML (2), Telemetry (1)</td>
</tr>
<tr>
<td>EL9410</td>
<td>Power supply terminals for E-bus (with diagnostics)</td>
<td>ML (2)</td>
</tr>
<tr>
<td>EL9505</td>
<td>Power supply terminals 5 V</td>
<td>ML (1)</td>
</tr>
</tbody>
</table>
Why semantics matter?

- What are semantic models?
- Where to apply them?
- How to apply them?
- How to build them?
- How to use them?
- Conclusions

Electrical design

I/O Module instance slot3

Digital input terminal to read the status of the SSI encoders of all 8 cover panels

System properties

Satisfies cover_sys:panel Design.requirements.absFeedbackStatus

Module type summary

<table>
<thead>
<tr>
<th>ID</th>
<th>EL1088</th>
</tr>
</thead>
<tbody>
<tr>
<td>Manufacturer</td>
<td>Beckhoff Automation</td>
</tr>
<tr>
<td>Description</td>
<td>8-channel digital input terminal 24V DC, negative switching</td>
</tr>
<tr>
<td>Used in</td>
<td>Cover (1), M9 (1)</td>
</tr>
</tbody>
</table>

Input 1

Input 2

Input 3

Input 4

Power contact 24V

Signal LED1

Signal LED2

Signal LED3

Signal LED4

Signal LED5

Signal LED6

Signal LED7

Signal LED8

[Diagram of I/O Module instance slot3]
Why semantics matter?

- What are semantic models?
- Where to apply them?
- How to apply them?
- How to build them?
- How to use them?
- Conclusions

Electrical design

Connections

<table>
<thead>
<tr>
<th>Type (EL1088)</th>
<th>Instance</th>
<th>Connected to</th>
</tr>
</thead>
<tbody>
<tr>
<td>Channel 1</td>
<td>Terminal 1</td>
<td>Symbol 1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>7</td>
<td>1</td>
</tr>
<tr>
<td>8</td>
<td>8</td>
<td>1</td>
</tr>
</tbody>
</table>

Interface

<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
<th>Description</th>
<th>Linked variable</th>
</tr>
</thead>
<tbody>
<tr>
<td>input1</td>
<td>BOOL</td>
<td>Input 1</td>
<td>interface.parts.cover.parts.top.parts.pl.encoderErrorSignal</td>
</tr>
<tr>
<td>input2</td>
<td>BOOL</td>
<td>Input 2</td>
<td>interface.parts.cover.parts.top.parts.pl.encoderErrorSignal</td>
</tr>
<tr>
<td>input3</td>
<td>BOOL</td>
<td>Input 3</td>
<td>interface.parts.cover.parts.top.parts.p3.encoderErrorSignal</td>
</tr>
<tr>
<td>input4</td>
<td>BOOL</td>
<td>Input 4</td>
<td>interface.parts.cover.parts.top.parts.pl.encoderErrorSignal</td>
</tr>
<tr>
<td>input5</td>
<td>BOOL</td>
<td>Input 5</td>
<td>interface.parts.cover.parts.bottom.parts.pl.encoderErrorSignal</td>
</tr>
<tr>
<td>input6</td>
<td>BOOL</td>
<td>Input 6</td>
<td>interface.parts.cover.parts.bottom.parts.pl.encoderErrorSignal</td>
</tr>
<tr>
<td>input7</td>
<td>BOOL</td>
<td>Input 7</td>
<td>interface.parts.cover.parts.bottom.parts.p3.encoderErrorSignal</td>
</tr>
<tr>
<td>input8</td>
<td>BOOL</td>
<td>Input 8</td>
<td>interface.parts.cover.parts.bottom.parts.pl.encoderErrorSignal</td>
</tr>
<tr>
<td>WsState</td>
<td>BOOL</td>
<td>EtherCAT Working counter state</td>
<td>interface.parts.cover.parts.io.parts.slot3.wsState</td>
</tr>
<tr>
<td>InfoDataState</td>
<td>UINT</td>
<td>EtherCAT state (INIT, PREOP, OP, ...)</td>
<td>interface.parts.cover.parts.io.parts.slot3.infoData</td>
</tr>
</tbody>
</table>
Why semantics matter?

• What are semantic models?
• Where to apply them?
• How to apply them?
• How to build them?
• How to use them?
• Conclusions
Software design

Why semantics matter?

- What are semantic models?
- Where to apply them?
- How to apply them?
- How to build them?
- How to use them?
- Conclusions

FunctionBlock SM_CoverPanel

Variables

<table>
<thead>
<tr>
<th>Variable</th>
<th>Name</th>
<th>Type</th>
<th>Initial value</th>
<th>Address</th>
<th>Description</th>
<th>Qualif</th>
</tr>
</thead>
<tbody>
<tr>
<td>VAR_INPUT</td>
<td>encoderErrorSignal</td>
<td>BOOL</td>
<td></td>
<td>%I1</td>
<td>Externally read error signal</td>
<td>OPC:UA:DA=1, DPC</td>
</tr>
<tr>
<td>VAR_IN_OUT</td>
<td>initializationStatus</td>
<td>InitializationStatus</td>
<td></td>
<td></td>
<td>INITIALIZING or INITIALIZING or ...</td>
<td></td>
</tr>
<tr>
<td></td>
<td>operatorStatus</td>
<td>OperatorStatus</td>
<td></td>
<td></td>
<td>TECH or OBSERVER or ...</td>
<td></td>
</tr>
<tr>
<td></td>
<td>operatingStatus</td>
<td>OperatingStatus</td>
<td></td>
<td></td>
<td>MANUAL or AUTO or NONE</td>
<td></td>
</tr>
<tr>
<td></td>
<td>config</td>
<td>CoverPanelConfig</td>
<td></td>
<td></td>
<td>Configuration of the panel</td>
<td></td>
</tr>
<tr>
<td></td>
<td>coverConfig</td>
<td>CoverConfig</td>
<td></td>
<td></td>
<td>Configuration of the cover</td>
<td></td>
</tr>
<tr>
<td>VAR_OUTPUT</td>
<td>actualStatus</td>
<td>STRING</td>
<td></td>
<td></td>
<td>Current status description</td>
<td>OPC:UA:DA=1, DPC</td>
</tr>
<tr>
<td></td>
<td>statuses</td>
<td>CoverPanelStatus</td>
<td></td>
<td></td>
<td>Statuses of the state machine</td>
<td></td>
</tr>
<tr>
<td></td>
<td>parts</td>
<td>CoverPanelParts</td>
<td></td>
<td></td>
<td>Parts of the state machine</td>
<td></td>
</tr>
<tr>
<td></td>
<td>processes</td>
<td>CoverPanelProcesses</td>
<td></td>
<td></td>
<td>Processes of the state machine</td>
<td></td>
</tr>
</tbody>
</table>
Software design

Why semantics matter?

- What are semantic models?
- Where to apply them?
- How to apply them?
- How to build them?
- How to use them?
- Conclusions

<table>
<thead>
<tr>
<th>VAR_INPUT</th>
<th>encoderErrorSignal</th>
<th>BOOL</th>
<th>%1*</th>
<th>Externally read error signal</th>
<th>OPC:UA:DA=1, DPC</th>
</tr>
</thead>
<tbody>
<tr>
<td>VAR_IN_OUT</td>
<td>initializationStatus</td>
<td>InitializationStatus</td>
<td>INITIALIZED or INITIALIZING or ...</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>operatorStatus</td>
<td>OperatorStatus</td>
<td>TECH or OBSERVER or ...</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>operatingStatus</td>
<td>OperatingStatus</td>
<td>MANUAL or AUTO or NONE</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>config</td>
<td>CoverPanelConfig</td>
<td>Configuration of the panel</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>coverConfig</td>
<td>CoverConfig</td>
<td>Configuration of the cover</td>
<td></td>
<td></td>
</tr>
<tr>
<td>VAR_OUTPUT</td>
<td>actualStatus</td>
<td>STRING</td>
<td>Current status description</td>
<td>OPC:UA:DA=1, DPC</td>
<td></td>
</tr>
<tr>
<td></td>
<td>statuses</td>
<td>CoverPanelStatuses</td>
<td>Statuses of the state machine</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>parts</td>
<td>CoverPanelParts</td>
<td>Parts of the state machine</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>processes</td>
<td>CoverPanelProcesses</td>
<td>Processes of the state machine</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Methods
- **startOpening()**

  ```
  Comment | Start opening the panel
  Return type | RequestResults
  Interface | Variable | Name | Type | Initial value | Address | Description | Qualifiers
  Implementation | startOpening := THIS#.processes.startOpening.request();
  ```

- **startClosing()**

  ```
  Comment | Start closing the panel
  Return type | RequestResults
  Interface | Variable | Name | Type | Initial value | Address | Description | Qualifiers
  Implementation | startClosing := THIS#.processes.startClosing.request();
  ```
Why semantics matter?

- What are semantic models?
- Where to apply them?
- How to apply them?
- How to build them?
- How to use them?
- Conclusions
Software design

Why semantics matter?

• What are semantic models?
• Where to apply them?
• How to apply them?
• How to build them?
• How to use them?
• Conclusions

FunctionBlock SM_CoverPanel

 encoderErrorSignal actualStatus
 initialStatus statuses
 operatorStatus parts
 operatingStatus processes
 config
 coverConfig

 startOpening()
 startClosing()

Variables

<table>
<thead>
<tr>
<th>Variable</th>
<th>Name</th>
<th>Type</th>
<th>Initial value</th>
<th>Address</th>
<th>Description</th>
<th>Qualif</th>
</tr>
</thead>
<tbody>
<tr>
<td>VAR_INPUT</td>
<td>encoderErrorSignal</td>
<td>BOOL</td>
<td></td>
<td>W1</td>
<td>Externally read error signal</td>
<td>OPC/UA:DA=1, OPC</td>
</tr>
<tr>
<td>VAR_IN_OUT</td>
<td>initializationStatus</td>
<td>InitializationStatus</td>
<td></td>
<td>W1</td>
<td>INITIALIZED or INITIALIZING or ...</td>
<td></td>
</tr>
<tr>
<td></td>
<td>operatorStatus</td>
<td>OperatorStatus</td>
<td></td>
<td>W1</td>
<td>TECH or OBSERVER or ...</td>
<td></td>
</tr>
<tr>
<td></td>
<td>operatingStatus</td>
<td>OperatingStatus</td>
<td></td>
<td>W1</td>
<td>MANUAL or AUTO or NONE</td>
<td></td>
</tr>
<tr>
<td></td>
<td>config</td>
<td>CoverPanelConfig</td>
<td></td>
<td></td>
<td>Configuration of the panel</td>
<td></td>
</tr>
<tr>
<td></td>
<td>coverConfig</td>
<td>CoverConfig</td>
<td></td>
<td></td>
<td>Configuration of the cover</td>
<td></td>
</tr>
<tr>
<td>VAR_OUTPUT</td>
<td>actualStatus</td>
<td>STRING</td>
<td></td>
<td>W1</td>
<td>Current status description</td>
<td>OPC/UA:DA=1, OPC</td>
</tr>
</tbody>
</table>
Software design

Why semantics matter?
- What are semantic models?
- Where to apply them?
- How to apply them?
- How to build them?
- How to use them?
- Conclusions

Library mtcs_cover

PLCopen XML serialization

<table>
<thead>
<tr>
<th>File</th>
<th>/home/wimpe/work/ont/ontomgement/env/ontomangement/generated/mtcs_cover.xml</th>
</tr>
</thead>
<tbody>
<tr>
<td>Status</td>
<td>File has been read</td>
</tr>
<tr>
<td>Code generation</td>
<td>Not running</td>
</tr>
</tbody>
</table>

Generate PLCopen XML  Download PLCopen XML

PyUAF serialization

<table>
<thead>
<tr>
<th>File</th>
<th>/home/wimpe/work/ont/ontomangement/env/ontomangement/generated/pyuaf/mtcs_cover.py</th>
</tr>
</thead>
<tbody>
<tr>
<td>Status</td>
<td>File has been read</td>
</tr>
<tr>
<td>Code generation</td>
<td>Not running</td>
</tr>
</tbody>
</table>

Generate pyUAF code  Download pyUAF code

File contents:

---
1. `<xml version="1.0" encoding="utf-8">`
2. `<project xmlns="http://www.pclext.com/xml/pcs_6300">
3. `<fileHeader company="Max Planck Society - Institute of Astronomy" productName="OntoManager" productVersion="0.0.1" creationDateTime="2015-10-09T01:23:49.793320">`
4. `<contentHeader name="mtcs_cover" modificationDateTime="2015-10-09T01:23:49.793320">`
5. `<coordinateInfo>`
6. `</coordinateInfo>`
7. `</project>`
---
Software design

Why semantics matter?
- What are semantic models?
- Where to apply them?
- How to apply them?
- How to build them?
- How to use them?
- Conclusions

Library mtcs_cover

PLCopen XML serialization

<table>
<thead>
<tr>
<th>File</th>
<th>/home/wimpe/work/onts/ontmanager/env/ontmanager/generated/mtcs_cover.xml</th>
</tr>
</thead>
<tbody>
<tr>
<td>Status</td>
<td>File has been read</td>
</tr>
<tr>
<td>Code generation</td>
<td>Not running</td>
</tr>
</tbody>
</table>

Generate PLCopen XML  Download PLCopen XML

PyUAF serialization

<table>
<thead>
<tr>
<th>File</th>
<th>/home/wimpe/work/onts/ontmanager/env/ontmanager/generated/pyuaf/mtcs_cover.py</th>
</tr>
</thead>
<tbody>
<tr>
<td>Status</td>
<td>File has been read</td>
</tr>
<tr>
<td>Code generation</td>
<td>Not running</td>
</tr>
</tbody>
</table>

Generate pyUAF code  Download pyUAF code

File contents:

```
1 /**
2 * PLCopen XML file for mtcs_cover
3 */
4
5 // Create a new PLCopen XML file
6
7 <xml version="1.0" encoding="utf-8" ?
8 <project xmlns="http://www.plexopen.org/xml/plcopen" tcs:03000>  
9 <project header="Institute of Astronomy" productVersion="0.0.1" creationDate="2015-09-21 23:49:79329" >  
10 <content header name="mtcs_cover" modificationDate="2015-09-21 23:49:79329" >  
11 <coordinateInfo>  
12 </coordinateInfo>  
13 <scaling x="1" y="1" />  
```
Why semantics matter?

• What are semantic models?
• Where to apply them?
• How to apply them?
• How to build them?
• How to use them?
• Conclusions
Why semantics matter?

• What are semantic models?
• Where to apply them?
• How to apply them?
• How to build them?
• How to use them?
• Conclusions
Software design

Why semantics matter?
• What are semantic models?
• Where to apply them?
• How to apply them?
• How to build them?
• How to use them?
• Conclusions
Why semantics matter?

- What are semantic models?
- Where to apply them?
- How to apply them?
- How to build them?
- How to use them?
- Conclusions
Why semantics matter?

- What are semantic models?
- Where to apply them?
- How to apply them?
- How to build them?
- How to use them?
- Conclusions
Why semantics matter?
• What are semantic models?
• Where to apply them?
• How to apply them?
• How to build them?
• How to use them?
• Conclusions

Software design
Software design

Why semantics matter?
• What are semantic models?
• Where to apply them?
• How to apply them?
• How to build them?
• How to use them?
• Conclusions
Why semantics matter?
• What are semantic models?
• Where to apply them?
• How to apply them?
• How to build them?
• How to use them?
• Conclusions
Why semantics matter?
- What are semantic models?
- Where to apply them?
- How to apply them?
- How to build them?
- How to use them?
- Conclusions

Software design

- Generated Python code (client side)
  - Based on our OPC UA library “UAF”: http://github.com/uaf/uaf
Software design

• Generated Python code (client side)
  – Based on our OPC UA library “UAF”: http://github.com/uaf/uaf
Software design

• Generated Python code (client side)
  – Based on our OPC UA library “UAF”: http://github.com/uaf/uaf
Software design

• Generated Python code (client side)
  – Based on our OPC UA library “UAF”: http://github.com/uaf/uaf
Software design

- Generated Python code (client side)
  - Based on our OPC UA library “UAF”: [http://github.com/uaf/uaf](http://github.com/uaf/uaf)
Software design

- Generated Python code (client side)
  - Based on our OPC UA library “UAF”: [http://github.com/uaf/uaf](http://github.com/uaf/uaf)
Why semantics matter?

- What are semantic models?
- Where to apply them?
- How to apply them?
- How to build them?
- How to use them?
- Conclusions

Generated Python code (client side)

- Based on our OPC UA library “UAF”: [http://github.com/uaf/uaf](http://github.com/uaf/uaf)
Software design

- Generated Python code (client side)
  - Based on our OPC UA library “UAF”: http://github.com/uaf/uaf

Why semantics matter?
- What are semantic models?
- Where to apply them?
- How to apply them?
- How to build them?
- How to use them?
- Conclusions
Software design

• Generated Python code (client side)
  – Based on our OPC UA library “UAF”: http://github.com/uaf/uaf
Why semantics matter?

- What are semantic models?
- Where to apply them?
- How to apply them?
- How to build them?
- How to use them?
- Conclusions

Generated Python code (client side)

- Based on our OPC UA library “UAF”: [http://github.com/uaf/uaf](http://github.com/uaf/uaf)
Why semantics matter?
- What are semantic models?
- Where to apply them?
- How to apply them?
- How to build them?
- How to use them?
- Conclusions

Software design

- Generated Python code (client side)
  - Based on our OPC UA library “UAF”: [http://github.com/uaf/uaf](http://github.com/uaf/uaf)
Software design

- Generated Python code (client side)
  - Based on our OPC UA library “UAF”: http://github.com/uaf/uaf

Why semantics matter?

- What are semantic models?
- Where to apply them?
- How to apply them?
- How to build them?
- How to use them?
- Conclusions
Software design

- Generated Python code (client side)
  - Based on our OPC UA library “UAF”: http://github.com/uaf/uaf

Why semantics matter?

- What are semantic models?
- Where to apply them?
- How to apply them?
- How to build them?
- How to use them?
- Conclusions
Generated Python code (client side)

Based on our OPC UA library “UAF”: http://github.com/uaf/uaf
Software design

• Generated Python code (client side)
  – Based on our OPC UA library “UAF”: http://github.com/uaf/uaf
Software design

- Generated Python code (client side)
  - Based on our OPC UA library “UAF”: http://github.com/uaf/uaf

```python
>>> import opcua
>>> c = opcua.buildClient()
>>> print c.read( opcua.MTCS.parts.m1.parts.axialSupport.regulatorPressure.average.bar.value.ADR() )
- overallStatus : Good
- requestHandle : 1
- targets[]
  - targets[0]
    - clientConnectionId : 0
    - status : Good
    - opcUaStatusCode : 0
    - data : 1.10534973145
    - sourceTimestamp : 2015-10-13T11:33:01.825Z
    - serverTimestamp : 2015-10-13T11:33:01.825Z
    - sourcePicoseconds : 0
    - serverPicoseconds : 0
```
Software design

- Generated Python code (client side)
  - Based on our OPC UA library “UAF”: http://github.com/uaf/uaf
Why semantics matter?

- What are semantic models?
- Where to apply them?
- How to apply them?
- How to build them?
- How to use them?
- Conclusions

Currently in operation:

- 1 PLC
- 5 subsystems
- 55 I/O modules
- 159 PLC Function Block definitions (626 instances)
Why semantics matter?

- What are semantic models?
- Where to apply them?
- How to apply them?
- How to build them?
- How to use them?
- Conclusions
Why semantics matter?

• What are semantic models?
• Where to apply them?
• How to apply them?
• How to build them?
• How to use them?
• Conclusions
Why semantics matter?

- What are semantic models?
- Where to apply them?
- How to apply them?
- How to build them?
- How to use them?
- Conclusions

User Interface (HMI) running on the PLC
Why semantics matter?

- What are semantic models?
- Where to apply them?
- How to apply them?
- How to build them?
- How to use them?

Conclusions

**Results**

- User Interface (HMI) running on the PLC
Why semantics matter?
• What are semantic models?
• Where to apply them?
• How to apply them?
• How to build them?
• How to use them?
• Conclusions

User Interface (HMI) running on the PLC
Why semantics matter?

• What are semantic models?
• Where to apply them?
• How to apply them?
• How to build them?
• How to use them?
• Conclusions
Why semantics matter?

- What are semantic models?
- Where to apply them?
- How to apply them?
- How to build them?
- How to use them?
- Conclusions
So, why semantics matter?

1. Because every piece of information is just one query “away”

   ➔ organize, integrate, browse, find (query) information

Why semantics matter?

• What are semantic models?
• Where to apply them?
• How to apply them?
• How to build them?
• How to use them?
• Conclusions
So, why semantics matter?

1. Because every piece of information is just one query “away”

   ➔ organize, integrate, browse, find (query) information

2. Because well defined semantics allow model verification

   ➔ verify information

Why semantics matter?

- What are semantic models?
- Where to apply them?
- How to apply them?
- How to build them?
- How to use them?
- Conclusions
So, why semantics matter?

1. Because every piece of information is just one query “away”
   - organize, integrate, browse, find (query) information

2. Because well defined semantics allow model verification
   - verify information

3. Because they’re a key enabling technology for future “smart” systems
   - share information
Thanks!

Any questions?

wim.pessemier@ster.kuleuven.be