### a demonstration on knowledge-based control system design

Wim Pessemier ICALEPCS 2015 Melbourne

- 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?

- Where to apply them?
- How to apply them?
- How to build them?
- How to use them?
- Conclusions

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

• What are semantic models?

\_

- Where to apply them?
- How to apply them?
- How to build them?
- How to use them?
- Conclusions

WEB3005

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

• 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)



socket\_Y

• What are semantic models?

- Where to apply them?
- How to apply them?
- How to build them?
- How to use them?
- Conclusions

WEB3005

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

plug_X	socket_Y
hasType	hasType
Plug	Socket

GENERIC	
hasType	

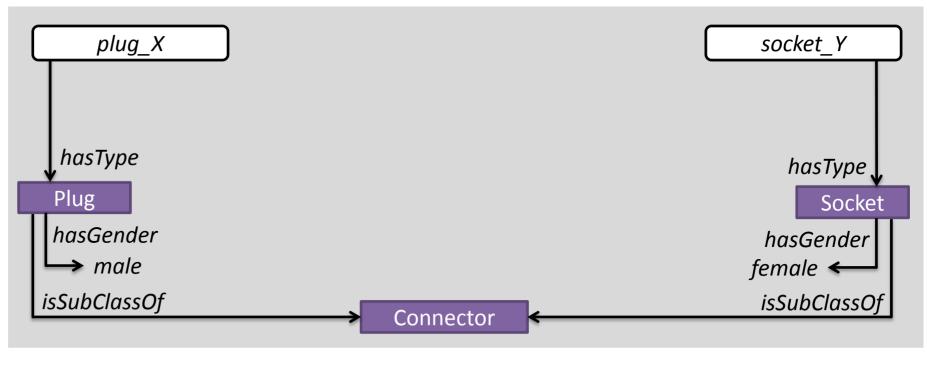
ELECTRIC	
Plug	
Socket	

• What are semantic models?

- Where to apply them?
- How to apply them?
- How to build them?
- How to use them?
- Conclusions

WEB3005

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





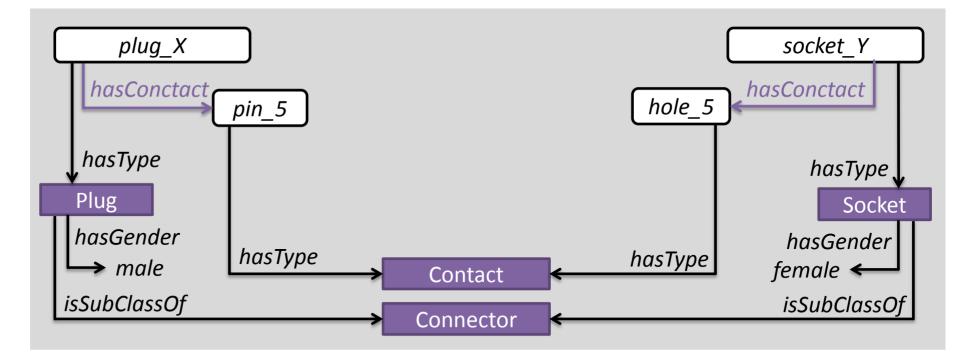
• What are semantic models?

- Where to apply them?
- How to apply them?
- How to build them?
- How to use them?
- Conclusions

WEB3005

8

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



GENERIC		_	ELECTRIC	
hasType hasGender	isSubClassOf		Plug Socket	Connector hasContact

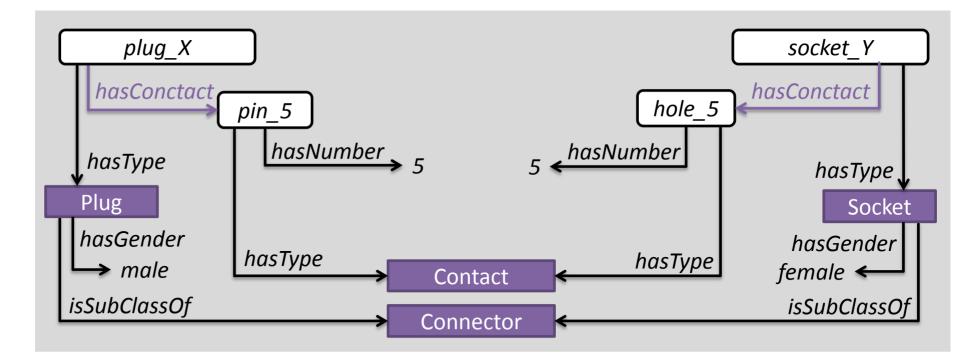
• What are semantic models?

- Where to apply them?
- How to apply them?
- How to build them?
- How to use them?
- Conclusions

WEB3005

### What are semantic models?

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



Contact

GENERIC		_	ELECTRIC	
hasType	isSubClassOf		Plug	Connector
hasGender	hasNumber		Socket	hasContact

• What are semantic models?

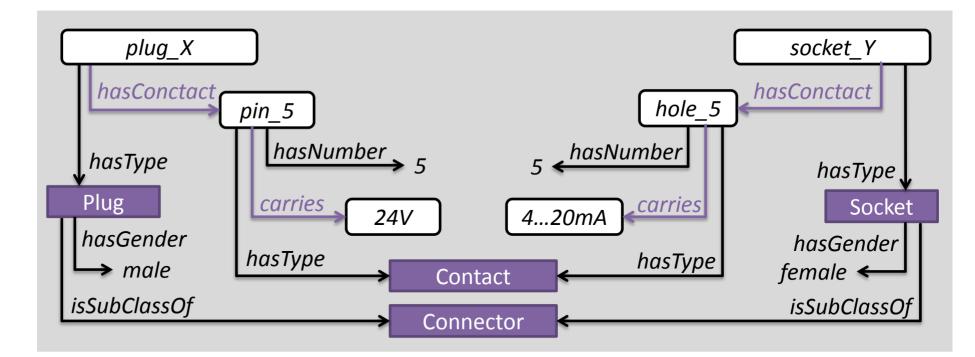
- Where to apply them?
- How to apply them?
- How to build them?
- How to use them?
- Conclusions

WEB3005

10

### What are semantic models?

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



Contact *carries* 

GENERIC		ELECTRIC	
hasType hasGender	isSubClassOf hasNumber	Plug Socket	onnector asContact

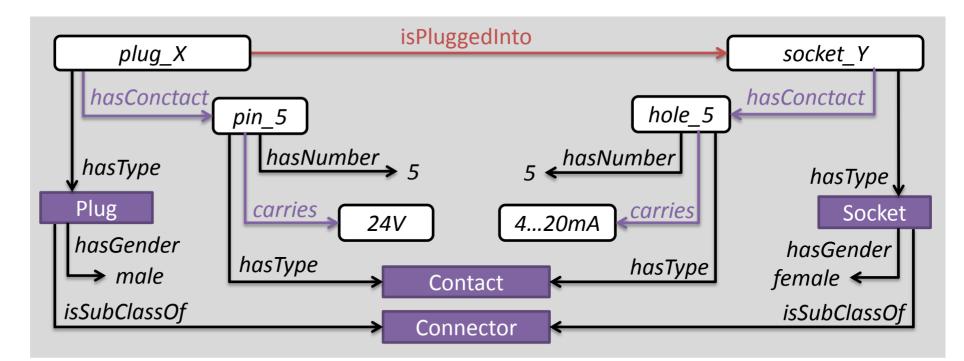
• What are semantic models?

- Where to apply them?
- How to apply them?
- How to build them?
- How to use them?
- Conclusions

WEB3005

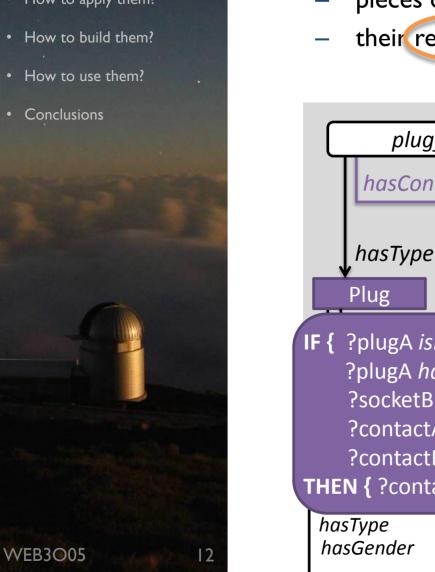
11

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

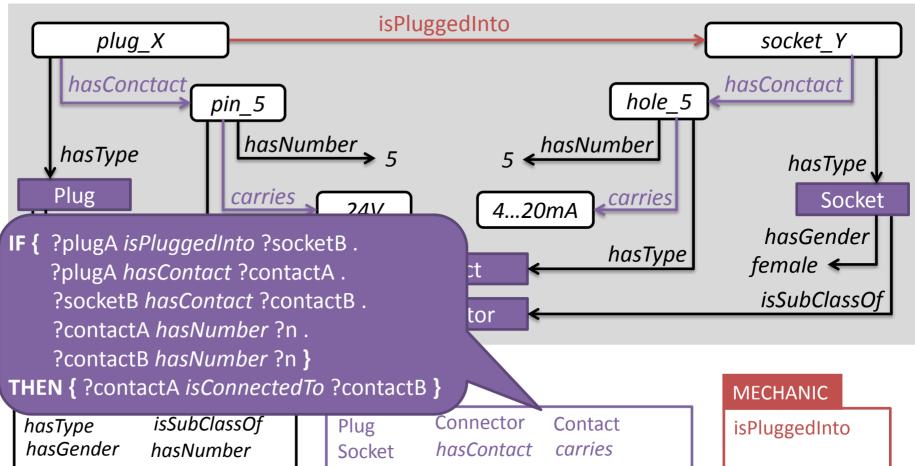


GENERIC		ELECTRIC			_	MECHANIC
hasType hasGender	isSubClassOf hasNumber	1108	onnector <i>asContact</i>	Contact carries		isPluggedInto

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



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

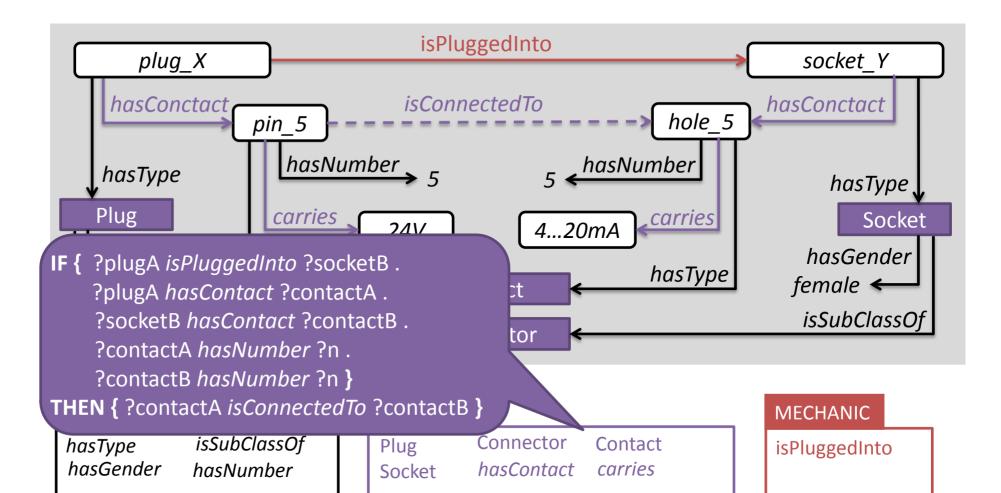


• What are semantic models?

- Where to apply them?
- How to apply them?
- How to build them?
- How to use them?
- Conclusions



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

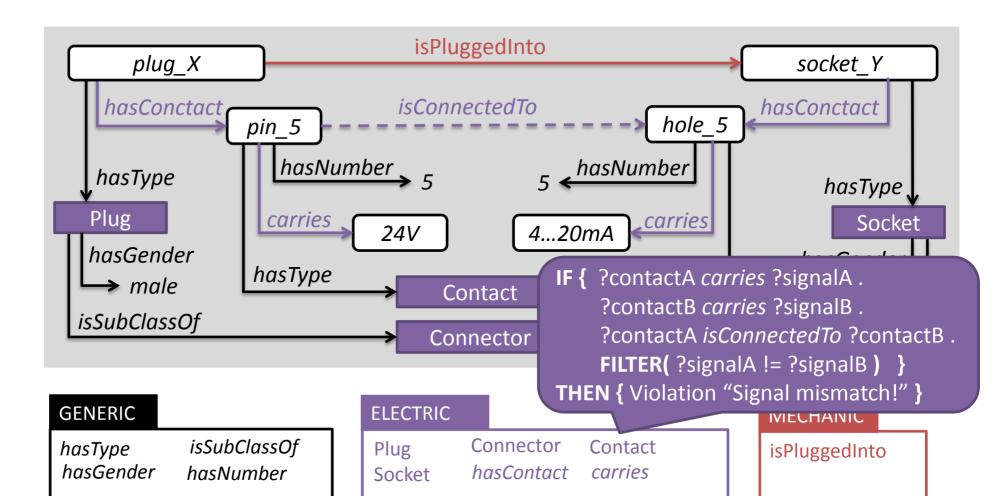


• What are semantic models?

- 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)



WEB3005

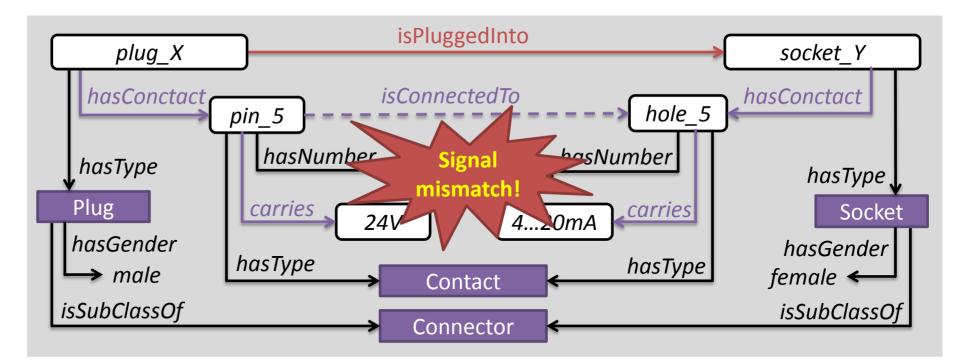
• What are semantic models?

- Where to apply them?
- How to apply them?
- How to build them?
- How to use them?
- Conclusions

WEB3005

15

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



GENERIC		ELECTRIC			_	MECHANIC
hasType hasGender	isSubClassOf hasNumber	Plug Socket	Connector hasContact	Contact carries		isPluggedInto

- 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?

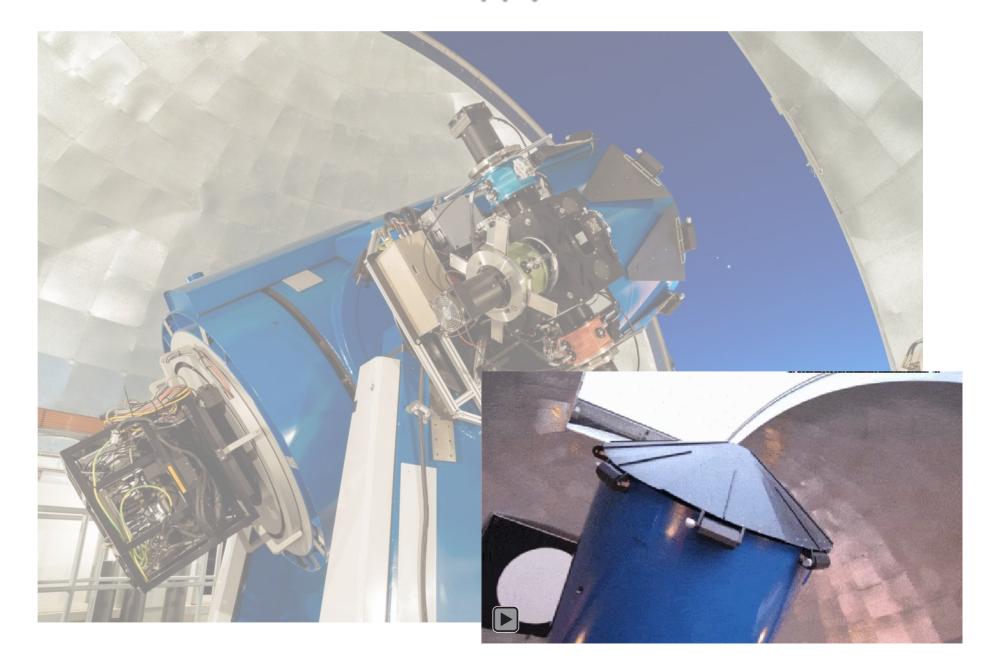


- What are semantic models?
- Where to apply them?
- How to apply them?
- How to build them?
- How to use them?
- Conclusions



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

# Where to apply them?



WEB3005

- 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?



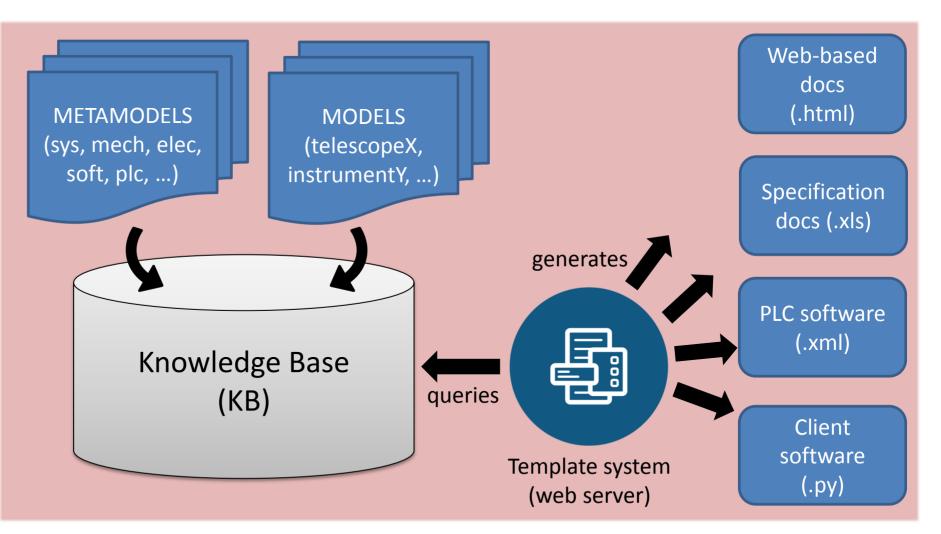
• What are semantic models?

 $\bullet$ 

- Where to apply them?
- How to apply them?
- How to build them?
- How to use them?
- Conclusions

# How to apply them?

Put them in a Knowledge Base and extract information!



### **OntoManager**

- 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?

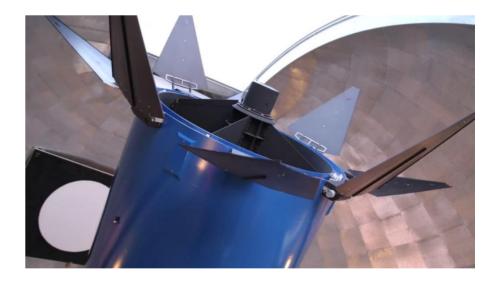
- What are semantic models?
- Where to apply them?
- How to apply them?
- How to build them?
- How to use them?
- Conclusions

### How to build them?

- 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, ...)

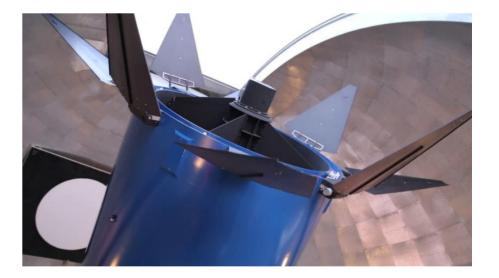


- What are semantic models?
- Where to apply them?
- How to apply them?
- How to build them?
- How to use them?
- Conclusions



## 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]

[1] J. M. Filgueira, "GMT software and controls overview", Proc. SPIE 8451, Amsterdam, July 2012, 845111

- 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**

#### 🖸 OntoManager @ Merc 🗙 📥 Extensions × → C f localhost:8080/home?show ontology=32 ☆ = OntoManager @ Mercator Telescope 0 Software Ontologies Dataset Problems Browse Ouerv Systems Mechanics Electronics Models Model: http://mercator.iac.es/onto/models/external/beckhoff http://mercator.iac.es/onto/models/common/roles Sources: [coffee] http://mercator.jac.es/onto/models/common/softwaremodels 230 - beckhoff.ADD IO MODULE TYPE( http://mercator.iac.es/onto/models/external/all 231 id "EL1088 "8-channel digital input terminal 24V DC, negative switching" 232 comment http://mercator.iac.es/onto/models/external/beckhoff 233 manufacturer : beckhoff.company http://mercator.jac.es/onto/models/external/binder 234 terminals: http://mercator.iac.es/onto/models/external/faulhaber 235 : -> symbol: "I1" , comment: "Input 1" http://mercator.iac.es/onto/models/external/harting 236 : -> sýmbol: "I2" , comment: "Triput 2" 237 : -> symbol: "I3" , comment: "Triput 3' http://mercator.iac.es/onto/models/external/itt 238 : -> symbol: "I4" , comment: "Triput 4" http://mercator.iac.es/onto/models/external/maxon : -> symbol: "I5" , comment: 239 "Trout 5" http://mercator.iac.es/onto/models/external/phoenix 240 : -> symbol: "I6" , comment: "Input 6" 241 : -> sýmbol: "I7" . comment: "Input 7" http://mercator.iac.es/onto/models/external/prehkeytec 242 : -> sýmbol: "I8" . comment: "Input 9 http://mercator.iac.es/onto/models/mtcs/common/all 243 channels: http://mercator.iac.es/onto/models/mtcs/common/hardware 244 1 : -> terminals: [ \$.terminals["1" http://mercator.iac.es/onto/models/mtcs/common/software 245 : -> terminals: [ \$.terminals["2" 246 : -> terminals: [ \$.terminals["3"] http://mercator.iac.es/onto/models/mtcs/common/statemachines 247 : -> terminals: [ \$.terminals["4"] http://mercator.iac.es/onto/models/mtcs/cover/electronics 248 : -> terminals: [\$.terminals["5"] http://mercator.iac.es/onto/models/mtcs/cover/mechanics 249 : -> terminals: [\$.terminals["6"] 250 7 : -> terminals: [ \$.terminals["7"] http://mercator.iac.es/onto/models/mtcs/cover/software 251 8 : -> terminals: [ \$.terminals["8"] http://mercator.iac.es/onto/models/mtcs/m1/electronics 252 soft interface: http://mercator.iac.es/onto/models/mtcs/m1/mechanics 253 254 255 inputl comment: 'Input l' : -> { type: t bool http://mercator.iac.es/onto/models/mtcs/m1/software input2 : -> { type: t bool comment Input 2' input3 type: t\_bool : -> { comment Triput 31 http://mercator.iac.es/onto/models/mtcs/m3/electronics 256 input4 : -> { type: t\_bool comment Trinut 4 http://mercator.iac.es/onto/models/mtcs/m3/mechanics 257 input5 : -> { type: t\_bool comment Input 5' http://mercator.iac.es/onto/models/mtcs/m3/software 258 input6 : -> { type: t bool comment Input 6' 259 input7 : -> { type: t bool comment Innut 7' http://mercator.iac.es/onto/models/mtcs/software 260 input8 : -> { type: t bool comment 'Input 8' http://mercator.iac.es/onto/models/mtcs/telemetry/electronics 261 WcState : -> { type: t bool , comment: 'EtherCAT Working counter state' } http://mercator.iac.es/onto/models/mtcs/telemetry/mechanics 262 InfoDataState : -> { type: t uint16, comment: 'EtherCAT state (INIT, PREOP, OP, ...)' } http://mercator.iac.es/onto/models/mtcs/telemetry/software 263 "EL1088" 264 http://mercator.iac.es/onto/models/mtcs/timing/electronics 265 - beckhoff.ADD IO MODULE TYPE( http://mercator.iac.es/onto/models/mtcs/timing/mechanics 266 id "EL2008' http://mercator.iac.es/onto/models/mtcs/timing/software 267 http://mercator.iac.es/onto/models/test/test http://mercator.iac.es/onto/models/test/test\_expressions

How to build them?

WEB3005

• What are semantic models?

 $\leftarrow \rightarrow$ 

٢

- 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**

🖪 OntoManager @ Merc 🗙 🌸 Exten	sions ×								٨
← → C 🖬 🗋 localhost:8080/	home?show_ont	ology=32							☆ =
		C	ntoMana	ager @ Merc	ator Tele	escope			<u>^</u>
0	Ontologies	Dataset	Problems	Browse Query	Systems	Mechanics	Electronics	Software	
Models			Model:	http://mercator.	iac.es/onto/	models/ext	ernal/beckh	off	
<ul> <li>http://mercator.iac.es/onto/models/co http://mercator.iac.es/onto/models/co</li> <li>http://mercator.iac.es/onto/models/ex</li> <li>http://mercator.iac.es/onto/models/m</li> <li>http://mercator.iac.es/ont</li></ul>	mmon/softwaremodel ternal/all ternal/binder ternal/binder ternal/harting ternal/harting ternal/harting ternal/phoenix ternal/phoenix ternal/prehkeytec tes/common/salt tes/common/salt tes/common/sattemat tes/cover/software tes/cover/software tes/cover/software tes/ma/nechanics tes/ma/software tes/m3/software tes/m3/software tes/m3/software tes/software tes/software tes/software tes/software tes/telemetry/electronics tes/telemetry/software tes/telemetry/software tes/timing/software	e chines ilcs ilcs	Sources: [ 230 ~ E 231 232 233 234 ~ 235 236 237 238 239 240 241 242 243 ~ 244 245 246 247 248 249 250 251 252 ~ 253 254 255 256 257 258 259 260 261 262 263 264	<pre>[coffee] beckhoff.ADD IO_MODULE id :" comment :" manufacture :" b terminals: 1 : -&gt; symbol: 3 : -&gt; symbol: 3 : -&gt; symbol: 4 : -&gt; symbol: 6 : -&gt; symbol: 7 : -&gt; symbol: 8 : -&gt; symbol: 8 : -&gt; symbol: channels: 1 : -&gt; termina 2 : -&gt; termina 3 : -&gt; termina 4 : -&gt; termina 5 : -&gt; termina 6 : -&gt; termina 8 : -&gt; termina 5 : -&gt; termina 6 : -&gt; termina 7 : -&gt; termina 8 : -&gt; termina 5 : -&gt; termina 8 : -&gt; termina 9 : input4 input5 input6 input7 : input8 WcState : ] "EL1088"</pre>	TYPE( EL1088" 8-channel digit: beckhoff.company "I1", comment "I2", comment "I3", comment "I4", comment "I5", comment "I6", comment "I6", comment "I6", comment "I7", comment "I8", comment sis: [\$.termina sis: [\$.	al input termin : "Input 1" : "Input 2" : "Input 2" : "Input 4" : "Input 5" : "Input 6" : "Input 6" : "Input 8" ls["2"] ] ls["3"] ] ls["3"] ] ls["4"] ] ls["5"] ] ls["6"] ]	<pre>inal 24V DC, nega i'Input 1' } i'Input 2' } i'Input 3' } i'Input 4' } i'Input 6' } i'Input 6' } i'Input 6' } i'Input 8' } i'EtherCAT Work.</pre>	_	, <b>,</b>

How to build them?

• 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**

→ C ☆ Diocalhost:808	0/home?show_ontology=3		\$
	official off		~
		OntoManager @ Mercator Telescope	
	Ontologies Datase	t Problems Browse Query Systems Mechanics Electronics Software	
dels		Model: http://mercator.iac.es/onto/models/externa <mark>l/beckhoff</mark>	
http://mercator.iac.e	'common/roles 'common/softwaremodels	Sources: [coffee]	
http://mercator.iac.	<ternal all<="" td=""><td>230 beckhoff.ADD IO_MODULE_TYPE( 231 id : "EL1088"</td><td></td></ternal>	230 beckhoff.ADD IO_MODULE_TYPE( 231 id : "EL1088"	
http://mercator.iac	srnal/beckhoff	231 comment : "8-channel digital input terminal 24V DC, negative switching"	
http://mercator.iac	nal/binder	233 manufacturer : beckhoff.company	
http://mercator.iac	al/faulhaber	234 terminals: 235 1 : -> symbol: "I1" , comment: "Input 1"	
http://mercator.ia	harting	236 2 : -> symbol: "I2" , comment: "Input 2"	
http://mercator.ia.	it.	237 3 : -> symbol: "I3", comment: "Input 3"	
http://mercator.ia	naxon	238 4 : -> symbol: "I4" , comment: "Input 4" 239 5 : -> symbol: "I5" , comment: "Input 5"	
http://mercator.ia	phoenix	240 6 : -> symbol: "I6" , comment: "Input 6"	
http://mercator.ii 👿 🕎 🚺	prehkeytec	241 7 : -> symbol: "I7" , comment: "Input 7" 242 8 : -> symbol: "I8" , comment: "Input 8"	
http://mercator.i	nmon/all	242 8 : -> symbol: "I8" , comment: "Input 8" 243 channels:	
http://mercator. 🌒 🖤 👘	mmon/hardware	244 1 : -> terminals: [ \$.terminals["1"] ]	
http://mercator	mmon/software	245 2 : -> terminals: [ \$.terminals["2"] ] 246 3 : -> terminals: [ \$.terminals["3"] ]	
http://mercator 15 16	mmon/statemachines	247 4 : -> terminals: [\$.terminals["4"]]	
http://mercator	over/electronics	248 5 : -> terminals: [ \$.terminals["5"] ]	
http://mercato	/over/mechanics /cover/software	249 6 : -> terminals: [ \$.terminals["6"] ] 250 7 : -> terminals: [ \$.terminals["7"] ]	
http://mercato	/m1/electronics	251 8 : -> terminals [ \$.terminals["8"] ]	
http://mercat	m1/mechanics	252 soft_interface:	
http://mercat	m1/software	253 input1 : -> { type: t_bool , comment: 'Input 1' } 254 input2 : -> { type: t_bool , comment: 'Input 2' }	
http://mercat	/m3/electronics	255 input3 : -> { type: t bool , comment: 'Input 3' }	
http://merca	ncs/mechanics	256 input4 :-> { type: t_bool , comment: 'Input 4' }	
http://merca	cs/m3/software	257 input5 : -> { type: t_bool , comment: 'Input 5' } 258 input6 : -> { type: t_bool , comment: 'Input 6' }	
http://mercialian	cs/software	259 input7 : -> { type: t_bool , comment: 'Input 7' }	
http://mercBECKHOFF	s/telemetry/electronics	260 input8 : -> { type: t_bool , comment: 'Input 8' } 261 WcState : -> { type: t_bool , comment: 'EtherCAT Working counter state' }	
http://merc	s/telemetry/mechanics	<pre>261 WcState : -&gt; { type: t_bool , comment: 'EtherCAT Working counter state' } 262 InfoDataState : -&gt; { type: t_uint16, comment: 'EtherCAT state (INIT, PREOP, OP,)' }</pre>	
http://me	ocs/telemetry/software	263 ) "EL1088"	
http://mercator.iac.	<pre>&gt;\cs/telemetry/software &gt;\cs/timing/electronics</pre>	264 265 - beckhoff.ADD IO MODULE TYPE(	
http://mercator.iac.es/onto/models/	mtcs/timing/mechanics	266 id : "EL2008"	
http://mercator.iac.es/onto/models/	mtcs/timing/software	267	

• 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 **instance**

🖪 OntoManager @ Merc 🗙 🙀 Extensions 🛛 🗙 🔪	
← → C f Dicalhost:8080/home?show_ontology=	=44 ☆
	OntoManager @ Mercator Telescope
Ontologies Data	iset Problems Browse Query Systems Mechanics Electronics Software
Models	Model: http://mercator.iac.es/onto/models/mtcs/cover/electronics
<ul> <li>http://mercator.iac.es/onto/models/common/roles</li> <li>http://mercator.iac.es/onto/models/cxternal/all</li> <li>http://mercator.iac.es/onto/models/external/beckhoff</li> <li>http://mercator.iac.es/onto/models/external/beckhoff</li> <li>http://mercator.iac.es/onto/models/external/beckhoff</li> <li>http://mercator.iac.es/onto/models/external/beckhoff</li> <li>http://mercator.iac.es/onto/models/external/beckhoff</li> <li>http://mercator.iac.es/onto/models/external/hanlbaber</li> <li>http://mercator.iac.es/onto/models/external/harting</li> <li>http://mercator.iac.es/onto/models/external/harting</li> <li>http://mercator.iac.es/onto/models/external/hoenix</li> <li>http://mercator.iac.es/onto/models/external/prenix</li> <li>http://mercator.iac.es/onto/models/external/prenix</li> <li>http://mercator.iac.es/onto/models/mtcs/common/all</li> <li>http://mercator.iac.es/onto/models/mtcs/common/software</li> <li>http://mercator.iac.es/onto/models/mtcs/common/software</li> <li>http://mercator.iac.es/onto/models/mtcs/cover/software</li> <li>http://mercator.iac.es/onto/models/mtcs/cover/software</li> <li>http://mercator.iac.es/onto/models/mtcs/cover/software</li> <li>http://mercator.iac.es/onto/models/mtcs/cover/software</li> <li>http://mercator.iac.es/onto/models/mtcs/cover/software</li> <li>http://mercator.iac.es/onto/models/mtcs/m1/software</li> <li>http://mercator.iac.es/onto/models/mtcs/m3/electronics</li> <li>http://mercator.iac.es/onto/models/mtcs/m3/electronics</li> <li>http://mercator.iac.es/onto/models/mtcs/m3/software</li> <li>http://mercator.iac.es/onto/models/mtcs/m3/software</li> <li>http://mercator.iac.es/onto/models/mtcs/m3/software</li> <li>http://mercator.iac.es/onto/models/mtcs/m3/software</li> <li>http://mercator.iac.es/onto/models/mtcs/telemetry/electronics</li> <li>http://mercator.iac.es/onto/models/mtcs/telemetry/electronics</li> <li>http://mercator.iac.es/onto/models/mtcs</li></ul>	Sources: [coffee] 115 116 117 119 118 118 119 119 119 119 119 119 119

WEB3005

• 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 **instance**

🖸 OntoManager @ Merc 🗙 🌸 Extensions 🛛 🗙 🗖	
← → C ☆ localhost:8080/home?show_ontology=4	44 ☆
	OntoManager @ Mercator Telescope
	Childhanager @ Mercalor Telescope
Ontologies Datase	et Problems Browse Query Systems Mechanics Electronics Software
Models	Model: http://mercator.iac.es/onto/models/mtcs/cover/electronics
<ul> <li>http://mercator.iac.es/onto/models/common/roles</li> </ul>	Sources: [coffee]
<ul> <li>http://mercator.iac.es/onto/models/common/softwaremodels</li> </ul>	115
<ul> <li>http://mercator.iac.es/onto/models/external/all</li> </ul>	116 - pover_elec.ADD_IO_MODULE_INSTANCE(
<ul> <li>http://mercator.iac.es/onto/models/external/beckhoff</li> </ul>	117 type : beckhoff.El1088 118 comment : "Digital input terminal to read the status of the SSI encoders of all 8 cover panels"
<ul> <li>http://mercator.iac.es/onto/models/external/binder</li> </ul>	119 comment - Digitat input terminar to requirements, absFeedbackStatus
<ul> <li>http://mercator.iac.es/onto/models/external/faulhaber</li> </ul>	120- terminals :
<ul> <li>http://mercator.iac.es/onto/models/external/harting</li> </ul>	121 1: -> symbol: "TC:T1:SSISTS", comment: "Top 1 SSI status", isConnectedTo: cover_elec.connectors.T1.pins[13] 122 2: -> symbol: "TC:T2:SSISTS", comment: "Top 2 SSI status", isConnectedTo: cover_elec.connectors.T2.pins[13]
<ul> <li>http://mercator.iac.es/onto/models/external/itt</li> </ul>	<pre>122 2: -&gt; symbol: "TC:T2:SSISTS", comment: "Top 2 SSI status" , isConnectedTo: cover_elec.connectors.T2.pins[13] 123 3: -&gt; symbol: "TC:T3:SSISTS", comment: "Top 3 SSI status" , isConnectedTo: cover_elec.connectors.T3.pins[13]</pre>
<ul> <li>http://mercator.iac.es/onto/models/external/maxon</li> </ul>	124 4: -> symbol: "TC:T4:SSISTS", comment: "Top 4 SSI status", isConnectedTo: cover elec.connectors.T4.pins[13]
<ul> <li>http://mercator.iac.es/onto/models/external/phoenix</li> </ul>	125 5: -> symbol: "TC:B1:SSISTS", comment: "Bottom SSI status", isConnectedTo: cover_elec.connectors.B1.pins[13]
<ul> <li>http://mercator.iac.es/onto/models/external/prehkeytec</li> </ul>	126 6: -> symbol: "TC:B2:SSISTS", comment: "Bottom SSI status", isConnectedTo: cover_elec.connectors.B2.pins[13] 127 7: -> symbol: "TC:B3:SSISTS", comment: "Bottom SSI status", isConnectedTo: cover_elec.connectors.B3.pins[13]
<ul> <li>http://mercator.iac.es/onto/models/mtcs/common/all</li> </ul>	127 3> symbol: "TC:B4:SSIST", comment: "Bottom SSI status", isConnectedTo: cover_elec.connectors.B4.pins[13]
<ul> <li>http://mercator.iac.es/onto/models/mtcs/common/hardware</li> </ul>	129 ) "slot3"
<ul> <li>http://mercator.iac.es/onto/models/mtcs/common/software</li> </ul>	130 131
<ul> <li>http://mercator.iac.es/onto/models/mtcs/common/statemachines</li> </ul>	131 132 - for [ slot, connector], connector2, panel1, panel2 ] in [ ['slot4', 'T1', 'T2', 'Top 1', 'Top 2'],
<ul> <li>http://mercator.iac.es/onto/models/mtcs/cover/electronics</li> </ul>	133 ['slot5', 'T3', 'T4', 'Top 3', 'Top 4'],
<ul> <li>http://mercator.iac.es/onto/models/mtcs/cover/mechanics</li> </ul>	134 ['slot6', 'B1', 'B2', 'Bottom 1', 'Bottom 2'],
<ul> <li>http://mercator.iac.es/onto/models/mtcs/cover/software</li> </ul>	135 136 - cover elec.ADD IO MODULE INSTANCE(
<ul> <li>http://mercator.iac.es/onto/models/mtcs/m1/electronics</li> </ul>	135 comment : "SSI module for #(panell) and #(panel2) encoders"
<ul> <li>http://mercator.iac.es/onto/models/mtcs/m1/mechanics</li> </ul>	138 type : beckhoff.EL5002
<ul> <li>http://mercator.iac.es/onto/models/mtcs/m1/software</li> </ul>	<pre>139 - terminals : 140 l: -&gt; symbol: "TC:#{connector1}:SSID+", comment: "#{panel1} SSI encoder Data +" , isConnectedTo: cover elec</pre>
<ul> <li>http://mercator.iac.es/onto/models/mtcs/m3/electronics</li> </ul>	140 1: -> symbol: "IC:#;connector1;:SsL+, comment: #;panel1; SsL encoder Data + , isConnected10: cover_elec 141 2: -> symbol: "IC:#;connector1;:SsL+, comment: #;panel1; SSL encoder Clock + , isConnectedT0: cover_elec
<ul> <li>http://mercator.iac.es/onto/models/mtcs/m3/mechanics</li> </ul>	142 3: -> symbol: "TC:#{connector2}:SSID+", comment: "#{panel2} SSI encoder Data +" , isConnectedTo: cover_elec
<ul> <li>http://mercator.iac.es/onto/models/mtcs/m3/software</li> </ul>	143 4: -> symbol: "TC:#{connector2}:SSIC+", comment: "#{panel2} SSI encoder Clock +", isConnectedTo: cover_elected and the symbol and the
<ul> <li>http://mercator.iac.es/onto/models/mtcs/software</li> </ul>	<pre>144 5: -&gt; sýmbol: "TC:#{connectorl}:SSID-", comment: "#{panell} SSI encoder Data -", isConnectedTo: cover_elec 145 6: -&gt; symbol: "TC:#{connectorl}:SSIC-", comment: "#{panell} SSI encoder Clock -", isConnectedTo: cover_elec</pre>
<ul> <li>http://mercator.iac.es/onto/models/mtcs/telemetry/electronics</li> </ul>	146 7: -> symbol: "TC:#{connector2}:SSID-", comment: "#{panel2} SSI encoder Data -", isConnectedTo: cover elec
<ul> <li>http://mercator.iac.es/onto/models/mtcs/telemetry/mechanics</li> </ul>	147 8: -> symbol: "TC:#{connector2}:SSIC-", comment: "#{panel2} SSI encoder Clock -", isConnectedTo: cover_elec
<ul> <li>http://mercator.iac.es/onto/models/mtcs/telemetry/software</li> </ul>	148 ) slot 149
<ul> <li>http://mercator.iac.es/onto/models/mtcs/timing/electronics</li> </ul>	149
<ul> <li>http://mercator.iac.es/onto/models/mtcs/timing/mechanics</li> </ul>	151
<ul> <li>http://mercator.iac.es/onto/models/mtcs/timing/software</li> </ul>	152~
<ul> <li>http://mercator.iac.es/onto/models/test/test</li> </ul>	
<ul> <li>http://mercator.iac.es/onto/models/test/test_expressions</li> </ul>	

WEB3005

• What are semantic models?

30

- 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 **instance**

	8
← → C ↑ Discalhost:8080/home?show_ontology=44	3 =
OntoManager @ Mercator Telescope	Î
Ontologies Dataset Problems Browse Query Systems Mechanics Electronics Software	
Model: http://mcrator.isc.es/onto/models/torman/isdes • http://mcrator.isc.es/onto/models/tormano/sdes • http://mcrator.isc.e	

WEB3005

- 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?

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



🗾 OntoMana	ager @ Merc ×										8			
←⇒Cr	🕈 🗋 localhost	:8080/elec								5	Ξ			
	OntoManager @ Mercator Telescope													
٥	Ontologies	Dataset	Problems	Browse	Query	Systems	Mechanics	Electronics	Software					
+ Cover + M1 + M3 + Telemetry + Timing	K													

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



🗾 OntoMana	iger @ Merc × 🚺									8
← → C' f	localhos	:8080/elec								☆ =
		(	OntoMai	nager @	Mer	cator Te	elescope			
٢	Ontologies	Dataset	Problems	Browse	Query	Systems	Mechanics	Electronics	Software	
+ Cover										
+ M1										
+ M3 + Telemetry	•									
+ Timing										

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



-> C n	🗋 localhost	:0000/elec	open=conng	guración, pac	.n=cover_	elec.covel				Ś
		(	OntoMai	nager (	D Mer	cator Te	elescope			
	Ontologies	Dataset	Problems	Browse	Query	Systems	Mechanics	Electronics	Software	
Cover										
+ VO module + terminals	S									
- + connectors	3									
M1 M3										
M3 Telemetry										
Timing										

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



+ M3 + Telemetry	- ⇒ C f	localhost									☆
Cover Solution			(	OntoMai	nager @	D Mer	cator Te	elescope			
<ul> <li>↓ Vo modules</li> <li>↓ Siot0</li> <li>↓ Siot2</li> <li>↓ Siot3</li> <li>↓ Siot6</li> <li>↓ Siot6</li> <li>↓ Siot8</li> <li>↓ Siot10</li> <li>↓ Siot11</li> <li>↓ Siot12</li> <li>↓ Siot13</li> <li>↓ Terminals</li> <li>↓ Termetry</li> </ul>	9	Ontologies	Dataset	Problems	Browse	Query	Systems	Mechanics	Electronics	Software	
A slot0         A slot1         A slot2         A slot3         A slot4         A slot5         A slot6         A slot6         A slot6         A slot6         A slot7         A slot8         A slot9         A slot9         A slot9         A slot9         A slot10         A slot11         A slot12         A slot13         A slot14         A slot12         A slot13         A slot14         A slot12         A slot13         A slot14         A slot15         A slot15         A slot14         A slot15         A slot15         A slot14         A slot15         A slot15         A slot16         A slot17         A slot18         A slot19         A slot19 </td <td>- Cover</td> <td></td>	- Cover										
<ul> <li>→ slot1</li> <li>→ slot2</li> <li>→ slot3</li> <li>→ slot4</li> <li>→ slot5</li> <li>→ slot6</li> <li>→ slot7</li> <li>→ slot8</li> <li>→ slot9</li> <li>→ slot10</li> <li>→ slot11</li> <li>→ slot12</li> <li>→ slot13</li> <li>→ slot13</li> <li>→ connectors</li> <li>→ MI</li> <li>→ Telemetry</li> </ul>		es									
<ul> <li>+ \$slot3</li> <li>+ \$slot3</li> <li>+ \$slot4</li> <li>+ \$slot5</li> <li>+ \$slot6</li> <li>+ \$slot7</li> <li>+ \$slot8</li> <li>+ \$slot9</li> <li>+ \$slot10</li> <li>+ \$slot11</li> <li>+ \$slot12</li> <li>+ \$slot13</li> <li>+ \$terminals</li> <li>+ \$connectors</li> <li>+ \$MI</li> <li>+ MI</li> <li>+ Telemetry</li> </ul>											
	- + slot2										
<ul> <li>* slot5</li> <li>* slot6</li> <li>* slot7</li> <li>* slot8</li> <li>* slot9</li> <li>* slot10</li> <li>* slot11</li> <li>* slot12</li> <li>* slot13</li> <li>* terminals</li> <li>* connectors</li> <li>* MI</li> <li>* MI</li> <li>* Telemetry</li> </ul>											
<pre></pre>											
<ul> <li>+ slot9</li> <li>+ slot10</li> <li>+ slot12</li> <li>+ slot13</li> <li>+ terminals</li> <li>+ connectors</li> <li>M1</li> <li>M3</li> <li>Telemetry</li> </ul>											
<ul> <li>+ slot10</li> <li>+ slot11</li> <li>+ slot12</li> <li>+ slot13</li> <li>+ terminals</li> <li>+ connectors</li> <li>+ M1</li> <li>+ M3</li> <li>+ Telemetry</li> </ul>											
+ slot11 + slot12 + slot13 + terminals + connectors M1 + M3 + Telemetry											
<pre></pre>											
<pre></pre>											
+ connectors + M1 + M3 + Telemetry											
± M1 ± M3 ± Telemetry											
+ M3 + Telemetry		s									
+ Telemetry											
	± Timing										

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



🗾 🖸 OntoMana	ger @ Merc × Ҭ										8
$\leftarrow \rightarrow$ C f	localhost	:8080/elec?	open=config	juration;pat	h=cover_	elec:Cover::t	erminals			53	≡
		(	OntoMar	nager @	Mero	cator Te	elescope				
•	Ontologies	Dataset	Problems	Browse	Query	Systems	Mechanics	Electronics	Software		
- Cover - VO module + slot0 - + slot1 - + slot2 - + slot3 - + slot3 - + slot4 - + slot5 - + slot6 - + slot7 - + slot8 - + slot10 - + slot11 - + slot12 - + slot13 - + slot12 - + slot13 - + slot3 - + slot3											
+ Telemetry											

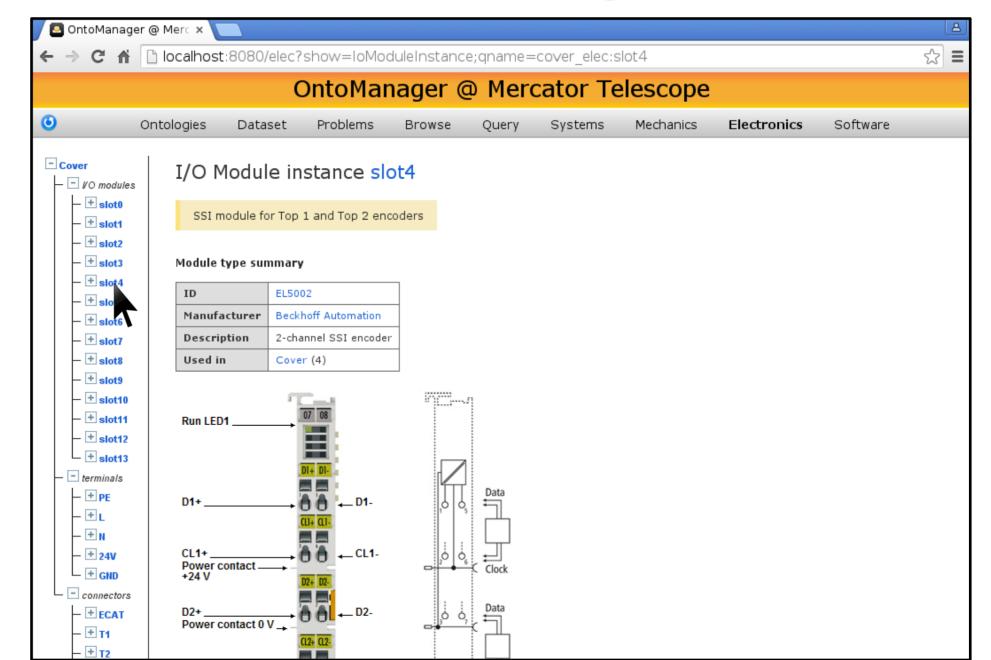
- What are semantic models?
- Where to apply them?
- How to apply them?
- How to build them?
- How to use them?
- Conclusions



_	ager @ Merc ×	+-8080/elec2		uration-pat	h=cover	elec:Cover	connectors			) کت
										22
		C	Intomar	lager @	<u>p</u> Mero	cator Te	elescope			
	Ontologies	Dataset	Problems	Browse	Query	Systems	Mechanics	Electronics	Software	
Cover										
– – VO modu	100									
- + slot1										
- + slot2										
- + slot3										
- + slot4										
- + slot5										
- 🕂 slot6										
- + slot7										
– 🕂 slot8										
- + slot9										
- + slot1										
- 🕂 slot1										
- 🕂 slot1										
L + slot1										
– 🖃 terminals										
— ⊡ PE — ± L										
- ± N										
- ± 24V										
+ GND										
- 💽 connecto	rs									
ECAT	г									
- ± 11										
- + T2										

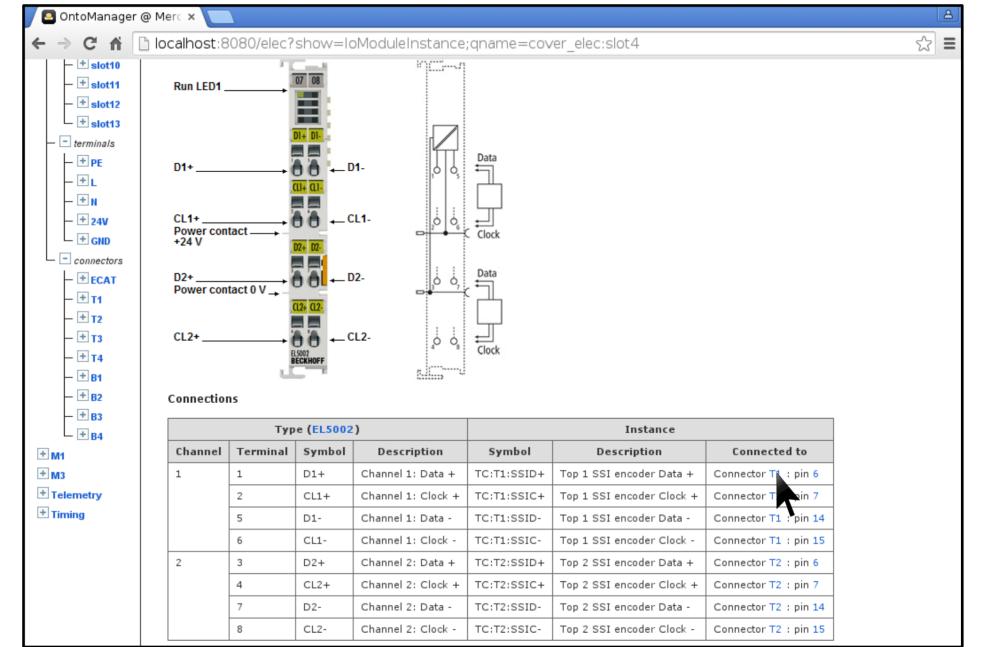
- What are semantic models?
- Where to apply them?
- How to apply them?
- How to build them?
- How to use them?
- Conclusions





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





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



🔁 Ont	toManage	er @ M	erc ×									l	8
← ⇒	C 🖬	🗋 lo	calho	st:8080/	/elec?show=	=ConnectorInst	ance;qname=cover	r_elec:c	connectors.T1	L		숬	≡
					Onto	Manager	@ Mercator	r Tel	lescope				
0						-			-				
0		Ontol	ogies	Data	set Prob	ems Browse	e Query Syste	ems	Mechanics	Electronic	s Software		
	VO modules		Con	necto	r instanc	e T1							
1 1	+ slot0		Conne	ector type	summary								
	+ slot1 + slot2		ID		D-sub 15 F								
	+ slot3		Gene	der	female								
1 1	+ slot4		Man	ufacturer	ITT Corporatio	on							
1 1	+ slot5		Desc	ription	D-sub 15 fem	ale connector							
	+ slot6 + slot7		Fits	to	D-sub 15 M								
1 1	+ slot7		Used	d in	Cover (8), M1	(1), M3 (2)							
1 1	+ slot9		_										
1 1	+ slot10 + slot11		8 15	••••	•••••	• ) 1 9							
	+ slot12		D	A-15S (Fem	ale Socket Front	View)							
	+ slot13 terminals		Conne	ections									
	+ PE												
	+ L			Type (D-s			Instand	ce					
	+ N		Pin	Symbol	Description	Symbol	Description		Connecte		•		
	+ 24V		1	1	Pin 1	TC:T1:GND HM	Top 1 GND of holding ma		Cover : terminal (				
	+ GND		2	2	Pin 2	TC:T1:GND MOT	Top 1 GND of motor		Cover : terminal (	GND			
	connectors + ECAT		3	3	Pin 3	TC:T1:MMON	Top 1 motor monitor						
	± T1		4	4	Pin 4	TC:T1:MDIR	Top 1 motor direction		I/O module slot1 :				
	± T2		5	5	Pin 5	TC:T1:GND ENC	Top 1 GND of encoder		Cover : terminal (	GND			

• What are semantic models?

- Where to apply them?
- How to apply them?
- How to build them?
- How to use them?
- Conclusions



🖉 OntoManager (	@ Merc :	×						8
← → C fi [	) local	nost:8080	/elec?show=	=ConnectorInst	ance;qname=cover_elec	:connectors.T1	5	3 =
- + slot5	De	scription	D-sub 15 fem	ale connector				
- + slot6	Fit	ts to	D-sub 15 M					
- + slot7	Us	ed in	Cover (8), M1	(1), M3 (2)				
- + slot8 - + slot9								
- ± slot10	8 (	•••		• 1				
- + slot11	15	••	• • • •	• / 9				
- + slot12		DA-15S (Fem	nale Socket Front	View)				
+ slot13	-							
terminals	Con	nections						
- ± L		Type (D-s	ub 15 F)		Instance			
— ± N	Pir	n Symbol	Description	Symbol	Description	Connected to		
- + 24V	1	1	Pin 1	TC:T1:GND HM	Top 1 GND of holding magnet	Cover : terminal GND		
+ GND	2	2	Pin 2	TC:T1:GND MOT	Top 1 GND of motor	Cover : terminal GND		
- connectors	3	3	Pin 3	TC:T1:MMON	Top 1 motor monitor			
- + ECAT	4	4	Pin 4	TC:T1:MDIR	Top 1 motor direction	I/O module slot1 : terminal 1		
- ± T1 - ± T2	5	5	Pin 5	TC:T1:GND ENC	Top 1 GND of encoder	Cover : terminal GND		
—	6	6	Pin 6	TC:T1:SSID+	Top 1 SSI Data +	I/O module slot4 : terminal 1		
- ± 14	7	7	Pin 7	TC:T1:SSIC+	Top 1 SSI Clock +	I/O module slot4 : terminal 2		
— + B1	8	8	Pin 8	TC:T1:PE	Top 1 Earth	Cover : terminal PE		
— + B2	9	9	Pin 9	TC:T1:+24V HM	Top 1 +24V of holding magnet	I/O module slot12 : terminal 2		
— <del>+</del> B3	10	10	Pin 10	TC:T1:+24V MOT	Top 1 +24V of motor	I/O module slot8 : terminal 2		
- + B4	11	11	Pin 11	TC:T1:MSPEED	Top 1 motor speed	I/O module slot2 : terminal 1		
+ M1	12	12	Pin 12	TC:T1:+24V ENC	Top 1 +24V of encoder	Cover : terminal 24V		
+ M3 + Telemetry	13	13	Pin 13	TC:T1:SSISTS	Top 1 SSI status	I/O module slot3 : terminal 1		
Telemetry     Timing	14	14	Pin 14	TC:T1:SSID-	Top 1 SSI Data -	I/O module slo		
	15	15	Pin 15	TC:T1:SSIC-	Top 1 SSI Clock -	I/O module slot4 : terminal 6		

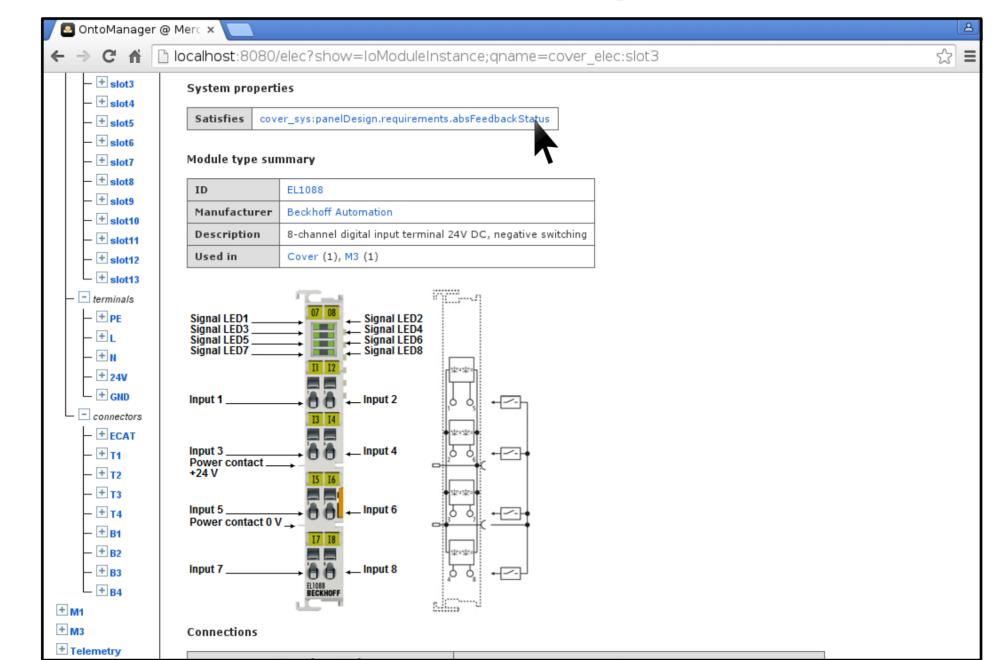
- What are semantic models?
- Where to apply them?
- How to apply them?
- How to build them?
- How to use them?
- Conclusions



OntoManager ← → C ☆		/elec?show=loModu	Jelostance:gna	me=cover el	ec:slot3			ا ک	III Do
				_	Telescope			~	-
<b>i</b> o	ntologies Data		Browse Qu			Electronics	Software		
Cover  VO modules  + slot0	I/O Modul	e instance <mark>slo</mark>	t3						
- + slot1 - + slot2 - + slot3 - + slot4	Digital input t System propert	erminal to read the stat	tus of the SSI enco	ders of all 8 cove	er panels				
- + slot5 - + slot6 - + slot7	Satisfies cov Module type su	er_sys:panelDesign.requir <b>nmary</b>	ements.absFeedbac	<status< td=""><td></td><td></td><td></td><td></td><td></td></status<>					
- + slot8 - + slot9	ID	EL1088							
- + slot10	Manufacturer	Beckhoff Automation							
- + slot11	Description	8-channel digital input te	erminal 24V DC, neg	ative switching					
- + slot12	Used in	Cover (1), M3 (1)							
+ slot13 - • terminals + • PE + L + 1 + N + 24V + GND • connectors + ECAT + T1 + T2	Signal LED1 Signal LED3 Signal LED5 Signal LED7 Input 1 Power contact +24 V	07 08 → Signal LE → Nignal LE → Signal LE	D6			ł			

- 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?
- Where to apply them?
- How to apply them?
- How to build them?
- How to use them?
- Conclusions



🔼 OntoManage	er @ Merc 🗙 📃									٩
← ⇒ C fi	localhost:808	30/sys?s	show=requir	ement;qnan	ne=cover_	sys:panelDe	esign.requirem	nents.absFeed	backStatus	☆ =
		C	OntoMan	ager @	Merc	ator Te	lescope			
0	Ontologies Da	ataset	Problems	Browse	Query	Systems	Mechanics	Electronics	Software	
+ Cover + M1	Requirem	ent <mark>al</mark>	osFeedba	ckStatus		7				
+ M3 + Telemetry + Timing	The status of	the abso	lute feedback sl	nall be known						
	Properties									
	Derives									
	Derived from		r_sys:concept.red r_sys:panelDesig							
	Satisfied by		er_sys:panelDesig er_elec:slot3	n.parts.encoder		_				
	Declared by		 er_sys:panelDesig	n						

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



		C	OntoMar	nager @	) Merc	ator Te	lescope			
	Ontologies	Dataset	Problems	Browse	Query	Systems	Mechanics	Electronics	Software	
+ Cover + M1	Requirer	nent <mark>al</mark>	osFeedba	ckStatus						
+ M3 + Telemetry + Timing	The status	of the abso	lute feedback s	hall be known						
	Properties									
	Derives									
	Derived from		r_sys:concept.re r_sys:panelDesig			_				
	Satisfied by	• cove	r_sys:panelDesig	n.parts.encoder		-				
		• cove	r_elec:slot3							
	Declared by	• cove	r_sys:panelDesig	jn						
						_				
			,							

• What are semantic models?

- Where to apply them?
- How to apply them?
- How to build them?
- How to use them?
- Conclusions



🖊 🖪 OntoManage	er @ Merc 🗙 🕻 🛔 Extension:	S		×													8
← → C fi	localhost:8080/sys	?sho	w=de	esign;	qnam	ne=co	over_s	sys:p	anelD	esigr	1						☆≡
		On	toM	lan	age	er @	D M	erc	ato	r T	eles	sco	ре				
٥	Ontologies Dataset	P	roblen	ns	Brov	wse	Que	ery	Sys	tems	M	lechar	nics	Elec	ctronic	cs Software	
+ Cover + M1	Design panelD	esig	ŋn														
+ M3 + Telemetry	The design of the tel	escop	e cove	r pane	ls												
+ Timing	Requirements derivat	tion m	natrix														
			1		panel	Design						con	ept				
		open	closed	eActuator	moveActuatorStatus	absFeedback	absFeedbackStatus	locking	aluminum	obstructing	sealing	controllable	monitorable	reliability	weight		
	open			•						¥							
	closed										~						
	moveActuator				~							~					
	moveActuatorStatus			*									~				
	absFeedback						~						~				
	<u>absFeedbackStatus</u> The status of the abs		faadh	- als als	- II ha	× .							~				
	aluminum	solute	reeab			KNOW								*	Ľ		

• What are semantic models?

- Where to apply them?
- How to apply them?
- How to build them?
- How to use them?
- Conclusions



📕 🖪 OntoManag	er @ Merc 🗙 🕻 🏞 Extension	s		×													8
← → C fi	localhost:8080/sys	?sho	w=de	esign;	qnar	ne=co	over_	sys:p	anelD	esigr	1						☆ =
		On	t∩M	lan	ade	or 6	M 6	erc	ato	rТ	مام	sco	ne				
			tor	an	uge		9 14	CIC	ato	•	CIC.	500	pe				
٥	Ontologies Dataset	P	robler	ns	Brow	wse	Qu	ery	Sys	tems	M	1echar	nics	Ele	ctroni	cs Software	
+ Cover + M1	Design panelD	)esig	gn														
+ M3 + Telemetry	The design of the tel	escop	e cove	r pane	ls												
+ Timing																	
	Requirements deriva	tion m	natrix														
					panel	Design	1					con	cept				
					itus		sn										
				-	orsta	<u> </u>	kStat										
				moveActuator	moveActuatorStatus	absFeedback	absFeedbackStatus		Ę	ting		controllable	monitorable	ity			
		en	closed	veA	veA	sFee	sFee	locking	aluminum	obstructing	sealing	ntrol	nito	reliability	weight		
		open	cle	ŭ	ŭ	ab	ab		alı	ob	se	CO	ŭ	re	we		
	open									~							
	closed										~						
	moveActuator				~							~					
	moveActuatorStatus			~									~				
	absFeedback						~						~				
	absFeedbackStatus					2							~				
	The status of ab	solute	feedb	ack sl	all be	know	n							~			
	aluminum														×	]	

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



		C	IntoMar	nager @	Merc	ator le	lescope			
	Ontologies	Dataset	Problems	Browse	Query	Systems	Mechanics	Electronics	Software	
+ Cover + M1	Requirer	nent <mark>a</mark> l	osFeedba	ckStatus						
+ M3 + Telemetry	The status	of the abso	lute feedback s	hall be known						
* Timing	Properties									
	Derives					]				
	Derived from	n ← cove	r_sys:concept.re	quirements.moni	itorable					
		← cove	r_sys:panelDesig	n.requirements.	absFeedback					
	Satisfied by	• cove	er_sys:panelDesi	gn.parts.encoder						
		• cove	er_elec:slot3							
	Declared by	• cove	er_sys:panelDesi	gn						
						_				

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



🖉 🖸 OntoManage	er @ Merc 🗙 📃								Ŀ	<u>م</u>
<rp>A ⇒ C A</rp>	🗋 localhost:8080,	/elec?show=loMod	uleInstance;«	qname=	cover_elec:s	lot3			5	≡
1		OntoMan	ager @	Merc	ator Te	lescope				
٥	Ontologies Data		-	Query	Systems	Mechanics	Electronics	Software		
Cover  Cover  VO modules  + slot0 + slot1	I/O Modu	le instance slo	t3			nels				
- + slot2 - + slot3 - + slot4 - + slot5	System propert Satisfies cov	<b>ies</b> er_sys:panelDesign.requi	rements.absFeed	lbackStatus						
- + slot6 - + slot7	Module type su	mma <b>ry</b>								
— 🕂 slot8 — 🕂 slot9	ID	EL1088								
- + slot10	Manufacturer	Beckhoff Automation								
- + slot11	Description	8-channel digital input t	erminal 24V DC,	negative sv	witching					
— 🕂 slot12	Used in	Cover (1), M3 (1)								
+ slot13 - terminals + PE + L + L + N + 24V + GND - connectors + ECAT - + T1 + T2	Signal LED1 Signal LED3 Signal LED5 Signal LED7 Input 1 Power contact +24 V		D4 D6	•						

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



🖉 🖸 OntoManage	er @ Merc × 📃								8
← → C fi	localhost:8080/	elec?show=loMod	uleInstance;q	name=cover	_elec:slot3				☆ =
		OntoMan	ager @	Mercato	or Teles	scope			
٩	Ontologies Data	set Problems	Browse (	Query Sys	tems Me	echanics	Electronics	Software	
Cover Cover Cover	I/O Modul	e instance <mark>slo</mark>	t3						
— (+) slot0 — (+) slot1	Digital input t	erminal to read the sta	tus of the SSI er	ncoders of all 8	over panels				
- + slot2 - + slot3 - + slot4	System propert	es							
- + slot5	Satisfies cov	er_sys:panelDesign.requi	rements.absFeedb	ackStatus					
—	Module type su	nmary							
— 🛨 slot8 — 🛨 slot9	ID	EL1088							
- + slot10	Manufacturer	Beckhoff Automation			_				
- 🕂 slot11	Description		erminal 24V DC, n	negative switching	_				
- + slot12	Used in	Cover (1), M3 (1)							
+ slot13 - = terminals + + PE + L + N + 24V	Signal LED1 Signal LED3 Signal LED5 Signal LED7	→ 07 08 → Signal LI → Signal LI → Signal LI → Signal LI	D6						
GND Gonnectors Connectors	Input 1		↓ ↓ + • *						
- ± T1 - ± T2	Input 3 Power contact +24 V	$\rightarrow$ $\bigcirc$ $\bigcirc$ $\leftarrow$ Input 4							

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

WEB3005



51

## Electrical design

		C	OntoMar	nager @	Merc	ator Te	elescope			
9	Ontologies	Dataset	Problems	Browse	Query	Systems	Mechanics	Electronics	Software	
Company	y Beckhoff									
compan	y Deckhon									
Logo										
DEC										
BEL	KHOF									
Summary										
Summary Short name	Beckhoff									
-	Beckhoff Beckhoff Automatic	on								
Short name			panels,							
Short name	Beckhoff Automatic		panels,							
Short name Long name Description	Beckhoff Automatic		panels,							
Short name Long name Description	Beckhoff Automatic	Cs, I/O, control	panels,				Used in			
Short name Long name Description Products	Beckhoff Automatic	Cs, I/O, control				over (1), M1 (1	Used in ), M3 (1), Telemetr	y (1), Timing (1)		
Short name Long name Description Products ID	Beckhoff Automatic Produces IPCs, PLC	Cs, I/O, control De with ID switch	scription	witching		over (1), M1 (1	), M3 (1), Telemetr	y (1), Timing (1)		
Short name Long name Description Products ID EK1101	Beckhoff Automatic Produces IPCs, PLC	Cs, I/O, control De with ID switch put terminal 24	scription V DC, negative st	witching	C		), M3 (1), Telemetr	y (1), Timing (1)		
Short name Long name Description Products ID EK1101 EL1088	Beckhoff Automatic Produces IPCs, PLC EtherCAT Coupler 8-channel digital in	Cs, I/O, control De with ID switch uput terminal 241 utput terminal 24	scription V DC, negative so 4V DC	witching	с с	over (1), M3 (1	), M3 (1), Telemetr	y (1), Timing (1)		
Short name Long name Description Products EK1101 EL1088 EL2008	Beckhoff Automatic Produces IPCs, PLC EtherCAT Coupler 8-channel digital in 8-channel digital ou 4-channel digital ou	De with ID switch put terminal 24 utput terminal 24 utput terminal 24	scription V DC, negative so 4V DC 24 V DC, 2 A	witching	с с м	over (1), M3 (1 over (1)	), M3 (1), Telemetr	y (1), Timing (1)		
Long name Description Products EK1101 EL1088 EL2008 EL2024	Beckhoff Automatic Produces IPCs, PLC EtherCAT Coupler 8-channel digital in 8-channel digital ou 4-channel digital ou	De with ID switch put terminal 24 utput terminal 24 utput terminal 24	scription V DC, negative so 4V DC 24 V DC, 2 A	witching	с с м	over (1), M3 (1 over (1)	), M3 (1), Telemetr	y (1), Timing (1)		
Short name Long name Description Products ID EK1101 EL1088 EL2008 EL2024	Beckhoff Automatic Produces IPCs, PLC EtherCAT Coupler 8-channel digital in 8-channel digital ou 4-channel digital ou	De with ID switch uput terminal 24 utput terminal 24 utput terminals 24 utput terminals 24 utput terminals 24 utput terminals 24	scription V DC, negative so 4V DC 24 V DC, 2 A 5 V DC		с с м м	over (1), M3 (1 over (1)	), M3 (1), Telemetr ) )	y (1), Timing (1)		

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



## Electrical design

OntoManager @ Merc × 👍 Extensions ×								
← → C 1	localhost:8080/org?show=Company;qname=beckho	off:company						
TID	Description	Used in						
EK1101	EtherCAT Coupler with ID switch	Cover (1), M1 (1), M3 (1), Telemetry (1), Timing (1)						
EL1088	8-channel digital input terminal 24V DC, negative switching	Cover (1), M3 (1)						
EL2008	8-channel digital output terminal 24V DC	Cover (1)						
EL2024	4-channel digital output terminals 24 V DC, 2 A	M1 (1)						
EL2124	4-channel digital output terminals 5 V DC	M1 (2)						
EL2622	2-channel relay	Cover (5), M1 (1)						
EL3024	4-channel analog input terminals 420mA, differential inputs, 12 bit	M1 (2), Telemetry (1)						
EL3102	2-channel analog input terminals -10+10 V, differential input, 16 bit	M1 (1)						
EL3164	4-channel analog input terminal 010 V, single-ended, 16 bit	M1 (1)						
EL3202-0010	2-channel input terminals PT100 (RTD) for 4-wire connection, high-precision	Telemetry (7)						
EL3351	1-channel resistor bridge terminal (strain gauge)	M1 (3)						
EL3681	Digital multimeter	Cover (1)						
EL4008	8-channel analog output terminal 010V, 12 bit	Cover (1)						
EL4022	2-channel analog output terminal 420 mA, 12 bit	M1 (1)						
EL5001	1-channel SSI encoder	M1 (1), M3 (1)						
EL5002	2-channel SSI encoder	Cover (4)						
EL5101	1-channel incremental encoder	M1 (1), M3 (1)						
EL6001	RS-232 serial communication	Timing (1)						
EL6688	IEEE 1588 external synchronisation interface	Timing (1)						
EL6751	CANopen master/slave controller	M3 (1)						
EL9070	Shield terminal	Telemetry (2)						
EL9186	Potential distribution terminal, 8 x 24V	M1 (1), Telemetry (1)						
EL9187	Potential distribution terminal, 8 x 0V	M1 (2), Telemetry (1)						
EL9410	Power supply terminals for E-bus (with diagnostics)	M1 (1)						
EL9505	Power supply terminals 5 V	M1 (1)						

E

☆ 〓

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



Cover     C	OntoManage	r @ Merc 🗙 📃								A
Ontologies       Dataset       Problems       Browse       Query       Systems       Mechanics       Electronics       Software         Cover <ul> <li> <i>P</i> or modules</li> <li> <i>P</i> stord</li> <li> <i>P</i> stord&lt;</li></ul>	⇒ C' ff	🗋 localhost:8080	/elec?show=loMod	uleInstance;	qname=	cover_elec:s	lot3			☆ =
Ontologies       Dataset       Problems       Browse       Query       Systems       Mechanics       Electronics       Software         Cover       F of oncodies       F of oncodies       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_systemalDesign.requirements.absFeedbackStatus         Module type summary       D       EL088         Satisfies       cover_locare       Beckhoff Automation         Description       8-channel digital input terminal 24V DC, negative switching         Used in       Cover (1), M3 (1)         Signal LED1       Signal LED2       Signal LED3         Signal LED3       Signal LED3       Signal LED4         Signal LED3       Signal LED4       Signal LED4         Signal LED3       Signal LED5       Signal LED4         Signal LED5       Signal LED5       Signal LED5         Signal LED2       Signal LED4       Signal LED5         Signal LED5       Signal LED5       Signal LED5	1		OntoMan	ager @	Merc	ator Te	elescope			
Image: Promodules       Image: Promodules         Image: Promodules       Digital input terminal to read the status of the SSI encoders of all 8 cover panels         Image: Promodules       System properties         Image: Promodule status       System properties         Image: Promodule status       Satisfies         Image: Promodule status       System properties         Image: Promodule status       Satisfies         Image: Promodule status       Satisfies         Image: Promodule status       Satisfies         Image: Promodule status       Satisfies         Image: Promodule status       Signal LED3	)	Ontologies Data		-				Electronics	Software	
Image: Digital input terminal to read the status of the SSI encoders of all 8 cover panels         Image: Digital input terminal to read the status of the SSI encoders of all 8 cover panels         Image: Digital input terminal to read the status of the SSI encoders of all 8 cover panels         Image: Digital input terminal to read the status of the SSI encoders of all 8 cover panels         Image: Digital input terminal to read the status of the SSI encoders of all 8 cover panels         Image: Digital input terminal to read the status of the SSI encoders of all 8 cover panels         Image: Digital input terminal to read the status of the SSI encoders of all 8 cover panels         Image: Digital input terminal to read the status of the SSI encoders of all 8 cover panels         Image: Digital ED3       Satisfies         Image: Digital ED3       Signal ED4         Image: Digital ED5       Signal ED4         Image: Digital ED5       Signal ED5         Image: Digital ED5       Signal ED5         Image: Digital ED5       Signal ED5         Image: Digital ED5       Signal ED6         Image: Digital ED5       Signal ED6         Image: Digital ED5       Signal ED5         Image: Digital ED5       Signal ED5         Image: Digital ED5       Signal ED6         Image: Digital ED5       Signal ED6         Image: Digital ED5       Signal ED6	- I/O modules	I/O Modu	le instance <mark>slo</mark>	t3						
- + slot3       System properties         + slot4       - f slot5         + slot5       Satisfies         - slot7       Module type summary         - + slot8       ID         - + slot8       ID         - + slot9       Manufacturer         - + slot10       Description         - + slot11       Description         - + slot12       Used in         - + slot13       cover (1), M3 (1)         - + slot13       signal LED2         - + slot14		Digital input f	terminal to read the sta	tus of the SSI	encoders of	f all 8 cover pa	nels			
Image: solution of the solution	- + slot2 - + slot3 System properties									
Module type summary   + slot3   + slot3   + slot10   - + slot11   - + slot12   - + slot13     - + slot13     - + pe   - + pe   - + pe   + l   + - + l   + - + - + - + - + - + - + - + - +		Satisfies cov	/er_sys:panelDesign.requi	rements.absFee	dbackStatus					
ID       EL1088         Manufacturer       Beckhoff Automation         Description       8-channel digital input terminal 24V DC, negative switching         Used in       Cover (1), M3 (1)         + slot13       -         - + pe       Signal LED1         - + pe       Signal LED3         - + N       Signal LED5         - + N       Signal LED5         - + N       Signal LED6         - + R       Signal LED7         - + N       Signal LED7         - + N       Signal LED3         - + R       Signal LED2         - + N       Signal LED3         - + N       Signal LED4         - + N       Signal LED5         - + N       Signal LED7         - + N       Signal LED7         - + N       Signal LED8         - + N       Signal LED7         - + N       - +	— 🕂 slot7	Module type su	mmary							
Imanufacturer       Beckhoff Automation         Description       8-channel digital input terminal 24V DC, negative switching         Used in       Cover (1), M3 (1)         Imaufacturer       Signal LED1         Imaufacturer       Signal LED1         Imaufacturer       Signal LED1         Imaufacturer       Signal LED1         Imaufacturer       Signal LED2         Imaufacturer       Signal LED3         Imaufacturer       Imaufacturer         Imaufacturer       Imaufacturer         Imaufacturer       Imaufacturer         Imaufacturer       Imaufacturer         Imaufacturer       Imaufacturer         Imaufacturer       Imaufacturer         Imaufacturer		ID	EL1088							
Used in Cover (1), M3 (1) Used in Cover (1), M3 (1) Used in Cover (1), M3 (1) Used in Cover (1), M3 (1) Signal LED1 Signal LED2 Signal LED3 Signal LED5 Signal LED5 Signal LED5 Signal LED7 Input 1 Cover (1), M3 (1) Used in Cover (1), M3 (1) Signal LED2 Signal LED2 Signal LED3 Signal LED4 Signal LED8 Used in Cover (1), M3 (1) Used in Cover (1), M3 (1) Used in Cover (1), M3 (1) Signal LED2 Signal LED3 Signal LED3 Signal LED4 Signal LED8 Used in Cover (1), M3 (1) Used in Cover (1), M3 (1) Used in Cover (1), M3 (1) Signal LED3 Signal LED3 Signal LED4 Signal LED5 Signal LED8 Signal LED8 Signa		Manufacturer	Beckhoff Automation							
Image: solution of the second seco		Description	8-channel digital input t	erminal 24V DC,	, negative sv	vitching				
Imput 1       Imput 1       Imput 2		Used in	Cover (1), M3 (1)							
$ \begin{array}{ c c c c c c c c c c c c c c c c c c c$										
Signal LED3 Signal LED5 Signal LED5 Signal LED7 + N + 24V + GND Input 1 - connectors Signal LED3 Signal LED5 Signal LED5 - Signal LED6 Signal LED8 - Input 2 - Input 2			07 08							
$ \begin{array}{ c c c c c c c c c c c c c c c c c c c$		Signal LED1 Signal LED3	Signal Li	D2 D4						
□ connectors Input 1 Input 2 ↓ □ connectors		Signal LED5	Signal Li	D8						
connectors	- + 24V		<b>II I2</b>	[±±±]						
	+ GND	Input 1	🖥 💧 🔔 Input 2		+[Z]h					
			13 14							
$- \div \text{ECAT}$ $- \div \text{Input 3} \longrightarrow 0 \oplus \leftarrow \text{Input 4} \qquad $		Input 3								
$ \begin{array}{c c} - \begin{array}{c} + & T_1 \\ \hline & Power \ contact \\ - \begin{array}{c} + & T_2 \end{array} \end{array}  \begin{array}{c} 0 \\ + & 24 \\ \end{array}  \begin{array}{c} 0 \\ + & 75 \\ \end{array}  \begin{array}{c} 0 \\ + & 75 \\ \end{array}  \begin{array}{c} 0 \\ - & 1 \\ \end{array}  \begin{array}{c} 0 \\ \end{array}  \begin{array}{c} 0 \\ - & 1 \\ \end{array}  \begin{array}{c} 0 \end{array}  \begin{array}{c} 0 \\ \end{array}  \begin{array}{c} 0 \end{array}  \begin{array}{c} 0 \\ \end{array}  \begin{array}{c} 0 \end{array}  \begin{array}{c}$		Power contact								

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



### **Electrical design**

숤

Ξ

#### 互 OntoManager @ Merc 🗙 🔽

→ C 🖌 🗋 localhost:8080/elec?show=loModuleInstance;qname=cover\_elec:slot3

M1
 M3
 Telemetry
 Timing

- D4

-

BECKHUFF	1
diama in the second second	1

#### Connections

	Type (	EL1088)			Instance	
Channel	Terminal	Symbol	Description	Symbol	Description	Connected to
1	1	I1	Input 1	TC:T1:SSISTS	Top 1 SSI encoder status	Connector T1 : pin 13
2	2	12	Input 2	TC:T2:SSISTS	Top 2 SSI encoder status	Connector T2 : pin 13
3	3	13	Input 3	TC:T3:SSISTS	Top 3 SSI encoder status	Connector T3 : pin 13
4	4	I4	Input 4	TC:T4:SSISTS	Top 4 SSI encoder status	Connector T4 : pin 13
5	5	15	Input 5	TC:B1:SSISTS	Bottom SSI encoder status	Connector B1 : pin 13
6	6	16	Input 6	TC:B2:SSISTS	Bottom SSI encoder status	Connector B2 : pin 13
7	7	17	Input 7	TC:B3:SSISTS	Bottom SSI encoder status	Connector B3 : pin 13
8	8	18	Input 8	TC:B4:SSISTS	Bottom SSI encoder status	Connector B4 : pin 13

#### Interface

Variable	Туре	Description	Linked variable
input1	BOOL	Input 1	interface.parts.cover.parts.top.parts.pt .encoderErrorSignal
input2	BOOL	Input 2	interface.parts.cover.parts.top.parts.p
input3	BOOL	Input 3	interface.parts.cover.parts.top.parts.p3.encoderErrorSignal
input4	BOOL	Input 4	interface.parts.cover.parts.top.parts.p4.encoderErrorSignal
input5	BOOL	Input 5	interface.parts.cover.parts.bottom.parts.p1.encoderErrorSignal
input6	BOOL	Input 6	interface.parts.cover.parts.bottom.parts.p2.encoderErrorSignal
input7	BOOL	Input 7	interface.parts.cover.parts.bottom.parts.p3.encoderErrorSignal
input8	BOOL	Input 8	interface.parts.cover.parts.bottom.parts.p4.encoderErrorSignal
WcState	BOOL	EtherCAT Working counter state	interface.parts.cover.parts.io.parts.slot3.wcState
InfoDataState	UINT	EtherCAT state (INIT, PREOP, OP,)	interface.parts.cover.parts.io.parts.slot3.infoData

• What are semantic models?

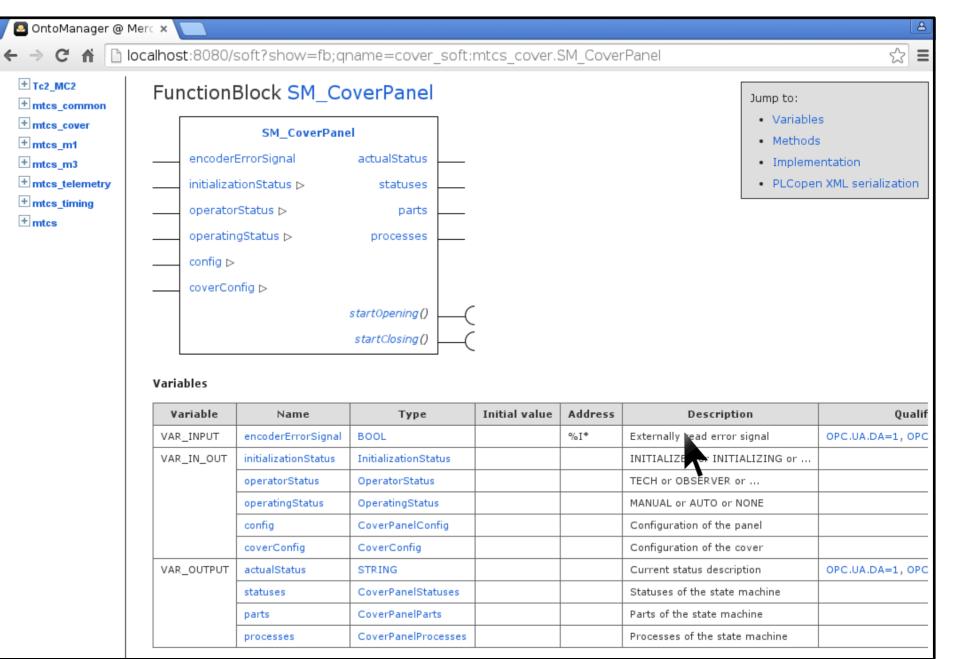
- Where to apply them?
- How to apply them?
- How to build them?
- How to use them?
- Conclusions



🛛 🖸 OntoMana	ger @ Merc ×									<u> </u>
← ⇒ C f	localho	st:8080/s	soft?show=fb;qi	name=cover_	soft:n	ntcs_cover.S	SM_Cove	rPanel		☆ =
			OntoMa	nager @	Me	ercator	Teles	cope		
•	Ontologies	Datas		Browse	Quer			chanics Electron	iics Sof	tware
<ul> <li>Tc2_MC2</li> <li>mtcs_comm</li> <li>mtcs_cover</li> <li>mtcs_m1</li> <li>mtcs_m3</li> <li>mtcs_telem</li> <li>mtcs_timing</li> <li>mtcs</li> </ul>	etry	encoderi initializat operator	nfig ⊳						Jump to: • Variable • Method: • Impleme • PLCoper	5
		ariable	Name	Туре		Initial value	Address	Descriptio		Qualif
		R_INPUT	encoderErrorSignal	BOOL		initial value	%I*	Externally read error s		OPC.UA.DA=1, OPC
		R_IN_OUT	initializationStatus	InitializationState	us			INITIALIZED or INITIA	-	
			operatorStatus	OperatorStatus				TECH or OBSERVER or		
			operatingStatus	OperatingStatus				MANUAL or AUTO or N	ONE	
			config	CoverPanelConfi	ig			Configuration of the pa	anel	
			coverConfig	CoverConfig				Configuration of the co	ver	
	VAR	R_OUTPUT	actualStatus	STRING				Current status descript	tion	OPC.UA.DA=1, OPC

- 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?
- Where to apply them?
- How to apply them?
- How to build them?
- How to use them?
- Conclusions



## Software design

🔼 OntoManager @ Mer	<ul> <li>×</li> </ul>						<u> </u>
← → C 🖌 🗋 loca	alhost:8080/s	soft?show=fb;qr	name=cover_soft:	mtcs_cover.SM	_Cove	rPanel	☆ =
	VAR_INPUT	encoderErrorSignal	BOOL	%	·I*	Externally read error signal	OPC.UA.DA=1, OPC
	VAR_IN_OUT	initializationStatus	InitializationStatus			INITIALIZED or INITIALIZING or	
		operatorStatus	OperatorStatus			TECH or OBSERVER or	
		operatingStatus	OperatingStatus			MANUAL or AUTO or NONE	
		config	CoverPanelConfig			Configuration of the panel	
		coverConfig	CoverConfig			Configuration of the cover	
	VAR_OUTPUT	actualStatus	STRING			Current status description	OPC.UA.DA=1, OPC
		statuses	CoverPanelStatuses			Statuses of the state machine	
		parts	CoverPanelParts			Parts of the state machine	
		processes	CoverPanelProcesses			Processes of the state machine	

#### Methods

startOpening()

Comment	Start openi	Start opening the panel								
Return type	RequestRe	equestResults								
Interface	Variable	Variable Name Type Initial value Address Description Qualifiers								
Implementation	start0pen	<pre>startOpening := THIS^.processes.startOpening.request();</pre>								

#### startClosing()

Comment	Start closin	Start closing the panel								
Return type	RequestRe	equestResults								
Interface	Variable	Variable Name Type Initial value Address Description Qualifiers								
Implementation	startClos	<pre>startClosing := THIS<sup>^</sup>.processes.startClosing.request();</pre>								

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



### Software design

### 🖪 OntoManager @ Merc 🗙 🔽

← → C f localhost:8080/soft?show=fb;qname=cover\_soft:mtcs\_cover.SM\_CoverPanel

implementation | startOpening := (Hist, processes, startOpening, request();

### <u>ک</u>

#### startClosing()

Comment	Start closin	Start closing the panel								
Return type	RequestRe	RequestResults								
Interface	Variable	Variable Name Type Initial value Address Description Qualifiers								
Implementation	startClos	<pre>startClosing := THIS^.processes.startClosing.request();</pre>								

#### Implementation

#### parts.axis( isEnabled := operatorStatus.tech AND (operatingStatus.manual AND initializationStatus.initialized), standstillTolerance := config.standstillTolerance); parts.motorRelay( isEnabled := parts.axis.isEnabled ); statuses.busyStatus( isBusy := parts.axis.statuses.busyStatus.busy OR parts.motorRelay.statuses.busyStatus.busy ); statuses.apertureStatus( isOpen := (ABS(config.openPosition - parts.axis.actPos.degrees.value)) < config.openTolerance, isClosed := (ABS(config.closedPosition - parts.axis.actPos.degrees.value)) < config.closedTolerance);</pre> statuses.healthStatus( isGood := parts.axis.statuses.healthStatus.isGood AND (NOT(encoderErrorSignal)) hasWarning := parts.axis.statuses.healthStatus.hasWarning); statuses.openingStatus( isOpening := parts.axis.statuses.motionStatus.backward, isClosing := parts.axis.statuses.motionStatus.forward); processes.startOpening( isEnabled := operatorStatus.tech AND (operatingStatus.manual AND initializationStatus.initialized) processes.startClosing( isEnabled := operatorStatus.tech AND (operatingStatus.manual AND initializationStatus.initialized)

#### PLCopen XML serialization

1 -	<pre>kpou name="SM_CoverPanel" pouType="functionBlock"&gt;</pre>
2 -	<interface></interface>
	<inputvars></inputvars>
4 -	<variable address="%I*" name="encoderErrorSignal"></variable>
5	<type><bool></bool></type>
6 -	<adddata></adddata>

• What are semantic models?

- Where to apply them?
- How to apply them?
- How to build them?
- How to use them?
- Conclusions



🗾 OntoManag	er @ Merc ×									<u> </u>
← → C fi	🗋 localhos	st:8080/s	soft?show=fb;qr	name=cover_s	soft:m	tcs_cover.S	GM_Cover	rPanel		☆ =
			OntoMa	nager @	Ме	rcator	Teles	соре		
٥	Ontologies	Datas	set Problems	Browse	Query	System	ns Med	chanics Electroni	cs Soft	ware
<ul> <li>Tc2_MC2</li> <li>mtcs_common</li> <li>mtcs_cover</li> <li>mtcs_min</li> <li>mtcs_telement</li> <li>mtcs_timing</li> <li>mtcs</li> </ul>		encoderi initializat operatori operatin config > coverCo	nfig Þ		(				Jump to: • Variable • Methods • Impleme • PLCoper	
	v	ariable	Name	Туре	1	(nitial value	Address	Description	n	Qualif
	VAR	_INPUT	encoderErrorSignal	BOOL			%I*	Externally read error si		OPC.UA.DA=1, OPC
	VAR	IN_OUT	initializationStatus	InitializationStatu	ıs			INITIALIZED or INITIA	IZING or	
			operatorStatus	OperatorStatus				TECH or OBSERVER or		
			operatingStatus	OperatingStatus				MANUAL or AUTO or NO	NE	
			config	CoverPanelConfi	g			Configuration of the pa	nel	
			coverConfig	CoverConfig				Configuration of the co	ver	
	VAR	_OUTPUT	actualStatus	STRING				Current status descripti	on	OPC.UA.DA=1, OPC

• What are semantic models?

- Where to apply them?
- How to apply them?
- How to build them?
- How to use them?
- Conclusions



📕 🖪 OntoManager @	Merc 🗙 📃								<u> </u>
← ⇒ C ⋒ 🗋	localhost:8080/soft	?show=library	/;qname=c	over_soft	mtcs_cover				☆ =
		OntoMar	ager @	Mer	ator Te	lescope			
•									
Oni Oni	tologies Dataset	Problems	Browse	Query	Systems	Mechanics	Electronics	Software	
+ Tc2_MC2 + mtcs_common + mtcs_cover	Library mtc	s_cover							
+ mtcs_m + mtcs_m3	PLCopen XML seria	alization							
t mtcs_telemetry	File	/lhome/wimpe/w	ork/onto/onton	nanager/env/	OntoManager/ont	tomanager/genera	ted/mtcs_cover.xm	1	
+ mtcs_timing	Status	File has been rea	d						
+ mtcs	Code generation	Not running							
	Generate PLCop	en XML Dow	vnload PLCop	en XML					
	PyUAF serialization	n							
	File	/lhome/wimpe/w	ork/onto/onton	nanager/env/	OntoManager/ont	tomanager/genera	ted/pyuaf/mtcs_cov	/er.py	
	Status	File has been rea	d						
	Code generation	Not running							
	Generate pyUAF	code Downl	oad pyUAF co	ode					
	File contents:								
	2 - <project x<br="">3 <filehea 4 - <content 5 - <coord 6 - <fbd< th=""><th>ion="1.0" encodi mlns="http://www der companyName= Header name="mto inateInfo&gt;</th><th>/.plcopen.org "Institute o s_cover" moo</th><th>)/xml/tc6_02 of Astronomy</th><th>productName</th><th>="OntoManager"   10-09T11:23:49.</th><th>productVersion=" 793320"&gt;</th><th>0.0.1" creatior</th><th>nDateTime='</th></fbd<></coord </content </filehea </project>	ion="1.0" encodi mlns="http://www der companyName= Header name="mto inateInfo>	/.plcopen.org "Institute o s_cover" moo	)/xml/tc6_02 of Astronomy	productName	="OntoManager"   10-09T11:23:49.	productVersion=" 793320">	0.0.1" creatior	nDateTime='

• What are semantic models?

- Where to apply them?
- How to apply them?
- How to build them?
- How to use them?
- Conclusions



🖊 🖪 OntoManager	@ Merc ×								٨
← ⇒ C' fi	🗋 localhost:8080/sof	t?show=library;	qname=co	over_soft:	mtcs_cover				☆ =
		OntoMana	ager @	) Mero	ator Te	elescope			
<b>6</b> c	Ontologies Dataset	Problems	Browse	Query	Systems	Mechanics	Electronics	Software	
+ Tc2_MC2 + mtcs_common + mtcs_cover	Library mto	cs_cover							
+ mtcs_m	PLCopen XML ser	ialization							
+ mtcs_m3 t mtcs_telemetry	File	/lhome/wimpe/wor	k/onto/ontom	anager/env/	OntoManager/on	tomanager/genera	ted/mtcs_cover.xm	I	
+ mtcs_timing	Status	File has been read							
+ mtcs	Code generation	Not running							
	Generate PLCo	oen XML Down	load PLCope	en XML					
	PyUAF serializati	on							
	File	/lhome/wimpe/wor	k/onto/ontom	anager/env/	OntoManager/on	tomanager/genera	ted/pyuaf/mtcs_cov	ver.py	
	Status	File has been read							
	Code generation	Not running							
	Generate pyUA	code Downlo	ad pyUAF co	de					
	File contents:								
	2 → <project 3 <filehe 4 → <conten 5 → <coor 6 → <fb< th=""><th>sion="1.0" encodin xmlns="http://www. ader companyName=" tHeader name="mtcs dinateInfo&gt;</th><th>g="utf-8"?&gt; plcopen.org, Institute o _cover" mod:</th><th>/xml/tc6_02 f Astronomy</th><th>" productName</th><th>="OntoManager" 10-09T11:23:49.</th><th>productVersion=" 793320"&gt;</th><th>0.0.1" creation</th><th>DateTime='</th></fb<></coor </conten </filehe </project 	sion="1.0" encodin xmlns="http://www. ader companyName=" tHeader name="mtcs dinateInfo>	g="utf-8"?> plcopen.org, Institute o _cover" mod:	/xml/tc6_02 f Astronomy	" productName	="OntoManager" 10-09T11:23:49.	productVersion=" 793320">	0.0.1" creation	DateTime='

• What are semantic models?

Readv

- Where to apply them?
- How to apply them?
- How to build them?
- How to use them?



#### Software design MTCS - Microsoft Visual Studio File Edit View Project Build Debug TwinCAT PLC Tools Scope Window Help : 🛅 • 🖽 • 🌈 🔲 🦼 👗 🐴 🛝 🤟 • 🔍 • 🗷 • 🔍 🕨 Release verifyOperatorPassword - | 🟹 🗄 🗄 TwinCAT RT (x86) - 080 🗄 🚟 🚟 📥 🚽 🔛 🚨 🗖 🥔 🔨 🎯 🚮 🐛 🛛 MTCSPLC \_\_\_\_\_ MTCS Solution Explorer CoverApertureProcedure × • 🗜 🗙 FUNCTION BLOCK CoverApertureProcedure EXTENDS SM\_CoverApertur -VAR INPUT Solution 'MTCS' (1 project) END VAR MTCS VAR OUTPUT SYSTEM END VAR A MOTION VAR D PLC ..... MTCS update(); MTCS Project External Types // run the procedure References CASE state OF DUTs CoverApertureProcedureStates.IDLE: GVLs idle(); MTCS CoverApertureProcedureStates.ABORTED: Common aborted(); Cover CoverApertureProcedureStates.PREPARE PROCESS: prepareProcess(); Generated Add CoverApertureProcedureStates.ENABLING RELAYS: A D StateMac Cover enablingRelays(); Import PLCopenXML... coverApertureProcedureStates.ENABLING MOTORS: Cover Export PLCopenXML... enablingMotors(); Cover Ctrl+X overApertureProcedureStates.OPENING TOP PANELS: Cover Cut openingTopPanels(); Ctrl+C 🔄 Cover 🗎 Copy overApertureProcedureStates.OPENING BOTH PANELS: X Delete Visu Del \_openingBothPanels(); Widgets Rename overApertureProcedureStates.ENABLING MAGNETS: 🛅 M1 enchlingMagnete() Properties Alt+Enter 🛅 M3 • 4 × ETTOT LISU MTCS 🙆 0 Errors 🔥 0 Warnings 🕕 0 Messages 🛛 Clear 🛅 Telemetry 📸 Error List 🔳 Output b image: Diminal Di

• What are semantic models?

or

Ready

- Where to apply them?
- How to apply them?
- How to build them?
- How to use them?



MTCS - Microsoft Visual Studio		
File Edit View Project Build Debug Tv	winCAT PLC Tools Scope V	Vindow Help
🛅 • 🖽 • 📂 😸 🍠 🗼 🗛 🖦 🗠 🔊 •	(° - 🖉 - 🖳 🕨 Release	🔹 TwinCAT RT (x86) 🔹 🧭 verifyOperatorPassword 🔹 🔩 🚆
	MTCSPLC	▼ -: MTCS
Solution Explorer	▼ 🕂 × CoverAper	tureProcedure ×
	1	FUNCTION BLOCK CoverApertureProcedure EXTENDS SM_CoverApertur
Solution 'MTCS' (1 project)	▲ 2 3	VAR_INPUT END VAR
MTCS	4	VAR OUTPUT
SYSTEM	5	END VAR
MOTION	6	VAR
	▲	
<ul> <li>Image: A market of the second s</li></ul>	= 1	_update();
Emiliar External Types	2	
References	3	// run the procedure CASE state OF
DUTs 🔁		CoverApertureProcedureStates.IDLE:
b GVLs	6	_idle();
MTCS	. 7	CoverApertureProcedureStates.ABORTED:
Common	8	_aborted();
⊿ 🦻 Cover	E 9	CoverApertureProcedureStates.PREPARE_PROCESS:
Generated A	Add	<pre>coverApertureProcedureStates.ENABLING_RELAYS:</pre>
	mport PLCopenXML	enablingRelays();
	Export PLCopenXML	CoverApertureProcedureStates.ENABLING_MOTORS:
Cover 👘		enablingMotors();
📑 Cover 🎽 C	Cut Ctrl+)	
	Copy Ctrl+	openingTopPanels(); CoverApertureProcedureStates.OPENING_BOTH_PANELS:
	Delete Del	openingBothPanels();
	Rename	CoverApertureProcedureStates.ENABLING_MAGNETS:
⊳ 🚞 M1 ⊳ 🛅 M3 🛛 🖶 Р	Properties Alt+E	nter v
▷ <u> </u>		• ₽ ×
D In CS D Telemetry	🛛 🙆 0 Error	rs 🔥 0 Warnings 🕕 0 Messages 🛛 Clear
D in Timing		ist 🗐 Output

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



👓 MTCS - Micros	soft Visual Studio					e X
File Edit View	Project Build Debug	TwinCAT PLC Tools Scope Window	w Help			
i 🛅 • 💷 • 📂	🚽 🦪 👗 🐴 🛍 🔊	- (≌ - ,⊒ - 🖳   ▶ Release -	TwinCAT RT (x86) 🔹 🚺	🆄 verifyOperatorPassw	vord 👻 🖣	₹11
: 🕮 🛗 🛓	i 🖈 🧧 🗖 🌮 📉 🎯	😚 🐾 🛛 MTCSPLC 👻 🚽	MTCS -		tine (b) (a)	i 🖆 🗒
Solution Explorer		▼ ₽ × CoverAperturePr	rocedure ×			÷
			ICTION BLOCK CoverAmerture	Procedure EVTENDS	M. Coverberty	r 🔺 🗽
Solution 'M	👓 Import PLCopenXML F	ile			<b>—</b> × <b>—</b>	
🖌 📑 MTCS		Downloads	<b>- - - - + - + - + - + - + + + + + + + + + +</b>	Search Downloads	Q	
b Contraction of the second						
MO MO PLC	Organize 🔻 New fo	older		8== 🔻		-
	👓 Microsoft Visual S	Name	Date modified	Туре	Size	
4	Projects	🖉 mtcs.xml	11/09/2015 1:00	XML Document	23 KB	<u></u>
Þ		mtcs_common.xml	11/09/2015 1:00	XML Document	353 KB	
4	🔆 Favorites	🖉 mtcs_cover.xml	11/09/2015 1:00	XML Document	88 KB	
Þ	🚺 Downloads	🖉 mtcs_m1.xml	11/09/2015 1:00	XML Document	92 KB	=
4	📗 Projects –	🚽 🔊 mtcs_m3.xml	11/09/2015 1:01	XML Document	213 KB	
		🔊 mtcs_telemetry.xml	11/09/2015 1:01	XML Document	81 KB	
	🥽 Libraries	🔊 mtcs_timing.xml	11/09/2015 1:01	XML Document	58 KB	
	🚮 Git					
	🔞 Homegroup	-				
		•				
	Fil	e name: mtcs_cover.xml	•	PLCopenXML files (*.xr	nl) 🔻	
				Open 🗸 🗸	Cancel	
	MI M3	20	enshlingMagnets().			Ψ.
	MTCS	Error List				<b>-</b> ₽ ×
	D Telemetry		1 0 Warnings (1) 0 Messages	Clear		
	D Timing	👻 👸 Error List 📘	Output			

• What are semantic models?

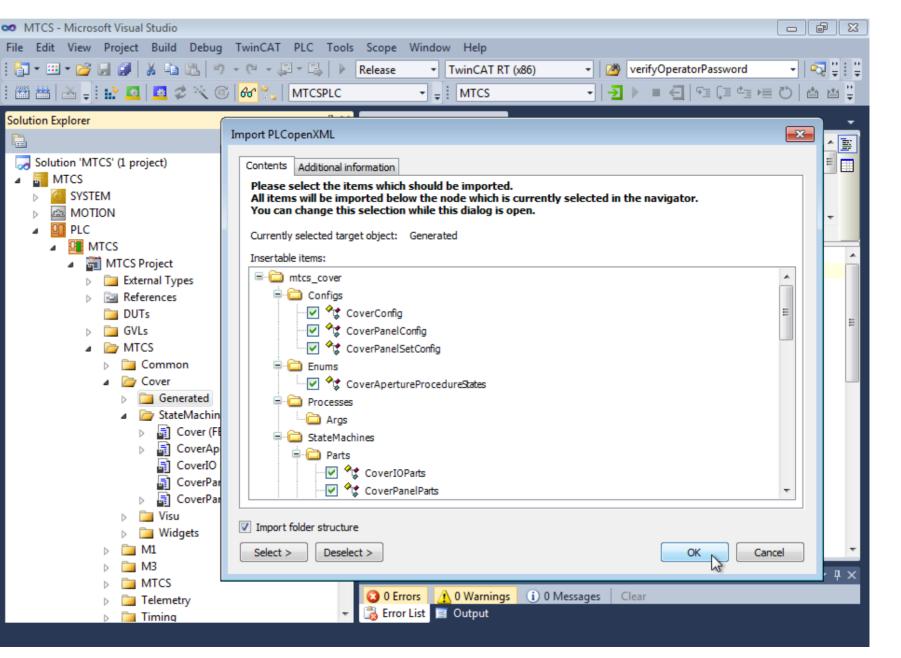
- Where to apply them?
- How to apply them?
- How to build them?
- How to use them?
- Conclusions



👓 MTCS - Micros	soft Visual Studio					di XX
File Edit View	Project Build Debug T	winCAT PLC Tools Scope	Window Help			
i 🛅 • 💷 • 💕	🚽 🥔 👗 🖦 🖄 🔊 🗸	(* - 🚑 - 🖳 🕨 Release	▼ TwinCAT RT (x86) ▼	verifyOperatorPas	sword 👻	
: 🕮 🛗 🛓	i 🖈 🖪   🖪 🤣 🖄 🎯	60 🚼 MTCSPLC	• . MTCS •	- <b>7</b> 🕨 = 🕣 🖓	,≣ de ne o   a	≛ ≝ 🖁
Solution Explorer		▼ ቑ × CoverAp	ertureProcedure ×			-
		1	FINCTION BLOCK CoverAperture	Procedure RVTRNDS	SM Coverlnert	ur 🔺 🛐
Solution 'M	👓 Import PLCopenXML File				×	
MTCS	COV kke +	Downloads	<b>−</b> +	Search Downloads	٩	
SYS MOL						
MO MO PLC	Organize 🔻 New fold	ler				-
4 🛄 I	👓 Microsoft Visual S	Name	Date modified	Туре	Size	
4	Projects	🖉 mtcs.xml	11/09/2015 1:00	XML Document	23 KB	<u> </u>
	_	mtcs_common	11/09/2015 1:00	XML Document	353 KB	
	🚖 Favorites 🧻	🖉 mtcs_cover.xml	11/09/2015 1:00	XML Document	88 KB	
D	🚺 Downloads	mtcs_m1 vml	11/09/2015 1:00	XML Document	92 KB	=
4	🌗 Projects 💷	🜌 mtcs_m3.xml	11/09/2015 1:01	XML Document	213 KB	
		💇 mtcs_telemetry.xml	11/09/2015 1:01	XML Document	81 KB	
	🛜 Libraries	💇 mtcs_timing.xml	11/09/2015 1:01	XML Document	58 KB	
	Git Git					
	🍓 Homegroup					
	<b>F</b> 21-			PLCopenXML files (*		
	File	name: mtcs_cover.xml	•	PLCopenXIVIL files (*	.xmi)	
				Open 🗸 🔻	Cancel	
				-0	h.	<b>.</b>
	M3	20	enshlingMagnete().			- 1 ×
	MTCS	Error List		L Char		<b>-</b> ₽ ×
	D Telemetry		ors 1 0 Warnings 1 0 Messages	Clear		
	D Timing					

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





• What are semantic models?

- Where to apply them?
- How to apply them?
- How to build them?
- How to use them?



#### MTCS - Microsoft Visual Studio File Edit View Project Build Debug TwinCAT PLC Tools Scope Window Help i 🛅 🕶 🐨 💕 💭 🥔 👗 🐴 🙈 🔊 🗉 🖓 🗉 🖳 🕨 Release WerifyOperatorPassword - | 🟹 🗄 🗄 TwinCAT RT (x86) 🗄 🎬 🚟 🚽 🔛 🖪 🗖 🥔 🔨 🎯 🔐 🐛 🛛 MTCSPLC - \_ MTCS Solution Explorer Import PLCopenXML × \* <u>B</u> = Solution 'MTCS' (1 project) Contents Additional information MTCS Please select the items which should be imported. SYSTEM All items will be imported below the node which is currently selected in the navigator. A MOTION You can change this selection while this dialog is open. Þ PLC Currently selected target object: Generated MTCS Insert-ue items: MTCS Project 🗀 mtcs\_cover . External Types 😑 🚞 Configs References 🗸 🔶 CoverConfig DUTs 🔽 🔩 CoverPanelConfig GVLs 🔽 🔩 CoverPanelSetConfig MTCS Common Enums Cover 12 CoverApertureProcedureStates Generate Processes 🗁 StateMaci in 🗀 Aras 🔄 Cover 🕻 StateMachines CoverAp 🚊 🚞 Parts CoverIO 🔽 🔩 CoverIOParts 🔄 CoverPar 🔽 🔩 CoverPanelParts 🔄 CoverPar Þ 🚞 Visu Import folder structure 🚞 Widgets 🛅 M1 Select > Deselect > OK Cancel M3 Ψ× MTCS 🔕 0 Errors 🔥 0 Warnings 🕕 0 Messages 🛛 Clear Telemetry $\triangleright$ 📸 Error List 🧧 Output b 🧰 Timing

• What are semantic models?

- Where to apply them?
- How to apply them?
- How to build them?
- How to use them?
- Conclusions



🖊 🖪 OntoManager @ N	Merc ×							<u>8</u>
← → C 🕯 🗋	ocalhost:8080/soft?	show=library;qr	name=cover_so	ft:mtcs_cover				☆ =
	(	OntoMana	aer @ Me	rcator Te	elescope			
Onto			-			Ele etre el es	Q = (have and	
	ologies Dataset	Problems B	srowse Query	Systems	Mechanics	Electronics	Software	
+ Tc2_MC2 + mtcs_common + mtcs_cqver	Library mtcs	_cover						
+ mtcs_m + mtcs_m3	PLCopen XML seria	lization					_	
+ mtcs_telemetry	File	/lhome/wimpe/work/	onto/ontomanager/e	nv/OntoManager/on	tomanager/genera	ted/mtcs_cover.xml		
+ mtcs_timing	Status	File has been read					_	
+ mtcs	Code generation	Not running						
	Generate PLCope	n XML Downlo	ad PLCopen XML					
	PyUAF serialization							
	File	/lhome/wimpe/work/	onto/ontomanager/e	nv/OntoManager/on	tomanager/genera	ted/pyuaf/mtcs_cove	er.py	
	Status	File has been read						
	Code generation	Not running						
	Generate pyUAF	code Download	pyUAF code					
	File contents:							
	2 - <project xm<br="">3 <filehead 4 - <contenth 5 - <coordi 6 - <fbd></fbd></coordi </contenth </filehead </project>	on="1.0" encoding= lns="http://www.pl er companyName="In eader name="mtcs_c nateInfo>	"utf-8"?> copen.org/xml/tc6 stitute of Astron over" modificatio	omy" productName	="OntoManager"   10-09T11:23:49.7	oroductVersion="0 793320">	.0.1" creation	DateTime='

• What are semantic models?

- Where to apply them?
- How to apply them?
- How to build them?
- How to use them?
- Conclusions



🗾 OntoManage	er @ Merc × 📃								<u>a</u>
← → C fi	🗋 localhost:8080/so	t?show=library	;qname=c	over_soft	:mtcs_cover				☆ =
		OntoMan	ager @	Mero	cator Te	elescope			
٥	Ontologies Dataset	Problems	Browse	Query	Systems	Mechanics	Electronics	Software	_
+ Tc2_MC2 + mtcs_commo + mtcs_cover + mtcs_m	Library mt								
+ mtcs_m3	File	/lhome/wimpe/wo	rk/onto/onton	nanager/env/	OntoManager/on	tomanager/genera	ted/mtcs_cover.xml	1	
+ mtcs_telemet	Status	File has been read	1					_	
+ mtcs	Code generation	Not running						_	
	Generate PLCo PyUAF serializati		nload PLCop	en XML					
	File	/lhome/wimpe/wo	rk/onto/onton	nanager/env/	OntoManager/on	tomanager/genera	ted/pyuaf/mtcs_cov	er.py	
	Status	File has been read	1						
	Code generation	Not running							
	Generate pyUA	F code Downlo	ad pyUAF co	ode					
	File contents:								
	Show PLCopen X								
	2 - <project 3 <filehe 4 - <conter 5 - <coor 6 - <ft< th=""><th>sion="1.0" encodi xmlns="http://www ader companyName= tHeader name="mtc: dinateInfo&gt; d&gt; scaling x="1" y=":</th><th>.plcopen.org "Institute o s_cover" moo</th><th>/xml/tc6_02 of Astronomy</th><th><pre>/" productName</pre></th><th>="OntoManager"   10-09T11:23:49.</th><th>productVersion="( 793320"&gt;</th><th>0.0.1" creation</th><th>DateTime='</th></ft<></coor </conter </filehe </project 	sion="1.0" encodi xmlns="http://www ader companyName= tHeader name="mtc: dinateInfo> d> scaling x="1" y=":	.plcopen.org "Institute o s_cover" moo	/xml/tc6_02 of Astronomy	<pre>/" productName</pre>	="OntoManager"   10-09T11:23:49.	productVersion="( 793320">	0.0.1" creation	DateTime='

• 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": <u>http://github.com/uaf/uaf</u>

09:26:43)	
5)] on linux2	
ense()" for more information.	
	09:26:43) 6)] on linux2 ense()" for more information.

70

• 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": <u>http://github.com/uaf/uaf</u> —

	DreamPie	$\odot$ $\odot$
<u>H</u> istory <u>E</u> dit <u>S</u> hell Help		
Python 2.7.8 (default, Apr 15 [GCC 4.9.2 20150212 (Red Hat 4	4.9.2-6)] on linux2	
	r "license()" for more information.	
DreamPie 1.1.1		
>>>		
import opcua		
• • • • • • •		

• 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": <u>http://github.com/uaf/uaf</u>

🗢 💿	DreamPie	$\odot$ $\odot$
<u>H</u> istory <u>E</u> dit <u>S</u> hell Help		
Python 2.7.8 (default, Apr [GCC 4.9.2 20150212 (Red Ha		

• 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": <u>http://github.com/uaf/uaf</u>

	DreamPie	$\odot$ $\odot$ $\otimes$
<u>H</u> istory <u>E</u> dit <u>S</u> hell Help		
Python 2.7.8 (default, Apr 15 2015, 09	:26:43)	
[GCC 4.9.2 20150212 (Red Hat 4.9.2-6)]		
Type "copyright", "credits" or "license	e()" for more information.	
DreamPie 1.1.1		
>>> import opcua		
>>>		
c = opcua.buildClient()		

• 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": <u>http://github.com/uaf/uaf</u>

<u>H</u> istory <u>E</u> dit <u>S</u> hell Help	5 2015 00.05.40	
Python 2.7.8 (default, Apr 13 [GCC 4.9.2 20150212 (Red Hat		
	or "license()" for more information.	
DreamPie 1.1.1		
>>> import opcua		
>>> c = opcua.buildClient()		
>>>		

• 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": <u>http://github.com/uaf/uaf</u> \_

	fault, Apr 15 2015, 09:26:43) 212 (Red Hat 4.9.2-6)] on linux2	
	, "credits" or "license()" for more informa	tion.
DreamPie 1.1.1		
>>> import opcua		
>>> c = opcua.bu	ildClient()	
>>>		
	MAIA	
	MTCS	
	MercatorConfig	
	buildClient	
	buildClientSettings	
	maia	
	models	
	mtcs	
	pyuaf U	
	~	
	> val	

• 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": <u>http://github.com/uaf/uaf</u> \_

	Descending	
<b>—</b> •	DreamPie	$\odot$ $\odot$ $\otimes$
<u>H</u> istory <u>E</u> dit <u>S</u> hell Help		
Python 2.7.8 (default, Ag		
[GCC 4.9.2 20150212 (Red		
DreamPie 1.1.1	cs" or "license()" for more information.	
>>> import opcua		
>>> c = opcua.buildClient	:()	
>>>		
	ADR	
	INFO	
	actualStatus	
	noOfFailedOperatorChang	
	parts	
	processes	
	statuses	
print c.read( opcua.MTCS		
prine erreau, opeaarmes		

• 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": <u>http://github.com/uaf/uaf</u>

<u>H</u> istory <u>E</u> dit <u>S</u> hell	•	
	lt, Apr 15 2015, 09:26:43)	
	(Red Hat 4.9.2-6)] on linux2 'credits" or "license()" for more information.	
reamPie 1.1.1	credits" of "license()" for more information.	
>> import opcua		
>> c = opcua.build	Client()	
>>		
	ADR	
	INFO	
	cover	
	ml	
	m3	
	telemetry	
	timing	
	J J J J J J J J J J J J J J J J J J J	

• 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": <u>http://github.com/uaf/uaf</u>

📼 💿	DreamPie	$(\mathbf{x}) (\mathbf{x})$
<u>H</u> istory <u>E</u> dit <u>S</u> hell Help		000
Python 2.7.8 (default, Apr 1	15 2015, 09:26:43)	
[GCC 4.9.2 20150212 (Red Hat		
	or "license()" for more information.	
DreamPie 1.1.1		
>>> import opcua		
>>> c = opcua.buildClient()		
>>>		
	ADR	
	INFO	
	actualStatus	
	config	
	editableConfig	
	parts	
	pneumaticSupplyPressure	
	processes	
	statuses	
which a needly show works works		
print c.read( opcua.MTCS.pa	rts.mi.	

• 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": <u>http://github.com/uaf/uaf</u>

<b>a</b> 🕢	DreamPie	$\odot \odot \otimes$
<u>H</u> istory <u>E</u> dit <u>S</u> hell Help		
<pre>Python 2.7.8 (default, Apr 15 2015, 09:28 [GCC 4.9.2 20150212 (Red Hat 4.9.2-6)] or Type "copyright", "credits" or "license() DreamPie 1.1.1 &gt;&gt;&gt; import opcua &gt;&gt;&gt; c = opcua.buildClient() &gt;&gt;&gt;</pre>	n linux2	
	ADR INFO axialSupport configManager inclinometer io radialSupport	
<pre>print c.read( opcua.MTCS.parts.ml.parts.</pre>		

• 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": <u>http://github.com/uaf/uaf</u>

	DreamPie	$\odot$ $\otimes$ $\otimes$
<u>H</u> istory <u>E</u> dit <u>S</u> hell Help		000
Python 2.7.8 (default, Apr 15 2	2015 09:26:43)	
[GCC 4.9.2 20150212 (Red Hat 4.		
	"license()" for more information.	
DreamPie 1.1.1		
>>> import opcua		
<pre>&gt;&gt;&gt; c = opcua.buildClient()</pre>		
>>>		
	manualPressureSetpoint	
	mirrorAverageForce	
	mirrorNorthEastForce	
	mirrorNorthWestForce	
	mirrorPressure	
	mirrorSouthForce	
	pressureSetpointOutput	
	processes	
	regulatorPressure	
	statuses	/
<pre>print c.read( opcua.MTCS.parts.</pre>	.m1.parts.axialSupport.	

• 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": <u>http://github.com/uaf/uaf</u>

	DreamPie		$\odot$ $\otimes$ $\otimes$
History Edit Shell Help			
Python 2.7.8 (default, Apr 15 2015, [GCC 4.9.2 20150212 (Red Hat 4.9.2- Type "copyright", "credits" or "lic DreamPie 1.1.1 >>> import opcua >>> c = opcua.buildClient()	-δ)] on linux2		
>>>			
		ADR	^
		INFO	Π
		actual	
		actualStatus	
		average	
		conversionFactor	
		error	
		overrange	L I
		raw/value	U
		statuses	$\hat{\mathbf{v}}$ —
<pre>print c.read( opcua.MTCS.parts.m1.p</pre>	parts.axialSupport.regulatorPress	ure.	

• 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": <u>http://github.com/uaf/uaf</u>

<b>-</b> 0	DreamPie	
History Edit Shell Help		000
Python 2.7.8 (default, Apr 15 201	.5, 09:26:43)	
[GCC 4.9.2 20150212 (Red Hat 4.9.	2-6)] on linux2	
Type "copyright", "credits" or "1	icense()" for more information.	
DreamPie 1.1.1 >>> import opcua		
>>> c = opcua.buildClient()		
>>>		
		ADR
		INFO
		actualStatus
		bar
		newBarValue
		pascal
		~
print c.read( opcua.MTCS.parts.ml	l.parts.axialSupport.regulatorPressure.a	verage.
		- 1

• 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": <u>http://github.com/uaf/uaf</u>

$\sim$ $\odot$	DreamPie	$\odot$ $\odot$
<u>H</u> istory <u>E</u> dit <u>S</u> hell Help		
ython 2.7.8 (default, Apr 15 2015,		
GCC 4.9.2 20150212 (Red Hat 4.9.2- ype "copyright", "credits" or "lid		
reamPie 1.1.1	cense() a for more information.	
>> import opcua		
>> c = opcua.buildClient()		
>>		
		ADR
		INFO actualStatus
		unit
		value
		Valac
rint c.read( opcua.MTCS.parts.m1.	parts.axialSupport.regulatorPressure.av	verage.bar.

• 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": <u>http://github.com/uaf/uaf</u>

<u>H</u> istory <u>E</u> dit <u>S</u> hell Help Python 2.7.8 (default, Apr 15	2015 00.26.42)	
[GCC 4.9.2 20150212 (Red Hat 4		
	"license()" for more information.	
DreamPie 1.1.1		
>>> import opcua		
<pre>&gt;&gt;&gt; c = opcua.buildClient() &gt;&gt;&gt;</pre>		
~~~		
maint a need onen WTCC next		
<pre>print c.read( opcua.MTCS.parts</pre>	s.ml.parts.axialSupport.regulatorPressure.avera	age.bar.value.ADR() 🌔

• 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": <u>http://github.com/uaf/uaf</u>

- 🖸	DreamPie	$\supset \bigcirc ($
<u>H</u> istory <u>E</u> dit <u>S</u> hell Help		
ython 2.7.8 (default, Apr 15	2015, 09:26:43)	
GCC 4.9.2 20150212 (Red Hat 4		
	"license()" for more information.	
reamPie 1.1.1		
>> import opcua		
<pre>&gt;&gt; c = opcua.buildClient()</pre>		
>> print c.read( opcua.MTCS.p	parts.ml.parts.axialSupport.regulatorPressure.average.bar.value.A	ADR()
- overallStatus	: Good	
<ul> <li>requestHandle</li> </ul>	: 1	
<pre>- targets[]</pre>		
<pre>- targets[0]</pre>		
- clientConnectionId		
- status	: Good	
- opcUaStatusCode		
	: 1.10534973145	
	: 2015-10-13T11:33:01.825Z	
	: 2015-10-13T11:33:01.8252	
- sourcePicoseconds		
- serverPicoseconds	: 0	
>>		

• 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": <u>http://github.com/uaf/uaf</u>

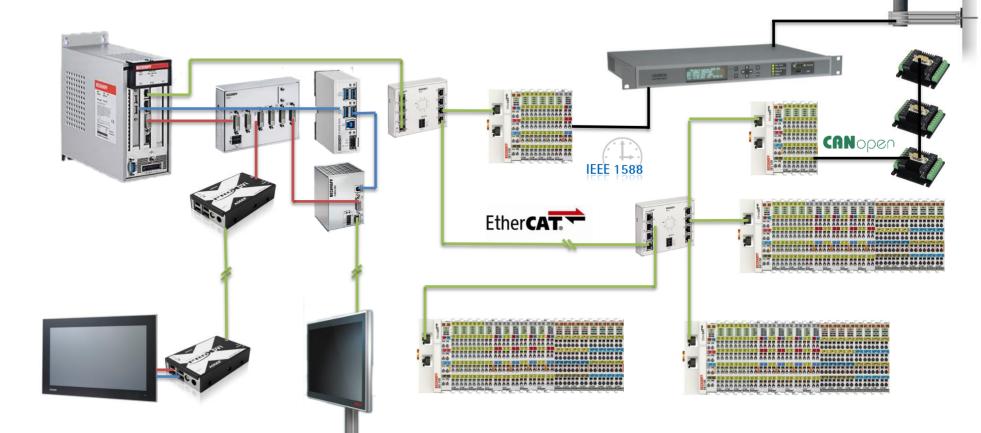
<pre>History Edit Shell Help Python 2.7.8 (default, Apr 15 2015, 09:26:43) [GCC 4.9.2 20150212 (Red Hat 4.9.2-6)] on linux2 Type "copyright", "credits" or "license()" for more information. DreamPie 1.1.1 &gt;&gt;&gt; import opcua &gt;&gt;&gt; c = opcua.buildClient() &gt;&gt;&gt; print c.read( opcua.MTCS.parts.ml.parts.axialSupport.regulatorPressure.average.bar.value.ADR() - overallStatus : Good - requestHandle : 1 - targets[] - clientConnectionId : 0 - status : Good - opcUaStatusCode : 0 - data : 1.10534973145 - sourceTimestamp : 2015-10-13T11:33:01.8252 - sourcePicoseconds : 0 - serverPicoseconds : 0 &gt;&gt;&gt;</pre>	🗢 🕑	DreamPie	0 0
<pre>[GCC 4.9.2 20150212 (Red Hat 4.9.2-6)] on linux2 Type "copyright", "credits" or "license()" for more information. DreamPie 1.1.1 &gt;&gt;&gt; import opcua &gt;&gt;&gt; c = opcua.buildClient() &gt;&gt;&gt; print c.read( opcua.MTCS.parts.ml.parts.axialSupport.regulatorPressure.average.bar.value.ADR() - overallStatus : Good - requestHandle : 1 - targets[] - targets[0] - clientConnectionId : 0 - status : Good - opcUaStatusCode : - data : 1.10534973145 - sourceTimestamp : 2015-10-13T11:33:01.8252 - serverTimestamp : 2015-10-13T11:33:01.8252 - sourcePicoseconds : 0 - serverPicoseconds : 0</pre>	<u>H</u> istory <u>E</u> dit <u>S</u> hell Help		
Type "copyright", "credits" or "license()" for more information. DreamPie 1.1.1 >>> import opcua >>> c = opcua.buildClient() >>> print c.read( opcua.MTCS.parts.ml.parts.axialSupport.regulatorPressure.average.bar.value.ADR() - overallStatus : Good - requestHandle : 1 - targets[] - targets[0] - clientConnectionId : 0 - status : Good - opcUaStatusCode : 0 - data : 1.10534973145 - sourceTimestamp : 2015-10-13T11:33:01.8252 - serverTimestamp : 2015-10-13T11:33:01.8252 - serverPicoseconds : 0	Python 2.7.8 (default, Apr 15	2015, 09:26:43)	
<pre>DreamPie 1.1.1 &gt;&gt;&gt; import opcua &gt;&gt;&gt; c = opcua.buildClient() &gt;&gt;&gt; print c.read( opcua.MTCS.parts.ml.parts.axialSupport.regulatorPressure.average.bar.value.ADR() - overallStatus : Good - requestHandle : 1 - targets[] - targets[0] - clientConnectionId : 0 - status : Good - opcUaStatusCode : 0 - data 1.10534973145 - sourceTimestamp : 2015-10-13T11:33:01.8252 - serverTimestamp : 2015-10-13T11:33:01.8252 - serverPicoseconds : 0 - serverPicoseconds : 0</pre>	[GCC 4.9.2 20150212 (Red Hat 4	.9.2-6)] on linux2	
DreamPie 1.1.1 >>> import opcua >>> c = opcua.buildClient() >>> print c.read( opcua.MTCS.parts.ml.parts.axialSupport.regulatorPressure.average.bar.value.ADR() - overallStatus : Good - requestHandle : 1 - targets[] - targets[0] - clientConnectionId : 0 - status : Good - opcUaStatusCode : D - data : 1.10534973145 - sourceTimestamp : 2015-10-13T11:33:01.8252 - serverTimestamp : 2015-10-13T11:33:01.8252 - sourcePicoseconds : 0 - serverPicoseconds : 0	Type "copyright", "credits" or	"license()" for more information.	
<pre>&gt;&gt;&gt; c = opcua.buildClient() &gt;&gt;&gt; print c.read( opcua.MTCS.parts.ml.parts.axialSupport.regulatorPressure.average.bar.value.ADR() - overallStatus : Good - requestHandle : 1 - targets[]     - targets[0]     - clientConnectionId : 0     - status : Good     - opcUaStatusCode : 0     - data : 1.10534973145     - sourceTimestamp : 2015-10-13T11:33:01.8252     - serverTimestamp : 2015-10-13T11:33:01.8252     - sourcePicoseconds : 0     - serverPicoseconds : 0</pre>			
<pre>&gt;&gt;&gt; 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 : 2015-10-13T11:33:01.8252 - serverTimestamp : 2015-10-13T11:33:01.8252 - serverPicoseconds : 0 - serverPicoseconds : 0</pre>	>>> import opcua		
<pre>- overallStatus : Good - requestHandle : 1 - targets[] - targets[0] - clientConnectionId : 0 - status : Good - opcUaStatusCode : 0 - data : 1.10534973145 - sourceTimestamp : 2015-10-13111:33:01.8252 - serverTimestamp : 2015-10-13T11:33:01.8252 - sourcePicoseconds : 0 - serverPicoseconds : 0</pre>	>>> c = opcua.buildClient()		
<pre>- requestHandle : 1 - targets[]     - targets[0]     - clientConnectionId : 0     - status : Good     - opcUaStatusCode : 0     - data : 1.10534973145     - sourceTimestamp : 2015-10-13T11:33:01.8252     - serverTimestamp : 2015-10-13T11:33:01.8252     - sourcePicoseconds : 0     - serverPicoseconds : 0</pre>	>>> print c.read( opcua.MTCS.p	arts.ml.parts.axialSupport.regulatorPressure.average.bar	.value.ADR() )
<pre>- targets[]   - targets[0]   - clientConnectionId : 0   - status : Good   - opcUaStatusCode : 0   - data : 1.10534973145   - sourceTimestamp : 2015-10-13111:33:01.8252   - serverTimestamp : 2015-10-13T11:33:01.8252   - serverPicoseconds : 0   - serverPicoseconds : 0</pre>	- overallStatus	: Good	
<pre>- targets[0] - clientConnectionId : 0 - status : Good - opcUaStatusCode : 0 - data : 1.10534973145 - sourceTimestamp : 2015-10-13111:33:01.8252 - serverTimestamp : 2015-10-13T11:33:01.8252 - sourcePicoseconds : 0 - serverPicoseconds : 0</pre>	- requestHandle	: 1	
<pre>- 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</pre>	<pre>- targets[]</pre>		
- status : Good - opcUaStatusCode : D - data : 1.10534973145 - sourceTimestamp : 2015-10-13T11:33:01.8252 - serverTimestamp : 2015-10-13T11:33:01.8252 - sourcePicoseconds : 0 - serverPicoseconds : 0	- targets[0]		
- opcUaStatusCode : 0 - data : 1.10534973145 - sourceTimestamp : 2015-10-13T11:33:01.825Z - serverTimestamp : 2015-10-13T11:33:01.825Z - sourcePicoseconds : 0 - serverPicoseconds : 0	- clientConnectionId	: 0	
- data : 1.10534973145 - sourceTimestamp : 2015-10-13T11:33:01.825Z - serverTimestamp : 2015-10-13T11:33:01.825Z - sourcePicoseconds : 0 - serverPicoseconds : 0	- status	: Good	
<ul> <li>sourceTimestamp : 2015-10-13T11:33:01.825Z</li> <li>serverTimestamp : 2015-10-13T11:33:01.825Z</li> <li>sourcePicoseconds : 0</li> <li>serverPicoseconds : 0</li> </ul>	<ul> <li>opcUaStatusCode</li> </ul>		
<pre>- serverTimestamp : 2015-10-13T11:33:01.825Z - sourcePicoseconds : 0 - serverPicoseconds : 0</pre>	- data	1.10534973145	
- sourcePicoseconds : O - serverPicoseconds : O	<ul> <li>sourceTimestamp</li> </ul>	: 2015-10-13711:33:01.8252	
- serverPicoseconds : O	<ul> <li>serverTimestamp</li> </ul>	: 2015-10-13T11:33:01.825Z	
	<ul> <li>sourcePicoseconds</li> </ul>	: 0	
>>>	<ul> <li>serverPicoseconds</li> </ul>	: 0	
	>>>		

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

# Currently in operation:

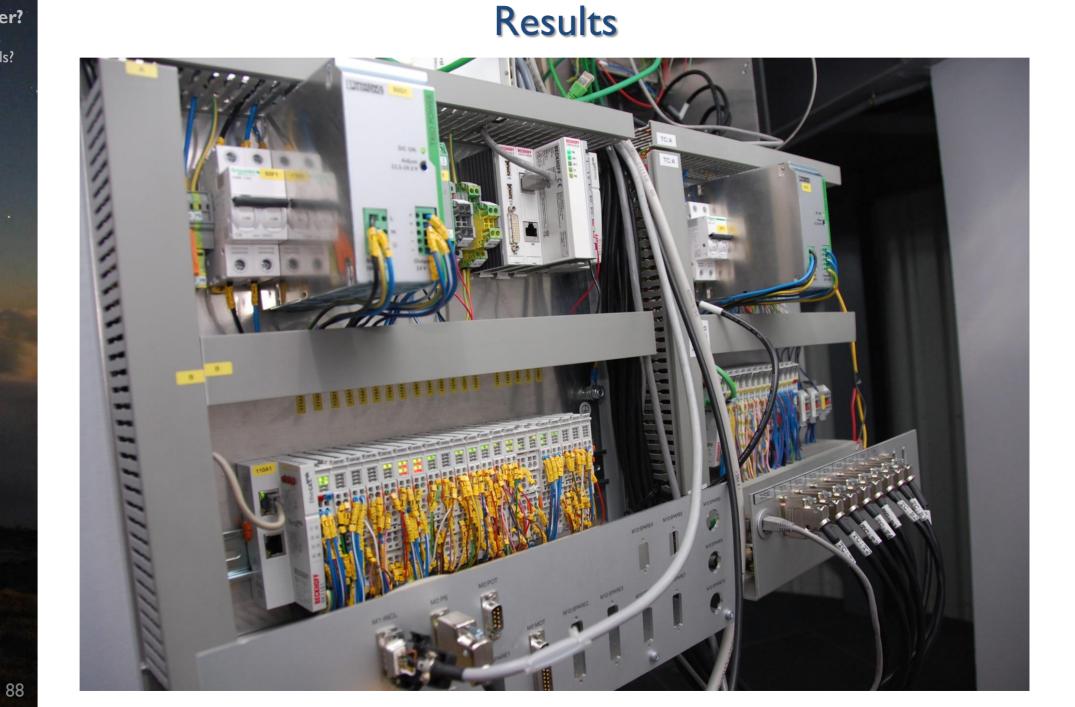
– I PLC

- 5 subsystems
- 55 I/O modules
- 159 PLC Function Block definitions (626 instances)

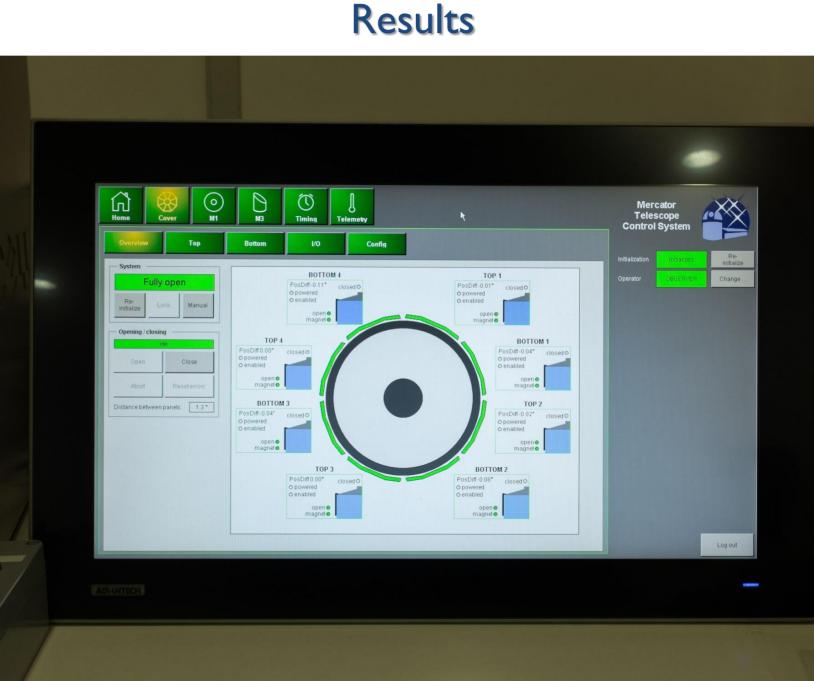


**Results** 

- 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?
- Where to apply them?
- How to apply them?
- How to build them?
- How to use them?
- Conclusions



SH

• What are semantic models?

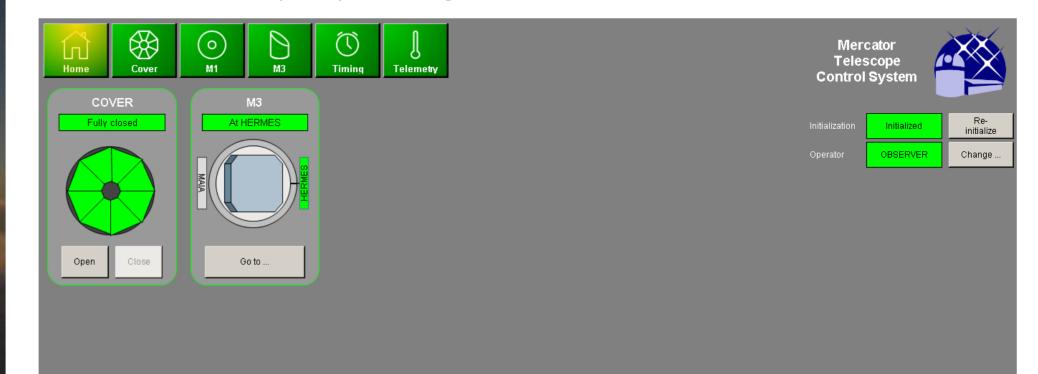
lacksquare

- Where to apply them?
- How to apply them?
- How to build them?
- How to use them?
- Conclusions

WEB3005

90

### User Interface (HMI) running on the PLC



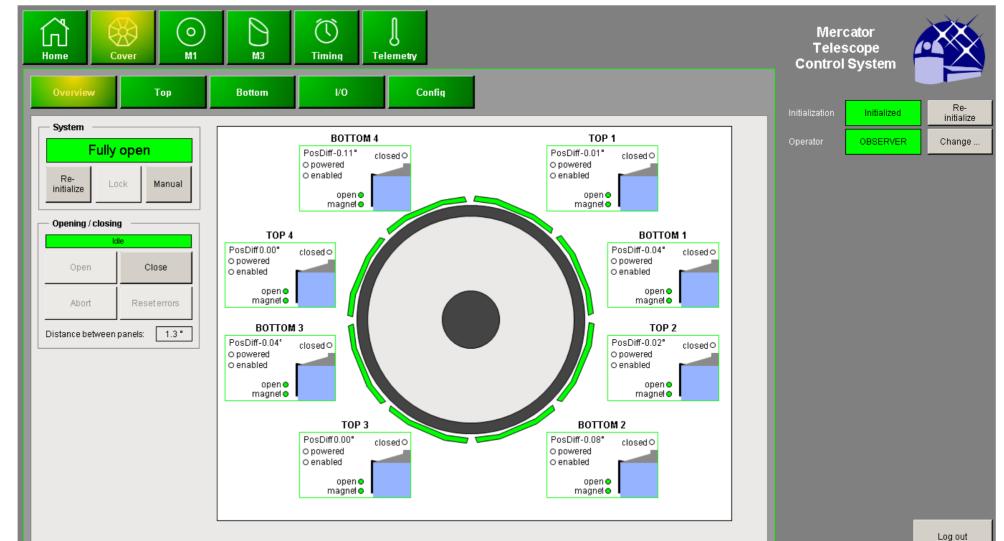
Results

• What are semantic models?

lacksquare

- Where to apply them?
- How to apply them?
- How to build them?
- How to use them?
- Conclusions

### User Interface (HMI) running on the PLC

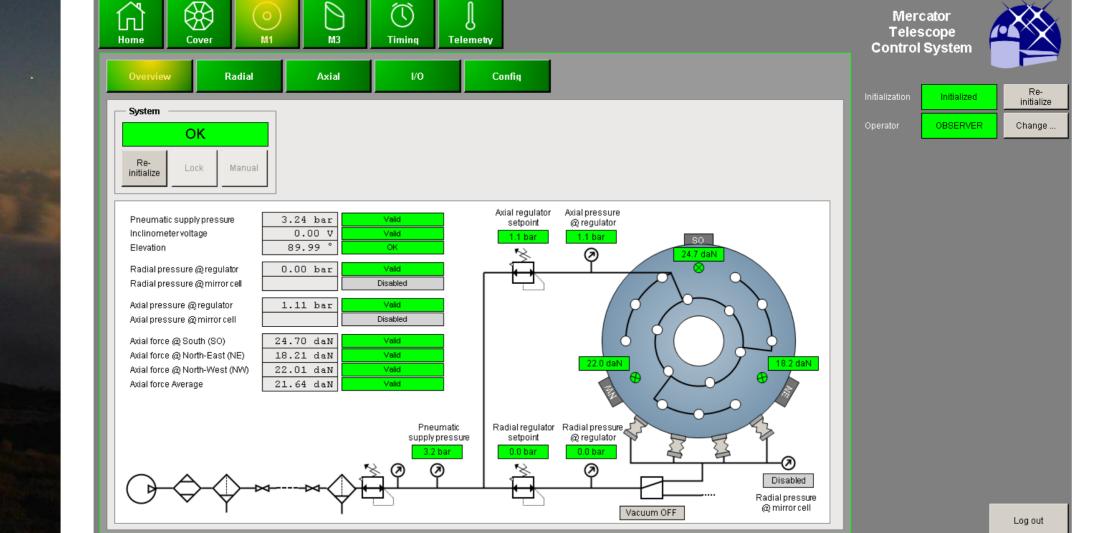


**Results** 

• 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?
- Where to apply them?
- How to apply them?
- How to build them?
- How to use them?
- Conclusions

# Conclusions

So, why semantics matter?

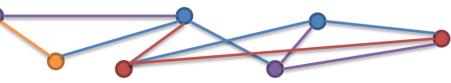
- What are semantic models?
- Where to apply them?
- How to apply them?
- How to build them?
- How to use them?
- Conclusions

WEB3005

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



I. Because every piece of information is just one query "away"

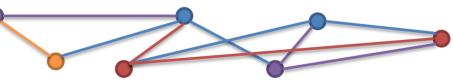


→ organize, integrate, browse, find (query) information

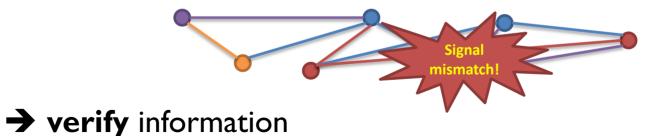
- What are semantic models?
- Where to apply them?
- How to apply them?
- How to build them?
- How to use them?
- Conclusions



Because every piece of information is just one query "away"



- → organize, integrate, browse, find (query) information
- 2. Because well defined semantics allow model verification

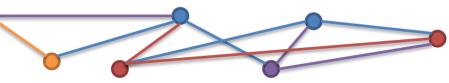


- What are semantic models?
- Where to apply them?
- How to apply them?
- How to build them?
- How to use them?



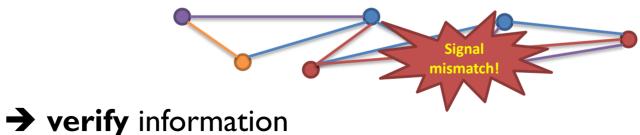
# So, why semantics matter?

Because every piece of information is just one query "away"

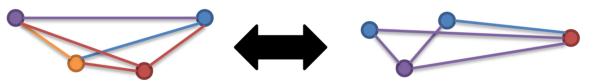


- → organize, integrate, browse, find (query) information
- 2. Because well defined semantics allow model verification

share information



3. Because they're a key enabling technology for future "smart" systems



# Thanks!

### Any questions?

wim.pessemier@ster.kuleuven.be