Capturing Industrial Information Models with Ontologies and Constraints

Evgeny Kharlamov
Senior Research Fellow
Department of Computer Science
University of Oxford

B. Cuenca Grau, E. Jimenez-Ruiz, S. Lamparter,
G. Mehdi, M. Ringsquandl, Y. Nenov, S. Grimm
M. Roshchin, Ian Horrocks

Smart Factory

Automation
- of various individual processes
  - production
  - warehouse

(Enterprise-wide) integration
- of machines and processes
- factory as one organism

Control
- machines and processes
- monitoring, analytics, and diagnostics
Smart Factory

Automation
- of various individual processes
- of production
- of warehouse

Integration
- (Enterprise-wide)
- of machines and processes
- of the factory as one organism

Control
- of machines and processes
- monitoring, analytics, and diagnostics

Smart factories are in the heart of Industry 4.0
Software View on Smart Factories

Smart factory is

- fully computerized
- software-driven (system)
Software View on Smart Factories

Smart factory is
- fully computerized
- software-driven (system)

Software levels
- embedded in machines

Ex: Conveyor belt system
- simple controlling
  - positioning
  - speed
  - safety: emergency stop

Management Layer

Supervisory Layer

Control Layer

Field Devices

Conveyor belt system
Software View on Smart Factories

Smart factory is
- fully computerized
- software-driven (system)

Software levels
- embedded in machines
- controlling several machines

Ex: Manufacturing conveying sub-system
- combines
  - Conveyer belt system
  - Routing system
  - Storage system
- orchestrated by complex controllers
Software View on Smart Factories

Smart factory is
- fully computerized
- software-driven (system)

Software levels
- embedded in machines
- controlling several machines
- controlling the whole plant

Supervisory level
- plant-wide
  - integration
  - orchestration of processes
- plant-wide
  - monitoring
  - diagnostics of machines and processes

Diagram:
- Management Layer
- Supervisory Layer
- Control Layer
- Field Devices
- SCADA computer system
Software View on Smart Factories

Smart factory is
- fully computerized
- software-driven (system)

Software levels
- embedded in machines
- controlling several machines
- controlling the whole plant
- management level software
  - ERP
    - Manufacturing resource planning
    - Finance
    - Human resources
Software View on Smart Factories

Smart factory
- fully computerized
- software-driven (system)

Software levels
- embedded in machines
- controlling several machines
- controlling the whole plant
- management level software
  - ERP
    - Manufacturing resource planning
    - Finance
    - Human resources
Software Challenges

Challenges

- Software development
- Software integration

Software development: ~40% of the price of manufacturing machines

estimated by Mechanical Engineering Industry Association (VDMA) [2011]
Information Models for Smart Factories

Factory-wide info. models
- address challenges
  - SW development
  - SW integration
- capture knowledge on all SW levels
Information Models for Smart Factories

Factory-wide info. models
- address challenges
  - SW development
  - SW integration
- capture knowledge on all SW levels
Information Models for Smart Factories

**Factory-wide info. models**
- address challenges
  - SW development
  - SW integration
- capture knowledge on all SW levels

**Industrial standardisation is critical**
- ensures: safety, security, robustness, …
- sets “good practices” for industrial automation
- bases for industrial information models

How well these models solve the problems?
Challenges with Existing Information Models

Reality of Information Models

- many types of models co-exist in one factory
- often incompatible models
  - independently developed
  - use different (often incompatible) formats
  - come from different types of proprietary software
  - may not come with a well-defined semantics
  - specification can be ambiguous

Consequences

- applications
  - ad hoc customization for various models
  - loosely integrated
- model management is a nightmare
  - development
  - maintenance
  - integration

Can Semantic Technologies make life easier?
Ontologies as Information Models

Industrial Adoption of Sem. Tech.
- A lot of research
- Industry started adapting Sem Tech
  - Statoil, Aibel, Siemens
- OWL 2 and RDF Benefits
  - W3C standard
  - a lot of tooling
  - clear (machine process.) semantics
  - flex. data standard: storing, exch.
Outline

Intro
- Smart factories and the role of software
- Industrial information models to facilitate smart factories
- Ontologies as industrial information models

Our project
- goals
- achievements

Capturing Industrial Information Models with Ontologies and Constraints
Our Project Goals

1. Ontology language for industrial info. models
   - better understanding
   - set **foundations** for ontologies capturing
     - master data ~ industrial standards
     - domain specific model ~ concrete factories
   - study
     - expressiveness
     - management tasks: ontology and data oriented
     - algorithms: to efficiently accomplish the tasks

2. Concrete ontologies
   - to show modeling capabilities and **practical benefits** for industry

3. Modelling Methodology and Tooling
   - **cost efficient** for creation & management of IIM – w/o SWeb background
Our Achievements

Ontology language for IIM
- expressiveness
- algorithms

Concrete ontologies
- 2 ontologies
- experiments

Modeling methodology and tooling
- SOMM systems

Goals
1. Onto language for IIM
2. Concrete ontologies
3. Modelling methodology and tooling
Ontology Language for Industrial Info Models

Analyzed two (sets of) industrial standards

- Manufacturing
  - IEC 62264 → ISA 88 and ISA 95
- Energy
  - IEC 81346 → ISO/TS 16952-10 → RDS PP and KKS
- Consolidated modeling requirements

ISA 88/95

![Diagram of ISA 88/95](image)

IEC 81346

ISO/TS 16952-10

RDS-PP
Energy

IEC 81346 ➔ ISO/TS 16952-10 ➔ RDS PP and KKS

<table>
<thead>
<tr>
<th>IEC 81346</th>
<th>Designation</th>
</tr>
</thead>
<tbody>
<tr>
<td>A Systems for common tasks</td>
<td>=MDA</td>
</tr>
<tr>
<td>B Systems of the main process (power plants)</td>
<td>=MD</td>
</tr>
<tr>
<td>C Electrical auxiliary power</td>
<td>=MDK</td>
</tr>
<tr>
<td>D Control and management</td>
<td>=MDY</td>
</tr>
<tr>
<td>E Functional allocation</td>
<td>=MDK56</td>
</tr>
<tr>
<td>F Fuel treatment and supply of energy sources</td>
<td>=MDK55</td>
</tr>
<tr>
<td>G Water supply and disposal</td>
<td>=MDK40</td>
</tr>
<tr>
<td>H Heat generation by combustion</td>
<td>=MDK30</td>
</tr>
<tr>
<td>I Nuclear heat generation</td>
<td>=MDK20</td>
</tr>
<tr>
<td>J Nuclear auxiliary systems</td>
<td>=MDK10</td>
</tr>
<tr>
<td>K Water, steam, condensate systems</td>
<td>=MDK01</td>
</tr>
<tr>
<td>L Medium supply systems</td>
<td>=C001</td>
</tr>
<tr>
<td>M Systems for generation to and transmission</td>
<td>=C002</td>
</tr>
<tr>
<td>N Central lubrication system</td>
<td>=C003</td>
</tr>
<tr>
<td>O Auxiliary systems</td>
<td>=C004</td>
</tr>
<tr>
<td>P Cooling water systems</td>
<td>=C005</td>
</tr>
<tr>
<td>Q Ancillary systems</td>
<td>=C006</td>
</tr>
<tr>
<td>R Flue gas extraction</td>
<td>=C007</td>
</tr>
<tr>
<td>S Coordination of elements</td>
<td>=C008</td>
</tr>
<tr>
<td>T Function</td>
<td>=G001</td>
</tr>
<tr>
<td>U Product</td>
<td>=G002</td>
</tr>
<tr>
<td>V Point of installation</td>
<td>=G003</td>
</tr>
<tr>
<td>W Site of installation</td>
<td>=G004</td>
</tr>
<tr>
<td>X Location</td>
<td>=G005</td>
</tr>
<tr>
<td>Y Company</td>
<td>=G006</td>
</tr>
<tr>
<td>Z Product classes</td>
<td>=G007</td>
</tr>
</tbody>
</table>

Wind Turbine Model

Conjoint designation for Wind Power Plant: #5154N00883E.DE_NW.EU_1WN

Main system designation e.g. for Wind Turbine Generator: =G001

System designation e.g. for Yaw System: =G001 MDL

Subsystem designation e.g. for Yaw Drive System: =G001 MDL10

Basic Function designation e.g. for Yaw Drive 1: =G001 MDL10 MZ010

Product designation e.g. for Yaw Motor 1: =G001 MDL10 MZ010–MA001

Product designation e.g. for Yaw Gear 1: =G001 MDL10 MZ010–TL001
Manufacturing

IEC 62264 → ISA 88 and ISA 95

Manufacturing Process Model

ISA 88/95

Product Segments
- Product Blueprints
- Process Blueprints

Process Segment
- Process Routing

Execution
- Process Execution
- Operational Data

Data-driven Model
- Legend:
  - Used in
  - Has part
  - Data flow

Level of Detail

Product 1
- Part A
- Part B

High-level Model
- Process 1
- Process 2
- Process 3

Low-level Model
- Operation 1
- Operation 2
- Operation 3

Operation 1
- Process 2

Operation 2
- Process 2

Operation 3
- Process 2

DB
How to Turn an ISA standard into an Ontology?

Vocabulary

- Classes
  - Main class, subclasses, type of provenance, prov. Results

- Properties
  - Data, Object, Annotation, with provenance

Axioms

- Properties “attached” to classes with
  - Typing
  - Default values
  - Uniqueness of prop. values
  - “Required” property
  - Cardinality restriction (?)

- Disjoint classes
  - E.g. personnel & equipment

Table 5 – Attributes of person

<table>
<thead>
<tr>
<th>Attribute name</th>
<th>Description</th>
<th>Examples</th>
</tr>
</thead>
</table>
| ID            | A unique identification of a specific person, within the scope of the information exchanged (production capability, production schedule, production performance, etc.). | 999-123-4567
                | The ID shall be used in other parts of the model when the person needs to be identified, such as the production capability for this person, or a production response identifying the person. | Jane W Smith – #2
                |                                              | Employee 23 |
| Description   | Additional information about the resource. | "Person information"
| Name          | The name of the individual. | Joe Smith III
                | This is meant as an additional identification of the resource, but only as information and not as a unique value. | Jane
                |                                              | Bubba |

Someone has to make a design choice on how to interpret a standard
A Possible Interpretation of ISA 88-95

ISA 88-95 modules
- Person, Equipment, Material

Classes:
- Person, PersonClass, PersonProvenanceType, PersonProvenanceTest
- Engineer, Plummer, etc

Properties
- ID, Description, Name
- DoB, Address

Attached properties for Person
- ID (Int): compulsory, unique
- Name (String)

Someone has to make a design choice on how to attach properties to classes
Classes

Class

- “Person” = \{bob, john, …\}

Class of classes (?) or jobs (?)

- “Personnel class” = \{Engineers, Pilots, …\}

Modeling in OWL

Class: Person

Class: Personnel

Class: Engineer, Pilot

Individual: Engineer

Types: Personnel

Individual: Pilot

Types: Personnel

Design choice is not trivial
Properties of Objects

Attributes: for objects
- “Default” for objects of a given class
  - E.g.: Bob has ID, Desc, Name
- Extra properties of objects (for objects of a given class)
  - Person prop. = {age, friend-of}
  - Defined via “templates”
  - Age has ID, Desc, Value, V. unit of meas.

Table 5 – Attributes of person

<table>
<thead>
<tr>
<th>Attribute name</th>
<th>Description</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>A unique identification of a specific person, within the scope of the information exchanged. (production capability, production schedule, production performance, etc.). The ID shall be used in other parts of the model when the person needs to be identified, such as the production capability for this person, or a production response identifying the person.</td>
<td>999-123-4567</td>
</tr>
<tr>
<td>Description</td>
<td>Additional information about the resource.</td>
<td>“Person information”</td>
</tr>
<tr>
<td>Name</td>
<td>The name of the individual. This is meant as an additional identification of the resource, but only as information and not as a unique value.</td>
<td>Joe Smith III, Jane, Bubba</td>
</tr>
<tr>
<td>Description</td>
<td>Additional information about the person property.</td>
<td>“Person information”</td>
</tr>
<tr>
<td>Value</td>
<td>The value, set of values, or range of the property. The value(s) is assumed to be within the range or set of defined values for the related personnel class property.</td>
<td>True, 4</td>
</tr>
<tr>
<td>Value unit of measure</td>
<td>The unit of measure of the associated property value, if applicable.</td>
<td>Boolean, h</td>
</tr>
</tbody>
</table>

Table 6 – Attributes of person property

<table>
<thead>
<tr>
<th>Attribute name</th>
<th>Description</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>An identification of the specific property.</td>
<td>Class 1 certified, Exposure hours available, Pager number</td>
</tr>
<tr>
<td>Description</td>
<td>Additional information about the person property.</td>
<td>“Indicates if the person is class 1 certified widget assembly operator”, “Indicates number of exposure hours available this month”, “Pager number”</td>
</tr>
<tr>
<td>Value</td>
<td>The value, set of values, or range of the property. The value(s) is assumed to be within the range or set of defined values for the related personnel class property.</td>
<td>True, 4, 800-555-1212</td>
</tr>
<tr>
<td>Value unit of measure</td>
<td>The unit of measure of the associated property value, if applicable.</td>
<td>Boolean, h, Phone number</td>
</tr>
</tbody>
</table>
Properties of Objects

Class: Person

HasKey: ID
SubClassOf: Description \textit{min 1}, description \textit{only} string
SubClassOf: Name \textit{exactly 1}, Name \textit{only} string

ObjectProperty: PagerNumber
Annotations: ID “pager number”
Description “descr of pager number”

Domain: Person
Range: PhoneNumber

Table 4 – Attributes of personnel class property

<table>
<thead>
<tr>
<th>Attribute name</th>
<th>Description</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>A unique identification of a specific person, within the scope of the information exchanged (production capability, production schedule, production performance, etc.). The ID shall be used in other parts of the model when the person needs to be identified, such as the production capability for this person, or a production response identifying the person.</td>
<td>999-123-4567, Jane W Smith – #2, Employee 23</td>
</tr>
<tr>
<td>Description</td>
<td>Additional information about the resource.</td>
<td>“Person information”, “Person information”</td>
</tr>
<tr>
<td>Name</td>
<td>The name of the individual. This is meant as an additional identification of the resource, but only as information and not as a unique value.</td>
<td>Joe Smith III, Jane, Bubba</td>
</tr>
</tbody>
</table>

Default

Table 5 – Attributes of person

<table>
<thead>
<tr>
<th>Attribute name</th>
<th>Description</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>An identification of the specific property.</td>
<td>Class 1 certified, Exposure hours available, Pager number</td>
</tr>
<tr>
<td>Description</td>
<td>Additional information about the person property.</td>
<td>“Indicates if the person is class 1 certified widget assembly operator”, “Indicates number of exposure hours available this month”, “Pager number”</td>
</tr>
<tr>
<td>Value</td>
<td>The value, set of values, or range of the property. The value(s) is assumed to be within the range or set of defined values for the related personnel class property.</td>
<td>True, 4, 800-555-1212</td>
</tr>
<tr>
<td>Value unit of measure</td>
<td>The unit of measure of the associated property value, if applicable.</td>
<td>Boolean, h, Phone number</td>
</tr>
</tbody>
</table>

Table 6 – Attributes of person property

<table>
<thead>
<tr>
<th>Attribute name</th>
<th>Description</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>An identification of the specific property.</td>
<td>Class 1 certified, Exposure hours available, Pager number</td>
</tr>
<tr>
<td>Description</td>
<td>Additional information about the person property.</td>
<td>“Indicates if the person is class 1 certified widget assembly operator”, “Indicates number of exposure hours available this month”, “Pager number”</td>
</tr>
<tr>
<td>Value</td>
<td>The value, set of values, or range of the property. The value(s) is assumed to be within the range or set of defined values for the related personnel class property.</td>
<td>True, 4, 800-555-1212</td>
</tr>
<tr>
<td>Value unit of measure</td>
<td>The unit of measure of the associated property value, if applicable.</td>
<td>Boolean, h, Phone number</td>
</tr>
</tbody>
</table>

AnnotationProperty: ID
Domain: PagerNumber, …
Range: integer[ > 0 ]

HasKey is not a constraint: does not enforce explicit ID for data
Properties of Classes

Attributes: for classes

- “Default” for specific class
- Bob has ID, Desc, Name
- Extra properties for classes
  - Person prop. = {age, friend-of}
  - Defined via “templates”
- Age has ID, Desc, Value, V. unit of meas.

Table 5 – Attributes of person

<table>
<thead>
<tr>
<th>Attribute name</th>
<th>Description</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>A unique identification of a specific person, within the scope of the information exchanged (production capability, production schedule, production performance, etc.). The ID shall be used in other parts of the model when the person needs to be identified, such as the production capability for this person, or a production response identifying the person.</td>
<td>999-123-4567 Jane W Smith – #2 Employee 23</td>
</tr>
<tr>
<td>Description</td>
<td>Additional information about the resource.</td>
<td>“Person information” “Person information” “Person information”</td>
</tr>
<tr>
<td>Name</td>
<td>The name of the individual. This is meant as an additional identification of the resource, but only as information and not as a unique value.</td>
<td>Joe Smith III Jane Bubba</td>
</tr>
</tbody>
</table>

Table 6 – Attributes of person property

<table>
<thead>
<tr>
<th>Attribute name</th>
<th>Description</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attribute name</td>
<td>Description</td>
<td>Examples</td>
</tr>
<tr>
<td>Description</td>
<td>Additional information about the person property.</td>
<td>“Indicates if the person is class 1 certified widget assembly operator” “Indicates number of exposure hours available this month” “Pager number”</td>
</tr>
<tr>
<td>Value</td>
<td>The value, set of values, or range of the property.</td>
<td>True 4 800-555-1212</td>
</tr>
<tr>
<td>Value unit of measure</td>
<td>The unit of measure of the associated property value, if applicable.</td>
<td>Boolean</td>
</tr>
</tbody>
</table>
Properties of Classes

AnnotationProperty: ID
  Domain: PersonnelClass
  Range: string

AnnotationProperty: Description
  Domain: PersonnelClass
  Range: string

DataProperty: ClassOneCertified
  Annotations: ID “Class One Certified”
  Description “Indicates the …”
  Domain: Engineer
  Range: Boolean

Class: Engineer
  SubClassOf: ClassOneCertified exactly 1 and
  ClassOneCertified exists {true, false}
  ClassOneCertified only {true, false}

Table 3 – Attributes of personnel class

<table>
<thead>
<tr>
<th>Attribute name</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>A unique identification of a specific personnel class. These are not necessarily job titles, but identify classes that are referenced in other parts of the model.</td>
<td>Widget assembly operator</td>
</tr>
<tr>
<td>Description</td>
<td>Additional information and description about the personnel class.</td>
<td>“General information about widget assembly operators.”</td>
</tr>
</tbody>
</table>

Table 4 – Attributes of personnel class property

<table>
<thead>
<tr>
<th>Attribute name</th>
<th>Description</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>An identification of the specific property, unique under the scope of the parent personnel class object. For example, the property &quot;has class 1 safety training&quot; (with values of yes or no) may be defined under several different personnel class definitions, such as fork lift operator and pipe fitter classes, but has a different meaning for each class.</td>
<td>Class 1 certified</td>
</tr>
<tr>
<td>Value</td>
<td>The value, set of values, or range of the property. This presents a range of possible numeric values, a list of possible values, or it may be empty if any value is valid.</td>
<td>(True, False)</td>
</tr>
</tbody>
</table>

Table 5 – Attributes of person

<table>
<thead>
<tr>
<th>Attribute name</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value unit of measure</td>
<td>The unit of measure of the associated property values, if applicable.</td>
<td>Boolean</td>
</tr>
</tbody>
</table>

Example

DIN EN 62264-2:2008-07
Inheritance of Properties

Q: What kind of inheritance do we need?
A: May make sense to allow different options

Table 3 – Attributes of personnel class

<table>
<thead>
<tr>
<th>Attribute name</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>A unique identification of a specific personnel class. These are not necessarily job titles, but identify classes that are referenced in other parts of the model.</td>
<td>Widget assembly operator</td>
</tr>
<tr>
<td>Description</td>
<td>Additional information and description about the personnel class.</td>
<td>&quot;General information about widget assembly operators.&quot;</td>
</tr>
</tbody>
</table>

Table 4 – Attributes of personnel class property

<table>
<thead>
<tr>
<th>Attribute name</th>
<th>Description</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>An identification of the specific property, unique under the scope of the parent personnel class object. For example, the property &quot;has class 1 safety training&quot; (with values of yes or no) may be defined under several different personnel class definitions, such as fork lift operator and pipe fitter classes, but has a different meaning for each class.</td>
<td>Class 1 certified, Night shift available, Monthly exposure hours maximum</td>
</tr>
<tr>
<td>Description</td>
<td>Additional information and description about the personnel class property.</td>
<td>&quot;Indicates the certification level of the operator.&quot; &quot;Indicates if operator is available for night shift.&quot; &quot;Indicates the maximum monthly exposure hours that can be used.&quot;</td>
</tr>
<tr>
<td>Value</td>
<td>The value, set of values, or range of the property. This presents a range of possible numeric values, a list of possible values, or it may be empty if any value is valid.</td>
<td>(True, False), (True, False), [0..20]</td>
</tr>
<tr>
<td>Value unit of measure</td>
<td>The unit of measure of the associated property values, if applicable.</td>
<td>Boolean, Boolean, n</td>
</tr>
</tbody>
</table>
Ontology Language for Industrial Info Models

Axioms

- assigning (relevant) properties to classes
  - If-Then by default (A-quantifier)
  - influence type of inheritance
  - domains and ranges of properties

Data Constraints

- Compulsory and default values
- # of compulsory values
- functional properties
- encoded as annotated standard axioms

<table>
<thead>
<tr>
<th>Axiom</th>
</tr>
</thead>
<tbody>
<tr>
<td>SubClassOf(Turbine Equipment)</td>
</tr>
<tr>
<td>SubDataPropertyOf(hasRotorSpeed hasSpeed)</td>
</tr>
<tr>
<td>TransitiveObjectProperty(hasPart)</td>
</tr>
<tr>
<td>InverseObjectProperties(hasPart partOf)</td>
</tr>
<tr>
<td>SubClassOf(Conveying)</td>
</tr>
<tr>
<td>ObjectAllValuesFrom(followedBy Packaging))</td>
</tr>
<tr>
<td>SubClassOf(Turbine SomeValuesFrom(R B))</td>
</tr>
<tr>
<td>SubClassOf(A HasValue(R b))</td>
</tr>
<tr>
<td>SubClassOf(A MaxCardinality(n R B))</td>
</tr>
<tr>
<td>SubClassOf(A MinCardinality(n R B))</td>
</tr>
<tr>
<td>FunctionalProperty(R)</td>
</tr>
<tr>
<td>SubClassOf(Turbine ObjectSomeValuesFrom(hasPart Rotor))</td>
</tr>
<tr>
<td>SubClassOf(TwoRotorTurbine ObjectMinCardinality(2 hasPart Rotor))</td>
</tr>
<tr>
<td>SubClassOf(TwoRotorTurbine ObjectMaxCardinality(2 hasPart Rotor))</td>
</tr>
</tbody>
</table>
Algorithms: Reasoning, Data Validation

Separate axioms and constr.
- using annotations
- axioms: reasoning
- constraints: data validation

Encode in Datalog
- gives a unified framework for axioms and constraints

Choose the right system
- triple store or rule inference system
- supporting
  - Datalog reasoning and
  - stratified negation-as-failure
- IRIS, RDFOx, etc

<table>
<thead>
<tr>
<th>OWL 2 Axiom</th>
<th>Datalog Rules</th>
</tr>
</thead>
<tbody>
<tr>
<td>SubClassOf(A B)</td>
<td>B(?x) ← A(?x)</td>
</tr>
<tr>
<td>SubPropertyOf(P1 P2)</td>
<td>P2(?x, ?y) ← P1(?x, ?y)</td>
</tr>
<tr>
<td>TransitiveObjectProperty(P)</td>
<td>P(?x, ?z) ← P(?x, ?y) ∧ P(?y, ?z)</td>
</tr>
<tr>
<td>InverseObjectProperties(P1, P2)</td>
<td>P2(?y, ?x) ← P1(?x, ?y) and P1(?y, ?x) ← P2(?x, ?y)</td>
</tr>
<tr>
<td>SubClassOf(A AllValuesFrom(P B))</td>
<td>B(?y) ← P(?x, ?y) ∧ A(?x)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>OWL Axiom</th>
<th>Datalog rules</th>
</tr>
</thead>
<tbody>
<tr>
<td>SubClassOf(A SomeValuesFrom(R B))</td>
<td>R.B(?x) ← R(?x, ?y) ∧ B(?y) and Violation(?x, α) ← A(?x) ∧ not R.B(?x)</td>
</tr>
<tr>
<td>SubClassOf(A HasValue(R b))</td>
<td>Violation(?x, α) ← A(?x) ∧ not R(?x, b)</td>
</tr>
<tr>
<td>FunctionalProperty(R)</td>
<td>R.2(?x) ← R(?x, ?y1) ∧ R(?x, ?y2) ∧ not owl:sameAs(?y1, ?y2) and Violation(?x, α) ← R.2(?x)</td>
</tr>
<tr>
<td>SubClassOf(A MaxCardinality(n R B))</td>
<td>R.(n+1).B(?x) ← (∃1≤i≤n+1 (R(?x, ?y1) ∧ B(?y1))) ∧ (∀1≤i,j≤n+1 (not owl:sameAs(?y1, ?yj))) and Violation(?x, α) ← A(?x) ∧ R.(n+1).B(?x)</td>
</tr>
<tr>
<td>SubClassOf(A MinCardinality(n R B))</td>
<td>R.n.B(?x) ← (∃1≤i≤n (R(?x, ?y1) ∧ B(?y1))) ∧ (∀1≤i,j≤n (not owl:sameAs(?y1, ?yj))) and Violation(?x, α) ← A(?x) ∧ not R.n.B(?x)</td>
</tr>
</tbody>
</table>
Our Achievements

Ontology language for IIM
- formalization
- algorithms

Concrete ontologies
- 2 ontologies
- experiments

Modeling methodology and tooling
- SOMM systems

Goals
1. Onto language for IIM
2. Concrete ontologies
3. Modelling methodology and tooling
Ontologies

Manufacturing ontology
- based on IEC 62264
- 79 standard axioms
- 20 constraints

Turbine ontology
- based on IEC 81346
- 121 standard axioms
- 25 constraints
Manufacturing Experiment

Manufacturing data
- simulated by Siemens
- two types of products
- two configurations
  - manufacturing that violates the model specifications (too much material is used)
  - manufacturing according to specifications
- 6 data sets: 50 → 1x10^6

3 monitoring queries
- Q1: find all products that use material from a given lot
- Q2: find all material lots used in a given product
- Q3: find the total quantity of material in lots of a specific kind

Results
- C. validation, Q. answering is feasible on stock hardware: 87s over data datasets with ~1 million triples
Gas Turbine Experiment

Anonymized dataset

- from 800 real gas turbines
- sensor readings (temperature, pressure, rotor speed and position)
- associated processes (e.g., expansion, compression, start up, shut down)
- converted from a relational DB into RDF
- 25,090 triples over 4,076 individuals.

3 monitoring queries

- Q1: find all core parts, equipment & current state of all turb. of a given type
- Q2: find all components involved in a compression process
- Q3: find temperature readings of turbines of a given type

Results

- Constraint checking and query answering: < 2s
- 1,582 constraint violations
Our Achievements

Ontology language for IIM
- formalization
- algorithms

Concrete ontologies
- 2 ontologies
- Experiments

Modeling methodology and tooling
- SOMM systems

Goals
1. Onto language for IIM
2. Concrete ontologies
3. Modelling methodology and tooling

See demo later today!
Summary

Use case analyses
- Smart factories and the role of info models
- Industrial standards
  - Manufacturing (IEC 62264), Energy (IEC 81346)

Foundations of ontology language to capture IIM
- Capturing with axioms and constraints
- Algorithms for constraint verification and query answering

Concrete ontologies
- 2 ontologies: Manufacturing, Energy
- experiments

Modeling methodology and tooling
- SOMM system

Goals
1. Onto language for IIM
2. Concrete ontologies
3. Modelling methodology and tooling