Ontology for sensors

MICHAL SIR
PETR FIEDLER
VAČLAV KACYMARCYZK
Control of Instrumentation
Brno University of Technology
Kolejni 2906/4 612 00 Brno
CZECH REPUBLIC
xsirmi00@stud.feec.vutbr.cz, fiedlerp@feec.vutbr.cz, xkaczm00@stud.feec.vutbr.cz

Abstract: - The aim of this paper is the description of our structure. This structure is created as an extension of Sensor ML. The first part of the paper is description of logic, which is used to describe the ontology. We describe rules for description logic and we define formal definition of ontology. The second part of the paper is about description of our structure. There are description tables which are used in structure and their properties.

Key-Words: - Ontology, Description logic, Predicate logic, Computing system, Sensor, Taxonomy

1 Introduction
From the historical point of view the ontology is a part of science that has many faces. Originally, ontology was the part of philosophy and was understood as science about “being”. Ontology is the universal system of knowledge, which describes objects, phenomena, and regularities of the world. The ontology is a part of science which interests about being and existence. Aristoteles named it as First philosophy.

The ontology allows creating metadata e.g. information about information. We can find ontology in different part of science today. In our case we are taking into account definition for information technology which Gruber described in [1]:

„Ontology is explicit specification of conceptualization“

Conceptualization is an abstract model of part of real world, which identifies relevant concept of the part. Explicitly means that it clearly defines type of concept and conditions of its use. The definition is often extended about word “formal”. It means that ontology can be computer-processable. The ontology is understated as taxonomy which describes hierarchy of determinate names. As we mentioned in definition of ontology, the ontology is rather connected with information technology today. From the knowledge engineering point of view, the ontology is defined as knowledge model i.e. it is the abstract description of knowledge system.

2 Ontology Description
The Ontology can be described in different ways. The first mentioned here is predicate logic of the first order.

2.1 Predicate logic of the first order
The fundamental difference between predicational and predicate logic is that the first mentioned is focused on the structure of sentences. The predicate logic differentiates elements in every sentence. The predicate is mentioned as the attribute or relationship. The majority of attributes, which are formulated in predicational logic, are valid in predicate logic as is written in [2]. We do not use predicational logic because it is not decidable logic. A lot of logical consequences does not verify in predicational logic at finite time.

2.2 Description logic
Description logics create the family of languages, which are used for knowledge structuring. The knowledge is used for the formally understandable representation of terminological knowledge from application domain.

Description logic is created as extension to semantic networks, which did not provide formal semantic. The formal semantic is not based on logic. The name of description logic was originated in 80’s. The description logic was early named “Terminological systems” and “Conceptual languages”. However, the description logic belongs to foundations of semantic web, because it can be used during the ontology design. There exist languages like OWL, OWL Lite and OWL DL
which are based on description logic as Prachař shown in [3].

The goal of the description logic is the interception of classes and relations similarly as ontology do. Description logics differ from classical ontological languages, because they have not early specified the relation between class and subclass. The evaluation between classes relations are performed based on their description. The description is represented by logical expressions as is written in [4].

The description logic is understood as a part of predicate logic of the first order i.e. the less expressive logic. We know some types of the description logic before we defines the basic description logic. Generally, we define that the logic has these rules

\[ \neg A, C \cap D, \forall R. C, \exists R. T, \]

where R is atomic role and A is atomic concept and C, D are concepts. The knowledge base of description logic is created by TBox and ABox (we sometime speak about RBox which contain axioms of role). TBox contains terminological axioms. ABox contains the existential axioms where the concepts and participation in role contain elements.

Another extension of the basic description logics is marked by the letter. The accessible constructions are described in table 1 (specific description logic is marked by combination of letters, for example ALCN).

<table>
<thead>
<tr>
<th>Name</th>
<th>Extension</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unification ( U )</td>
<td>( C \cup D )</td>
</tr>
<tr>
<td>Full existential quantifier ( E )</td>
<td>( \exists R. C )</td>
</tr>
<tr>
<td>Numerical restriction ( N )</td>
<td>( \leq nR a \geq nR )</td>
</tr>
<tr>
<td>Negation ( C )</td>
<td>( \neg C )</td>
</tr>
</tbody>
</table>

Tab. 1: Extension of description logic

The symbol \( S \) is the description logic \( \mathcal{ALC} \). This logic is scaled up about axioms of transitive roles. Symbolic marks of other description logics are shown in table 2[5].

<table>
<thead>
<tr>
<th>Appellation</th>
<th>Admissible extension</th>
</tr>
</thead>
<tbody>
<tr>
<td>( S )</td>
<td>( S = \mathcal{ALC} + \text{axioms of transitive roles} )</td>
</tr>
<tr>
<td>( SH )</td>
<td>( SH = S + \text{axioms of hierarchy (inclusion) roles} )</td>
</tr>
<tr>
<td>( SHF )</td>
<td>( SHF = SH + \text{functional axioms of roles} )</td>
</tr>
<tr>
<td>( SHO )</td>
<td>( SHO = SH + \text{nominal axioms} )</td>
</tr>
<tr>
<td>( SHOI )</td>
<td>( SHOI = SHO + \text{inversive roles} )</td>
</tr>
<tr>
<td>( SHOIN )</td>
<td>( SHOIN = SHOI + \text{numerical restriction} )</td>
</tr>
<tr>
<td>( SHOIQ )</td>
<td>( SHOIQ = SHOI + \text{qualified numerical restriction} )</td>
</tr>
</tbody>
</table>

Tab. 2: Other extension of description logic

To creation of ontology, we use OWL DL language, which is described by SHOIQ description logic.

3 Ontology

There is no common formal definition of what the ontology is. However, most approaches share a few core items: concepts, a hierarchical IS-A-relation, and further relations. For sake of generality, we do not discuss more specific features like constraints, functions, or axioms here. We formalize the core in the following way. The ontology is a tuple

\[ O := (C, is_a, \mathcal{R}, \sigma), \]

where \( C \) is a set whose elements are called concepts, \( is_a \) is a partial order on \( C \) (i.e., a binary relation \( is_a \subseteq C \times C \) which is reflexive, transitive, and antisymmetric), \( \mathcal{R} \) is a set whose elements are called relation names (or relations for short), and \( \sigma : \mathcal{R} \to C^+ \) is a function which assigns to each relation name its arity[6].

The concept itself can be defined similarly as the ontology was.

\[ c := (O, T, P, I) \]

- \( O \) is ontology, where the concept is defined
- \( T \) is a set of taxonomies, i.e. relationships for hierarchical structure definition
- \( P \) is a set of attributes, i.e. binary relations
- \( I \) is a set of instances of concept c. This concept contains elements.

Ontologies are created with respect to specific subject or domain. Ontologies are created as hierarchy or as networks.
4 Future work
We create system which connects ontology and computing system. The system checks the state of the device (in our case, the device is sensor). The whole system is shown in the figure 1.

The sensor sends data to A/D converter, which is connected to the computer system. The system with ontology gathers these data:

- Processional data of sensor
- Information about sensor
- Information about environment

These data are metadata which ontology use for its work. The metadata are sent by sensor or engineer with domain knowledge. The metadata are saved to database where the system find them and can reuse them.

5 Ontology a SensorML
We are interested in language which has been developed specifically to work with sensors data. A SensorML language has been developed by Open Geospatial Consortium. It focuses on processes data visualization on web site because it was designed on XML language structure. SensorML can provide general data about sensor, processing support and measurement analysis. It provides functional properties as accuracy, threshold etc. as shown in [7].

5.1 Structure for automation
Based on the SensorML language, whose structure is shown in the figure 2, we create a simple structure for sensor.

We create two concepts of structure. The first concept is built on principle of MySql database while the second one on principle of OWL DL language. Tables in our structure have unique names (for example “Device”). The first item in table is identification number. The number is used to correct data save and reuse.

5.2 Structure description
The structure is based on the table “Device”. This table is initially used as the signpost for further information and tables, which describe other parts of device (sensor).

The second table of our structure is table called “Description” which is used for general description of sensor. There are items called “LongName” and “ShortName”. These items, which are adopted from Sensor ML, are used for device name storage. Next item in the table is “KeyWords” item, where we store general key words describing the device. For example, Item “KeyWords” is presented as type string and is contain comma-delimited text. We will have pressure sensor and item “KeyWords” can look as follows: pressure sensor, analog output, piezoresistive sensor etc. The whole table is shown in the figure 3.

We create two concepts of structure. The first concept is built on principle of MySql database while the second one on principle of OWL DL language. Tables in our structure have unique names (for example “Device”). The first item in table is identification number. The number is used to correct data save and reuse.
Following tables are used for description of power system. The first table, called “TypeOfPowerSupplies”, is mentioned to store the kind of a power system. There are two possibilities which we can be chosen. If we have energy from battery, we set the item Battery on True or if we have energy from supply mains, we write its type. For example: TNC.

Table called Environment contains three items. These items are of type BOOL and we choose type of environment where the power system is located.

The next tables of our structure describe Input and Output channel. The first table is called “InputOutput” and is shown in the figure 4.

**InputOutput**

<table>
<thead>
<tr>
<th>InputOutputId</th>
<th>NameOfPhenomena</th>
<th>UnitOfPhenomena</th>
<th>TimeStamp</th>
</tr>
</thead>
</table>

Fig. 4: Table of Input/Output channel

The first item in this table is used for description of physical phenomena used by the device. For example, we can write “Pressure. “UnitOfPhenomena” is table, which is used for description of physical unit. In case of the pressure sensor, the unit will be equal to [Pa]. The item “TimeStamp” is time stamp of the measurement. The item describes period of measurement. The table, which is closely connected with “InputOutput” table, is called “PhysicalValue”. This table is used for save of sensor data. The table has two items. The first item is called Value and it is used to save sensor data, while the second is used for record of uncertainty.

Sensors often use communication protocol. This problem is covered in table called “CommunicInterfaces” which is shown in the figure 5.

**CommunicInterfaces**

<table>
<thead>
<tr>
<th>CommunicInterfacesId</th>
<th>NameOfProtocol</th>
<th>TypePhysicalLayer</th>
<th>TypeOfConnector</th>
</tr>
</thead>
</table>

Fig. 5: Table for communication protocol

This table contains items which are used to find general information about communication protocol. The first item is “NameOfProtocol”, which is used for the name of the protocol. For example, we can use As-Interface. The item called “TypePhysicalLayer” stores the type of physical layer. The last item is type of connector which is used to connect cables and devices on physical layer. For example, we can use connector Click&Go, which is used by As-Interface.

The last table in our structure is table called “CpuModul”. This table shows information about cpu module embedded in the device. The table is voluntary because some devices have not cpu module. The figure 6 shows the whole table for description of the cpu module.

**CpuModul**

<table>
<thead>
<tr>
<th>CpuModulId</th>
<th>Producer</th>
<th>ShortNameCpu</th>
<th>TypeOfCpu</th>
<th>Frequency</th>
</tr>
</thead>
</table>

Fig. 6: Description of Cpu module

The table contains the item “Producer, where we write name of manufacturer, e.g. Atmel. The next item called “ShortNameCpu” is used for description of processor name. The “TypeOfCpu” is used for type of processor unit. The last item is used for description of processor frequency.

### 6 Our ontology

Our created structure is used for derivation of ontology. The ontology is focused on sensor as we mentioned in the paper. The figure 7 shows the all taxonomy of proposal of ontology. The ontology contains tables, which are used in our structure, and items, which are adopted Sensor ML.
7 Conclusion
The taxonomy of ontology is the foundation of the system able to detect sensors faults. We want to detect faults of the whole device in future.

The future work is the computer system design and storing data to database. We have to choose type of data storage. The storage kind will be determined and the system functionality problems will be solved. One of possible problems can be unavailable storage and thus missing data.

The environment influencing the sensor should be taken into account as well. This can be solved by measurement uncertainties, which cover different types of influences.

8 Acknowledgement
This work has been supported in part by Ministry of Education, Youth and Sports of the Czech Republic (Research Intent MSM0021630529 Intelligent systems in automation (VZ UAMT), Grant Agency of the Czech Republic (102/09/H081 SYNERGY – Mobile Sensoric Systems and Network) and grant "Supporting Research of Modern Methods and Approaches in Automation" from the Internal Grant Agency of Brno University of Technology (grant No. FEKT-S-11-6).

References:

Fig. 7: Taxonomy of ontology