Process Instrumentation Ontology Based Web Scrapping

B. Sreedhar¹ & Dr. K. Venkata Ramana²
Asst.G.M.(Inst.) Andhra Petrochemicals Limited
Asst. Professor, Andhra University

Abstract: Process Instrumentation (PRIN) ontology is a store of vocabulary meant for exploiting advantages of semantic web technologies, by introducing semantics in process instrumentation vendor’s webpages in a standard way, to facilitate semantic based search across web pages by machines.

1. Introduction

PRIN ontology is an OWL ontology which describes semantics of Process Instrumentation. Process Instrumentation means application of instrumentation engineering for process plants viz. Oil refineries, petrochemical industries. PRIN ontology is basically derived from SSN[1] ontology and DUL[2] ontology since these ontologies contain vocabularies related to sensors and their characteristics.

Hundreds of process instrumentation manufacturers produce vast range of products that are suitable for wide variety of applications in process industries. These products have several brand names and series names even though manufacturer is same. Thus it become very difficult for instrument engineer to select a product which is a best match for his application as vendor’s product information spread over several web pages and technical documents. PRIN ontology standardizes process instrumentation vocabulary so that a computer program can scrap required information from vendor’s web pages and makes it available to the user.

1.1 Development of PRIN ontology

SSN ontology[3] which was developed by W3C is the main inspiration for the development of PRIN ontology. The author is an experienced professional in process instrumentation and

1.2 PRIN ontology

PRIN ontology defines terms related to process instrumentation and positions them under suitable classes of SSN ontology. PRIN ontology was divided into three major sections: Measurement devices, Control system and Final control elements.

1.2.1 Classes

Classes related to all three major sections are inherited from DUL#Physical Object. A PRIN#Category class, to classify the three major sections of the ontology, is inherited from DUL#Entity.

Fig 1: Measurement section of PRIN

The class PRIN#Instrument is the core object representing measurement device section of the ontology (see Fig 1), which is further divided to cover a strata of instruments meant for measuring process variables like pressure, temperature, flow, level etc., An equivalent class PRIN#Meter to cater user specific needs.

Fig 2. SSN#Output hierarchy
SSN#Output class (Fig 2.) is further expanded to suit various outputs, [PRIN#AnalogOutput, PRIN#DigitalOutput, PRIN#PneumaticOutput] that are implemented in PRIN#Instrument, PRIN#ControlSystem, PRIN#Positioner etc.

Fig. 3: PRIN#Communication Hierarchy

Class PRIN#Communication (Fig.3) is, inherited from DUL#Entity, used to describe the type of communication implemented in the PRIN#Instrument. This class is an union of classes PRIN#Channel and PRIN#Protocol.

Fig. 4: PRIN#FinalcontrolElement Hierarchy

Class PRIN#FinalcontrolElement (Fig. 4) is meant for describing several final control elements in an abstract way. A PRIN#ControlValve and PRIN#ShutdownValve are the major final control elements used in process industries to control the process and represented as a union of PRIN#Actuator and PRIN#Valve.

Fig. 5: SSN#System

SSN#System class is expanded to include all sensors related to process instrumentation viz., PRIN#ConductivitySensor, PRIN#MassFlowSensor.

1.2.2 Object Properties

SSN#Property which is used to describe various SSN#Quality features of different PRIN class objects is only sufficient to describe PRIN#Instrument, (measurement instrumentation section) class objects.

A new SSN#Property which is PRIN#ControlProperty is defined to describe SSN#Quality characteristics of PRIN#FinalcontrolElement [PRIN#ControlValve, PRIN#ShutdownValve].

DUL#hasProperty, DUL#hasPart object properties are suitable to describe most of the PRIN# object properties, but certain object properties like PRIN#hasCategory, PRIN#hasImplemented, PRIN#hasSeries are defined to describe the exclusive object properties of PRIN# classes.

Object property: PRIN#hasCategory is used to describe category under which a PRIN# class object belongs to, e.g., PRIN#PressureTransmitter object belongs to PRIN#PressureMeasurement which is a PRIN#Category.

Object property: PRIN#hasImplemented is used to describe category under which a PRIN# class object belongs to, e.g., PRIN#PressureTransmitter object belongs to PRIN#PressureMeasurement which is a PRIN#Category.

Object property: PRIN#hasSeries is used to describe the type of PRIN#Communication that was implemented in the PRIN#Instrument, PRIN#Positioner and PRIN#ControlSystem.

2. Need for Web scrapping

At present all web pages are hard coded html pages to present the information in different style formats, instrumentation vendor’s web pages are not exceptional to this. Semantics related to process instrument products (manufactured by a particular vendor) is hidden in the web pages in the form of
hard coded html. Hence web scrapping methods needs to be developed to extract semantic data of these products.

2.1 Method for web scrapping

Information extraction based on Extraction ontologies\(^4\) is the basic principle behind the proposed method for web scrapping.

![Web scrapping using PRIN ontology](image)

Fig. 6: Web scrapping using PRIN ontology

The basic method followed to extract data from target web page is shown in Fig. 6 and the algorithm of the method is given below

Algorithm:
1. Display all class names of PRIN ontology as keywords
2. User select a class, equivalent classes fetched to form a [search key]
3. Scrap base URL of vendor’s page to match pages ‘title’ with [search key] If found go to step 7.
4. Scrap base URL to match link having text = [search key] If not found fetch for ‘hasCategory’ property of the [search key]
5. Scrap for a link using the Category fetched and update URL otherwise “Data not found”
6. If URL is updated fetch Object properties of [search key] from PRIN ontology.
7. Scrap the updated URL for Object properties among the html elements <table>, <p> etc.,
8. Convert data into RDF format
9. Use SPARQL to query RDF data.

RDF data generated for each base URL is vendor specific and strictly comply to PRIN ontology.

2.2 Tools used

Protégé\(^8\) and Python\(^9\) are the mail tools used to implement web scrapping based on PRIN ontology. The details are given below

1. Protégé: It is developed by Stanford University to create ontologies in several formats. PRIN ontology is created using OWL/XML format. SPARQL bundled with this package can be used to run queries.
2. Python package, Owlready 0.2: This python package gives APIs to access, create and manipulate ontologies. Classes of ontology can be treated as Python classes \(^5\) in this package.
3. Python package, Beautiful Soup\(^10\): This is a powerful python package which works on DOM principle to scrap web pages to extract required data hidden behind various <html> elements.
4. Python package, RE\(^11\): Regular Expressions required for matching and extracting data from <html> elements can be built using this package.

3. Data hidden under html code

Detailed study of several web pages of process instrumentation vendors viz., Emerson\(^6\), Yokogawa\(^7\) reveals the following facts

1. <a href> elements contain all the links to their individual product ranges classified into several categories that are described in PRIN ontology
2. <td> elements defined under <table><tr> tags contains actual data relevant to product properties (ObjectProperty) like Accuracy, Drift, Measurement range
3. <p> elements contains information regarding particular product or brand predominant features
4. <ul> unordered lists starts with <li>, contains a list of links associated with vendor’s web pages, language wise links etc.,
5. The above given elements are repetitive under block level elements <div>
6. Technical specifications of particular product can be found in separate pdf files that are downloadable.

4. Psuedo-code

To find a URL of a selected PRIN Class

```python
PRIN = get_ontology("%path%PRIN.owl").load()
ssn = get_ontology("%URL%ssn.owl").load()
DUL = get_ontology("%URL%DUL.owl").load()

#Example URL
baseURL = "http://www.emerson.com"

#Example search class
searchKey = "PressureTransmitter"

princlasses = PRIN.classes
propertyClass = PRIN.properties
for classItems in princlasses:
```

for classItems in princlasses:
searchClass=find(searchKey)

#Calling scrapping function

url = Scraplink(baseURL,searchClass)

If url is none

equivalentclass = searchClass.equivalent

#Calling Scraplink iteratively

url = Scraplink(baseURL,equivalentclass)

If url is none

Parentclass = searchClass.isa

#Calling scraplink iteratively

url = Scraplink(baseURL,parentclass)

if url is none

print “key not found”
else

url found

URL found:


Using url found call property finding function to scrap data for properties like PRIN#Accuracy, PRIN#Drift

5. Conclusion

Web scrapping is sometimes may result in errors due to the fact that web pages are hard coded in html, hence web developer style reflects in coding style and manufacturers may modify web pages. Implementation of semantic web standards like adding RDF data to the existing process instrumentation vendor’s web pages can help in development of Mash up’s for process instrumentation product range similar to e-commerce product range which will help instrument engineers to take decisions in choosing right product suitable to their application.

6. References


https://www.w3.org/2005/Incubator/ssn/ssnx/ssn


Martin Labský, Vojtěch Švátek, and Marek Nekvšíl


[9] Python documentation: https://docs.python.org/3/

[10] Beautiful Soup Documentation:

https://www.crummy.com/software/BeautifulSoup/bs4/doc

[11] Python RE package:

https://docs.python.org/2/howto/regex.html