

AVanT Tutorial

OWL and SKOS Application in Automotive Domain

Overview

Problem and Motivation

Quick introduction to semantic technologies

- OWL
- SKOS

Modelling JCB automotive terminology/vocabularies

- JCB Terminology (Technical Dictionary)
- Subject Version Revision (SVR)
- Parts Index List (PIL)
- SVR to PIL Mapping

Navigation and Demo

Problem and Motivation

Snap-on Business solutions produce a variety of products supporting Original Equipment Manufacturers (OEMs), including JCB, in managing an increasingly large and complex body of knowledge on a complex and diverse range of products.

JCB produced a number of different models of the components of particular vehicles including an Electronic Parts Catalogue (EPC), Subject Version Revision (SVR) and Parts Index List (PIL). These **offer different perspectives on the structures and relationships between components**, but have **inflexible hierarchical structures** and **little cross-linking**.

A new model is needed that:

- (1) supports **more flexible hierarchical structures**
- (2) enables **better cross-linking** between resources
- (3) **still allows for multi-dimensional hierarchies** to enable users to search and navigate using different views and perspectives.

We will show how to exploit semantic technologies to create such a model.

Semantic Technologies

Semantic technologies enable creation of data stores on the Web (the so called Semantic Web or Linked Data), build vocabularies, and write rules for handling data.

W3C Semantic Web standards include RDF, SPARQL, OWL, and SKOS.

RDF provides the foundation for publishing and linking data. In the RDF data model, statements about (web) resources are in the form of subject–predicate–object expressions – known as triples in RDF terminology.

SPARQL is the query language for the RDF.

OWL and SKOS are Semantic Web languages built on top of RDF.

Protégé is an editor and browser for OWL/SKOS.

<http://www.w3.org/standards/semanticweb/>

Web Ontology Language (OWL)

OWL is a Semantic Web language designed to represent rich and complex knowledge about things, groups of things, and relations between things.

OWL is a computational logic-based language such that knowledge expressed in OWL can be exploited by computer programs, called OWL Reasoners, e.g., to verify the consistency of that knowledge or infer new knowledge – discover new relationships and make implicit knowledge explicit.

OWL development is supported by OWL API.

OWL documents, known as ontologies, can be published in the World Wide Web and may refer to or be referred from other OWL ontologies.

<http://www.w3.org/2001/sw/wiki/OWL>

Simple Knowledge Organisation System (SKOS)

SKOS is a common data model for sharing and linking knowledge organization systems via the Web. Many knowledge organization systems, such as [thesauri](#), [taxonomies](#), [classification schemes](#) and [subject heading systems](#), share a similar structure, and are used in similar applications. SKOS captures much of this similarity and makes it explicit, to enable data and technology sharing across diverse applications.

SKOS provides a [lightweight, intuitive language](#) for developing and sharing new knowledge organisation systems. It may be used on its own, or in combination with OWL.

SKOS development is supported by [SKOS API](#).

<http://www.w3.org/2001/sw/wiki/SKOS>

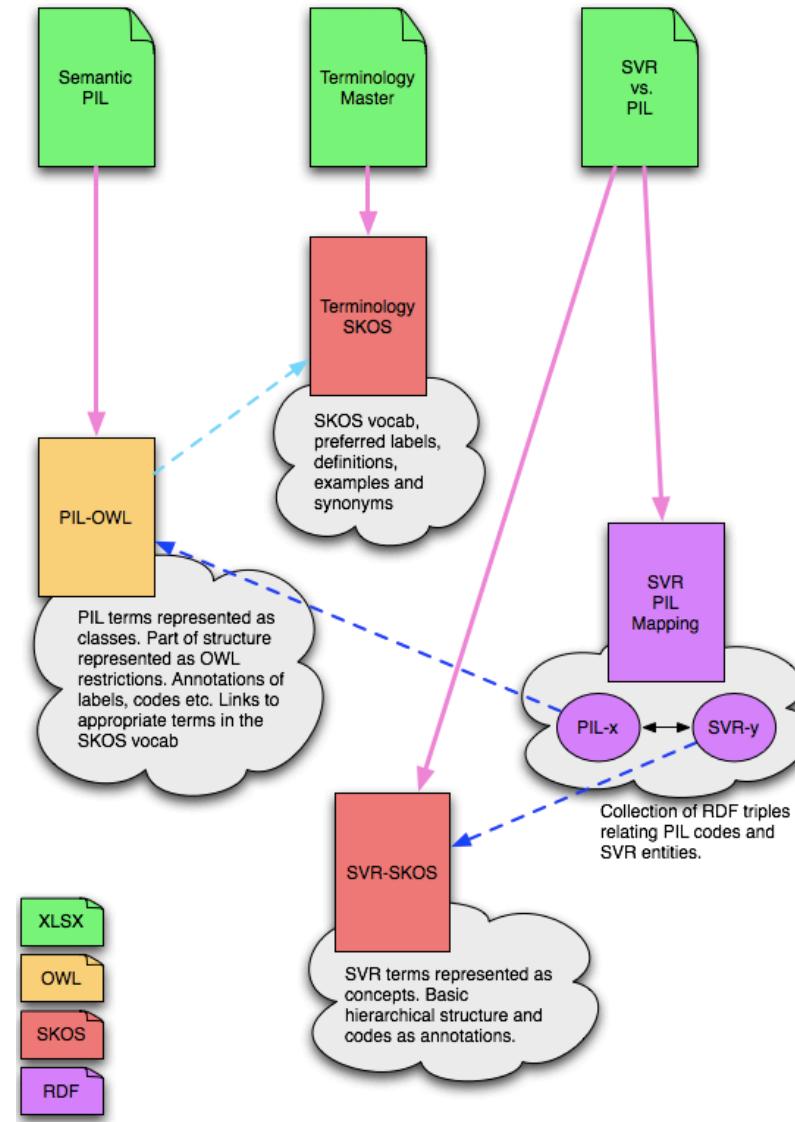
Advantages of OWL/SKOS

SKOS/OWL satisfy our model requirements and provide even more!

Their models enable:

- Separation of entity and label
- Better support for multiple languages and synonymy, thus enhancing search
- Explicit capture and modeling of entities, relationships and terminology
- Inference of even more relationships between entities by OWL reasoning
- Recognition of entities and providing “related” entities to browse
- Easy cross-linking between different SKOS/OWL documents
- Export to simple “tree” structures
- Higher flexibility to changes in products being described

Converting JCB Vocabularies to OWL/SKOS



JCB Terminology (Technical Dictionary)

JCB Terminology is stored in an excel sheet whose every row contains information on a JCB relevant term. Its columns include: the [preferred term](#), [definition](#), [example](#), [definition source](#), a [list of synonyms](#).

For every JCB term we define [a new SKOS concept](#) and assign it [a unique URL](#).

The column values can easily be modelled in SKOS using the [pre-defined annotation properties](#):

Preferred Term => [skos:prefLabel](#)

Definition => [skos:definition](#)

Example => [skos:example](#)

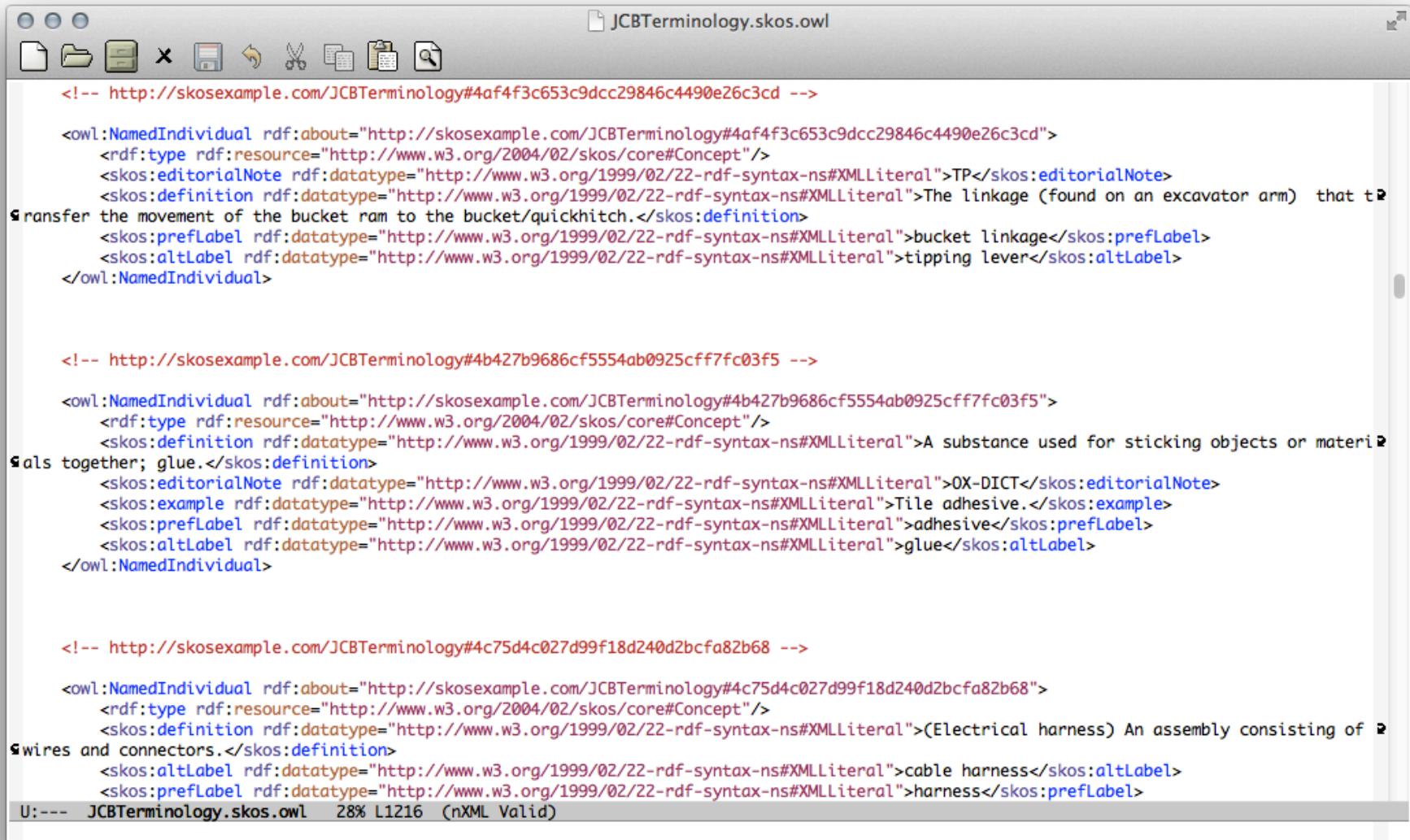
Synonym => [skos:altLabel](#)

Source => [skos:editorialNote](#)

Terminology Master Excel

	A	B	C	D	E	F	G	H	I	J
1	JCB Technical Dictionary									
2	To identify new/amended terms, search for the '#' character.									
3	Updated: 4/10/12									
4										
5	Term (preferred term) ⁽¹⁾	Type ⁽²⁾	Part Desc ⁽³⁾	Definition	Approved Example	Translate?	Source	Synonyms (non-preferred term)		
6										
7	A									
26	adaptor	noun	Yes	A device that connects pieces of equipment that were not originally designed to be connected.		Yes	OX-DICT	adapter	(noun)	
#	adhesive	noun	Yes	A substance used for sticking objects or materials together; glue.	Tile adhesive.	Yes	OX-DICT	glue	(noun)	
#	adhesive tape	noun	Yes	A strip of paper or plastic coated with adhesive and sold in a roll, used to stick things together.		Yes	OX-DICT			
30	# alarm	noun	Yes	A warning sound or device.	A burglar alarm.	Yes	OX-DICT			
51	# alternator	noun	Yes	A dynamo that generates an alternating current.		Yes	OX-DICT			
60	# antenna	noun	Yes	A rod, wire, or other structure by which signals are transmitted or received as part of a radio or television transmission or receiving system. Chiefly North American or technical: another term for aerial.	A TV antenna.	Yes	OX-DICT	aerial	(noun)	
75	antifreeze	noun	Yes	A liquid added to water to prevent it from freezing, especially as used in the radiator of a motor vehicle.		Yes	OX-DICT	anti-freeze	(noun)	
76	# arm	noun	Yes	A thing resembling an arm in form or function.	The completed machine could roam around and had a fully functional arm.	Yes	OX-DICT			
89	armrest	noun	Yes	The devices on either side of the seat which support the driver's arms when he is seated.		Yes	JCB/78/764/EEC Annex 4.1	seat armrests	(noun)	
91	ARV	abbrv	Yes	Abbreviation for 'auxiliary relief valve'						
96	attachment	noun	Yes	Assembly of components that can be mounted onto the base machine or equipment for specific use.		Yes	ISO 6746-2:2003(E/F) 3.5			
106	auger	noun	Yes	A tool resembling a large corkscrew, for boring holes....		Yes	OX-DICT	earth drill	(noun)	
107	axle	noun	Yes	A rod or spindle passing through the centre of a wheel or group of wheels.		Yes	OX-DICT			
118	B									
119	backrest	noun	Yes	The almost vertical area of the seat supporting the		Yes	78/764/EEC Annex			
120										

SKOS model of the JCB Terminology



The screenshot shows a Java Swing application window titled "JCBTerminology.skos.owl". The window contains the RDF XML code for the SKOS model of the JCB Terminology. The code is organized into three main sections, each starting with a comment indicating the URL of the individual:

- Section 1:** <!-- http://skoseexample.com/JCBTerminology#4af4f3c653c9dcc29846c4490e26c3cd -->
An owl:NamedIndividual node with rdf:type skos:Concept. It has an editorialNote (TP) and a definition (The linkage (found on an excavator arm) that transfers the movement of the bucket ram to the bucket/quick hitch). It also has a prefLabel (bucket linkage) and an altLabel (tipping lever).
- Section 2:** <!-- http://skoseexample.com/JCBTerminology#4b427b9686cf5554ab0925cff7fc03f5 -->
An owl:NamedIndividual node with rdf:type skos:Concept. It has an editorialNote (OX-DICT), an example (Tile adhesive), a prefLabel (adhesive), and an altLabel (glue).
- Section 3:** <!-- http://skoseexample.com/JCBTerminology#4c75d4c027d99f18d240d2bcfa82b68 -->
An owl:NamedIndividual node with rdf:type skos:Concept. It has a definition (An assembly consisting of wires and connectors), an altLabel (cable harness), and a prefLabel (harness).

The status bar at the bottom of the window indicates "U:--- JCBTerminology.skos.owl 28% L1216 (nXML Valid)".

JCBTerminology.skos.owl in Protégé

The screenshot shows the Protégé ontology editor interface with the following details:

- Tab Bar:** Annotation Properties, Individuals, SKOS view (selected), OWLViz, DL Query, OntoGraf, SPARQL Query, Ontology Differences.
- Left Sidebar:** Class hierarchy, showing a tree structure starting from Thing, with 'adhesive' selected.
- Annotations Tab:** Shows annotations for the concept 'adhesive'.
 - Annotations:** altLabel [type: XMLLiteral] glue
 - definition:** A substance used for sticking objects or materials together; glue.
 - editorialNote:** OX-DICT
 - example:** Tile adhesive.
 - prefLabel:** adhesive
- Other Panels:** Description: adhesive (Types: Concept), Property assertions: adhesive (Object property assertions, Data property assertions).
- Bottom Status Bar:** No Reasoner set. Select a reasoner from the Reasoner menu. Show Inferences

JCB Terminology in SKOS

SKOS Advantages:

Separating entity identity (URL) from its label (preferred term). Preferred term can be altered or replaced by a synonym without changing the entity identity.

SKOS allows for multilingual labels – this means that preferred terms can be defined for many different languages.

SKOS documents can easily be linked to other SKOS or OWL documents – more details will follow.

Part Index List (PIL)

The PIL Structure is the 3-Level navigation (classification) structure created for service information procedures in the service manual and for service bulletins.

3 Levels roughly correspond to the notions of: System, Assembly, Part.

Relationships between the levels differ and are not specified in the original PIL document.

JCB provided relationships between Levels 2-1 and 3-2 in PIL, thus creating the so-called Semantic PIL.

The relationships between the parts that occur in the Semantic PIL are: [kind_of](#), [part_of](#), [functionally_part_of](#), [mechanism_of](#), [power_source_of](#), [diagnoses](#), [carries](#) and [contains](#).

Original PIL Excel

A	B	C
12 03 - Attachments	24 - Auxiliary Circuit	00 - General 06 - Handheld Hydraulic Tool 09 - Bidirectional
16 03 - Attachments	27 - Bucket	00 - General 03 - Standard 06 - Jaw 15 - Grapple 90 - Teeth
22 03 - Attachments	30 - Shovel	00 - General 03 - Standard 06 - Side-Tip 09 - 4-in-1 12 - 6-in-1 15 - Grapple 18 - Grading 21 - Beet Basket 60 - Toeplate 90 - Teeth
33 03 - Attachments	33 - Fork	00 - General 03 - Carriage 06 - Standard 09 - Manure 12 - Pallet 15 - Grass 18 - Bale 60 - Frame 90 - Level Indicator 93 - Guard
44 03 - Attachments	36 - Side-shift Carriage	

Semanticised PIL Excel

	B	C	D	E	F
1	1st Level - System		2nd Level - Assembly	3rd Level - Part	Notes
105	03 - Attachments	power source	24 - Auxiliary Circuit	00 - General 06 - Handheld Hydraulic Tool 09 - Bidirectional	
106			kind_of		
107			kind_of		
108					
109	03 - Attachments	kind_of	27 - Bucket	00 - General 03 - Standard 06 - Jaw 15 - Grapple 90 - Teeth	
110			kind_of		
111			kind_of		
112			kind_of		
113			part_of		
114					
115	03 - Attachments	kind_of	30 - Shovel	00 - General 03 - Standard 06 - Side-Tip 09 - 4-in-1 12 - 6-in-1 15 - Grapple 18 - Grading 21 - Beet Basket 60 - Toeplate 90 - Teeth	
116			kind_of		
117			kind_of		
118			kind_of		
119			kind_of		Clam Shovel
120			kind_of		Clam Shovel
121			kind_of		
122			kind_of		
123			kind_of		
124			part_of		
125			part_of		
126	03 - Attachments	kind_of	33 - Fork	00 - General 03 - Carriage 06 - Standard 09 - M	
127			carries		
128			kind_of		
129					

Modelling PIL in OWL

PIL Concepts

For every PIL code/concept, an **OWL class** is created. Every class is uniquely defined by its **URL**, e.g.

```
<owl:Class rdf:about="http://www.semanticweb.org/ontologies/PIL.owl#PIL_03_30_03">
```

kind_of relationship

The **kind_of** relationship is modelled by means of **subClassOf** axioms in OWL.

```
Standard Shovel subClassOf Shovel
```

Meaning: Every “Standard Shovel” is a “Shovel”. This means that everything that holds for a Shovel, holds for the “Standard Shovel” as well.

Modelling PIL in OWL

Other relationships (part_of, functionally_part_of, mechanism_of, ...)

For the part_of relationship, a new OWL object property part_of is created. Similar is done for other relationships. We state:

Shovel Tooth subClassOf part_of some Shovel

Meaning: Every “Shovel Tooth” is a part of some “Shovel”.

Note: This does not imply that every “Shovel” has a part “Shovel Tooth”!
In order to model this, we would have to state:

has_part = inverse(part_of)
Shovel subClassOf has_part some Shovel Tooth

Then it would be inferred that every Shovel, including Standard, Side-Tip, 4-in-1, 6-in-1 etc. has a part Shovel Tooth.

As the relationships between PIL levels were stated only in one direction in the Semantic PIL, we adopted a cautious approach and modelled them in the same way in order to avoid possible unwanted inferences. This means that we have not introduced the has_part and other inverse properties in our OWL model.

Modelling PIL in OWL

Transitive/indirect part_of

In order to model both the direct and the transitive `part_of` relationship, we introduce a new property `trans_part_of` and state:

```
part_of subPropertyOf trans_part_of
transitive(trans_part_of)
```

OWL reasoners can handle transitive relationships.

Modelling PIL in OWL

Keeping track of the original PIL structure

OWL annotations are used to model the original PIL 3-Level Structure. We define new OWL annotation properties pilCode, pilLevel, pilLabel, and pilParent.

We can state for the Standard Shovel

(OWL class with the URL http://www.semanticweb.org/ontologies/PIL.owl#PIL_03_30_03)

pilCode: „03_30_03”
pilLevel: „PART”
pilLabel: „Standard”

Besides built-in datatypes such as strings, URLs can be used as annotation values. Thus we can use one OWL class (URL) to annotate another OWL class:

pilParent: http://www.semanticweb.org/ontologies/PIL.owl#PIL_03_30_00
(the URL of “Shovel”)

Modelling PIL in OWL

Mapping PIL to the JCB Terminology

Similarly, OWL Annotations are used to map PIL concepts to the JCB Terminology terms. We define a new annotation property [terminology](#).

PIL concept “Standard Shovel” is mapped to the JCB term “shovel”.

Similar to pilParent, terminology values are URLs of the JCB Terminology SKOS concepts

terminology: <http://skosexample.com/JCBTerminology#2c997ee34e3a72c17421d81c1c3716ae>
(the URL of the SKOS concept “shovel”)

PIL.owl in Text Editor: Standard Shovel

```
<!-- http://www.semanticweb.org/ontologies/PIL.owl#PIL_03_30_03 -->

<owl:Class rdf:about="http://www.semanticweb.org/ontologies/PIL.owl#PIL_03_30_03">
  <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Standard Shovel [Attachment]</rdfs:label>
  <rdfs:subClassOf rdf:resource="http://www.semanticweb.org/ontologies/PIL.owl#PIL_03_30_00"/>
  <www:pilCode rdf:datatype="http://www.w3.org/2001/XMLSchema#string">03_30_03</www:pilCode>
  <www:pilLevel rdf:datatype="http://www.w3.org/2001/XMLSchema#string">PART</www:pilLevel>
  <www:origin rdf:datatype="http://www.w3.org/2001/XMLSchema#string">PIL_ORIGINAL</www:origin>
  <www:pilLabel rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Standard</www:pilLabel>
  <www:terminology rdf:resource="http://skosexample.com/JCBTerminology#2c997ee34e3a72c17421d81c1c3716ae"/>
  <www:pilParent rdf:resource="http://www.semanticweb.org/ontologies/PIL.owl#PIL_03_30_00"/>
</owl:Class>
<owl:Axiom>
  <www:origin rdf:datatype="http://www.w3.org/2001/XMLSchema#string">SEMANTIC_PIL</www:origin>
  <owl:annotatedTarget rdf:resource="http://www.semanticweb.org/ontologies/PIL.owl#PIL_03_30_00"/>
  <owl:annotatedSource rdf:resource="http://www.semanticweb.org/ontologies/PIL.owl#PIL_03_30_03"/>
  <owl:annotatedProperty rdf:resource="http://www.w3.org/2000/01/rdf-schema#subClassOf"/>
</owl:Axiom>
```

PIL.owl in Protégé: Standard Shovel

The screenshot shows the Protégé ontology editor interface with the following details:

- Active Ontology:** PIL (<http://www.semanticweb.org/ontologies/PIL.owl>)
- Entities Tab:** Entities
- Classes Tab:** Classes
- Object Properties Tab:** Object Properties
- Data Properties Tab:** Data Properties
- Annotation Properties Tab:** Annotation Properties
- SKOS view Tab:** SKOS view
- Individuals Tab:** Individuals
- OWLviz Tab:** OWLviz
- DL Query Tab:** DL Query

Class hierarchy: Standard Shovel [Attachment]

- 'Hook [Attachment]'
- 'Hose Reel [Attachment]'
- 'Impact Wrench [Attachment]'
- 'Jib [Attachment]'
- 'Mower [Attachment]'
- 'Patch Planer [Attachment]'
- 'Pole Planter [Attachment]'
- 'Pump [Attachment]'
- 'Rake [Attachment]'
- 'Rockbreaker [Attachment]'
- 'Rotary Drill [Attachment]'
- 'Rotary Tiller [Attachment]'
- Shovel [Attachment]**
 - '4-in-1 Shovel [Attachment]'
 - '6-in-1 Shovel [Attachment]'
 - 'Beet Basket Shovel [Attachment]'
 - 'Grading Shovel [Attachment]'
 - 'Grapple Shovel [Attachment]'
 - 'Side-Tip Shovel [Attachment]'
 - Standard Shovel [Attachment]**
- 'Spike [Attachment]'
- 'Spreader [Attachment]'
- 'Sweeper Collector [Attachment]'
- 'Trailer [Attachment]'
- 'Trencher [Attachment]'
- 'Weed Cutter [Attachment]'
- 'Winch [Attachment]'
- 'Work Platform [Attachment]'

Annotations: Standard Shovel [Attachment]

- label** [type: string]
Standard Shovel [Attachment]
- origin** [type: string]
PIL_ORIGINAL
- pilCode** [type: string]
03_30_03
- pilLabel** [type: string]
Standard
- pilLevel** [type: string]
PART
- pilParent**
 - Shovel [Attachment]**
- terminology**
 - shovel**

Description: Standard Shovel [Attachment]

- SubClass Of**
 - Shovel [Attachment]**

Reasoner active Show Inferences

PIL.owl in Text Editor: Shovel Tooth

```
<!-- http://www.semanticweb.org/ontologies/PIL.owl#PIL_03_30_90 -->

<owl:Class rdf:about="http://www.semanticweb.org/ontologies/PIL.owl#PIL_03_30_90">
  <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Shovel Tooth [Attachment]</rdfs:label>
  <rdfs:subClassOf rdf:resource="http://www.semanticweb.org/ontologies/PIL.owl#77ec14e9c85fac87542b984e8988b422"/>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="http://www.semanticweb.org/ontologies/PIL.owl#part_of"/>
      <owl:someValuesFrom rdf:resource="http://www.semanticweb.org/ontologies/PIL.owl#PIL_03_30_00"/>
    </owl:Restriction>
  </rdfs:subClassOf>
  <www:pilCode rdf:datatype="http://www.w3.org/2001/XMLSchema#string">03_30_90</www:pilCode>
  <www:pilLevel rdf:datatype="http://www.w3.org/2001/XMLSchema#string">PART</www:pilLevel>
  <www:origin rdf:datatype="http://www.w3.org/2001/XMLSchema#string">PIL_ORIGINAL</www:origin>
  <www:pilLabel rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Teeth</www:pilLabel>
  <www:terminology rdf:resource="http://skosexample.com/JCBTerminology#28b26be59c986170c572133aaace31c2"/>
  <www:pilParent rdf:resource="http://www.semanticweb.org/ontologies/PIL.owl#PIL_03_30_00"/>
</owl:Class>
<owl:Axiom>
  <www:origin rdf:datatype="http://www.w3.org/2001/XMLSchema#string">SEMANTIC_PIL</www:origin>
  <owl:annotatedSource rdf:resource="http://www.semanticweb.org/ontologies/PIL.owl#PIL_03_30_90"/>
  <owl:annotatedProperty rdf:resource="http://www.w3.org/2000/01/rdf-schema#subClassOf"/>
  <owl:annotatedTarget>
    <owl:Restriction>
      <owl:onProperty rdf:resource="http://www.semanticweb.org/ontologies/PIL.owl#part_of"/>
      <owl:someValuesFrom rdf:resource="http://www.semanticweb.org/ontologies/PIL.owl#PIL_03_30_00"/>
    </owl:Restriction>
  </owl:annotatedTarget>
</owl:Axiom>
<owl:Axiom>
  <www:origin rdf:datatype="http://www.w3.org/2001/XMLSchema#string">AUTOMATICALLY_GENERATED</www:origin>
  <owl:annotatedTarget rdf:resource="http://www.semanticweb.org/ontologies/PIL.owl#77ec14e9c85fac87542b984e8988b422"/>
  <owl:annotatedSource rdf:resource="http://www.semanticweb.org/ontologies/PIL.owl#PIL_03_30_90"/>
  <owl:annotatedProperty rdf:resource="http://www.w3.org/2000/01/rdf-schema#subClassOf"/>
</owl:Axiom>
```

PIL.owl in Protégé: Shovel Tooth

Class hierarchy: 'Shovel Tooth [Attachment]'

- ▶ Tank
- ▶ Thermostat
- ▶ Thinner
- ▶ **Throttle**
- ▶ **'Throttle Position'**
- ▶ Thrust
- ▶ Tilt
- ▶ Tipper Body
- ▶ **Tool [Body/Framework]**
- ▶ **Tool Carrier [Attachment]**
- ▶ **Tool Toolbox [Body/Framework]**
- ▶ **Tooth**
 - ▶ **'Bucket Tooth [Attachment]'**
 - ▶ **'Shovel Tooth [Attachment]'**
 - ▶ Top Grille
 - ▶ Topper
 - ▶ **'Torque Converter [Driveline]'**
 - ▶ Touch Screen
 - ▶ Touch-Up
- ▶ **'Tow Hitch [Attachment]'**
- ▶ **'Track [Driveline]'**
- ▶ **'Track Gearbox Sprocket [Driveline]'**
- ▶ Transmission
- ▶ **'Transmission Disconnect'**
- ▶ **'Transmission Flush'**
- ▶ Travel-Speed Selector
- ▶ Trunnion
- ▶ Turbine

Annotations: 'Shovel Tooth [Attachment]'

Annotation	Type	Value	Buttons
label	[type: string]	Shovel Tooth [Attachment]	@ X O
origin	[type: string]	PIL_ORIGINAL	@ X O
pilCode	[type: string]	03_30_90	@ X O
pilLabel	[type: string]	Teeth	@ X O
pilLevel	[type: string]	PART	@ X O
pilParent	[type: string]	'Shovel [Attachment]'	@ X O
terminology	[type: string]	tooth	@ X O

Description: 'Shovel Tooth [Attachment]'

Property	Value	Buttons
SubClass Of	<ul style="list-style-type: none">▶ part_of some 'Shovel [Attachment]'▶ Tooth	? @ X O ? @ X O

Enriching the PIL

The 3-Level PIL structure is rather limiting as it allows only to define a relationship between a PIL concept and its parent.

However, there are many more implicit relationships hidden in PIL labels. While a human could capture some of them, it would be very difficult for a machine to do so.

Example 1:

03 Attachments

kind_of 03_30 Shovel

kind_of 03_30_12 6-in-1

6-in-1 Shovel



part_of ?

30 Hydraulics

part_of 30_63 Hose Burst Check Valve (HBCV)

kind_of 03_63_39 6-in-1 Shovel

6-in-1 Shovel Hose Burst Check Valve

Enriching the PIL (more implicit relationships)

Example 2:

30 Hydraulics

part_of 30_00_03 Smooth Ride System (SRS)

30 Hydraulics

part_of 30_18 Accumulator

kind_of 30_18_06 SRS



Example 3:

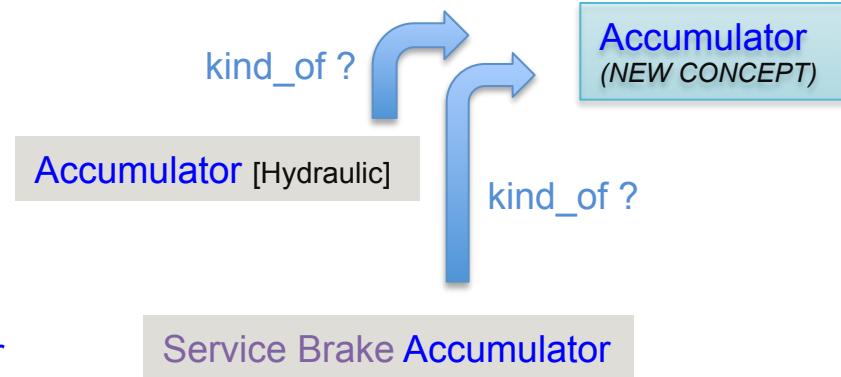
30 Hydraulics

part_of 30_18 Accumulator

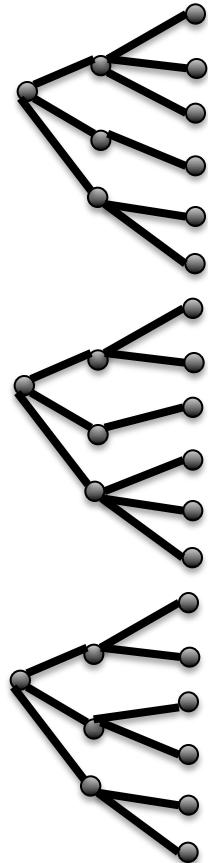
24 Brakes

part_of 24_03 Service Brake

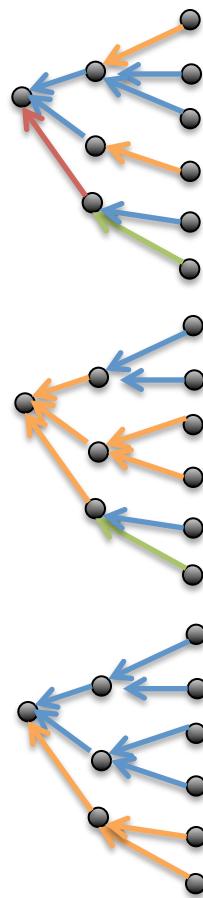
part_of 24_03_24 Accumulator



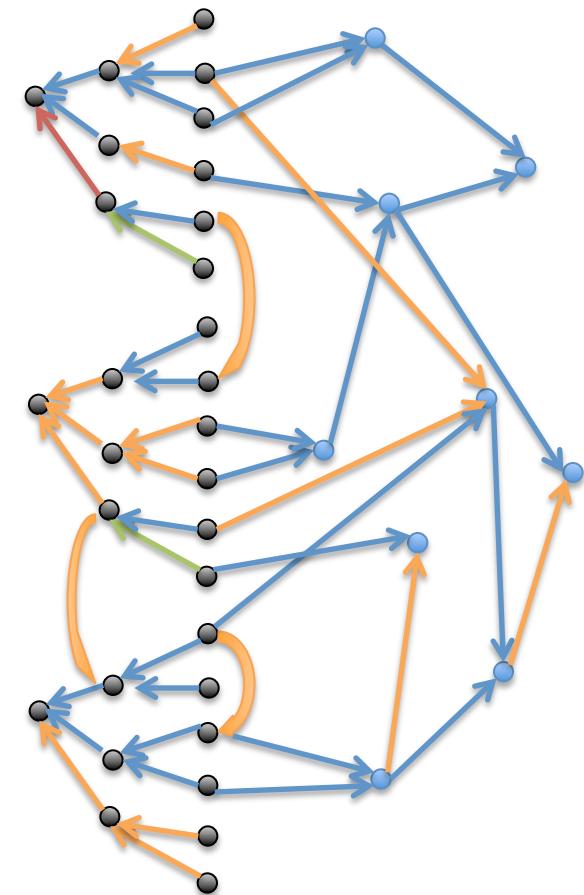
PIL (tree)



Semantic PIL



Enriched PIL (graph)



PIL Working Labels

In the previous examples we replaced original: short-hand, context-dependent PIL labels with new: longer, normalised, context-independent ones (shown in the grey boxes). We will refer to them as working labels. By doing a lexical analysis of working labels we can generate and suggest new concepts and relationships.

Automatic generation of working labels

(1) Normalise (norm(label)) PIL labels on levels 1-3 :

- convert them to a singular form (Attachments-> Attachment, Work Lights -> Work Light)
- resolve acronyms (SRS -> Smooth Ride System)

(2) Working Label of PIL Level 3 concepts constructed to capture the context of Level 2:

Merge normalised Level 3 labels with the normalised label of their parent on Level 2, depending on the relationship between the concepts.

- If the relationship is **part_of**, the working label is:
 $\text{norm(Level2Label)} \text{ norm(Level3Label)}$

Accumulator (part_of Service Brake): Service Brake Accumulator

Harness (part_of Work Lights): Work Light Harness

PIL Working Labels (continued)

- If the relationship is **kind_of**, the working label is usually:
`norm(Level3Label) norm(Level2Label)`

`6-in-1 (kind_of Shovel): 6-in-1 Shovel`

`SRS (kind_of Accumulator): Smooth Ride System Accumulator`

Note: Sometimes we get strange results:

`Accelerator Pedal (kind_of Controls): Accelerator Pedal Control`

We make **exceptions** in cases where Level-3 label and Level-2 label **end with the same word** (here we recognise that Level-3 label is not a qualifier of Level- 2 label as assumed in the previous cases)

If Level-3 label appears only once in PIL, we do not change it:

`Rear Work Light (kind_of Exterior Light): Rear Work Light`

If Level-3 label appears more than once in PIL, we carefully merge it with Level-2 label:

`Mechanical Quickhitch (kind_of Excavator Arm Quickhitch): Mechanical Excavator Arm Quickhitch`

`Mechanical Quickhitch (kind_of Lift Arm Quickhitch): Mechanical Lift Arm Quickhitch`

PIL Full Labels

Full labels are obtained by adding the normalised PIL Level 1 labels in square brackets to the working labels of PIL Level 2-3 concepts. While two different PIL concepts may have the same working labels, the full labels are unique as they also capture the context of the PIL Level 1 for every concept. We are not using full labels in the lexical analysis, but simply for disambiguation and display.

Full labels needed for disambiguation (PIL Level 1 defines context):

Tank [Hydraulic]

Tank [Fuel]

Gearbox [Hydraulic]

Switch [Engine]

Full labels not really needed for disambiguation:

Exterior Light [Electrical]

Smooth Ride System Accumulator [Hydraulic]

Alternatively, we could have added PIL 1 Labels as prefixes/suffixes to all 2- and 3- Level labels, but decided against it as it would have largely complicated extraction of new relationships.

Lexical Analysis of Labels

By doing lexical analysis of working labels we can suggest new concepts and relationships.

It holds often that:

(1) YX kind_of X

Service Brake Accumulator kind_of Accumulator
Accumulator [Hydraulic] kind_of Accumulator



create groupings (new concepts) and new kind_of relationships

(2) YX part_of Y

Smooth Ride System Accumulator part_of Smooth Ride System



create part_of relationships and new part candidate concepts

Enriching the PIL (continued)

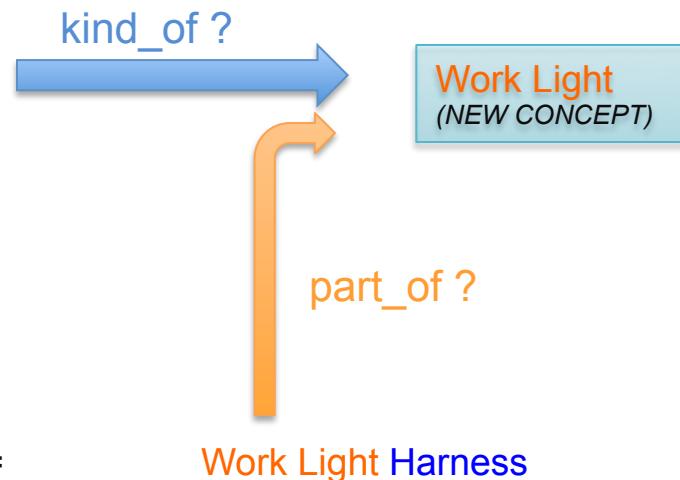
Example 4:

33 Electrical

part_of 33_42 Exterior Light

kind_of 33_42_15 Front Work Light

kind_of 33_42_16 Rear Work Light



Enriching the PIL (continued)

Example 5:

30 Hydraulics

part_of 30_42 Gearbox [Hydraulic]

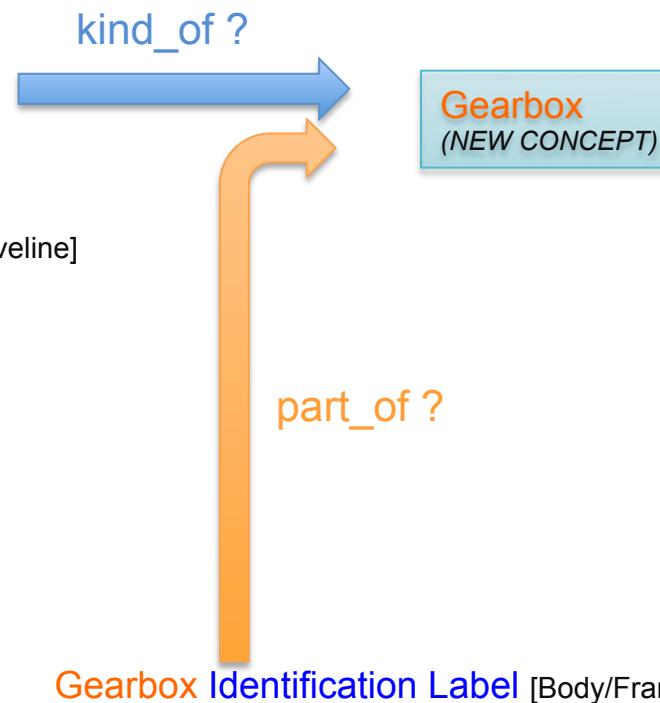
27 Driveline

part_of 27_03 Manual Gearbox [Driveline]
part_of 27_06 Semi-Automatic Gearbox [Driveline]
part_of 27_09 Automatic Gearbox [Driveline]
part_of 27_11 Transfer Gearbox [Driveline]
part_of 27_12 Transfer Gearbox [Driveline]
part_of 27_17 PTO Gearbox [Driveline]

06 Body and Framework

part_of 06_63 Identification Label

kind_of 06_63_12 Gearbox



Enriching the PIL (continued)

Example 6:

30 Hydraulics

part_of 30_60 Flow Control Valve

kind_of 33_60_81 Engine Fan = Engine Fan Flow Control Valve

part_of ?



Engine Fan
(NEW CONCEPT)

33 Electrical

part_of 33_33 Console Switch

kind_of 33_33_92 Engine Fan

=

Engine Fan Console Switch



part_of ?

Generated Concepts

We generate new concepts in two ways:

(1) New part candidates: Normalised PIL Level-3 labels that may denote missing part_of fillers

33 Electrical

part_of 33_33 Console Switch

kind_of 33_33_92 Engine Fan Engine Fan Console Switch

As Engine Fan is not in the set of working labels of original PIL concepts, we introduce a new concept with the label Engine Fan. This concept will be used to introduce a new part_of relationship.

Obviously, we will introduce some unwanted concepts this way, whose labels do not describe part names, but simply qualify the term on PIL Level 3. Examples of these are “6-in-1” (qualifies Shovel), “Standard” (qualifies Shovel, Fork, Bucket), location qualifiers such as “Rear”, “Front”, “Side” or various Sensor qualifiers including “Fuel Temperature” and “Oil Pressure”. We recognise and exclude some of these qualifiers programmatically but the rest needs to be manually reviewed and rejected if necessary.

Generated Concepts (continued)

(2) Groupings:

For any two PIL concepts (original and new), we compute the **maximal common suffix** of their **working labels** (it can be equal to either of them), and create a new concept. Moreover, we create **new concepts for non-maximal common suffices** that correspond to normalised original PIL labels.

Gearbox Filter is the maximal common suffix for:

Manual Gearbox Filter

Automatic Gearbox Filter

Semi-Automatic Gearbox Filter

Transfer Gearbox Filter

Manual Gearbox Filter

Automatic Gearbox Filter

Semi-Automatic Gearbox Filter

Transfer Gearbox Filter

We will also introduce a new concept for the non-maximal common suffix Filter because it is the original PIL Level-3 label in the above examples.

These new concepts will be used to group existing PIL concepts. Those that do not make sense or are redundant (equivalent to another PIL concept, original or new) will be rejected or merged during the reviewing process.

Generated Relationships

Generate

- (1) a **kind_of** relationship between an **original** and a **new** concept
if the working label of an original PIL concept is equal or ends with the
label of the new concept:

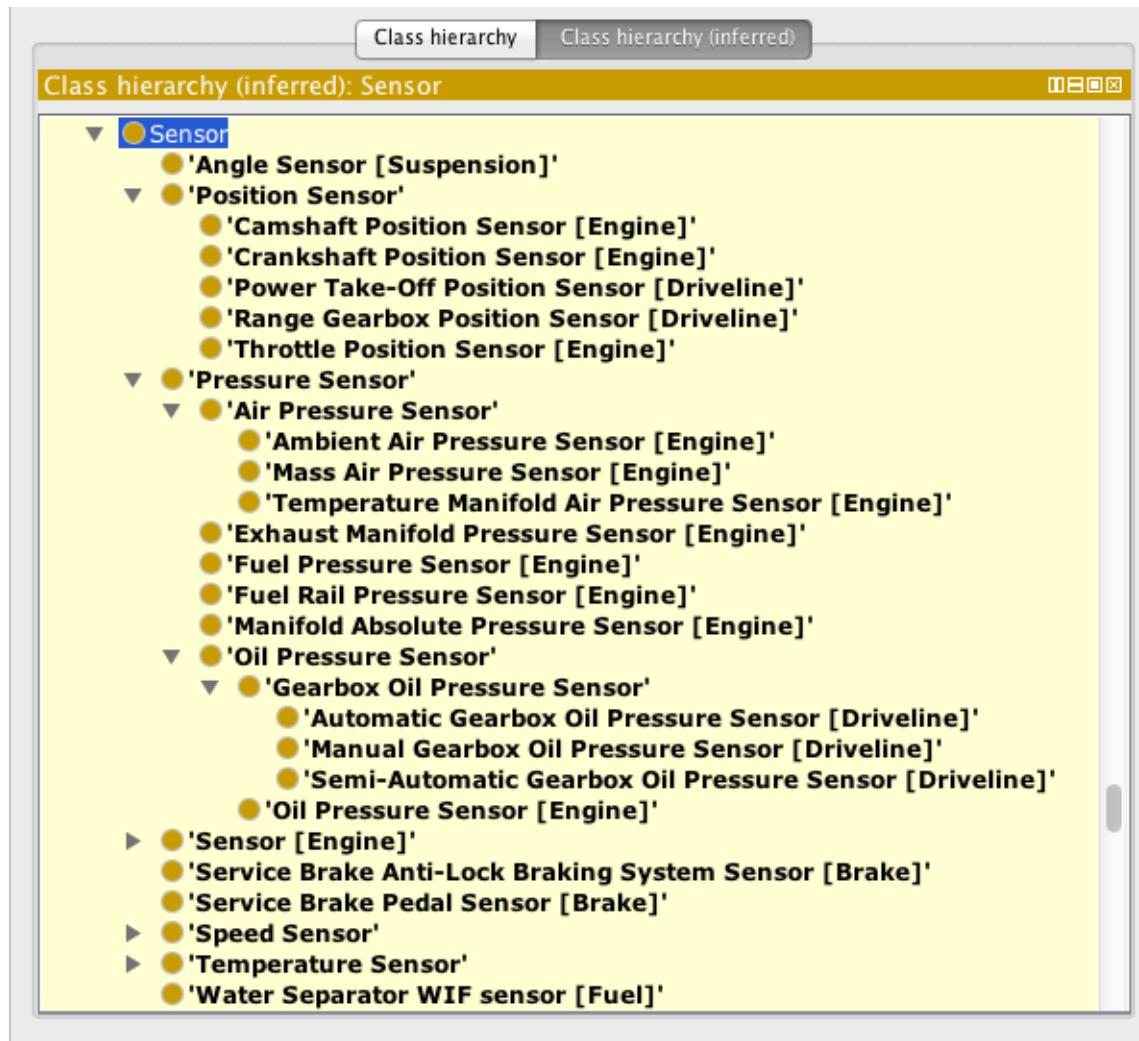
Sensor [Engine]	kind_of Sensor
Angle Sensor [Suspension]	kind_of Sensor
Range Gearbox Position Sensor [Driveline]	kind_of Position Sensor

- (2) a **kind_of** relationship **between new** concepts
if one's label is a proper suffix of the other's label:

Position Sensor **kind_of** Sensor

Note: in order to minimise the number of generated **kind_of** relationships to be manually reviewed, if we generated “X kind_of Y” and “Y kind_of Z”, we will not generate “X kind_of Z”. Similar is done if “X kind_of Y” is already stated in the Semantic PIL.

Sensor Classification in the Enriched PIL



Generated Relationships (continued)

- (3) a **part_of** relationship between any two concepts C1 and C2 if C2's (working) label is a proper prefix of C1's (working) label.

Note: If C1 is an original PIL concept, we will generate the relationship only if there is no original PIL concept C3 whose working label equals the (working) label of C2 and "C1 part_of C3" is already stated in the Semantic PIL. Moreover, if C2 is an original PIL concept, we generate the relationship only if there is no new (grouping) concept with the same label as C2.

6-in-1 Shovel	Hose Burst Check Valve [Hydraulic]	part_of	6-in-1 Shovel	[Attachment]
Work Light Harness	[Electrical]	part_of	Work Light	
Gearbox Filter		part_of	Gearbox	

The generated relationships that do not make sense will be rejected during the reviewing process. In some cases, it will make sense to rename **part_of** into another relationship property, such as **functionally_part_of**.

- (4) a **REL** relationship between the new concept **C1** and the original concept **C2** if all original PIL concepts that belong to the new grouping concept **C1** are related by the same relationship property **REL** with **C2** in the Semantic PIL.

Exterior Light [Electrical]
kind_of Front Side Light [Electrical]
kind_of Rear Side Light [Electrical]



Exterior Light [Electrical]
kind_of Side Light
kind_of Front Side Light [Electrical]
kind_of Rear Side Light [Electrical]

Reviewing Process

The generated concepts and relationships have to be **manually reviewed**.

For every **generated concept and relationship**, it needs to be stated if it should be **included in the ontology (Yes or No)**. Only non-rejected concepts and relationships will be included in the revised version of PIL ontology. All relationships of rejected concepts will be rejected as well.

It can also be stated that the **generated concept is the same as another (original or new) PIL concept**. In that case, the concepts will be merged and the non-rejected relationships of the redundant concept will still be included.

For example, if we have generated a new concept **Main Frame** with the relationships

PIL_06_03_06 Chassis Main Frame [Body/Framework] kind_of Main Frame
PIL_33_12_03 Main Frame Harness [Electrical] part_of Main Frame

and we state that “**Main Frame same_as PIL_06_03_06**”, we will obtain:

PIL_33_12_03 Main Frame Harness [Electrical] part_of
PIL_06_03_06 Chassis Main Frame [Body/Framework]

Recording Concept/Relationship Origin in PIL.owl

For every newly generated PIL concept we generate a new unique URL and create a new OWL Class with that URL.

In order to record the origin of PIL concepts, we [annotate OWL Classes](#) using the new annotation property [origin](#). We may state

[origin: PIL_ORIGINAL](#) or

[origin: AUTOMATICALLY_GENERATED](#) or

[origin: AUTOMATICALLY_GENERATED_APPROVED](#)

Similarly, we [annotate OWL axioms](#) (relationships) to record their origin.

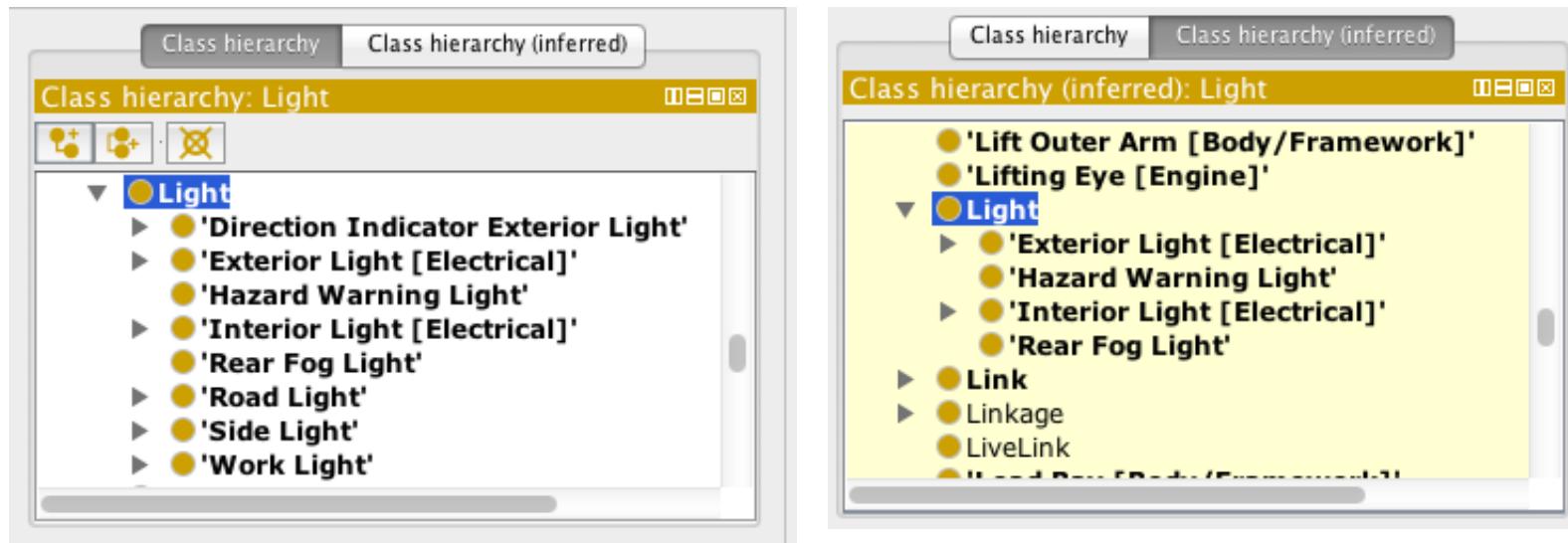
```
<!-- http://www.semanticweb.org/ontologies/PIL.owl#2fa3609fafd409f1344229d78b4a1f9f -->

<owl:Class rdf:about="http://www.semanticweb.org/ontologies/PIL.owl#2fa3609fafd409f1344229d78b4a1f9f">
  <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Position Sensor</rdfs:label>
  <rdfs:subClassOf rdf:resource="http://www.semanticweb.org/ontologies/PIL.owl#6b185256c71c1aec263c6e22bf8ef6b"/>
  <www:origin rdf:datatype="http://www.w3.org/2001/XMLSchema#string">AUTOMATICALLY_GENERATED</www:origin>
  <www:terminology rdf:resource="http://skosexample.com/JCBTerminology#a5fe26d5d09b736a77f4345e9f80b951"/>
</owl:Class>
<owl:Axiom>
  <www:origin rdf:datatype="http://www.w3.org/2001/XMLSchema#string">AUTOMATICALLY_GENERATED</www:origin>
  <owl:annotatedSource rdf:resource="http://www.semanticweb.org/ontologies/PIL.owl#2fa3609fafd409f1344229d78b4a1f9f"/>
  <owl:annotatedTarget rdf:resource="http://www.semanticweb.org/ontologies/PIL.owl#6b185256c71c1aec263c6e22bf8ef6b"/>
  <owl:annotatedProperty rdf:resource="http://www.w3.org/2000/01/rdf-schema#subClassOf"/>
</owl:Axiom>
```

Navigation and Reasoning

OWL Reasoner is used to compute the inferred PIL concept (`kind_of`) hierarchy. The inferred PIL hierarchy is “cleaner” than the original one as it correctly shows children (direct descendants) and parents (direct ascendants). OWL reasoner also infers new `kind_of` relationships (computing all descendants and ascendants for a given concept).

Example:



In the inferred hierarchy `Light` has only 4 children. The others from the original hierarchy such as `Side Light` are actually children of `Exterior Light [Electrical]`.

Navigation and Reasoning

For every PIL concept, OWL reasoning is used to answer questions:

What is it a part of? What is it a mechanism of? What is it a power source of? ...

By using reasoning we infer new relationships as well, i.e. we get more than what we explicitly stated in the PIL ontology.

Pick-Up Tow Hitch is a **mechanism of** Attachment:

Pick-Up Tow Hitch **kind_of** Tow Hitch
Tow Hitch **mechanism_of** Attachment

Rear Door Gas Strut is a **part of** Body/Framework:

Rear Door Gas Strut **kind_of** Gas Strut
Gas Strut **part_of** Body/Framework

OWL reasoning also helps us answer the question:

What is a PIL concept an indirect part of?

Stabilizer Leg is an **indirect part of** Body/Framework:

Stabilizer Leg **part_of** Stabilizer
Stabilizer **part_of** Body/Framework

Navigation and Reasoning

Structural axiom look-up (querying)

Question: What are the parts of X?

As answer we give all Y such that X `subClassOf part_of some` Y is an axiom in the PIL ontology. Here we do not use reasoning as we decided against introducing the property `has_part` to the ontology.

Combination of OWL reasoning and structural axiom look-up:

Question: Is X an assembly or a part? (Something is an assembly if it has parts)

We use reasoning to compute all `kind_of` descendants of X. If X or one of its descendants has parts (checked by a structural look up), then we conclude that X is an assembly.

Reading OWL annotations

Task: build the original PIL navigation tree

The original 3-Level PIL structure and labels are stored in `PIL.owl` by means of OWL annotations – these are used to build the original PIL navigation tree.

Advantages of Modelling PIL in OWL

Separating identity from the label. Alternative labels (synonyms, keywords, translations to other languages) can easily be added.

Generating a much richer and more flexible structure.

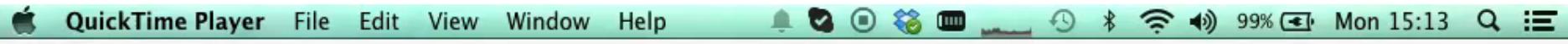
Original 3-Level PIL structure is still captured.

Explicitly defining relationships between concepts.

Knowing when two concepts are in a kind_of relationships enables creation of a rich concept (kind_of) hierarchy which is useful for navigation and search purposes.

OWL reasoning can be used to infer new relationships from the existing ones.

Easy linking to other OWL/SKOS resources, e.g. JCB Terminology.



AVAnT

Quick Access

PIL SVR

▶03_00_00 Attachments
▶06_00_00 Body and Framework
▶09_00_00 Operator Station
▶12_00_00 HVAC
▶15_00_00 Engine
▶18_00_00 Fuel
▶21_00_00 Cooling
▶24_00_00 Brakes
▶25_00_00 Steering
▶27_00_00 Driveline
▶28_00_00 Suspension
▶30_00_00 Hydraulics
▶33_00_00 Electrical
▶72_00_00 Fasteners and Fixings
▶75_00_00 Consumables

A screenshot of the AVAnT application window. The title bar says "AVAnT". The menu bar includes "File", "Edit", "View", "Window", "Help", and a "Quick Access" search field. On the left is a sidebar with tabs for "PIL" and "SVR", currently showing "PIL". The main pane displays a hierarchical list of categories, each preceded by a triangle icon indicating they are expandable. The categories listed are: 03_00_00 Attachments, 06_00_00 Body and Framework, 09_00_00 Operator Station, 12_00_00 HVAC, 15_00_00 Engine, 18_00_00 Fuel, 21_00_00 Cooling, 24_00_00 Brakes, 25_00_00 Steering, 27_00_00 Driveline, 28_00_00 Suspension, 30_00_00 Hydraulics, 33_00_00 Electrical, 72_00_00 Fasteners and Fixings, and 75_00_00 Consumables. A cursor arrow is visible near the bottom left of the sidebar area.

Subject Version Revision (SVR)

The SVR Structure is the 3-Level navigation (classification) structure used for parts pages in the JCB parts catalogue.

3 Levels are: Section, Subsection, Subject.

Relationships between the levels have loose semantics: a section is broader than the related subsections, and a subsection is broader than the related subjects.

We used the [SKOS pre-defined broader/narrower relationships](#) to model them.

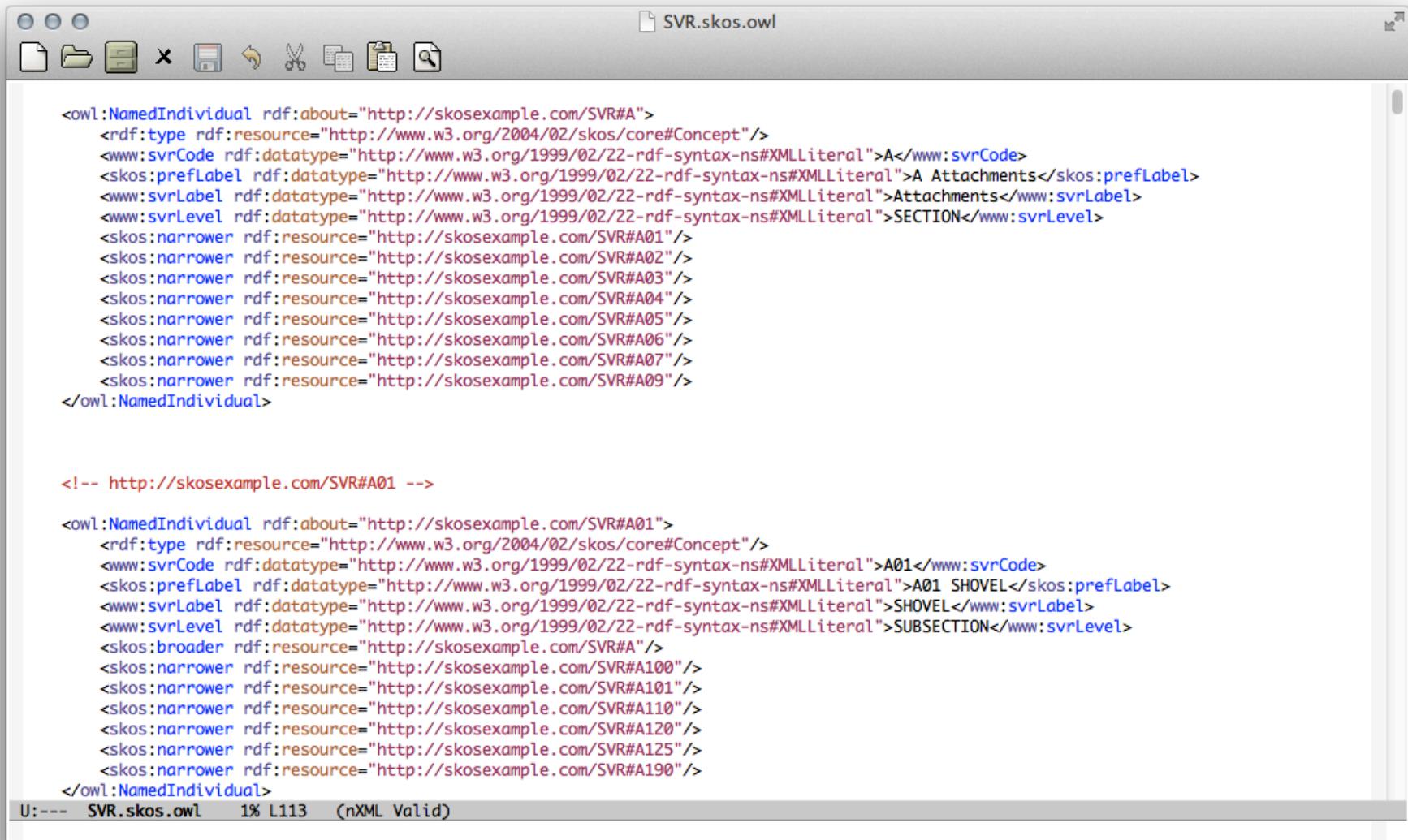
We defined new annotation properties to store SVR Label, SVR Level and SVR codes.

SVR in Excel

The screenshot shows a Microsoft Excel spreadsheet titled "Book1.xlsx (Read-Only)". The ribbon menu is visible at the top, showing tabs for Home, Layout, Tables, Charts, SmartArt, Formulas, Data, and Review. The Home tab is selected. The formula bar shows "D60". The main area contains a table with the following data:

	A	B	C	D	E	F
1	Prefix	Dictionary	Subsection	SubsectionDesc	Section	SectionDesc
2	A100	SHOVEL INSTALLATION ASSEMBLIES	A01	SHOVELS	A	ATTACHMENTS
3	A100	SHOVEL INSTALLATION ASSEMBLIES	A01	SHOVEL	A	ATTACHMENTS
4	A100	SHOVEL INSTALLATION ASSEMBLIES	A01	SHOVEL	A	SHOVEL
5	A100	SHOVEL INSTALLATION ASSEMBLIES	A01	SHOVELS	A	SHOVEL
6	A101	SHOVEL SALES SPECIALS	A01	SHOVELS	A	ATTACHMENTS
7	A110	GENERAL PURPOSE SHOVEL	A01	SHOVEL	A	ATTACHMENTS
8	A110	GENERAL PURPOSE SHOVEL	A01	SHOVELS	A	ATTACHMENTS
9	A110	GENERAL PURPOSE SHOVEL	A01	SHOVELS	A	SHOVEL
10	A110	GENERAL PURPOSE SHOVEL	A01	SHOVEL	A	SHOVEL
11	A120	CLAM SHOVEL ASSEMBLY	A01	BUCKETS	A	ATTACHMENTS
12	A120	CLAM SHOVEL ASSEMBLY	A01	SHOVEL	A	ATTACHMENTS
13	A120	CLAM SHOVEL ASSEMBLY	A01	SHOVEL	A	SHOVEL
14	A120	CLAM SHOVEL ASSEMBLY	A01	SHOVELS	A	ATTACHMENTS
15	A120	CLAM SHOVEL ASSEMBLY	A01	SHOVELS	A	SHOVEL
16	A125	CLAM SHOVEL DETAILS	A01	BUCKETS	A	ATTACHMENTS
17	A125	CLAM SHOVEL DETAILS	A01	SHOVEL	A	ATTACHMENTS
18	A125	CLAM SHOVEL DETAILS	A01	SHOVELS	A	ATTACHMENTS
19	A125	CLAM SHOVEL DETAILS	A01	SHOVELS	A	SHOVEL
20	A190	OTHER SHOVELS	A01	SHOVEL	A	ATTACHMENTS
21	A190	OTHER SHOVELS	A01	SHOVELS	A	ATTACHMENTS
22	A190	OTHER SHOVELS	A01	SHOVEL	A	SHOVEL
23	A200	BUCKET INSTALLATION ASSEMBLIES	A02	Bucket	A	ATTACHMENTS
24	A200	BUCKET INSTALLATION ASSEMBLIES	A02	BUCKETS	A	ATTACHMENTS
25	A201	BUCKET SALES SPECIALS	A02	BUCKETS	A	ATTACHMENTS
26	A210	STANDARD BUCKET	A02	BUCKETS	A	SHOVEL
27	A210	STANDARD BUCKET	A02	Bucket	A	ATTACHMENTS
28	A210	STANDARD BUCKET	A02	BUCKETS	A	ATTACHMENTS
29	A220	GRADING BUCKET	A02	BUCKETS	A	ATTACHMENTS

SKOS model of SVR



The screenshot shows a window titled "SVR.skos.owl" containing an RDF/XML document. The document defines a SKOS-based vocabulary for SVR (Software Versioning and Revision) concepts. It includes a main concept "A" with various labels and narrower terms, and a detailed description of the "A01" term.

```
<owl:NamedIndividual rdf:about="http://skosexample.com/SVR#A">
  <rdf:type rdf:resource="http://www.w3.org/2004/02/skos/core#Concept"/>
  <www:svrCode rdf:datatype="http://www.w3.org/1999/02/22-rdf-syntax-ns#XMLLiteral">A</www:svrCode>
  <skos:prefLabel rdf:datatype="http://www.w3.org/1999/02/22-rdf-syntax-ns#XMLLiteral">A Attachments</skos:prefLabel>
  <www:svrLabel rdf:datatype="http://www.w3.org/1999/02/22-rdf-syntax-ns#XMLLiteral">Attachments</www:svrLabel>
  <www:svrLevel rdf:datatype="http://www.w3.org/1999/02/22-rdf-syntax-ns#XMLLiteral">SECTION</www:svrLevel>
  <skos:narrower rdf:resource="http://skosexample.com/SVR#A01"/>
  <skos:narrower rdf:resource="http://skosexample.com/SVR#A02"/>
  <skos:narrower rdf:resource="http://skosexample.com/SVR#A03"/>
  <skos:narrower rdf:resource="http://skosexample.com/SVR#A04"/>
  <skos:narrower rdf:resource="http://skosexample.com/SVR#A05"/>
  <skos:narrower rdf:resource="http://skosexample.com/SVR#A06"/>
  <skos:narrower rdf:resource="http://skosexample.com/SVR#A07"/>
  <skos:narrower rdf:resource="http://skosexample.com/SVR#A09"/>
</owl:NamedIndividual>

<!-- http://skosexample.com/SVR#A01 -->

<owl:NamedIndividual rdf:about="http://skosexample.com/SVR#A01">
  <rdf:type rdf:resource="http://www.w3.org/2004/02/skos/core#Concept"/>
  <www:svrCode rdf:datatype="http://www.w3.org/1999/02/22-rdf-syntax-ns#XMLLiteral">A01</www:svrCode>
  <skos:prefLabel rdf:datatype="http://www.w3.org/1999/02/22-rdf-syntax-ns#XMLLiteral">A01 SHOVEL</skos:prefLabel>
  <www:svrLabel rdf:datatype="http://www.w3.org/1999/02/22-rdf-syntax-ns#XMLLiteral">SHOVEL</www:svrLabel>
  <www:svrLevel rdf:datatype="http://www.w3.org/1999/02/22-rdf-syntax-ns#XMLLiteral">SUBSECTION</www:svrLevel>
  <skos:broader rdf:resource="http://skosexample.com/SVR#A"/>
  <skos:narrower rdf:resource="http://skosexample.com/SVR#A100"/>
  <skos:narrower rdf:resource="http://skosexample.com/SVR#A101"/>
  <skos:narrower rdf:resource="http://skosexample.com/SVR#A110"/>
  <skos:narrower rdf:resource="http://skosexample.com/SVR#A120"/>
  <skos:narrower rdf:resource="http://skosexample.com/SVR#A125"/>
  <skos:narrower rdf:resource="http://skosexample.com/SVR#A190"/>
</owl:NamedIndividual>
```

U:--- SVR.skos.owl 1% L113 (nXML Valid)

SVR.skos.owl in Protege

SVR (<http://skosexample.com/SVR>) : [/Users/maja/SNAP-ON/snaponskos/output/SVR.skos.owl]

Annotation Properties SKOS view Individuals OWLViz DL Query OntoGraf Ontology Differences SPARQL Query

Asserted Concept Hierarchy Concept Scheme List View

Asserted Concept Hierarchy: 'A01 SHOVEL'

A Attachments A01 SHOVEL A02 BUCKETS A03 FORKS A04 ROCKBREAKER/POWERBREAKER A05 HAND HELD TOOLS A06 AGRICULTURAL A07 UTILITY ATTACHMENTS A09 ATTACHMENTS

'A100 SHOVEL INSTALLATION ASSEMBLIES'
'A101 SHOVEL SALES SPECIALS'
'A110 GENERAL PURPOSE SHOVEL'
'A120 CLAM SHOVEL ASSEMBLY'
'A125 CLAM SHOVEL DETAILS'
'A190 OTHER SHOVELS'

'A200 BUCKET INSTALLATION ASSEMBLIES'
'A201 BUCKET SALES SPECIALS'
'A210 STANDARD BUCKET'
'A220 GRADING BUCKET'
'A230 HEAVY DUTY BUCKET'
'A240 TAPER DITCHING BUCKET'
'A250 JAW BUCKET'
'A260 RIPPER TOOTH'
'A290 OTHER BUCKETS'

'A01 SHOVEL'

SKOS Object Property Assertions: 'A01 SHOVEL'

SKOS in scheme assertion +
SKOS top concept of assertion +
SKOS broad match assertion +
SKOS narrow match assertion +
SKOS related match assertion +
SKOS exact match assertion +
Other property assertions +

narrower 'A120 CLAM SHOVEL ASSEMBLY'
narrower 'A100 SHOVEL INSTALLATION ASSEMBLIES'
narrower 'A190 OTHER SHOVELS'
broader 'A Attachments'
narrower 'A125 CLAM SHOVEL DETAILS'
narrower 'A110 GENERAL PURPOSE SHOVEL'
narrower 'A101 SHOVEL SALES SPECIALS'

No Reasoner set. Select a reasoner from the Reasoner menu Show Inferences

SVR to PIL Mapping

SVR to PIL mapping is provided in the excel sheet “SVR vs PIL.xls”, where SVR subjects are mapped to PIL codes.

The exact semantics of these mappings is not specified, i.e. it is not known whether a mapping is to an equivalent, broader or a narrower concept.

We created a new RDF/OWL file `SVR2PIL.owl` that simply uses annotations with **new annotation properties `pilMap` and `svrMap`** in order to establish links in both directions.

SVR vs PIL excel

SVR vs PIL.xls

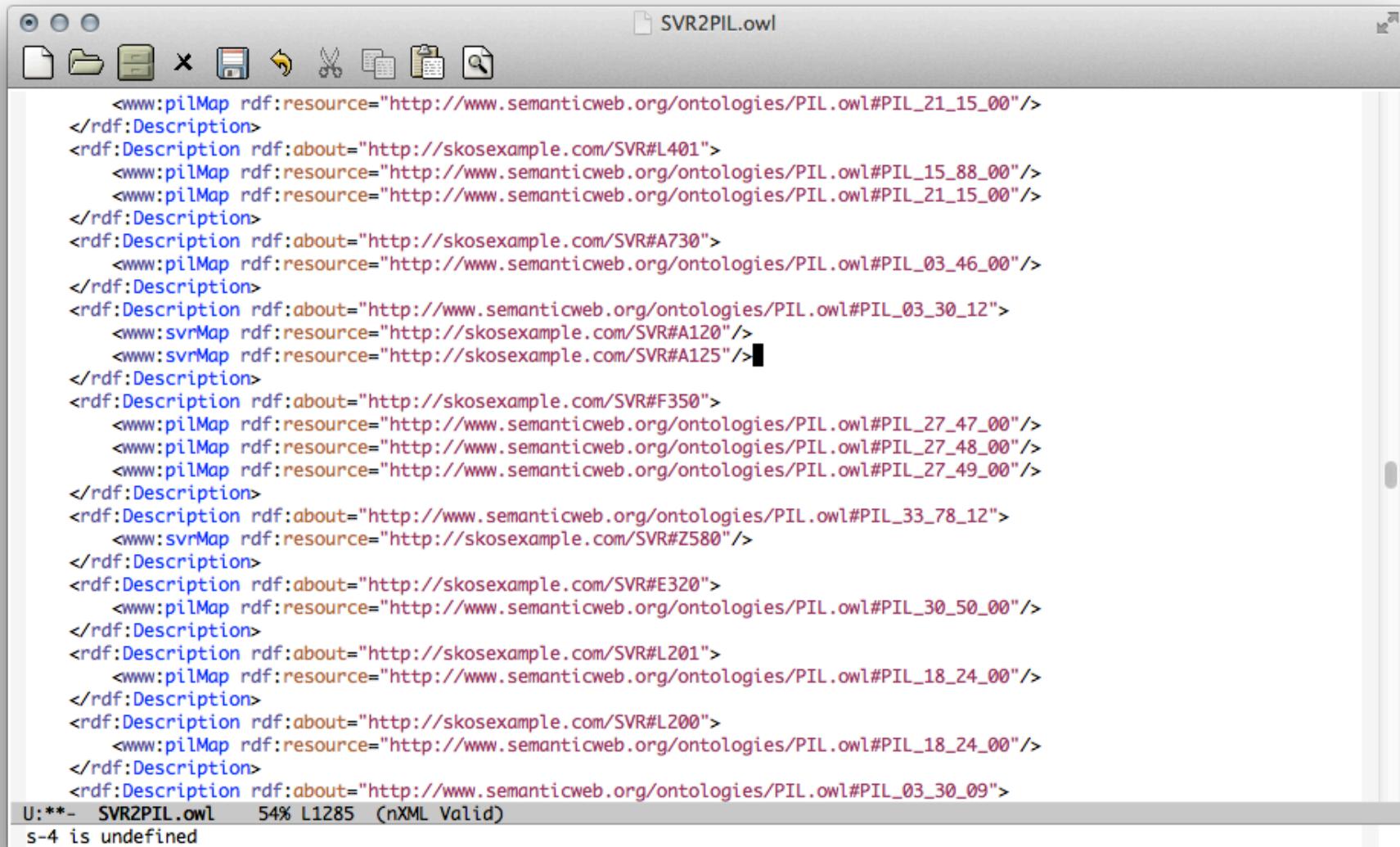
The screenshot shows a Microsoft Excel spreadsheet titled "SVR vs PIL.xls". The search bar at the top right contains the text "Engine Fan". The ribbon menu is visible with tabs for Home, Layout, Tables, Charts, SmartArt, Formulas, Data, and Review. The Home tab is selected.

The spreadsheet contains data in columns A through I. Column A lists part numbers (e.g., A100, A101, A110, etc.). Column B lists descriptions (e.g., SHOVEL INSTALLATION ASSEMBLIES, SHOVEL SALES SPECIALS, GENERAL PURPOSE SHOVEL, etc.). Column C indicates the source of the code: "FROM COMBINATION MASTER" or "PIL". Column D contains the SVR codes, and Column E contains the PIL codes. Some entries in Column E are underlined, suggesting they are future values or pending updates.

	A	B	C	D	E	F	G	H	I
1									
2	A100	SHOVEL INSTALLATION ASSEMBLIES	PIL	03_30					
3	A101	SHOVEL SALES SPECIALS		03_30					
4									
5	A110	GENERAL PURPOSE SHOVEL		03_30_00	03_30_03				
6	A120	CLAM SHOVEL ASSEMBLY		03_30_09	03_30_12				
7	A125	CLAM SHOVEL DETAILS		03_30_09	03_30_12				
8	A190	OTHER SHOVELS		03_30_00					
9									
10	A200	BUCKET INSTALLATION ASSEMBLIES	FROM COMBINATION MASTER	03_27_00					
11	A201	BUCKET SALES SPECIALS		03_27_00					
12									
13	A210	STANDARD BUCKET		03_27_03					
14	A220	DITCHING/GRAVING BUCKET		FUTURE					
15	A230	HEAVY DUTY BUCKET		X					
16	A240	TAPERED BUCKET		X					
17	A250	JAW BUCKET		03_27_06					
18	A260	RIPPER TOOTH		03_27_90					
19	A280	OTHER BUCKETS		03_27_00					
20									
21	A300	HITCH INSTALLATION ASSEMBLIES	FROM COMBINATION MASTER						
22	A301	HITCH SALES SPECIALS							
23									
24	A310	FRONT HITCH		03_12_00?					
25	A320	REAR HITCH		03_12_00?					
26	A330	EXCAVATOR QUICK HITCH		03_10_00					
27	A340	LOADER QUICK HITCH		03_09_00					
28	A350	TOOL CARRIER		03_06_00					
29	A360	CARRIAGE		03_36_00?					
30	A370	FORKS		03_33_00					
31	A380	HITCH LOCKING		03_12_00?					
32	A390	OTHER HITCHES		03_12_00?					
33									
34	A400	ROCK BREAKERS	FROM COMBINATION MASTER	03_10_00					

Normal View Ready Sum=0

SVR vs PIL Mappings in RDF/OWL



The screenshot shows a code editor window titled "SVR2PIL.owl". The window contains an RDF/OWL XML document with numerous `<www:pilMap` and `<www:svrMap` statements. The code is color-coded, with blue for namespaces and red for URIs. The XML structure consists of multiple `<rdf:Description>` blocks, each containing one or more `<www:pilMap rdf:resource="..."/>` or `<www:svrMap rdf:resource="..."/>` elements. The document is scrollable, with a vertical scrollbar visible on the right side of the editor.

```
<www:pilMap rdf:resource="http://www.semanticweb.org/ontologies/PIL.owl#PIL_21_15_00"/>
</rdf:Description>
<rdf:Description rdf:about="http://skosexample.com/SVR#L401">
    <www:pilMap rdf:resource="http://www.semanticweb.org/ontologies/PIL.owl#PIL_15_88_00"/>
    <www:pilMap rdf:resource="http://www.semanticweb.org/ontologies/PIL.owl#PIL_21_15_00"/>
</rdf:Description>
<rdf:Description rdf:about="http://skosexample.com/SVR#A730">
    <www:pilMap rdf:resource="http://www.semanticweb.org/ontologies/PIL.owl#PIL_03_46_00"/>
</rdf:Description>
<rdf:Description rdf:about="http://www.semanticweb.org/ontologies/PIL.owl#PIL_03_30_12">
    <www:svrMap rdf:resource="http://skosexample.com/SVR#A120"/>
    <www:svrMap rdf:resource="http://skosexample.com/SVR#A125"/>
</rdf:Description>
<rdf:Description rdf:about="http://skosexample.com/SVR#F350">
    <www:pilMap rdf:resource="http://www.semanticweb.org/ontologies/PIL.owl#PIL_27_47_00"/>
    <www:pilMap rdf:resource="http://www.semanticweb.org/ontologies/PIL.owl#PIL_27_48_00"/>
    <www:pilMap rdf:resource="http://www.semanticweb.org/ontologies/PIL.owl#PIL_27_49_00"/>
</rdf:Description>
<rdf:Description rdf:about="http://www.semanticweb.org/ontologies/PIL.owl#PIL_33_78_12">
    <www:svrMap rdf:resource="http://skosexample.com/SVR#Z580"/>
</rdf:Description>
<rdf:Description rdf:about="http://skosexample.com/SVR#E320">
    <www:pilMap rdf:resource="http://www.semanticweb.org/ontologies/PIL.owl#PIL_30_50_00"/>
</rdf:Description>
<rdf:Description rdf:about="http://skosexample.com/SVR#L201">
    <www:pilMap rdf:resource="http://www.semanticweb.org/ontologies/PIL.owl#PIL_18_24_00"/>
</rdf:Description>
<rdf:Description rdf:about="http://skosexample.com/SVR#L200">
    <www:pilMap rdf:resource="http://www.semanticweb.org/ontologies/PIL.owl#PIL_18_24_00"/>
</rdf:Description>
<rdf:Description rdf:about="http://www.semanticweb.org/ontologies/PIL.owl#PIL_03_30_09">
```

U:**- SVR2PIL.owl 54% L1285 (nXML Valid)
s-4 is undefined

Protégé Map View – from SVR

SVR (<http://skosexample.com/SVR>) : [/Users/maja/SNAP-ON/snaponskos/output/SVR.skos.owl]

SVR (<http://skosexample.com/SVR>) Search for entity

Active Ontology Entities Classes Object Properties Data Properties Annotation Properties SKOS view Individuals OWLViz ►

Individuals: 'A120 CLAM SHOVEL ASSEMBLY'

- ◆ 'A101 SHOVEL SALES SPECIALS'
- ◆ 'A110 GENERAL PURPOSE SHOVEL'
- ◆ **'A120 CLAM SHOVEL ASSEMBLY'**
- ◆ 'A125 CLAM SHOVEL DETAILS'
- ◆ 'A190 OTHER SHOVELS'
- ◆ 'A200 BUCKET INSTALLATION ASSEMBLIES'
- ◆ 'A201 BUCKET SALES SPECIALS'
- ◆ 'A210 STANDARD BUCKET'
- ◆ 'A220 GRADING BUCKET'
- ◆ 'A230 HEAVY DUTY BUCKET'
- ◆ 'A240 TAPER DITCHING BUCKET'
- ◆ 'A250 JAW BUCKET'
- ◆ 'A260 RIPPER TOOTH'
- ◆ 'A290 OTHER BUCKETS'
- ◆ 'A300 HITCH INSTALLATION ASSEMBLIES'
- ◆ 'A310 FRONT HITCH'
- ◆ 'A320 REAR HITCH'
- ◆ 'A330 EXCAVATOR QUICK HITCH'
- ◆ 'A340 LOADER QUICK HITCH'
- ◆ 'A350 TOOL CARRIER'
- ◆ 'A360 CARRIAGE'
- ◆ 'A370 FORKS'
- ◆ 'A380 HITCH LOCKING'

Annotations: 'A120 CLAM SHOVEL ASSEMBLY'

Annotations	Usage
pilMap	@ X O
● '4-in-1 Shovel [Attachment]'	
pilMap	@ X O
● '6-in-1 Shovel [Attachment]'	
prefLabel [type: XMLLiteral]	@ X O
A120 CLAM SHOVEL ASSEMBLY	
svrCode [type: XMLLiteral]	@ X O
A120	
svrLabel [type: XMLLiteral]	@ X O
CLAM SHOVEL ASSEMBLY	
svrLevel [type: XMLLiteral]	@ X O
SUBJECT	

Description: 'A120 CLAM'

Types +

- Concept ? @ X O

Property assertions: 'A120 CLAM SHOVEL ASSEMBLY'

Object property assertions +

- broader 'A01 SHOVEL' ? @ X O

No Reasoner set. Select a reasoner from the Reasoner menu Show Inferences

Protégé Map View – from PIL

PIL (<http://www.semanticweb.org/ontologies/PIL.owl>) : [/Users/maja/ SNAP-ON/snaponSKOS/output/PIL.owl]

Active Ontology Entities Classes Object Properties Data Properties Annotation Properties Individuals SKOS view OWLViz ▶

🔍 Shovel

Class hierarchy Class hierarchy (inferred)

Class hierarchy (inferred): '4-in-1 Shovel [Attachment]'

- 'Pole Planter [Attachment]'
- ▶ ● 'Pump [Attachment]'
- 'Rake [Attachment]'
- 'Rockbreaker [Attachment]'
- 'Rotary Drill [Attachment]'
- 'Rotary Tiller [Attachment]'
- ▼ ● 'Shovel [Attachment]'
 - '4-in-1 Shovel [Attachment]'
 - '6-in-1 Shovel [Attachment]'
 - 'Beet Basket Shovel [Attachment]'
 - 'Grading Shovel [Attachment]'
 - 'Grapple Shovel [Attachment]'
 - 'Side-Tip Shovel [Attachment]'
 - 'Standard Shovel [Attachment]'
- ▶ ● 'Spike [Attachment]'
- ▶ ● 'Spreader [Attachment]'
- 'Sweeper Collector [Attachment]'
- 'Trailer [Attachment]'
- 'Trencher [Attachment]'
- 'Winch [Attachment]'
- 'Work Platform [Attachment]'
- Auto
- 'Automatic Gearbox Clutch Forward [Driveline]'
- 'Auxiliary Hydraulic'

Annotations: '4-in-1 Shovel [Attachment]'

Annotation	Type	Value	Actions
origin	[type: string]	PIL_ORIGINAL	@ X O
pilCode	[type: string]	03_30_09	@ X O
pilLabel	[type: string]	4-in-1	@ X O
pilLevel	[type: string]	PART	@ X O
pilParent	[type: string]	'Shovel [Attachment]'	@ X O
svrMap	[type: string]	'A120 CLAM SHOVEL ASSEMBLY'	@ X O
svrMap	[type: string]	'A125 CLAM SHOVEL DETAILS'	@ X O
terminology	[type: string]	'shovel'	@ X O

Reasoner active Show Inferences



AVAnT

Quick Access

PIL SVR

- ▶ 03_00_00 Attachments
- ▶ 06_00_00 Body and Framework
- ▶ 09_00_00 Operator Station
- ▶ 12_00_00 HVAC
- ▶ 15_00_00 Engine
- ▶ 18_00_00 Fuel
- ▶ 21_00_00 Cooling
- ▶ 24_00_00 Brakes
- ▶ 25_00_00 Steering
- ▶ 27_00_00 Driveline
- ▶ 28_00_00 Suspension
- ▶ 30_00_00 Hydraulics
- ▶ 33_00_00 Electrical
- ▶ 72_00_00 Fasteners and Fixings
- ▶ 75_00_00 Consumables

Indexing and Search

We used the [JCB Terminology](#) to create synonyms for PIL and SVR concepts at indexing. PIL acronyms are extracted and added to the Terminology as new synonyms.

Example:

Smooth Ride System Hose Burst Check Valve

has the following synonyms

Smooth Ride System HBCV

Smooth Ride System Hose Burst Protection Valve

Smooth Ride System Lock Out Valve

SRS Hose Burst Check Valve

SRS HBCV

SRS Hose Burst Protection Valve

SRS Lock Out Valve



AVAnT

Quick Access

PIL SVR

- ▶ 03_00_00 Attachments
- ▶ 06_00_00 Body and Framework
- ▶ 09_00_00 Operator Station
- ▶ 12_00_00 HVAC
- ▶ 15_00_00 Engine
- ▶ 18_00_00 Fuel
- ▶ 21_00_00 Cooling
- ▶ 24_00_00 Brakes
- ▶ 25_00_00 Steering
- ▶ 27_00_00 Driveline
- ▶ 28_00_00 Suspension
- ▶ 30_00_00 Hydraulics
- ▶ 33_00_00 Electrical
- ▶ 72_00_00 Fasteners and Fixings
- ▶ 75_00_00 Consumables

Search View

Search

Further Ideas

- Add missing Terminology and SVR concepts to the enriched PIL.
E.g. Clam Shovel (includes 4-in-1 Shovel, 6-in-1 Shovel).
- Once the enriched PIL structure and concepts are set, define preferred terms and synonyms (possibly multilingual) for PIL concepts. Similar can be done for SVR.
- Besides the PIL and SVR trees, add the enriched PIL concept hierarchy as the third means of navigation.
- Use concept hierarchy for indexing: e.g. index PIL concepts “4-in-1 Shovel” and “6-in-1 Shovel” with the term “Clam Shovel” so that they appear high on the search result list for “Clam Shovel”.