'Configuration as Linked Data' ontology

IRI:
http://purl.org/configurationontology

Authors:
http://data.semanticweb.org/person/edouard-chevalier
http://data.semanticweb.org/person/francois-paul-servant
http://dbpedia.org/resource/Renault

Ontology source:
Turtle
RDF-XML

Abstract

Ontology for the description of customizable products. It models the configuration process as the traversal of a graph of partially defined products, or "Configurations".

Ranges of customizable products are described rather effectively, for human users, by means of dedicated web applications called configurators. A configurator helps a user interactively define a product step by step, each step typically describing a valid partially defined product (or "Configuration"), with a start price and a list of remaining choices given all previous selections. Each of these choices links to another configuration until completion. Thus, the configuration process traverses a graph whose nodes are configurations. Now identify each configuration with a URI returning the list of the configurations it is linked to, among other relevant information: what you have is a description of the range as Linked Data.

The main classes of this ontology are Configuration (a state of the configuration process), Specification (a possible value of a characteristic of the configured product) and ConfigurationLink (which models a change of the state of the configuration process).

Table of Content

Classes

Completely defined configuration[c]

back to ToC or Class ToC

IRI: http://purl.org/configurationontology#CompletelyDefinedConfiguration

has super-classes

Configuration[c]

Configuration[c]

back to ToC or Class ToC

IRI: http://purl.org/configurationontology#Configuration

A state of the configuration process, assumed to correspond to a valid partially defined product - valid in the sense that it can be completed into an existing fully specified product, one that can be ordered, without changing any of the current selections.

A configuration is defined by a Lexicon (the set of the descriptive attributes of the configured product, cf. the "lexicon" property), and by the set of choices that have been made so far (typically selected specifications), cf. the "definingChoice" property.

Note: A configurator application may conceivably have to handle “invalid configurations”, that is, combinations of specifications that are impossible. This can happen, for instance, if the user is allowed to begin the configuration process by choosing features without any control of their compatibility; or if she is allowed to choose a feature incompatible with her previous selections. In this case, it is the responsibility of the configurator application to restore the consistency of the configuration, necessarily by excluding some of the previous user selections. This Configuration class excludes such invalid combinations.

Relation with GoodRelations and schema.org vocabularies :

because a configuration mainly describes a Partially Defined Product, it can be seen as a GoodRelations "ProductOrServiceModel" (or a schema.org "ProductModel): "an intangible entity that specifies some characteristics of a group of similar, usually mass-produced products, in the sense of a prototype". The suffix "Model" may seem misleading when used for a Configuration, as it suggests something such as "Ford T", and not "Ford T with sunroof and climat control (itself not a completely defined product - you still can choose, well, the color: it is a "prototype of similar products").

On the other hand, a Configuration has a price (more precisely, a "from price" - the minimum of the "suggested retail prices" of all the matching completely defined products). As such, a Configuration may be seen as a commercial offer. It can also be used to represent a customer's wish list, constrained by the definition of the range. It could therefore be considered as a gr:Offering (or a schema:Offer) as well.

But gr:ProductOrService and gr:Offering are disjoint classes: a given Configuration cannot be both. Depending on the use case, one publisher of configuration data may want to type her Configurations with either of these classes.

has sub-classes

Completely defined configuration^c

is in domain of

alternative^op, choice sequence^op, chosen specification^op, completed^op, completed at the same price^op, configuration link^op, default^op, defining choice^op, implied specification^op, impossible^op, max price^op, possible^op, proposal^op

is in range of

linked configuration^op, proposal^op

Configuration Link^c

back to ToC or Class ToC

IRI: http://purl.org/configurationontology#ConfigurationLink

Models the link between a given Configuration and another one (a change of state in the configuration process): typically, the association of a Specification to be selected, and of a resulting Configuration.

has super-classes

linked configuration^op **exactly** 1 Configuration^c

is in domain of

added specification^op, linked configuration^op, removed specification^op

is in range of

alternative^op, configuration link^op, impossible^op, possible^op

Configuration variable^c

back to ToC or Class ToC

IRI: http://purl.org/configurationontology#ConfigurationVariable

A variable in a Lexicon, eg. the fuel type, or the body color. Automotive ranges typically only use discrete variables: the possible values corresponding to one ConfigurationVariable are alternative specifications (for instance "Diesel", "Gasoline", etc. for the fuel type variable)

has super-classes

lexicon^op **exactly** 1 Lexicon^c

is in domain of

:confVarId^dp, has value^op

Lexicon^c

back to ToC or Class ToC

IRI: http://purl.org/configurationontology#Lexicon

The set of specifications (and of their corresponding configuration variables) used in the configuration process: all the possible descriptive attributes of the configured product.

has super-classes
owl:Thing
is in range of
lexicon^{op}

Model specification^{c}

back to ToC or Class ToC

IRI: http://purl.org/configurationontology#Model

A specification that identifies a model (like "Ford T"). Corresponding configurations are supposed to share the same lexicon.

has super-classes
Specification^{c}
lexicon^{op} **exactly** 1 Lexicon^{c}

Specification^{c}

back to ToC or Class ToC

IRI: http://purl.org/configurationontology#Specification

Feature, option, any value of a characteristic of a product that can be chosen during the configuration process. For instance: "Automatic gearbox", "Blue body color", "CO2 emission value < 100 g of CO2"

has super-classes
gr:QualitativeValue **or** gr:QuantitativeValue
has sub-classes
Model specification^{c}
is in domain of
:specId^{dp}
is in range of
added specification^{op}, chosen specification^{op}, default^{op}, has value^{op}, implied specification^{op}, removed specification^{op}

Object Properties

removed specification

added specification op

back to ToC or Object Property ToC

IRI: http://purl.org/configurationontology#specToBeAdded

Specification to be added to a Configuration in a ConfigurationLink

has domain

Configuration Link c

has range

Specification c

alternative op

back to ToC or Object Property ToC

IRI: http://purl.org/configurationontology#alternative

Models the possibility to change a previous choice (links the configuration to a similar one, with one of the definingChoices removed or changed. This property may not be used when the chosen specification in question happens to be implied by the other choices

has super-properties

configuration link op

has domain

Configuration c

has range

Configuration Link c

choice sequence op

back to ToC or Object Property ToC

IRI: http://purl.org/configurationontology#choiceSeq

Choices are made one at a time and in a given order, which may matter. Of course it doesn’t impact the characteristics of the product in any way, but it can be used by some applications.

has super-properties

defining choice op

has domain

Configuration c

has range

rdf:Seq

chosen specification op

back to ToC or Object Property ToC

IRI: http://purl.org/configurationontology#chosenSpec

A SubProperty of co:definingChoice listing the specifications selected by the user:

If two or more of the cold:chosenSpec of a Configuration correspond to the same cold:ConfigurationVariable, by convention they are to be interpreted as ORed (even XORed, by the way, as such specifications are alternative).

So:

ex:AConf coold:chosenSpec ex:SimpleRadio, ex:RadioMP3.

means that the car has either a ex:SimpleRadio, or a ex:RadioMP3, not both.

has super-properties

defining choice op

has domain

Configuration c

has range

Specification c

completed op

back to ToC or Object Property ToC

IRI: http://purl.org/configurationontology#completed

Links a Configuration to a completely defined one (that is, a Completely Defined Configuration that matches the configuration in question). The price of the linked configuration may be higher than this configuration price. Configurations completed at the same price are given by the completedAtSamePrice property.

has sub-properties

completed at the same price op

has domain

Configuration c

completed at the same price op

back to ToC or Object Property ToC

IRI: http://purl.org/configurationontology#completedAtSamePrice

Links a Configuration to a completely defined one, which has the same price.

has super-properties

completed op

has domain

Configuration c

configuration link op

back to ToC or Object Property ToC

IRI: http://purl.org/configurationontology#link

Parent property of all the properties linking a Configuration to another one in the configuration process.

The HTML display of a cold:ConfigurationLink corresponds to an hypertext link, whose href is the value of the cold:linkedConf property. As for the text of this link, the rdfs:label of the cold:specToBeAdded value is quite adequate. It can be directly included in the RDF as the rdfs:label of the cold: ConfigurationLink

has sub-properties
alternative^op, impossible^op, possible^op

has domain
Configuration^c

has range
Configuration Link^c

default^op

back to ToC or Object Property ToC

IRI: http://purl.org/configurationontology#defaultSpec

Specification included by default in a Completely Defined Product matching this configuration

has domain
Configuration^c

has range
Specification^c

defining choice^op

back to ToC or Object Property ToC

IRI: http://purl.org/configurationontology#definingChoice

Parent to all properties specifying the choices that define the Configuration: a Configuration is defined by the list of triples it is the subject of, and which have a co:definingChoice as their predicates.

has super-properties
owl:topObjectProperty

has sub-properties
choice sequence^op, chosen specification^op, max price^op

has domain
Configuration^c

has value^op

back to ToC or Object Property ToC

IRI: http://purl.org/configurationontology#hasValue

Used in a Lexicon to list the values (specifications) of a ConfigurationVariable. Alternatively, we could say that these specifications have rdf:type the ConfigurationVariable in question.

has domain

Configuration variable^c

has range

Specification^c

implied specification^op

back to ToC or Object Property ToC

IRI: http://purl.org/configurationontology#impliedSpec

A Specification that is implied by the conjunction of the definingChoice(s) of this Configuration.

has domain

Configuration^c

has range

Specification^c

impossible^op

back to ToC or Object Property ToC

IRI: http://purl.org/configurationontology#impossible

Models a choice that conflicts with the definition of this configuration: when a specification is not compatible with a configuration, the configuration engine can nevertheless provide a way to select it - of course, at the cost of discarding some of the previous selections; there is a conflict, to be resolved by removing or changing some of the cold:definingChoice(s).

has super-properties

configuration link^op

has domain

Configuration^c

has range

Configuration Link^c

lexicon^op

back to ToC or Object Property ToC

IRI: http://purl.org/configurationontology#lexicon

Used to link a Configuration, a ConfigurationVariable or a Specification to a Lexicon.

has domain

Configuration^c **or** Configuration variable^c **or** Specification^c

has range

Lexicon c

linked configuration op

back to ToC or Object Property ToC

IRI: http://purl.org/configurationontology#linkedConf

the linked configuration that contains choices of current Configuration plus the specifications that will be added and listed in the configuration link it belongs to.

has domain

Configuration Link c

has range

Configuration c

max price op

back to ToC or Object Property ToC

IRI: http://purl.org/configurationontology#maxPrice

An upper limit set on the price of the configuration.

has super-properties

defining choice op

has domain

Configuration c

has range

gr:UnitPriceSpecification

possible op

back to ToC or Object Property ToC

IRI: http://purl.org/configurationontology#possible

Models a choice (or a set of choices) that can be made without changing any of the previous choices done in the configuration: links to a Configuration whose list of definingChoice(s) contains the list of definingChoice(s) of this configuration.

has super-properties

configuration link op

has domain

Configuration c

has range

Configuration Link c

proposal op

back to ToC or Object Property ToC

IRI: http://purl.org/configurationontology#proposedConf

Links a Configuration to another one supposed to be of interest for a customer at this point in the configuration process. The semantics is rather vague, leaving open the possibility to link to a Configuration that matches the defining choices of the subject configuration, or not (to be used by marketing people for bargains, 'upselling', etc.)

has domain

Configuration c

has range

Configuration c

removed specification op

back to ToC or Object Property ToC

IRI: http://purl.org/configurationontology#specToBeRemoved

Specification to be removed from a Configuration in a ConfigurationLink

has domain

Configuration Link c

has range

Specification c

Data Properties

:confVarId :specId

:confVarId dp

back to ToC or Data Property ToC

IRI: http://purl.org/configurationontology#confVarId

Code identifying a ConfigurationVariable within the Lexicon it belongs to.

has characteristics: functional

has domain

Configuration variable c

has range

rdfs:Literal

:specId dp

back to ToC or Data Property ToC

IRI: http://purl.org/configurationontology#specId

Code identifying a Specification within the Lexicon it belongs to.

has characteristics: functional

has domain

Specification[c]

has range

rdfs:Literal

Annotation Properties

vs:term_status

vs:term_status[ap]

back to ToC or Annotation Property ToC

IRI: http://www.w3.org/2003/06/sw-vocab-status/ns#term_status

Namespace Declarations

back to ToC

default namespace	http://purl.org/configurationontology#
cc	http://creativecommons.org/ns#
dc	http://purl.org/dc/terms/
gr	http://purl.org/goodrelations/v1#
owl	http://www.w3.org/2002/07/owl#
person	http://data.semanticweb.org/person/
purl-org	http://purl.org/
rdf	http://www.w3.org/1999/02/22-rdf-syntax-ns#
rdfs	http://www.w3.org/2000/01/rdf-schema#
resource	http://dbpedia.org/resource/
vann	http://purl.org/vocab/vann/
vs	

http://www.w3.org/2003/06/sw-vocab-status/ns#

xsd

http://www.w3.org/2001/XMLSchema#
