Describing Customizable Products on the Web of Data
Linked Data On the Web Workshop - Rio de Janeiro - May 14th, 2013

François-Paul Servant (francois-paul.servant@renault.com)
01 Publishing descriptions of customizable products
A challenging issue

02 Configuration as Linked Data
The Configuration Ontology

03 Designing sharable vocabularies
Requirements
Publishing product descriptions on the web of data

- has recently gained momentum
  - schema.org
  - GoodRelations

- Use rich data for Web marketing

- in order to increase the visibility of commercial offers
  - talking to all kinds of devices and agents
  - search engines, price comparators, recommendation applications
  - SEO ("search engine optimization")
Google search results for "the pleasure of finding things out".
What's about cars?

Google search terms: small new car gas air conditioning sunroof mp3 connection plug
For cars, especially for new cars, results are a bit disappointing...
Books

- Completely Defined Products
- Few search criteria
- Comparisons of the offers on a small number of criteria
- ISBN
- <130,000,000 different books*


Cars

- Partially Defined Products
- Many criteria
- Comparisons on many criteria
- No id
- and... a huge diversity
Product ranges in the automotive industry are huge

$10^{20}$

different cars for sale at Renault
Product ranges in the automotive industry are huge

Body styles \times Engines \times Gearboxes

\times Colors
\times Air Conditioning systems
\times Radios
\times Navigation systems
\times Electronic Stability System?
\times ABS?
\times Sun-roof?
\times ...

= 10^{25}
Huge, and complex

Body styles \( \times \) Engines \( \times \) Gearboxes
\( \rightarrow \) x Colors
\( \rightarrow \) x Air Conditioning systems
\( \rightarrow \) x Radios
\( \rightarrow \) x Navigation systems
\( \rightarrow \) x Electronic Stability System?
\( \rightarrow \) x ABS?
\( \rightarrow \) x Sun-roof?
\( \rightarrow \) ... 

\[ 10^{25} \]

Every combination of features is not possible:
- technical, industrial and legal constraints, marketing policy
- "Unglazed rear doors exclude rear wipers"
- "Rear electric windows imply front electric windows"
- ...
Huge, and complex

Body styles \times Engines \times Gearboxes
\times Colors
\times Air Conditioning systems
\times Radios
\times Navigation systems
\times Electronic Stability System?
\times ABS?
\times Sun-roof?
\times ...

= \ 10^{25}

1 chance upon 100,000 to get an existing car, if you choose its features without taking the constraints into account.

= \ 10^{20}
Description of an automotive range

- The range cannot be enumerated: defined "in intention"
- A set of Variables and Constraints between their values:
  - a "Constraint Satisfaction Problem" (CSP)
    - Computationally hard!
- Reasoning software required
How to publish such descriptions?

- The CSP can be represented using Semantic Web languages

- But publishing such data on the web would be too demanding of the clients
  - Reasoning better hosted on the server

- So?
Configurators: an effective way of presenting a range to human users

<table>
<thead>
<tr>
<th>LA GAMME RENAULT</th>
<th>Véhicules Particuliers</th>
</tr>
</thead>
<tbody>
<tr>
<td>TWINGO</td>
<td>NOUVELLE TWINGO</td>
</tr>
<tr>
<td>WIND</td>
<td>RENAULT WIND</td>
</tr>
<tr>
<td>CLIO</td>
<td>CLIO CAMPUS BYE BYE</td>
</tr>
<tr>
<td></td>
<td>CLIO</td>
</tr>
<tr>
<td></td>
<td>CLIO ESTATE</td>
</tr>
<tr>
<td>MODUS</td>
<td>MODUS</td>
</tr>
<tr>
<td></td>
<td>GRAND MODUS</td>
</tr>
<tr>
<td>KANGOO</td>
<td>KANGOO</td>
</tr>
</tbody>
</table>

LDOW 2013
Configurators: interactive definition of a car, one choice after the other
The configuration process

- at each step: list of possible choices, given the choices made so far
A configurator as a REST service

- **ConfigurationService?spec=spec1&spec=spec2&...**  
  
  Previous selections

- returns: **Possible choices**

- = specifications compatible with the previous selections

- to choose specX: append "&spec=specX" to (*)

- (*) identifies a state of the configuration process

- = a "Configuration"

- = a Partially Defined Car

- = a subset of the range

- (*) : URI of configuration
Configuration as Linked Data

- `ConfigurationService?spec=spec1&spec=spec2&... (*)`
  - returns the list of (compatible specification, URI of the refined configuration)
    - eg. `(specX, ConfigurationService?spec=spec1&spec=spec2&...&spec=specX)`

- Configuration Process = Traversal of a graph of linked Configurations
  - = Linked Data!

- Reasoning handled by the server, complexity hidden to the client
  - a GUI just has to display the links
Entry point: the list of models

Possible choices:

the list of models

I want a Laguna Hatchback...
The "Laguna Hatchback" configuration

I want an automatic gearbox...

Previous selections (encoded in the URI, in clear in the data)

Possible choices

<http://uk.co.rplug.renault.com/c/BAv/AAI#this>
a co:Configuration ;
co:chosenSpec <http://uk.co.rplug.renault.com/spec/BAv/LA3#this> ;
co:possible
  [ a co:ConfigurationLink ;
    co:linkedConf <http://uk.co.rplug.renault.com/c/BAv/AAMDg#this> ;
    co:specToBeAdded <http://uk.co.rplug.renault.com/spec/BAv/PT1633_automatic_gearbox#this>
  ] ;
co:possible
  [ a co:ConfigurationLink ;
    ...

<http://uk.co.rplug.renault.com/spec/BAv/PT1633_automatic_gearbox#this>
a co:Specification ;
rdfs:label "Automatic Gearbox"@en .
The "Laguna Hatchback, automatic gearbox" configuration

<a href="http://uk.co.rplug.renault.com/c/BAv/AAMDg#this">
  a co:Configuration ;
  co:chosenSpec
    <http://uk.co.rplug.renault.com/spec/BAv/LA3#this> ,
    <http://uk.co.rplug.renault.com/spec/BAv/PT1633_automatic_gearbox#this>
  co:possible
    [ a co:ConfigurationLink ;
      co:linkedConf <http://uk.co.rplug.renault.com/c/BAv/AAMDg#this> ;
      co:specToBeAdded <http://uk.co.rplug.renault.com/spec/BAv/PT1628_diesel#this>
    ] ;
  co:possible
    [ a co:ConfigurationLink ;
      ...  

gr:hasPriceSpecification [gr:hasCurrencyValue "21795"^^http://www.w3.org/2001/XMLSchema#float] ;
co:impliedSpec <http://uk.co.rplug.renault.com/spec/BAv/PT1627_direct_common_rail_with_turbo#this> ;
co:impossible
  [ a co:ConfigurationLink ;

Previous selections (encoded in the URI, in clear in the data)

Possible choices

More info
Configuration ontology

- http://purl.org/configurationontology

- The configuration process as the traversal of a graph of Configurations
  - 3 main classes: Specification, Configuration and ConfigurationLink
  - properties: chosenSpec, impliedSpec, possible, impossible, defaultSpec, etc
    - the different kinds of relations between a Configuration and a Specification

- A generic, domain independent ontology
  - not limited to the automotive industry
  - doesn't depend on the vocabulary used for the specifications
  - most online configurator applications could use it
Benefits

- Improved Architecture of the configuration server
  - Web architecture

- Decreased development costs of web applications
  - No need to understand the concepts underlying configuration
  - No need to learn an API
  - Just "display the data and follow the links"

- Data published on the web of data for e-business
  - Accurate description of the range, that can be explored by crawlers
    - just a matter of following links.

- Configuration URI: a global identifier for Partially Defined Products
Configuration URI: a global identifier for Partially Defined Products

- A Configuration
  - = a "Partially Defined Product"
  - a commercial offer
    - has a "from price"
    - can be described using GoodRelations
  - a customer’s wish list (constrained by the definition of the range)
    - BTW: an important thing, in a marketing point of view!
Aggregation of data from different publishers

- Range comparators, market places, etc.

- Companies will publish data using their own terms (URIs of specifications)
  - many specifications are unique to a company
    - and needs to be described, and therefore identified
  - the ultimate goal of the configuration process is an order, defined in the manufacturing company's terms
  - no additional cost
  - precision is lost when mapping to other vocabularies
    - my:GorgeousPanoramicSunroof is more than a dbpedia:Sunroof

- Mapping required
  - Use dbpedia URIs? Hmm...

- Reference thesaurus of specifications
  - some shared way to say "diesel" or "CO2 emission level"
Reference thesaurus of Specifications

- There's none
- Some vocabularies attempt to describe cars
  - but do not support Partially Defined Products (PDP)
    - A PDP is not a completely defined one with some properties left undocumented!
- VSO ("Vehicle Sales Ontology")
  - provides some terms, but uses a pattern than doesn't work for PDP:
    - vso:fuelType rdfs:range vso:FuelTypeValue.
    - foo:ACar vso:fuelType dbpedia:Diesel.
    - No room left for the kind of the relation between the configuration and the specification (possible, implied, etc)
      - Should be enough to state once for all that: dbpedia:Diesel a vso:FuelTypeValue.
- Shift from vocabularies describing products to vocabularies defining classes of Specifications
  - hierarchies of terms, etc.
Conclusion

- Ranges of customizable products can be described as Linked Data
- URIs for Partially Defined Products
- A generic ontology
- Renault publishes data about its range
  - $10^{20}$ cars (and more configurations) fully described in RDF
  - http://{uk,br,fr,de,es,it}.co.rplug.renault.com/docs#this
    (set ACCEPT HTTP Header to application/rdf+xml or to text/turtle)
  - quick start guide (and a javascript configurator based on this data):
    http://purl.org/configurationontology/quickstart
- Agents can crawl and use these data
  - a challenge for Search Engines ($10^{20}$ is huge!)
- A reference thesaurus of classes of specifications would be helpful for Range comparators
Range description (source data) → Compilation (offline) → Compiled Range

Configuration Engine

Range as Linked Data (generated on the fly)

Jersey REST service

Configurator web app, Crawler, Agent

HTTP connections
Indexing configurations

- Accurate description of the range, that can be explored by crawlers
  - just a matter of following links.

- But $10^{20}$ is huge!
  - Partial indexing
  - Based on the specifications
  - Beware to the semantics of the properties!
    - spec1 and spec2 can both be compatible with a given configuration, but not (spec1 and spec2) -
      - only way to know: query the configuration service

- Choose the indexing strategy
  - some specifications have more value than others

- Sitemap
  - which configurations should be included to get the most of it from a marketing point of view?