Semantic variant management
... and the automotive industry’s cry for help

September 2019
95% dark stuff

5% normal stuff
Semantic variant management | Situation
Mass individualization makes automotive manufacturers face immense data complexity.

- 30k Parts
- 300k Part numbers in store
- 10k Software parameter
- 1000 Supplier participate in production
- $10^6$ Technical configuration rules
- 40-50 Decisions have to be made by user to configure a car
- 10-30 years spare parts are kept in stores
- $10^{20}$ possible Configurations of final product possible
- $50^{12} \approx 10^{20}$
Semantic variant management | Situation
Configuration knowledge management is one of the most complex areas in PDM.

Example: Car seats

- 50 customer feature decisions
- 10 thousand configuration rules
- 6.5 millions seat variants
- 1 feature affects 90% of the cars components
- 10% complexity cost (CapEx, direct / R&D cost)

Source: cc.porsche.com
Semantic variant management | Complication
Inconsistent rule sets lead to high financial distress due to consideration of not existing configurations across the entire value chain.

Representation of configuration knowledge

<table>
<thead>
<tr>
<th>System m</th>
<th>System m+1</th>
<th>...</th>
<th>System n</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mechanic</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Software</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Electronic</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Complication
- Non compliant rules sets in different systems and functional areas (mechanics, electronics, software)
- Satisfiability check for completeness, consistency and redundancy not/partly possible
- Huge resources waste at simulation, verification and homologation of not existent configurations
- Errors and waste in production due to e.g. wrong software parameter for a configuration
- Inefficiencies in storage management due to lack of transparency in required parts
Semantic variant management | Complication

Inconsistent rule sets lead to high financial distress due to consideration of not existing configurations across the entire value chain.

Representation of configuration knowledge

Complication

- Non compliant rules sets in different systems and functional areas (mechanics, electronics, software)
- Satisfiability check for completeness, consistency and redundancy not/partly possible
- Huge resources waste at simulation, verification and homologation of not existent configurations
- Errors and waste in production due to e.g. wrong software parameter for a configuration
- Inefficiencies in storage management due to lack of transparency in required parts
Semantic variant management | Solution
Representation of rules as objects with semantic context incl. effectivities, versions and dependencies allows rule satisfiability analysis.

Transformation of configuration rules to a Knowledge Graph

Solution
- Integrated representation of rules, configuration parameters and their manifestations
- Rule engine can perform satisfiability checks of rule set within milliseconds
- Graphical representation and navigation in Browser
- Navigation and backtracking configuration logic
- Recommendation engine to resolve consistency conflicts
Semantic variant management | Solution
Representation of rules as objects with semantic context incl. effectivities, versions and dependencies allows rule satisfiability analysis.

Transformation of configuration rules to a Knowledge Graph

Solution
- Integrated representation of rules, configuration parameters and their manifestations
- Rule engine can perform satisfiability checks of rule set within milliseconds
- Graphical representation and navigation in Browser
- Navigation and backtracking configuration logic
- Recommendation engine to resolve consistency conflicts
Semantic variant management | Solution
Simple example

### Dependency matrix

<table>
<thead>
<tr>
<th>TOP</th>
<th>Motor</th>
<th>Sports Package</th>
<th>Stainless Steel</th>
<th>SPOILER</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>Variant A</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>Variant B</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>Variant C</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>Variant A</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>Variant C</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>Variant C</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>Variant C</td>
</tr>
</tbody>
</table>

### Code

```java
if (top == 0 && sports == 0 && steel == 1 ||
    top == 1 && sports == 0 && steel == 1 ||
    top == 0 && sports == 1 && steel == 1 ||
    top == 1 && sports == 1 && steel == 1 ) {
    spoiler = "variant a"
}

else if (top == 0 && sports == 0 && steel == 0 ||
    top == 1 && sports == 0 && steel == 0 ||
    top == 1 && sports == 1 && steel == 1 ) {
    spoiler = "variant b"
}

else if (top == 0 && sports == 1 && steel == 0 ||
    top == 1 && sports == 1 && steel == 0 ) {
    spoiler = "variant c"
}
```

### Graph

- **TOP**
- **Basis**
- **Sport**
- **Comfort**
- **Stainless**
- **Aluminum**

- 2 restrictions = 2 combinations
- 3 restrictions = 1 combination
- 1 restriction = 4 combinations

---

2019 Deloitte
Conflict detection and recommendation to resolve conflict.

Graph extract containing all nodes related to configuration conflict.

Configuration Rule Knowledge Graph for **frontend** – contains all dependencies related to frontend.

**recommendation**: add missing configuration rule.
Semantic variant management | Outlook
Knowledge based variant management enable software dependency management, supply chain risk analysis and portfolio evaluation.

Critical success factors to support highly complex decision making are:

A. Intelligent representation of real world context

B. Explanation of results to human decision maker

Traditional AI trained to solve complex problems can't explain its results to a human - semantic technologies are designed to fit both requirements.
Business Transformation with Knowledge Graphs
Deloitte supports its customers all around development, evaluation and implementation of semantic technologies within enterprise ecosystems.

<table>
<thead>
<tr>
<th>Strategy &amp; Operations</th>
<th>Underlying Technology</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEVELOPMENT OF NEW BUSINESS MODEL</td>
<td></td>
</tr>
<tr>
<td>PRODUCT AND SERVICE PORTFOLIO ROADMAP</td>
<td></td>
</tr>
<tr>
<td>BUSINESS TRANSFORMATION</td>
<td></td>
</tr>
<tr>
<td>SEMANTIC TECHNOLOGIES INCUBATION</td>
<td></td>
</tr>
<tr>
<td>360° ENTERPRISE ONTOLOGY DEVELOPMENT</td>
<td></td>
</tr>
<tr>
<td>KNOWLEDGE GRAPH MATURITY ASSESSMENT</td>
<td></td>
</tr>
<tr>
<td>DATA INTEGRATION &amp; GOVERNANCE</td>
<td></td>
</tr>
<tr>
<td>DATA ANALYTICS AND MACHINE LEARNING</td>
<td></td>
</tr>
<tr>
<td>ARCHITECTURE DESIGN</td>
<td></td>
</tr>
</tbody>
</table>
SPDM | Project overview
Creating knowledge out of unstructured data for enabling AI based solutions

The Case
1. Creation of metalevel ontology graph
2. Assignment of enterprise data to metalevel ontology
3. Implementation of semantic reasoner
4. Customized visualisation of data and creation of data analysis cockpit

Benefits for our Clients
- Connecting existing data and creating a single point of truth
- Inter-divisional impact analysis of product changes and decisions
- Real-time data analysis and discovery of hidden knowledge
- Usage of human-like language for querying
- Flexibility of data model and scalability of database

Our Services
- Creation of customized, graph-based metalevel ontology and implementation in a graph database
- Data transformation to graph and definition of rules for semantic reasoning
- Use case specific algorithm creation
- Customised visualisation of data and creation of data analysis cockpit
Building an ecosystem?
Connect the dots.