Ontologies and Reasoning to Capture Product Complexity in Automation Industry

Stefan Elmer[1], Foued Jrad[1], Thorsten Liebig[2], Anees ul Mehdi[1], Michael Opitz[2], Thomas Stauß[1], and Dirk Weidig[1]

[1] Festo AG & Co. KG, Esslingen, Germany
[2] derivo GmbH, Ulm, Germany

Abstract. The variety of components and the complexity of technical solutions in factory automation push information management based on relational databases to it's limits in terms of maintenance complexity and usage flexibility. Semantic Technologies account for maintainable, comprehensible and rich schema descriptions as well as state-of-the-art reasoning and SPARQL engines claim to deliver compelling performance. In the following we briefly report on applying ontologies and reasoning for managing complex product data in the automation domain.

1 Motivation

As a global leader in factory automation, Festo offers a wide spectrum of pneumatic and electric drive products to serve application domains ranging from microelectronics over food industry to heavy engineering. An electric drive train is composed of basic products: an axis, a mounting kit, optionally a gear, a motor and a controller. Festo manufactures thousands of such products which differ technically in frame size, speed, voltage etc. The modular design causes a combinatorial explosion resulting in millions of technically valid drive trains. According to the usual practice, product data and product compatibility codes are managed within a relational DB. In order to deliver tolerable response time for the compilation of valid drive trains, an enormous effort is required in terms of computing sets of materialized views that contain lists of feasible drive train elements for various tools and services. This approach is computationally expensive and requires a complex data maintenance and deployment management.

2 Dealing with Complexity in Electric Drives with Ontologies

As part of the Festo Semantic Platform (FSP) initiative OWL 2 and SWRL is used to better capture Festo's automation products. In the drives ontology, drive elements are organized according to their product family, type etc. They also have data property characteristics such as weight, size, etc. In case of a mechanically and electrically compatibility of two elements they are asserted to be direct compatible in the ontology (depicted as ⇹ in Figure 1). Property chains describe indirect compatibility between not neighboring components. The latter are automatically inferred by a reasoner which, in a relational model, would require joins within each particular application. Furthermore, the declarative information model with explicit classification of product characteristics increase the comprehensibility of the data and the development speed of applications.

The ontology schema also contains properties to represent part-of relationships (→ in Fig. 1) as well as aggregations (-►) to describe drive trains used in portals. Furthermore a 2D or 3D portal is concatenating several drive trains (-▷▷). SWRL is used to describe constraints imposed by product management to restrict the technically possible to the economically reasonable portal systems. The SWRL rules were iteratively worked out by ontology experts from informal statements verbalized on PowerPoint and Excel sheets. In addition, the ontology contains axioms to express implicit dependencies between components, e.g. to propagate information from drive elements to drive trains or portals. The schema part of the ontology was built using the Protégé ontology editor, contains about 500 classes and 60 object properties and is of horn-$\mathcal{SRIF}$(D) expressivity fully complying with OWL 2 RL plus SWRL.

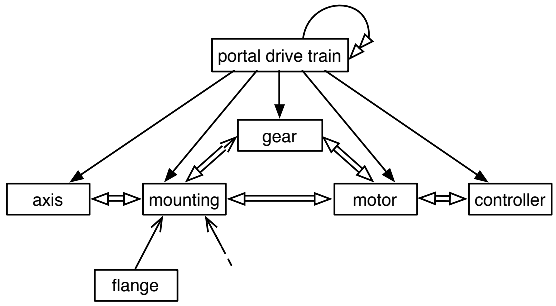

Fig. 1. Excerpt of the electric drives ontology

3 Fast and Flexible Knowledge Access with Reasoning and SPARQL

The ontology is populated via an import pipeline with instance data from different sources including master data from SAP and product development data from an Oracle DB. The very first benefit of the semantic approach was an increase in data quality. Many data flaws and inconsistencies were regularly reported with the help of automated tests that incorporate integrity checks based on SPARQL queries and entailment tests.

All inferences are materialized with the help of the RDFox rule engine prior to querying. After materialization querying for all valid drive trains or for 3D portals comprising axes of a particular family or transmission type is a matter of a few seconds. Furthermore, since we have now a declarative, comprehensible product model SPARQL queries are much more easier to write than the SQL queries before. In addition, extensions or changes in the information model due to technical or business demands are much faster accomplished and validated. That is a significant increase in flexibility as well as saves time and computing resources. For daily business a set of Web services and a SPARQL endpoint allow to deliver on-demand "views" on the complete product line taking account of all technical and business related constraints in contrast to the previous solution that had to run overnight to materialize predefined export queries.

Since the whole architecture is based on standards such as OWL, SWRL, and SPARQL as well as REST, JSON, etc. on top of a reasoning aware SPARQL backend there are no dependencies on particular vendors or software products. Furthermore, due to the compactness of the schema and data part of the ontology as well as the availability of multi-platform, easy deployable OWL RL/SPARQL engines the semantic solution is also ready to serve any on- and offline product configuration application.