Product Information Management Systems Powered by Knowledge Graphs

Amir Laadhar[1], Nikhil Acharya[1], Johann Wagner[1], and Martin Ley[1,2]

[1] PANTOPIX GmbH & Co. KG, Josephine-Hirner-Str. 2, 88131 Lindau, Germany
{amir.laadhar, nikhil.acharya, johann.wagner, martin.ley}@pantopix.com
[2] Munich University of Applied Sciences, Lothstraße 34, 80335 München, Germany
{martin.ley}@hm.edu

1 Introduction

Industries rely on Product Information Management (PIM) [1] systems to create, manage, and distribute product information across various channels. Traditional PIM platforms encounter numerous issues due to the complexity of product portfolios and an increasing heterogeneity of data sources. Moreover, there is the need to compare competitors' product offerings with the current product offering to achieve market and portfolio intelligence. Other issues include inconsistent data quality and cumbersome manual workflows faced by sales managers when there is no centralized source of truth. This leads to inefficiency in accessing necessary product data by company departments, detrimental to decision-making and competitiveness. Exploring potential solutions to these challenges led us to develop PIM systems based on Knowledge Graphs [5] for our industrial clients.

2 Product Navigator Application for Sales Managers

Sales managers are often faced with the hard task of collecting product offering information from multiple sources, which is a time-consuming and error-prone process. In addition to navigating several internal data sources, they have the added responsibility of manually tracking competitor product offerings across different external websites, further complicating their task. This makes it difficult to find accurate information quickly, deteriorating the ability to build strong customer relationships and stay competitive.

To cope with the mentioned issues, we are employing a knowledge graph as the heart of the PIM system. The knowledge graph contains or integrates product information originating from different data silos, represented in a unified structure in a central location. The product navigator web application serves as a tailor-made point of access to the knowledge graph for the specific needs of the sales managers.

3 PIM Network of Ontologies

We rely on a set of five ontologies as the backbone of the PIM knowledge graph. Figure 1 shows the main classes and properties of the PIM network of ontologies. This network contains five integrated ontologies to improve maintainability.

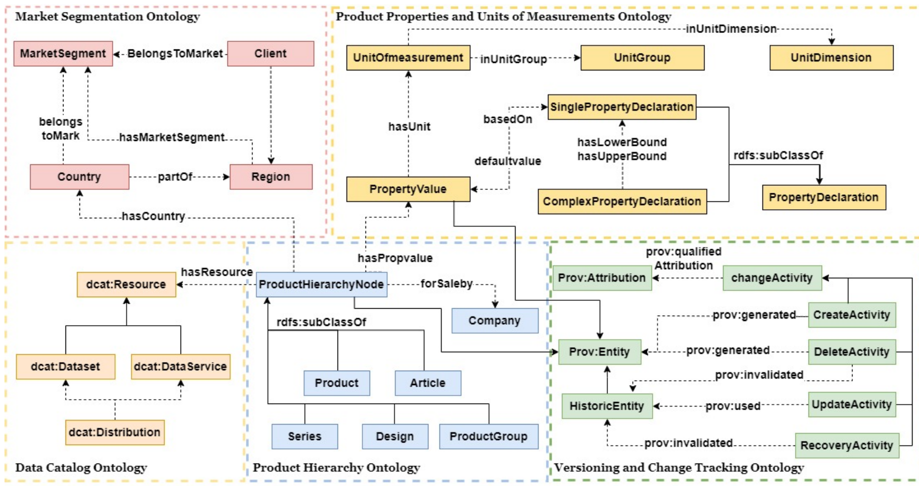

Fig. 1. PIM ontologies network

The main class of the product hierarchy ontology is `ProductHierarchyNode`. This class has sub-classes referring to different levels of the product hierarchy. The `ProductHierarchyNode` superclass is integrated with the other ontologies. It can be associated with `PropertyValue`, which is the value related to a specific product technical feature (e,g., pressure level of 6 bar). This ontology can reuse existing ontologies such as Building Product Ontology [6] or PRONTO [7].

The product properties and units of measurement ontology aims to model property values (`PropertyValue`) related to a given property declaration. A property declaration defines a technical feature that can be used to describe products, such as a nominal engine power. A product can be represented with an arbitrary number of property values associated to property declarations. Property values often require a reference to a unit of measurement. For example, an article can be associated with a property value instance with a value of 24.58 via the object property `hasPropvalue`. This Property value is `basedOn` a single property declaration "engine power".

We define a versioning and change tracking ontology based on the W3C PROV Ontology [2]. It records change activities made to the set of articles as well as their associated property values and property declarations. For instance, we can track historical changes made to an Article by the users.

We also integrate the W3C Data Catalog Vocabulary (DCAT) [3] in the PIM ontologies network. A `ProductHierarchyNode` can also be associated with a `dcat:Resource` via the object property `hasResource`. For instance, we can associate an article with data sheets, images, and datasets.

The market segmentation ontology models the different markets (i.e., construction market), countries, and regions. For example, a product can be sold for a specific country by a specific company. This allows the comparison of articles sold by competitors made for different countries or market segments.

4 PIM System Architecture

In Figure 2, we define the PIM system architecture overview, which has mainly four main layers: data sources, data onboarding, middleware, and serving layer.

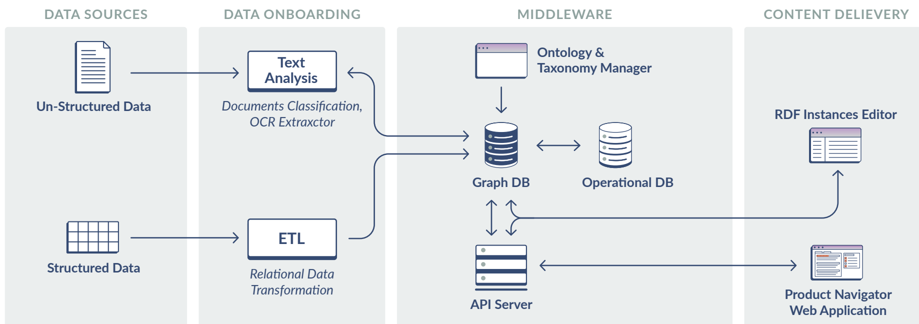

Fig. 2. PIM system architecture overview

4.1 Data Sources

We distinguish between two types of data sources: structured data and unstructured data. Unstructured data comprises mainly competitors' technical documents and websites. They are all classified as unstructured data because they contain paragraphs mixed with tables related to the technical features of different products and articles. We mainly use these documents to extract competitors' specific products and property descriptions for products that can be stored as RDF into the Knowledge Graph.

The structured data primarily comprises product datasheets and relational data sources containing predefined attributes having continuous or categorical values for each property of specific products like refrigerators and dryers. These structured data sources mainly have product property information like efficiency, rating, and pressure of specific products.

4.2 Data Onboarding

The goal of the unstructured data onboarding pipeline is to continuously extract competitors' products data and store it in the PIM knowledge graph. The stages to achieve unstructured data onboarding comprise documents crawling, documents classification, data extraction, and RDF generation. We begin by scraping competitors' websites searching and storing relevant public technical product documents. These documents provide a rich description of competitors' products. Then, we use a trained machine learning classifier to separate relevant product documents (i.e., products PDFs, product manuals) from irrelevant documents. Next, we rely on a trained OCR model to fetch relevant information from technical product documents. This data includes the article name, its technical properties, market, region, and manufacturer. We finally generate RDF triples from the extracted data and store them as competitors' articles in the PIM knowledge graph.

Structured data sources such as relational databases, and SAP APIs data contain specific product properties and values that we transform and onboard into the knowledge graph. This data onboarding pipeline is triggered automatically and periodically and involves a data validation process using SHACL [4].

4.3 Middleware

The middleware layer includes mainly an ontology and a taxonomy manager, a triple-store, a relational database, and an API server. The ontology and taxonomy manager is used to maintain the PIM ontologies and taxonomies. The relational database is used to store operational data such as access and control lists. The knowledge graph stores all the RDF data in different interconnected graphs, which increases the efficiency of the RDF querying. The API server defines a set of predefined API calls to be used by the serving layer for different applications including the product navigation web application.

5 Evaluation and Lessons Learned

To assess the knowledge graph's quality, we employed automated testing with predefined SPARQL templates. The templates dynamically generate SPARQL queries, which were then automatically evaluated against expected results. We have also used SHACL [4] for periodic validations of the knowledge graph.

Building a knowledge graph within an agile development methodology represents both benefits and challenges. Agile methodology allows adaptability to changing requirements, which takes advantage of the knowledge graph flexibility to onboard changes. This flexibility also brings the challenge of maintaining the integrity of the knowledge graph data. Therefore, we define three working environments (i.e., development, quality assessment, and deployment), and each of them is divided into three graph repositories (i.e., ingestion, maintenance, and consumption). Knowledge graphs are advantageous for PIM systems since they allow the handling of very complicated product portfolios and competitors' data, which open perspectives for more use cases.

References

1. Forza et al. Product information management for mass customization: connecting customer, front-office and back-office for fast and efficient customization. 2006.
2. W3C PROV-O Working Group. PROV-O: The PROV Ontology." W3C Recommendation, W3C, April 2013.
3. W3C Data Catalog Vocabulary (DCAT) Working Group. "Data Catalog Vocabulary (DCAT) – Version 2." W3C Recommendation, W3C, December 2020.
4. W3C Data Shapes Working Group. "Shapes Constraint Language (SHACL)." W3C Recommendation, W3C, July 2017.
5. Aidan Hogan et al. 2021. Knowledge Graphs. ACM Comput. Surv. 54, 4, May 2022
6. Wagner et al. "BPO: The building product ontology for assembled products." Proceedings of the 7th Linked Data in Architecture and Construction workshop (LDAC 2019)', Lisbon, Portugal. 2019.
7. Vegetti et al. "PRONTO: An ontology for comprehensive and consistent representation of product information." Engineering Applications of Artificial Intelligence 24.8, 2011