Transforming Domain Data into Semantic Models for Content Delivery Applications

Lars Tempelfeld [1,*] , Celine Wiedewilt [1,**], Anna-Maria Kruse [1,***] and Wolfgang Ziegler [1,****]

Author Affiliations
[1]*Karlsruhe University of Applied Sciences, Faculty of Information Management and Media, 76133 Karlsruhe, Germany*

Author Emails
[*]*tela1014@h-ka.de*
[**]*wice1011@h-ka.de*
[***]*kran1035 @h-ka.de*
[****]*wolfgang.ziegler@h-ka.de*

Abstract. This research focuses on transforming metadata from different systems in energy technology for transportation systems to create an ontology to dynamically provide cross-system information. Thereby, different approaches for transforming domain data and domain knowledge into ontology are investigated. We therefore analysed, which approach is best suited to the given business situation. In addition, the study and discussion address various challenges related to metadata extraction and transformation. Finally, this paper emphasizes the central role of ontologies and SCR for automation in the technical documentation and highlights the potential resulting from this integration strategy.

1 INTRODUCTION

In the field of technical information management, the need to provide precise information in the context of product information to users has strongly increased. In recent scenarios, this involves the development of an appropriate ontology model to compile and provide for example dynamically linked information using Semantic Correlation Rules (SCR) [1,2]. SCR are one of the basic possibilities in integrating semantics into content delivery. These rules can play a key role in automated processes for processing and providing technical information in digital information services. However, an important prerequisite is that the required metadata originating from different systems is exported, processed, and transformed to fit into the format and modelling patterns of ontologies. An ontology serves as a structured framework that defines the semantic relationships between different concepts and entities in technical documentation. This paper focuses primarily on the process of transforming metadata from a Spare Parts Catalogue (SPC) and a Component Content Management System (CCMS) into an ontology.

A SPC is a list of replacement components for a product or system. CCMS are generally used to create, classify, and publish information in a standardized manner. Content Delivery Portals (CDP) can be connected to the CCMS or SPC to display the required information to the end user. The framework of an ontology, which consists of classes and instances, is structured using the Resource Description Framework (RDF) markup language.

This research focuses on data from the field of energy technology for transportation systems where metadata has been applied to topics based on the PI-Class method used in many cases in CCMS and the hierarchically nested metadata of the SPC. In addition, we even show the handling with iiRDS, which is a standard of the European association of technical communication and originates also from the PI-Class methodology. We will show how multiple RDF files can be merged into one corporate ontology. Afterwards, the resulting ontology can be migrated to SPARQL for visualization and query construction purposes.

Semantic technologies to support model-based scenarios are increasingly being utilized. The aim of this paper is to document and evaluate the process of creating an automated ontology using different sets of metadata from various interfaces, with a particular focus on metadata based on the PI-Class method.

In summary, we will show the process identification of complex structures of classes and individuals and to recognize patterns in them as well as to establish relationships and connections between them.

2 CONCEPTUAL BACKGROUND

The PI-Class method is a standardized approach for categorizing metadata associated with modular content within CCMS [3]. Implementing the PI-Class method offers several benefits and improves the content creation, management, and consumption.

The PI-Class method essentially consists of two main classes: the product class and the information class. Each of these classes can be further divided into the categories extrinsic and intrinsic, resulting in a total of four classes:

- The intrinsic product class refers to product components such as the “Fuse box” or “Base frame”.
- The intrinsic information class defines the type of information associated with the content. For example, some parts may be designated for descriptions, while others may contain procedural explanations or safety instructions.
- The extrinsic product class deals with the affiliation or relationship of topics to certain product classes, for example, the link to the product class "SBE 920".
- The extrinsic information class is used to associate topics with specific media or document types, such as the document type "User Manual".

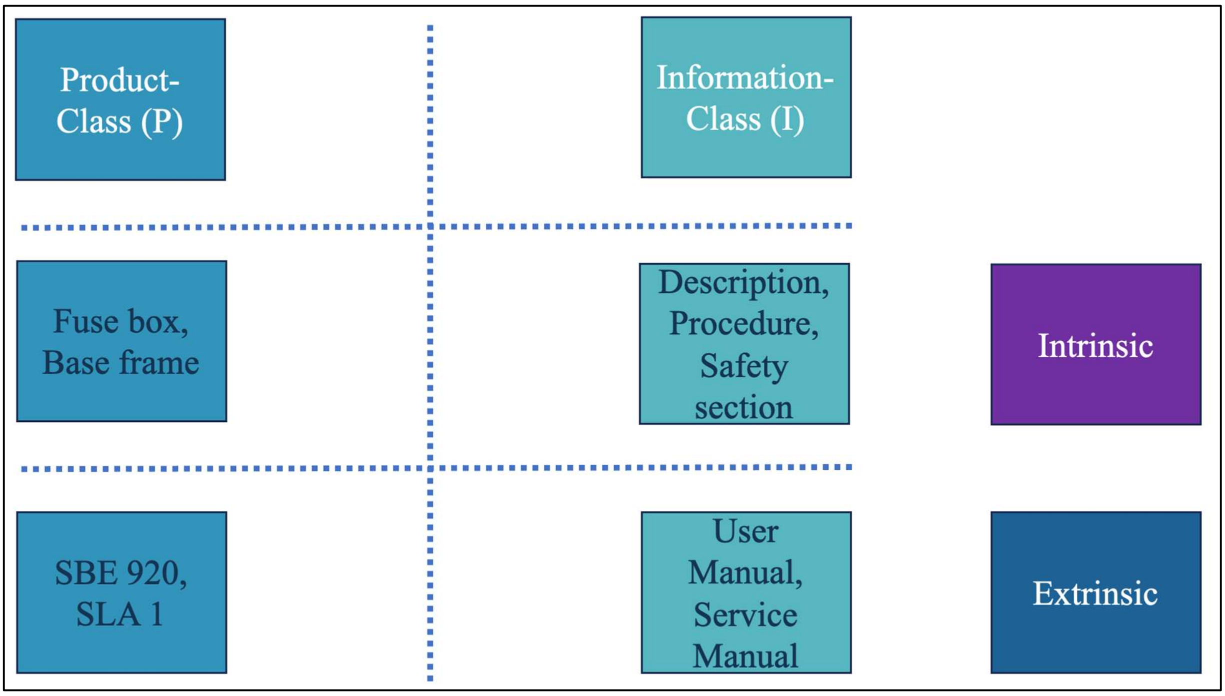

Figure 1. PI-Class

The PI-Class method enables organizations to establish a structured framework for the efficient organization and management of their content in the CCMS. This classification system facilitates optimized content creation, improves content management processes, user experience and access to information for users.

Two additional dimensions can extend the PI-Class framework:

- Functional metadata is used to manage tasks and schedules, for example, using items such as a maintenance plan or a checklist for start-up procedures.
- Conversely, variant features are suitable for specific product configurations, as illustrated by the AC and DC current examples, each of which includes variations in high and low voltage levels.

An ontology serves as a structured framework that defines the semantic relationships between different concepts and entities in the technical communication. There are four main components within an ontology:

- Object classes represent categories or types of instances.
- Instances are specific individual entities or examples that belong to defined object classes.
- Properties define relationships and attributes between classes and instances.
- Relations that are a specific property type display connections or associations between instances.

RDF and OWL syntax are employed for constructing ontologies.

3 PREPARING AND BUILDING AN AUTOMATED ONTOLOGY

The methodology for developing an ontology for SCR use consists of four main steps, as shown in the following graphic:

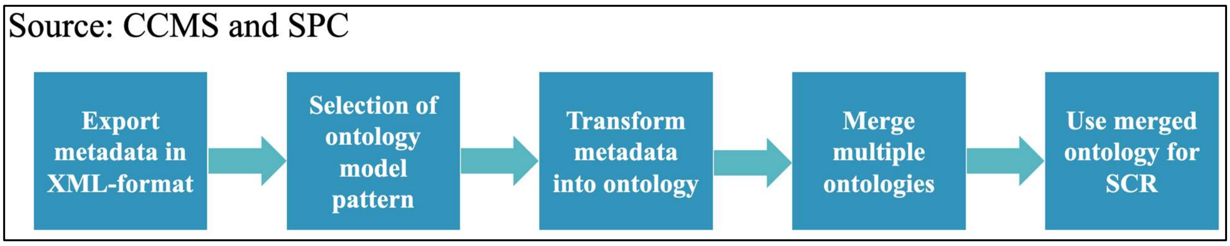

Figure 2. Workflow of ontology implementation

Export Metadata in XML Format

In this initial step, the primary objective is to obtain metadata in a standardized format, e.g. XML to enable a subsequent transformation of the metadata. While certain source systems can export metadata directly to the native XML format, other export formats may require the use of specific programs or scripts for XML transformation. For instance, using MS Excel allows for the conversion of XLSX files into XML format. It is worth noting that the used CCMS natively exports XML data, whereas the SPC was exported as XLSX.

Selection of Ontology Model Pattern

The following step includes the manual creation of the ontology structure. An individual ontology structure is required for each set of metadata. Tools such as the open-source program Protégé can be used to support this process in providing content and visualizing capabilities. By analysing previously extracted metadata, the components of the ontology can be determined, and their relationships can be defined accordingly. The metadata from the CCMS is based in our research project and the given industry data on the PI-Class method, the metadata from the SPC is hierarchically nested. Either a class-based or an instance-based approach can be considered for structuring the ontology. On the one hand, a class-based approach means nested classes to display the relations. On the other hand, in an instance-based approach, a main instance is used to contain all instances and the relations are placed directly on the instances.

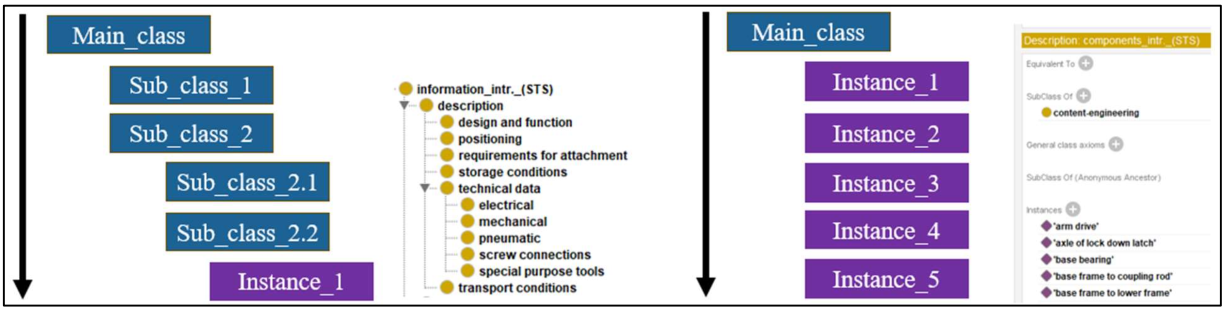

Figure 3. Class-based approach versus instance-based approach

Transform Metadata into an Ontology

Once the structure of the ontology is modelled, the metadata is transformed from XML to RDF and OWL syntax. The transformation from XML to RDF/OWL can be performed using an XSLT script or a tool like OpenRefine. Using tools can bypass the need to write scripts, but finding and learning to use a suitable tool can present challenges.

In addition, compared to programs, scripts can be tailored much more precisely to individual ontologies. Therefore, an XSLT script was written to transform metadata based on this research.

The following script parts are used to create the OWL class “content-engineering" and the relation property "has_parent":

```
<owl:Class rdf:about="https://metadata.schunk-group.com/sts#c_information_intr._(STS)">
   <rdfs:subClassOf rdf:resource="https://metadata.schunk-group.com/sts#content-engineering"/>
   <rdfs:label xml:lang="en">information_intr._(STS)</rdfs:label>
</owl:Class>
```

Figure 4. OWL class creation: „content-engineering“

```
<owl:ObjectProperty rdf:about="https://metadata.schunk-group.com/sts#has_parent">
   <rdfs:subPropertyOf rdf:resource="http://www.w3.org/2002/07/owl#topObjectProperty"/>
</owl:ObjectProperty>
```

Figure 5. OWL relation property creation: „has_parent“

The class "content-engineering" was designed to encompass the metadata of the CCMS, whereas the metadata of the SPC is included within the class "prod-engineering," with "prod" being short for product. The reason for this is to maintain the separation of metadata based on their system origin when merging both CCMS and SPC ontologies. This ensures clear attribution and identification of metadata even after the ontologies are combined.

The relationship property "has_parent" is universally applicable for representing relationships between instances. This means that an instance is a child of another instance, which in turn can also have a parent instance. For example, within the SPC ontology, the product part "bearing" has the parent "upper frame bearing," and this instance has the parent "lower frame complete”. Due to its general applicability, this type of relationship can also represent the relationships of CCMS metadata. For example, the instance "assembly" has the parent "procedure".

In RDF, direct relationships between classes are avoided as RDF is primarily used for describing instances and their properties. Instead, relationships between instances are modelled to express specific information about resources. Placeholder instances can be utilized to indirectly represent relationships between classes, serving as an intermediary step. For example, the class "adjustment" includes the placeholder instance "adjustment," which has the parent "procedure", itself being the placeholder instance of the corresponding class.

Merge Multiple Ontologies

The transformation can result in multiple ontologies. These individual ontologies can be merged into a single comprehensive ontology, which facilitates processing in subsequent steps. The merge process can be managed and executed using an XSLT script. As a basis for the research, an ontology from the CCMS and two ontologies from the SPC are combined. as illustrated in the following graphic:

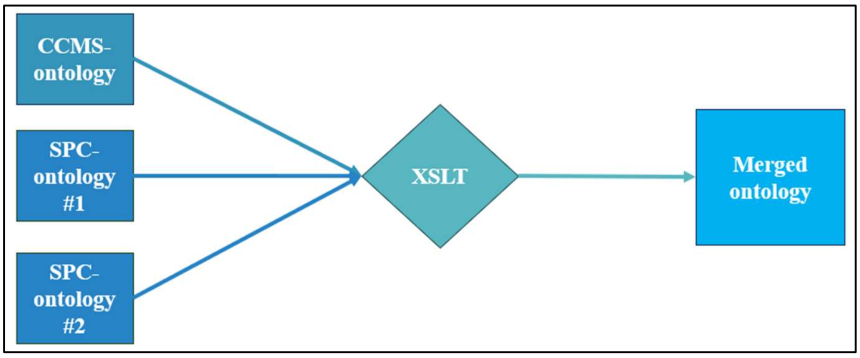

Figure 6. Merging different ontologies into one ontology

Use Merged Ontology for SCR

The merged ontology can be used for the implementation of SCR, which were discussed in [4]. An additional sophistication in the research project was the implementation of iiRDS. iiRDS is an „intelligent information request and delivery standard" [5]. The standard includes elements from PI-Class, RDFS, and its own elements (classes and instances). iiRDS can be used to standardize the exchange of delivery formats. In the research use case, the standard was used to establish communication between the underlying CCMS and the CDP via class mapping mechanisms from PI-Class to iiRDS. The reason for this was the use of iiRDS as an import format for content integration.

4 KEY FINDINGS

Creating an ontology based on metadata in accordance with the PI-Class can be achieved through scripting methodologies. These scripts automate the extraction and organization of metadata according to the PI-Class schema, ensuring that the ontology aligns with existing data structures and standards. In an industrial environment, the number of scripts will be limited by the finite number of CCMS and their initial metadata concepts. In this case, the use of PI-Class or similar concepts simplifies the standardization of the export routines. On the other side, in the case of a data governance approach encountering a company-ontology as the model base, the export routines depend again on the model patterns and not on the CCMS.

In our case, implementing additionally iiRDS requires further mapping of metadata elements to iiRDS specifications, guaranteeing compatibility with iiRDS-compliant systems and tools. However, developing an ontology that is not rooted in metadata based on the PI-Class might demand different approaches. Instead of relying on pre-existing metadata structures, this method involves defining ontology classes and properties independently. It necessitates employing knowledge engineering techniques such as domain analysis and expert input to construct an ontology from scratch. One of the challenges with metadata-based ontologies is the potential for metadata to change or expand over time. As such, flexibility in ontology design and versioning mechanisms becomes crucial to accommodate evolving metadata requirements. Moreover, when merging ontologies, coordination and adaptation consist of an agile and dynamic processes. Integrating multiple ontologies requires careful alignment of concepts, properties, and relationships to ensure coherence and consistency across the merged ontology.

When considering the ontology modelling approach, it's essential to decide whether to represent concepts as instances or classes, especially concerning classes within the PI-Class schema, such as "Intrinsic Information" and "Component Intrinsic." For "Intrinsic Information", modelling it as the class-based approach allows for abstractive relationships. This offers the advantage of providing a clear overview of instances. However, it comes with the disadvantage of lacking direct relations between these classes. To overcome this limitation, incorporating generic instances might be necessary. On the other hand, "Component Intrinsic", can be modelled as instances. This establishes partitive relationships, making it easier to implement relations. Nonetheless, this instance-based approach may result in a less clear overview of instances. Therefore, the choice between modelling as classes or instances within the PI-Class schema depends on specific ontology requirements and the nature of the metadata being represented. As an outlook, the research has shown, that CCMS providers should aim to develop export functions for ontology formats based on the stored metadata with the optional ability to cover the iiRDS format. This would facilitate the manual process of selecting ontology structures and transforming metadata into ontologies.

ACKNOWLEDGMENTS

This work was supported by Schunk Transit Systems and the information they provided. The authors gratefully acknowledge the helpful discussions with Marcus Bartel, Stephan Bartl, Sebastian Ries and Nino Tiezel (all Schunk Transit Systems). Furthermore, we acknowledge the technical support by Florian Rommel, Maurice Daum and Lena Wenner (all ICMS). We would also like to thank the system providers involved, namely Empolis and Quanos for the provision and adaption of their systems.

Additionally, we want to thank the International Office of Karlsruhe University of Applied Sciences and the study program of Communication and Media Management for the financial support.

REFERENCES

1. F. Rommel and W. Ziegler, *A Microservice-based Infrastructure for the Academic Research on Semantic Knowledge Graphs and Derived Applications.* Proceedings of the ETLTC ACM Chapter International Conference. Aizuwakamatsu, Japan. (2022) https://doi.org/10.1051/shsconf/202213902002
2. W. Ziegler "Basic Concepts Which Support the Management and Delivery of Intelligent Content" in *Frontier* (Official Journal of Japan Technical Communicators Association JTCA, 2017), p. 84-91
3. W. Ziegler, *Semantic Correlation Rules as a Logic Layer between Content Management and Content Delivery*. Proceedings of the ETLTC ACM Chapter International Conference. Aizuwakamatsu, Japan. (2021) https://doi.org/10.1051/shsconf/202110202007
4. A. Kruse, C. Wiedewilt, L. Tempelfeld, *Implementing Semantic Correlation Rules in the Domain of Energy Technology for Transportation Systems.* Proceedings of the ETLTC ACM Chapter International Conference. Aizuwakamatsu, Japan. (2024)
5. W. Ziegler, *Extending intelligent content delivery in technical communication by semantics: microdocuments and content services*. Proceedings of the ETLTC ACM Chapter International Conference. Aizuwakamatsu, Japan. (2020) https://doi.org/10.1051/shsconf/20207703009
6. L. Brecht and M. Derwand, *Ontology Modelling and Testing of Semantic Correlation Rules in an Open-Source Content Delivery Portal*. Proceedings of the ETLTC ACM Chapter International Conference. Aizuwakamatsu, Japan. (2023) https://doi.org/10.1063/5.0183079