

Ontology-based modelling for content delivery systems

Lena Mareike Padeken^{1*} and Annika Johanna Bürkel^{1**}

¹Karlsruhe University of Applied Sciences, Faculty of Information Management and Media, 76133 Karlsruhe, Germany

Abstract. Instead of managing metadata in a content management system (CMS) for content delivery, this research uses an ontology to make use of augmented intelligence. Hence, the retrieval of specific information in a content delivery system (CDS) is provided by a concise metadata concept. Ontology-based metadata concepts can have advantages in comparison to taxonomy-based metadata concepts because of the possibilities they offer to display linked information. The creation of an ontology depends on the vendor's system, the vendor's approach and most importantly the particular use case. Due to the ontology modelling, a data transformation for an external CDS is needed. When working with different vendors, requirements and considerations need to be discussed because the outcome can add value to user experience. The goal of this research is to identify the advantages of an ontology system over metadata management in a CMS. These advantages can be extracted by opportunities offered by the ontology modelling and by the transformation process.

1 Introduction

An easy accessed and customized retrieval of information is highly demanded in times of information 4.0 and intelligent information. Users do not want to be overwhelmed by too much information; they need the right information at the right time. Due to changes in user behaviour and technologies, content delivery systems (CDS) are used to enable web access to information depending on specific use cases [1, 2]. In most cases, content management systems (CMS) are used to manage modular content and the corresponding metadata. These metadata can also be used to retrieve content modules in a CDS. Another way to provide information for a CDS is using a knowledge network like an ontology. Ontologies are more reality-oriented as they not only represent the hierarchical structure of metadata but also realise cross-linked relationships between elements of different levels. For the ontology modelling process, it is key to make observations about potential use cases. This helps specifying the user's needs and simultaneously gain understanding about the needed search and filter options.

To deploy information in a CDS, vendors' either must own a CDS or need to carry out a transformation process to import content into a CDS [2]. Running one's own transformation process offers the chance of balancing the content and context deployment for a CDS. Possible advantages can be found by analysing the ontology output file and the requirements for the data import into an external CDS. This way, content, context and facets can be adjusted for an ideal user experience.

2 Context of research

This paper resulted out of a collaboration between students from the master's program at Karlsruhe University of Applied Sciences, Germany, and students from University of Aizu, Japan. As part of the course *XML-Based Information and Content Management* by Professor Wolfgang Ziegler, these students developed an ontology as a metadata model for content delivery on the subject of *Smart Home Technologies*. This knowledge network helps to retrieve intelligent information on a CDS. Therefore, a transformation from the data given in the ontology to the external CDS was needed.

In order to carry out this research, the students from Karlsruhe got access to the ontology system i-views Knowledge Builder and the content delivery system DOCUFY TopicPilot. Thus, this research mainly refers to the above-mentioned vendors' options and functions.

3 Theoretical overview

3.1 Importance of metadata

Metadata are additional information to describe, manage and identify objects such as modules in a CMS. Additionally, metadata are also used to aggregate product or context specific output in a CDS. In fact, cross-linked metadata allow a complex and automated aggregation of information. Therefore, metadata are the basis of intelligent content management and content delivery [3]. This intelligence manifests in the extraction of specific information through facets in content delivery systems and in the automation of product documentation [4].

*Lena Mareike Padeken: pale1014@hs-karlsruhe.de

**Annika Johanna Bürkel: buan1024@hs-karlsruhe.de

3.2 Ontologies as a concept of organising metadata

Metadata can be organised in a hierarchal system like a taxonomy. It describes a two-dimensional visualisation of elements containing classes and subclasses in a parent-child relationship. But when it comes to relationships between subclasses of the same or different class, a taxonomy reaches its limits. An ontology, in contrast, is extended with relations between elements on different levels. If the same underlying ontology is used, information of different origins can be combined in a common output medium [5]. Also, the usage of an ontology simplifies the search process in a content delivery scenario. Through the linking of contextual relevant information, a more precise search result can be displayed. Furthermore, facets can be managed in an ontology and adapted in a content delivery solution.

Ontologies are already widely used in the area of knowledge management and several other areas. The next step then would be to apply the benefits of ontologies to technical communication. However, the concepts of ontologies are relatively new in this field. But the demands of contextual linked information in search scenarios of content delivery systems are growing. In content management as well as content delivery systems, metadata models can be extended using an ontology and thus creating a more intelligent content. The benefits of ontologies are the following: Due to the usage of formal language to describe them, they are machine readable and therefore a basis of machine learning and artificial intelligence [6]. For technical communication this means that new opportunities can be created for knowledge sharing and displaying contents.

Yet, there is neither standardised methodology in ontology engineering nor a standard language to document ontology modelling. However, the two most common modelling languages used are RDF (Resource Description Framework) and OWL (Web Ontology Language).

3.3 Importance of content, context and facets

Deploying information to users is often defined by too much content and too few context [7]. Considering a CDS, topics can be displayed individually by running a search, selecting facets or as hierarchical structures like table of contents or directories. Hierarchical structures often imply a document-oriented CDS approach. This document-oriented approach is for example used by the CDS of SCHEMA [2]. In contrast, the CDS TopicPilot of DOCUFY uses a topic-oriented approach, but also displays (hierarchical and non-hierarchical) directories next to each topic. Providing hierarchical structures can result in too much content for users because they contain non-relevant information, whereas topics without references create not enough context. The later explained transformation process introduces an approach to enrich information deployment by using the connection between an ontology and a CDS (see chapter 7).

3.4 Thematic classification into the project

In this research project, the ontology marks the intermediate step between content creation management and content delivery (Fig. 1). For content creation and management, the modular-based content management system Smart Media Creator (SMC) was used. Different content modules about the topic *Smart Home Technologies* were collated. The aim of this project was to enrich the content we created in the CMS with the metadata from the ontology system to display user specific information in an external CDS.

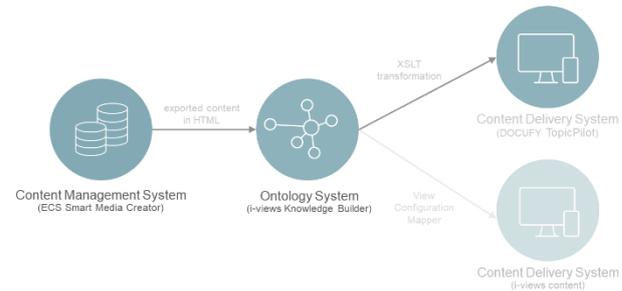


Fig. 1. Correlation of the software systems.

For this research, the data from the ontology system was used to feed an external CDS. Reasons for using an external system are that company owned CDS are often aligned to their own company language and company philosophy including their approach. This leads to a one-way thinking which channels processes in one direction and results in lost and missed chances of identifying and using better alternatives. In this research, we figure out possibilities, chances and advantages by using an ontology with linkage to an external CDS. It is our goal to find out how individually we can feed a CDS by following the requirements.

4 Ontology modelling with the i-views Knowledge Builder

4.1 General introduction in i-views

Intelligent views GmbH (i-views) is a German company founded in 1997 that is specialised in providing semantic technologies [8]. Part of the i-views solution is the Knowledge Builder which allows the modelling of metadata in the form of a knowledge network. The i-views Knowledge Builder allows the creation, editing and graphical visualisation of metadata and their relations in an ontology. Users can work in the textual and in the graphical interface (see Fig. 2). Both options offer different aspects and overviews. Also, contents from various CMS can be integrated via common standards and cross-linked to related elements in the ontology. The corresponding CDS ‘i-viewscontent’ allows to display the imported content which is directly linked to the metadata from the ontology system. For more information about the i-views internal delivery system, see “Ontologies and use case based planning of content delivery” (Burkhardt and Clesle) in this volume. Logical queries help retrieving use

case specific information in the knowledge network as well as in the i-views content delivery solution. These queries can be realised as facets, guiding the user through possibilities of content search. Also, these facets from the ontology system can be reused for facet creation in an external CDS (for more information, see chapter 6.3.3).

Within the scope of the course, the i-views Knowledge Builder was chosen as the main tool to model the ontology. To understand further approaches in the ontology creation process, a short introduction into the internal terminology of the Knowledge Builder is given. A knowledge network in the Knowledge Builder consists of four main elements. 'Types of Objects' (Types) describes the superior area an object belongs to [9]. In the area of Smart Home Technologies, the Type 'PhysicalObject' can have multiple objects, for example 'SmartHomeDevice' and 'ControlDevice'. These hierarchical relations are reflexive. The lowest element in such a hierarchy is called 'Instance' (or special object), it classifies the most specified object of a chain of objects. These instances can be connected through a 'Relationship'. Relationships can have different validities and therefore be valid for selected instances of objects [8].

4.2 Use case oriented ontology modelling

Due to simplicity, the following process description is shown in chronological order. In the practical realisation of this project, the different steps depended on one another and resulted in retroactive changes and adaptations.

As a preparatory step, we collected content about the topic *Smart Home Technologies* in the CMS Smart Media Creator, which served as the basis for the actual ontology modelling. At this point, we were supported by the students from Aizu University, Japan, that collected and prepared topics about Smart Home Technologies and provided them for us. Then, the topics were exported without metadata in HTML-format and manually transformed to be imported into the i-views Knowledge Builder.

To get a more precise metadata concept and on this basis a purposeful outcome, different use cases must be taken into consideration in advance. We collected information about the user's knowledge level, their intends, the user's input and the displayed content output as well as the information search scenario. The created use cases mainly focus on gaining first insight and general information about the topic Smart Home Technologies from non-specialists.

For more details about the use cases created for the ontology modelling in the Knowledge Builder refer to the article "Ontologies and use case based planning of content delivery" (Burkhardt and Clesle) in this volume. In this paper, we use one use case as an example to outline the functionalities of the modelling software i-views Knowledge Builder.

The next step marks the creation of 'Objects', their 'Instances' and their connection through 'Relationships'. Therefore, the first approach is a hierarchical organisation of 'Objects'. These parent-child relationships are called 'is object of' and 'has object' and are automatically drawn

when creating objects in a hierarchical order. For example, the object 'SmartHomeDevice' has the instance 'HomeSecurityAndMonitoringSystem'. Furthermore, additional relationships can be created to connect instances of the same or different objects. To continue with the example, the instance 'HomeSecurityAndMonitoringSystem' is connected by the relationship 'Serves' to the instances 'Comfort', 'ProtectionAgainstAttacks', 'ProtectionAgainstFire' and 'ProtectionAgainstWaterDamage' of the object 'Purpose', which are represented by the red elements in the graphic below.

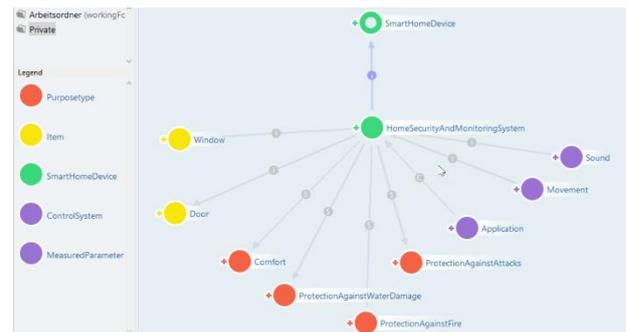


Fig. 2. Graph editor view in the i-views Knowledge Builder.

After modelling the metadata concept, the exported content modules from the CMS have to be implemented into the ontology. Therefore, the i-views Knowledge Builder offers a section called 'Chapter' to upload HTML-content. At the time of this research, an iiRDS-export has not been possible yet, therefore the content modules had to be inserted manually. Every module from the CMS was implemented as a 'Chapter'-element which could then be connected to the objects and instances of the ontology via the relationship 'has facet term'. Every chapter could then be connected to the corresponding metadata and in this way enable a three-dimensional metadata concept.

5 Process for the data import into the content delivery system

The process of providing information in a CDS by using a CMS is the same as providing information by using an ontology. In a heterogeneous system landscape [2], the information of a CMS or an ontology needs to be imported by a content package with the use of an own transformation. The only difference between a CMS and an ontology during this process is the export file format. The CMS export file is usually provided in an xml-format while the ontology export file usually is provided in a descriptive language format. This leads to different adjustments of the transformation process and to different possibilities of utilising the ontology output data. To transform the content of the export file, XSLT is used to generate files for the content package. Content packages are needed to create content in a CDS.

6 Content delivery system planning

The transformation process depends on many considerations and needs to be adjusted to the given conditions. An example for difficulties while planning is that the ontology system i-views Knowledge Builder describes the content in a book-oriented way and the CDS TopicPilot displays the content in a topic-oriented way [2]. Therefore, the export data and the import data for the CDS need to be analysed closely. Furthermore, the possibilities of maintaining content, context and the resulting facets in a balanced way will be discussed.

6.1 Structure of the ontology output

The significant components of the output file for the transformation can be divided into four parts. These parts are responsible for satisfying the content package requirements (see 6.2.2). The first part describes the book structure by *document roots*. *Document roots* are the main chapters inside a book and include *containing chapters* which are described individually in the second part as *chapters*. These *chapters* can have facet-like elements which are attached to a *chapter* as a *facet term*. The content of each *chapter* is described in the third part in hidden HTML (HTML displayed by entities). The fourth part contains the definition of each *facet term*.

6.2 Components of DOCUFY TopicPilot

6.2.1 Displaying content

The CDS TopicPilot of DOCUFY GmbH offers three general ways of finding information in the CDS [10]. The content is displayed in topics which can be accessed by searching for specific keywords, filtering metadata and navigating through related topics inside a *Publication*.

When *searching* for specific content by using the search field, the CDS TopicPilot offers keywords for matching and completing the written letters. The CDS TopicPilot searches through topics, trees and metadata [10]. *Facets* are used to filter content in a CDS and to give an orientation to users to find information according to a specific topic and its related information [1]. This enriches the searching process. Directories are placed next to each topic to enhance the *navigation* process between connected topics. All connected topics belong to the same content package and are also displayed inside a *Publication* container (*tree*) on top of the CDS [2, 10].

6.2.2 Requirements for data import

To import data into a CDS, vendors have different approaches (e. g. document-oriented or topic-oriented) and requirements because there is no standardised structure or content for a content package [2]. Adding content to the CDS TopicPilot is realised by uploading content packages in ZIP-format. These content packages consist of (1) files for the content (topic), (2) a filter hierarchy (facet-data), (3) a reference structure of the

topics (tree), (4) a media folder which contains all media files and (5) a manifest file (MANIFEST.MF) for declaring the version of the manifest itself and the DFXML-version. DFXML is a mark-up language and was developed by DOCUFY based on XHTML.

6.3 Usage of content, context and facets

The structure of the ontology output file and the requirements of the content package for the CDS give an insight to the chances that can enrich content deployment by adjusting the data transformation. These chances are discussed considering possible implementations of content, context and resulting facets.

6.3.1 Content

Since the CDS TopicPilot is topic-oriented, there is no high risk of having too much content. Each topic content is attached to a chapter in the ontology output file. The additional directory shown in the CDS is built by the tree file in the CDS TopicPilot and is based on all topics displayed in the CDS. It is mandatory to upload a tree file, which on the other hand makes it difficult to implement major changes in the directory representation.

6.3.2 Context

The ontology output file provides nested structures between chapters and facets as well as relations between facets. Nested structures can add context information by static linking in the CDS TopicPilot to generate links between topics. Relations between chapters can also add context information because most *chapters* contain *facet term* elements. These *facet terms* have relations between each other, which can be used for dynamic linking in the CDS TopicPilot. Dynamic links provide shorter access to information by a direct facet link.

6.3.3 Facets

Facet term elements are also of great importance to the facet structure in the CDS TopicPilot because the export file of the ontology does not provide a facet structure itself. The book structure of the ontology output file offers a structure consisting of *document roots* and *containing chapters*. These *document roots* and *containing chapters* may have defined facets which are declared within an own superordinate local name (namespace) at the end of the document. This rough structure already includes a facet structure based on the document structure itself and based on the facets of the *document roots* and *containing chapters*. Using facets based on the document structure is undesirable for a CDS because these facets would be displayed as a table of contents. Therefore, the facet structure should resemble the structure of the attached *facet terms* of the *chapters*. Allocated facets for the *document roots* cannot be used, because these *document roots* do not include any HTML. Thus, only facets for

elements to link and nest given topics into a reference structure. It contains an *ID* attribute which is used for internal representation [2, 10]. Inside the <node> element, <topicref> elements refer to a specific topic using the *href* attribute. Based on the given structure of the ontology output file, the tree structure will not be nested because only *chapters* will be used (see chapter 6.3.3).

7.4 Facet file

The facet file consists of nested metadata attached as facet definitions in the first part of the document and of references between the defined facets and the assigned topics in the second part of the document [2]. Topics reference to facets with the lowest hierarchical level. Not all topics need to have a facet and not all facets need to be referenced by a topic which is why only referenced facets and topics will be displayed in the CDS [10]. The structure of the ontology output file for the facet structure is a combination of the facet terms of the relevant chapters and the relations between the facet terms. Each facet is assigned to one of four <facet-tree> elements because each type attribute of a <facet-tree> represents one of the four basic classes of the PI-Class® method (<https://www.i4icm.de/en/research-transfer/pi-classification/>).

8 Outlook

This research paper only shows the realisation of one content delivery solution based on an ontology. The data transformation, however, can be adjusted for multiple CDS and the ontology can be used as a model for different content inputs. Due to the concrete specifications of the knowledge network, content elements are machine readable, which enables the usage of this content according to artificial intelligence.

9 Conclusion

This research has shown that ontologies do not only grant advantages in contrast to a CMS but also offer transformation processes for the data import into a CDS that lead to unforeseen advantages. Ontologies as knowledge networks use relationships to set logical references between elements and use relations in more than one context. These relationships were used for the transformation process and were able to enrich context information for users by offering static and dynamic linking inside topics. The filtering of the facet file is restricted because of the considerations regarding the adoption of the given book structure. Since a book structure is not helpful in enriching the information deployment, an ontology sets clear restrictions of how to be able to use context information. This has the advantage that users do not have too many options when searching for information, helping them to find information with as little effort and little time as possible. Considering this aspect and the representation of content inside a CDS, the possibility of not being stuck to a document or book structure is given within this research. Further research on

using even deeper relations inside an ontology is recommended to enrich the user experience even more.

We would like to give thanks to all the institutions which rendered our journey to Japan possible and helped with the realisation of this project. We therefore thank the International Office, the Faculty of Information Management and Media, the AStA Studierendenschaft HsKA, the Master's program Communication and Media Management for financial support, Dominik Kremer from DOCUFY GmbH and Achim Steinacker from intelligent views GmbH for great collaboration and insightful discussions. Especially, we would like to thank Prof. Dr. Ziegler of Karlsruhe University of Applied Sciences, Germany, and Prof. Dr. Roy of University of Aizu, Japan, for organising the exchange between the countries and for managing the whole project.

References

1. A. Ahmadpour and A. Wilhelm, *From Information Modeling of Technical Content to Intelligent Delivery*, in ACM Chapter Proceedings on Educational Technology, Language & Technical Communication, ACM 2019, pp. 3–9 (2019)
2. K.-B. Hanczaryk, *XML Content Transformation and Import Interfaces for Content Delivery Portals*, in ACM Chapter Proceedings on Educational Technology, Language & Technical Communication, ACM 2019, pp. 25–34 (2019)
3. W. Ziegler, *Intelligente Informationen*. in Schriften zur Technischen Kommunikation, Volume 22, pp. 51-53 (2017)
4. W. Ziegler, *Basic Concepts Which Support the Management and Delivery of Intelligent Content*, In Frontier, Official Journal of Japan Technical Communicators Association JTCA, p. 89 (2017)
5. N. F. Noy and D. L. McGuinness, *Ontology Development 101: A Guide to Creating Your First Ontology*, Stanford University, Stanford, CA, https://protege.stanford.edu/publications/ontology_development/ontology101-noy-mcguinness.html (n.d.)
6. D. Gašević, D. Djuric, and V. Devedžic, *Model Driven Architecture and Ontology Development*, Springer Verlag, Berlin, Heidelberg, p. 46 (2009)
7. W. Ziegler, *Delivery zwischen Kontext und Content*, in technische kommunikation, 41(6), pp. 58–61 (2019)
8. intelligent views GmbH, *About us*, <https://i-views.com/en/about-us/> (2019)
9. intelligent views GmbH, *i-views 5.3*, <http://documentation.i-views.com/manual/5.3/86cc538a-370b-4899-b711-191aca85bdf#4aa2898f-2bf5-4cdc-b41a-aebd09b6c56a-chapter> (n.d.)
10. DOCUFY GmbH, *TopicPilot 3.5 – Importspezifikationen*, pp. 8–12, 17, 20 [unpublished] (2019)
11. DOCUFY GmbH, *Spezifikation – DFXML2*, pp. 22, 104 [unpublished] (2018)