AIP Conference Proceedings

RESEARCH ARTICLE | JUNE 16 2026

Aligning the provisioning of product information with agile software development by semantics and correlation rules

Sarina Muench; Katrin Eckert; Lejla Bratic; Wolfgang Ziegler

AIP Conf. Proc. 3432, 060006 (2026)
https://doi.org/10.1063/5.0329892

View Online Export Citation

Articles You May Be Interested In

Standardized generation of semantic domain models from content management metadata

AIP Conf. Proc. (June 2026)

Implementing cross-domain linking of service information in the aviation security sector

AIP Conf. Proc. (June 2026)

Challenges and opportunities of semantic information management and cross domain linking in the context of manufacturing

AIP Conf. Proc. (June 2026)

18 June 2026 16:39:25

AIP Publishing

Aligning the Provisioning of Product Information With Agile Software Development by Semantics and Correlation Rules

Sarina Muench [a)], Katrin Eckert [b)], Lejla Bratic [c)] and Wolfgang Ziegler [d)]

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

[a)] musa1020@h-ka.de
[b)] ecka1011@h-ka.de
[c)] brle1021@h-ka.de
[d)] Corresponding author: wolfgang.ziegler@i4icm.de

Abstract. This paper addresses the implementation of a semantic information model for the use case-based content delivery of product information in the software industry, considering agile software development and its software-specific requirements and circumstances (like feature-based delivery processes and role management). The solution provided offers an approach for the flexible delivery of content, differentiating between different roles and systems across a company's system environment using a common semantic information model for central governance. A standardized XSLT Stylesheet serves as a foundation for content structuring, while the integration of sub-ontologies enables adaptation to software-specific requirements, such as role management and feature-based correlations. In addition, the establishment of semantic correlation rules facilitates the delivery of content in both context-oriented and process-oriented ways. This ensures that information is adapted dynamically to user roles, system environments, and the customer journey.

18 June 2026 16:39:25

INTRODUCTION

Digitalization is having a profound impact on the field of Technical Communication (TC). The range of digital information products is expanding to include semantic technologies and digital services, which in turn necessitates the development of more intelligent information [1]. The objective is to facilitate applications in accessing data from diverse sources within a company and delivering customized information, for instance via Content Delivery Portals (CDP) [1, 2].

One method for achieving this involves the use of Semantic Information Models (SIM). These form the basis for Semantic Correlation Rules (SCR), which are constructed from use cases on top of the SIM [1]. Previous studies have explored this method across different industries [1, 3]. However, the applications in software industry have not yet been covered in previous studies. Since working methods, tools and products are different compared to those in other industries, the implementation of a SIM might also differ.

The aim of this paper is to report on the development process of a Semantic Information Model (SIM) for a project and research partner from software domain. From this experience, we derive practical guidelines for a SIM implementation in software technology domains, including an overview of the required tools. Software development is characterized by its agile nature [4]. This field is characterised by the continuous development of new products, the utilization of software development tools as a support mechanism, and the presence of multifaceted roles among employees [5, 6, 7].

The paper investigates the implementation of a SIM in agile software development, with a particular focus on role management, role-specific content delivery and its integration into the system environment considering agile feature-based development processes and the developer's environment. We will discuss the derivation of use cases from agile development user stories, the incorporation of role management into a SIM for tailored content delivery, and the extension of content delivery by considering the customer's context and journey within the system using different approaches for a flexible and customer-friendly content delivery. By addressing these issues, the paper demonstrates

ETLTC2025 International Conference Series
AIP Conf. Proc. 3432, 060006-1–060006-10; https://doi.org/10.1063/5.0329892
Published under an exclusive license by AIP Publishing. 978-0-7354-5404-0/$30.00

how role-specific content can be dynamically adapted based on user interactions and use cases, leading to a more efficient, user-centric approach to content management across multiple systems.

The paper is organized as follows: It starts with the conceptual background, which is important to set the base for the implementation of a SIM and to understand the following steps of the research. The next section is about the method used in the research. Afterwards, the paper deals with the initial situation. The section is divided into the preparation part which describes the collection of the use cases and the part about the planning and adjustment of metadata and ontologies. This is followed by the implementation section. It is divided into six subsections about the process with steps of the implementation addressing requirements, how the SRC are defined and the integration of sub-ontologies in the SIM. The paper concludes with a summary of the key findings of the research, an outlook on the implementation of a CDP, and future goals.

CONCEPTUAL BACKGROUND

In this section the focus is on background knowledge that is important for implementing a SIM and SCR for managing metadata.

In the domain of TC, the utilization of component content management systems (CCMS) has become a common method for the management of topic-based content [3]. The difference between a CCMS and a Content Management System (CMS) is that a CCMS is primarily about information management. The term came about because CMSs were often understood to be web content management systems [8]. Another significant platform is the CDP, which facilitates the dynamic search for information by customers [1, 9].

A useful way of structuring information such as metadata is to incorporate PI-classification [2]. This model contains four classes that label information and improve the efficiency of information retrieval [2]. We used the methodologies described in [1, 2] as the basis for both the conceptual framework for the research covering ontology models, microDocs, and how to implement Semantic Correlation Rules SCR [1, 2].

Ontology models play a crucial role in structuring metadata and enabling efficient content management. An ontology is a structured framework for organizing and representing knowledge in a formal, machine-readable, and human-interpretable way [2]. It defines concepts, their properties, and the relationships between different entities, thereby enabling a more precise and meaningful organization of information.

A key mechanism for leveraging metadata within content management and content delivery is the use of SCR. These rules define correlations between metadata entities, which are derived from both CCMS and CDP platforms and delivered as microDocs [1]. SCR allows for dynamic content associations, ensuring that relevant information is linked and presented based on user interactions. SCR is built upon two fundamental components: InRules and OutRules [1]. InRules define the selection criteria for a primary object, such as a topic currently viewed by a user. OutRules, on the other hand, determine the selection of secondary objects that are contextually related to the primary object, such as reference materials or logically related topics [1]. By applying these rules, metadata-driven content delivery can be enhanced, improving information retrieval and user experience [1].

This structured approach, incorporating ontology models and SCR, are a possible foundation for implementing intelligent content strategies, enhanced information accessibility and digitalization scenarios.

METHOD

The methodology for developing an ontology for SCR typically consists of key steps: exporting metadata in XML format, developing a universal transformation script (XSLT), defining universal ontology patterns for consistent modelling, applying Semantic Correlation Rules (SCR), transforming metadata into an ontology, and merging multiple ontologies. [10]

The process begins with exporting metadata from various source systems into a standardized format such as XML (Figure 1). This ensures that metadata can be efficiently transformed and utilized in subsequent stages. To achieve this, a standardized XSLT stylesheet, used across different sectors, is applied to transform the XML data. This transformation results in an OWL file, which serves as the basis ontological representation of the domain metadata.[11]

The next step involves the selection of model patterns. An individual ontology pattern is required for each set of domain metadata. By analyzing previously extracted metadata, the components of the ontology can be determined, and their relationships defined accordingly. For structuring the ontology, either a class-based or an instance-based approach can be considered. For this ontology, we used the class-based approach. [11]

18 June 2026 16:39:25

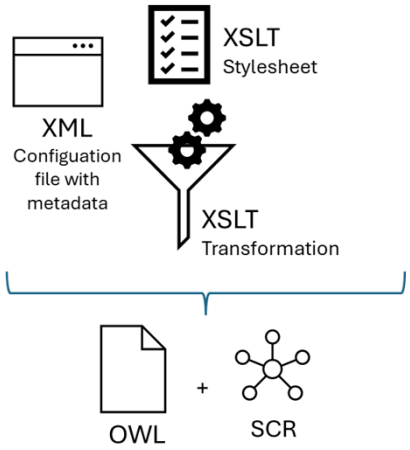

FIGURE 1. Example of the Process of developing a SIM

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. 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 resulting sub-ontologies as well as the basic OWL definition of SCR were merged in the cloud-based database environment using the GraphDB software [10, 11].

After the basic ontology is created, use cases can be implemented in a graph database, using SPARQL to query specific information from the ontology. For the implementation of SCR, a standard modelling pattern is used, where the InRules are defined first, requiring a unique name as well as the metadata IDs of the starting point and the correlated OutRules. Similarly, the OutRules are inserted into the graph database, selecting the metadata IDs of the targeted endpoint.[11]

INITIAL SITUATION

Due to the type of product, adjustments are required to ensure the ontology can be implemented in an agile software environment. The required 'features' of a software product are realized through development sprints. They are collected in the product backlog. Since one sprint is about 1-4 weeks, features that need to be developed are divided into smaller units that are easier to implement within the duration of a 'sprint'. These backlog items are called 'user stories' and each contain requirements for the software product. Then, the development team defines the tasks that are required to implement those items in a sprint backlog for the next sprint [12]. A common tool for managing feature-based development processes and sprint planning is Microsoft's Azure DevOps for the developing environment. [13]

Preparation and Prerequisites: Collecting Use Cases for Software Products

In preparation, the implementing company needs to first collect the use cases required for the content delivery. The user stories from feature-based development are useful for content planning because they provide an overview of all required topics for the product documentation: each user story represents one required content module with metadata in the CCMS. Further, this collection of the modules, resp. user stories also represent the required use cases for the content delivery.

As the features are divided into user stories, they can be grouped according to the feature to which they belong. Since each user story is represented by a module, the modules that belong to one feature can be seen as a set of content that is collectively considered to be a single entity. So, the required context information for a module is already given: each set of user stories equals a set of microDocs. Therefore, the modules should be connected to the other modules

18 June 2026 16:39:25

within a feature group. The user stories and their related content serve as a basis for the use cases planned for its content delivery. Each user story and its related module are represented by a use case.

Planning and Adjustment of Metadata and Ontologies

Due to the differences to other sectors and the special features of software products, the OWL file generated by standard XSLT transformation required adjustments and additions for the SIM to be implementable for software companies.

Role Management: The software product can be used by different user groups who perform different tasks in the system and therefore have differentiated access to features depending on their role. To only deliver the content relevant to the user and his activities, special consideration had to be given to the role management for the product information and needed to be added to the use cases planning and implementation as well.

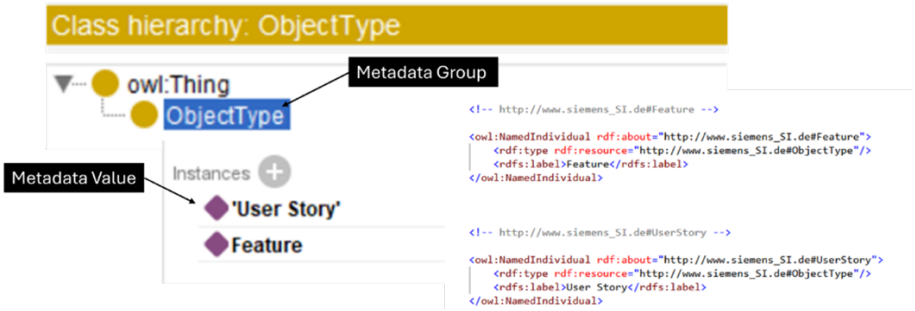

FIGURE 2. Example of a Sub-ontology for Object Types

Differentiating between object types: As described earlier, the modules align with the user stories from the development process. Therefore, this differentiation between user stories and features is also considered for the content side by differentiating between the two object types “user story” and “feature”. This differentiation has strategic benefits for increasing the use and provision of the product information within the system landscape of a company, by making the content available to developers within their familiar development environment. On the one side, technical writers can be more involved in the development process, on the other side, their content is more transparent for the developers. This can have a positive impact on the collaboration between developers and technical writers, making their work more transparent and simplifying communication.

Considering the system environment: The implementation of the object types enables the use of product information within multiple systems. Therefore, it must be possible to distinguish between the systems where content has been accessed. In our case, this could cover Azure DevOps as a development platform, the software environment itself and CDP applications. The same systematics can be used to determine for which system the information should be used in the delivery process.

Managing multiple product types: If there are multiple different product types, additional extrinsic metadata is required to differentiate the validity and belonging of the content for the different product variants described by variant parameters. In case of this research project, no additional variant parameter, resp. extrinsic metadata was not required.

IMPLEMENTATIONS

Implementing Additional Requirements

To cover the software specific requirements, to manage different roles, object types and multiple systems, firstly, new metadata groups and suitable metadata values are defined. Secondly, additional sub-ontologies are built specifically for those instances. In the modelling tool, each metadata group becomes a class and the according, pre-defined values become instances in the sub-ontologies.

For better maintainability of the data and more efficiency when generating and exchanging data, it turned out to be most effective to build separate ontologies for the metadata sub-ontologies which are then uploaded separately to the

18 June 2026 16:39:25

graph database. This makes the data easier to exchange and check if errors occur, compared to when all data is collected in one single OWL file.

Use Case Planning

After the required metadata is added, preliminary considerations can be incorporated into the concrete use case planning. To accelerate the implementation process, it is recommended to first collect and discuss the required use cases. For planning the use cases, additional tools can be helpful for the visualization and understanding of the processes and requirements. For agile environments tools like Conceptboard can be used for the preparation to first create a use case diagram. (Figure 3)

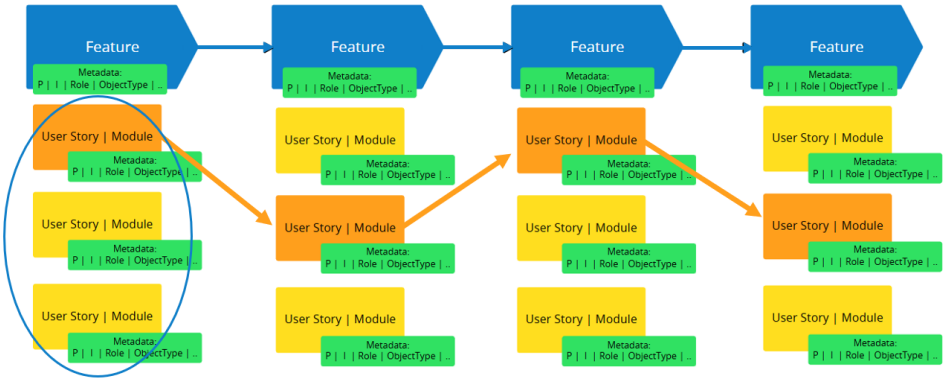

FIGURE 3. Visualization of an Excerpt of a Software Model covering Features and UserStories for planning Semantic Correlations

This diagram can be used as a preparation for developing use cases of the semantic information model and the definition of the correlation rules. It visualizes the user stories and their according modules, as well as the features they belong to and the according metadata.

Each yellow shape represents a user story, a module in the CCMS and a use case of the content delivery and can be identified by their metadata according to the PI Classification [3] and additionally defined metadata groups and values. Different users or parts of the process can be differentiated visually by colour coding.

Defining and Adjusting SCRs

For each use case of the content delivery, the SCRs need to be defined by specifying which identifying data needs to be selected for each InRule and OutRule. For each InRule addressing a content module, the required metadata values are selected for each metadata group. The additionally defined metadata which considers role, object type or the system environment are also selected by the InRule. The defined use cases can be implemented by defining semantic correlation rules manually in a modelling tool or by coding the rules with SPARQL insert statements, as shown in figure 4 and 5.

```
#OutRules
scr:In_Ceating-Station scr:hasCorrelation scr:Out_Deleting-Station.
scr:In_Ceating-Station scr:hasCorrelation scr:Out_Modifying-Station.
scr:In_Ceating-Station scr:hasCorrelation scr:Out_Viewing-Station.
context-orientated

scr:In_Ceating-Station scr:hasCorrelation scr:Out_Adding-Device.
process-orientated (next step)
```

FIGURE 4. Example of the Definition of an OutRule

18 June 2026 16:39:25

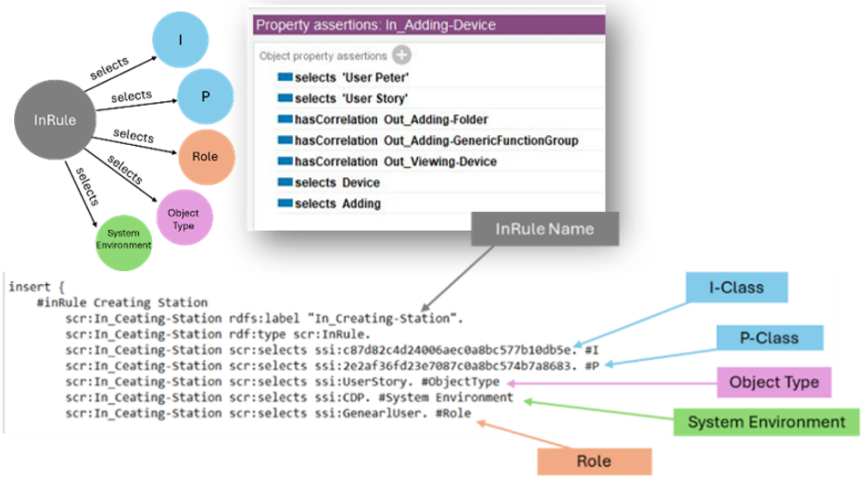

FIGURE 5. Example of the Definition of an InRule

For the definition of SCRs two different types of InRules and OutRules were defined using a two-stage principle: One type focuses on user stories and the other type focuses on feature-based connection to directly and separately address user stories and Features with the InRules. This was possible because of the newly defined metadata group "objecttype".

For the definition of OutRules different approaches were followed for the selection of correlated content, because it can include two different types of MicroDocs for the product information following two different approaches for the selection of the MicroDocs:

"Context-orientated MicroDocs" to select all modules that are grouped by a feature because they are directly related to the starting point in terms of content and provide the context information for a module.

"Process-orientated MicroDocs" select modules that are part of a specific process. In this case the modules have a predefined chronological order that aligns with the order of the customer's task in the system along the customer journey and includes all user stories that belong to a specifically defined process of tasks.

This influences how granularly differentiated the content is delivered. In the CDP the module next in the customer's journey can selectively be delivered as a "next step", independently from the feature it belongs to. So, the customer can be guided according to the tasks he performs in the system and the required product information can be delivered adapted to this.

Definition of Use Cases

The use cases can be implemented in the use case table template, to define their related MicroDocs for their implementation in the semantic model. For each use case every selected metadata value that are required to specifically identify the content are defined for the InRules and OutRules.

General Information					Target point			
Use case ID	Use case name	List of metadata start	InRule name	List of selects InRule	List of metadata target	OutRule name	List of selects OutRule
1	Create Station: Creating station	I: Creating; P:Station, O:UserStory	In_Creating-Station	I: Creating; P:Station	I: Deleting; P:Station; O:UserStory	Out_Deleting-Station	I: Deleting; P:Station; O: UserStory
1	Create Station: Creating station	I: Creating; P:Station, O:UserStory	In_Creating-Station	I: Creating; P:Station	I: Modifying; P: Station; O:UserStory	Out_Modifying-Station	I: Modifying; P: Station; O:UserStory
1	Create Station: Creating station	I: Creating; P:Station, O:UserStory	In_Creating-Station	I: Creating; P:Station	I: Adding; P: Device; O: UserStory; R: User_Peter	Out_Adding-Device	I: Adding; P: Device; O: UserStory; R: User_Peter

FIGURE 6. Example of Use Cases defined in the Template

18 June 2026 16:39:25

Integration of Supplementary Sub-Ontologies into the SIM

As a result, four different sub-ontologies were created in the modelling tool alongside the generated OWL file. These sub-ontologies can then be integrated into the existing OWL file by importing them into the modelling tool or uploading them directly into the graph database. The sub-ontologies can be merged into a single, comprehensive ontology to streamline further processing. This merging process can be managed and executed using a XSLT script.

Testing the API and SCR-Deduced Linking

Application Programming Interfaces (APIs) have been created to ensure a dynamic system and to simplify the search of metadata stored on different platforms. To ensure that the APIs work the platform Postman was used [14]. It simulates a person searching on a platform (for example CCMS, CDP) for specific information.



FIGURE 7. Cutout of the result of API testing with Postman while searching for the SCR use case 'create station'

Figure 7 shows an example of the search of the metadata defined for the SCR, in this case the search of the InRule 'Creating-Station'. The screenshot shows that an OutRule ('Out_Modyfying-Station') was found with the linking to the connected metadata as well as the assigned PI-classification.

KEY FINDINGS

The product information from feature-based development processes can be used as a basis for the planning of content delivery for software product information. If user stories are represented in the CCMS by modules with metadata, each group of user stories belonging to a feature, equals a set of microDocs and provides the required context for the delivery.

The standardized XSLT Stylesheet serves as a common starting point for companies across different sectors and can also be used for software industry domains. However, this requires further adjustments within the software and software development environment for calling the SCR-API. In addition to the generated, standardized OWL file, the use of multiple sub-ontologies offered flexibility to implement these required adjustments for software industries, like role management and feature-based developing processes using individual sub-ontologies for every aspect.

The integration of additional metadata and supplementary ontologies and the definition of SCRs following different approaches (context-orientated and process-orientated) provides flexibility for the content delivery of use case specific content and enables the differentiation of content depending on different accessing roles and calling systems. Additionally, the definitions of SCRs consider feature-based content correlations and specific processes along the customer's journey, providing new opportunities and further guidance for the customer in the content delivery.

The developed SIM and the SCR contained therein, serves as an approach that has strategic benefits for the provision of software product information and connecting information within the system environment. It increases the use, range of product information and flexibility for content delivery, to differentiate the validity of content for different systems within a company, and to make the content available to different users and roles within different systems. This further

18 June 2026 16:39:25

enhances the use of one SIM with a company's system environment to increase helpfulness, acceptance, and usability of software product information by enhancing collaboration, simplifying communication and gaining transparency.

CONTENT DELIVERY

A CDP provides further features and options for the access of product information. Key functions are facetted search, full text search and navigation based on the content metadata. [2]

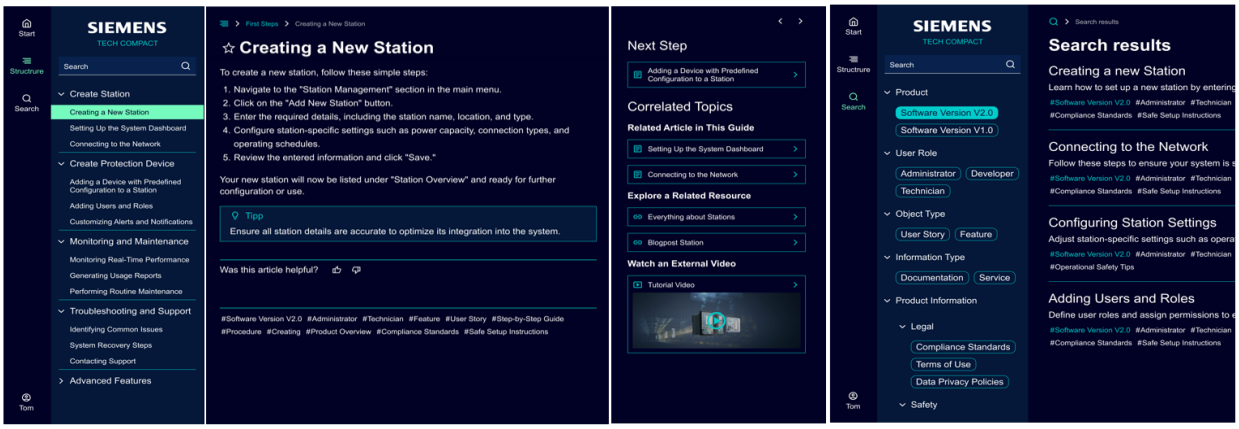

FIGURE 8. Mock-Up: Prototype of a CDP demonstrating Options and Features for accessing the Product Information
(! Disclaimer: The CDP and Software mentioned in this context is a mock-up prototype used solely for demonstration purposes. It does not represent a real system or contain actual data, but the concept is designed analog to the real environment.)

To demonstrate this, a prototype has been developed that provides different access options (Figure 8). It consists of an Online Help and an additional entry with faceted Search and Navigation. For the delivery of suitable context information, the features include advanced search, role-based access, and contextual linking to related topics, process steps, and learning materials, as well as interactive features such as feedback components, and multimedia integration. Therefore, the content can be provided more user-friendly and delivered differentiated by user's role and tasks considering their rights and requirements.

Furthermore, the content can be accessed from different systems within the system environment. The same content within its context can be delivered to different users inside and outside the company and can be accessed from the software product directly (Figure 9) by the customer, or from the development environment (for example Microsoft Azure DevOps) by the developer (Figure 10).

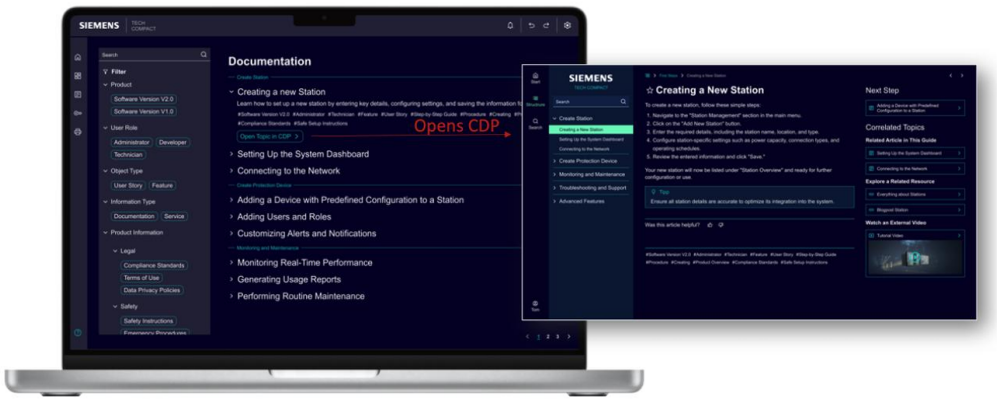

FIGURE 9. Access of Product Information in CDP via corresponding Interface of the Software Product by the user

18 June 2026 16:39:25

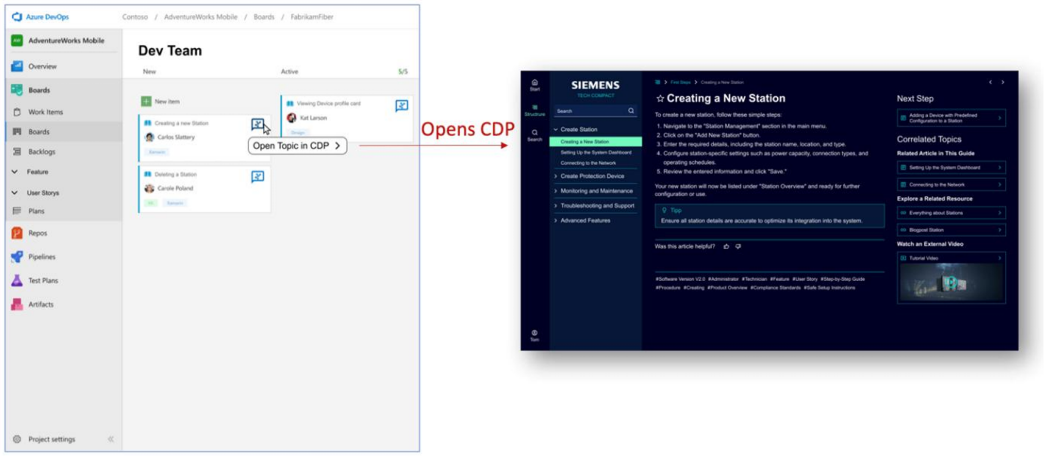

FIGURE 10. Mock-Up: Access of product information in CDP via corresponding interface of the development environment

OUTLOOK

Future goals for the delivery of product information are to further improve the new delivery solution for software products by offering CDP (Figure 8) and SmartHelp within the software product (Figure 9). Therefore, the goal is to provide information from multiple sources, make it accessible for multiple systems (for customers and employees of different roles) to increase user guidance and support through customer-specific, context-specific, target group-orientated content.

In many companies, it is a task of digitalization approach to establish a central (meta)data governance for aligning and interconnecting the various data sources. This can technically be done by using for example a unified semantic information model that enables communication between involved systems within the system landscape as described in the paper.

ACKNOWLEDGEMENT

We would like to acknowledge the technical support provided by Maurice Daum who developed the SCR API and the Proricon company for providing the cloud environment (Imaya). In addition, we would like to acknowledge the cooperation with Carina Schade and Timo Hantschmann from Siemens, who provided data and answers for our research. Furthermore, we would like to thank the International Office, the study program KMM, the General Students' Committee, and the Student Council of the Faculty of Information Management and Media at the Hochschule Karlsruhe - University of Applied Sciences for their financial support.

REFERENCES

[1]. W. Ziegler, *Semantic Correlation Rules as a Logic Layer between Content Management and Content Delivery* (ETLTC International Conference on Information and Communications Technology, Aizuwakamatsu, Japan, 2021), pp. 1–3, https://doi.org/10.1051/shsconf/202110202007

[2]. W. Ziegler, *Extending Intelligent Content Delivery in Technical Communication by Semantics: Microdocuments and Content Services* (ETLTC International Conference on Information and Communications Technology, Aizuwakamatsu, Japan, 2020), pp. 1–3, https://doi.org/10.1051/shsconf/20207703009

[3]. A.-M. Kruse, C. Wiedewilt, and L. Tempelfeld, *Implementing Semantic Correlation Rules in the Domain of Energy Technology for Transportation Systems* (ETLTC International Conference on Information and Communications Technology, Aizuwakamatsu, Japan, 2024), https://doi.org/10.1051/shsconf/202419402005

[4]. A. Komus and M. Kuburg, *Status Quo (Scaled) Agile 2019/20 – 4th International Survey: Benefits and Challenges of (Scaled) Agile Approaches* (University of Applied Sciences Koblenz, Koblenz, 2020), https://www.hs-koblenz.de/index.php?id=7169

18 June 2026 16:39:25

[5]. K. Beck, M. Beedle, A. van Bennekum *et al.*, *Principles Behind the Agile Manifesto* (2001), http://agilemanifesto.org/principles.html
[6]. S. Sawalha and H. AbdelNabi, *Agile Software Development: Methodologies and Trends* (2020), https://doi.org/10.3991/ijim.v14i11.13269
[7]. M. Gaborov, D. Karuovic, D. Radosav, L. Berek, and E. Bartha, *The Difference Between Scrum and Other Agile and Traditional Methodologies* (8th International Joint Conference on Environmental and Light Industry Technologies, Budapest, Hungary, 2021), pp. 29–38
[8]. P. Drewer and W. Ziegler, *Technische Dokumentation* (Vogel Buchverlag, Würzburg, 2014), p. 290
[9]. W. Ziegler, *New Roles and Competencies in Technical Communication Induced by Semantics and Analytics* (ETLTC International Conference on Information and Communications Technology, Aizuwakamatsu, Japan, 2022), p. 2, https://doi.org/10.1051/shsconf/202213902004
[10]. S. Sommer, M. Teichmann, S. Kopf, L. Bratic, R. Erb, and W. Ziegler, *Standardized Generation of Semantic Domain Models from Content Management Metadata* (ETLTC International Conference on Information and Communications Technology, Aizuwakamatsu, Japan, 2025)
[11]. L. Tempelfeld, C. Wiedewilt, and A.-M. Kruse, *Transforming Domain Data into Semantic Models for Content Delivery Applications* (ETLTC International Conference on Information and Communications Technology, Aizuwakamatsu, Japan, 2024), pp. 3–5, https://doi.org/10.1051/shsconf/202419402004
[12]. T. Stober and U. Hansmann, *Agile Software Development: Best Practices for Large Software Development Projects* (Springer, Berlin Heidelberg, 2010), pp. 42–46, p. 127
[13]. Microsoft, *Azure Boards* (2025), https://azure.microsoft.com/en-us/products/devops/boards/
[14]. Postman, *What is Postman?* (2025), https://www.postman.com/product/what-is-postman/

18 June 2026 16:39:25