

Model-based documentation

Feroz Farazi, Craig Chapman, Raju Pathmeswaran and William Byrne

[†] *Knowledge Based Engineering Lab, Birmingham City University, Millennium Point
Birmingham, B4 7XG, United Kingdom
{mohammad.farazi, craig.chapman, path.raju, william.byrne}@bcu.ac.uk*

Received Day Month Year; Revised Day Month Year; Accepted Day Month Year; Published Day Month Year

Knowledge acquisition is becoming an integral part of the manufacturing industries, which rely on domain experts in various phases of product life cycle including design, analysis, manufacturing, operation and maintenance. It has the potential to enable knowledge reuse, however, poorly managed knowledge can cause information loss and inefficiency. If technical documentation is managed well in the manufacturing industries, intended piece of knowledge can easily be located, used and reused for purpose and as a result, the corresponding industry can be benefited. Some examples of technical documentation are design specification, operating manual and maintenance manual. Model-based documentation (MBD) is a documentation approach that uses model to provide structure to the data of the documents. MBD can be thought of as a way to better organize knowledge thereby knowledge identification and retrieval become easier, faster and efficient. In this paper, we propose model-based documentation and its extension as a potential solution to overcome the issues involved in the typical technical documentation approaches.

Keywords: Semantic technology, ontology, engineering design knowledge, knowledge modeling, model-based documentation.

1. The Problem

Manufacturing industries all over the world are going through a difficult time mainly because of the high production cost. The production time influences the production cost and usually the former is proportional to the latter. Therefore, there is a pressing need for reducing the production time in order to survive in the highly competitive global market. In this ever-changing world, products also evolve continuously. Complex products which keep evolving and which require deep knowledge and detailed instruction to operate and maintain are necessarily released with the operating and maintenance manual. An aero-engine is an example of a complex product. The production time reasonably includes the product manual development time. Writing manuals is an onerous and time consuming job and performing this job after the manufacturing will delay the product release that might make manufacturing companies losing customers and profit. To reduce the total duration of the product release, documentation and manufacturing can proceed concurrently [1]. But the efforts required and the total man-months needed for the documentation remain unchanged.

Problem 1: Therefore the challenge is how to minimize the efforts and man-months in authoring and producing documentation?

Design specification is a documentation used in the design phase of a product. It contains engineering design choices, rules and rationales. The choices, which were made, the rules, which were defined by the expert engineers in producing an earlier version of the product and the rationales behind both the choices and rules, are of the utmost importance for the engineers who have replaced them. The rules included in the design specification are identifiable by humans, however reading hundreds of or thousands of pages for detecting the rules lowers the efficiency of the design engineers. Tools developed using NLP (Natural Language Processing) techniques would be helpful in automating for example the rule detection process, unfortunately, these tools still underperform and sometimes the accuracy that they demonstrate is unacceptable. Automatic identification is crucial as it creates the environment for programmatic import of the rules to the engineering design automation tools such as Technosoft's AML^{*}, Siemens NX[†] and CATIA[‡].

Problem 2: The challenge, therefore, to overcome here is how to remove the barrier that hinders automatic identification of the rules?

These rules are often a set of statements written in natural language. For making them actionable, in the current practice, the engineering workforce codifies them manually into the native platform of the design automation tools.

Problem 3: The challenge to address here is how to automatically integrate the rules retrieved from the technical documentation with the design automations tools?

^{*} <http://www.technosoft.com/application-software/adaptive-modeling-language/>

[†] https://www.plm.automation.siemens.com/en_gb/products/nx/

[‡] <http://www.3ds.com/products-services/catia/>

The 2012 workshop on technical documentation challenges in aviation maintenance discussed the problems that have been identified and that lead to errors in maintenance, rework and maintenance delays. The participants of the workshop were mainly from the aero-engine producing companies (e.g., GE Aviation and Pratt & Whitney), airlines companies (e.g., US Airways and Delta) and airframe industries (e.g., Airbus and Boeing). In the workshop, they also discussed that while producing maintenance documents relying on low cost resources such as Microsoft Word for authoring and Adobe pdf for delivery prohibits to be cost-effective in the long term as the produced content lacks structure and makes the information management and search difficult [2].

Problem 4: The challenge is how to provide structure to the information codified in the technical documentation in order to make the management and search activities easier?

2. State-of-the-art

This section is divided into the following two parts: documentation approaches and documentation specifications. In the former part, a brief description of a number of document generation techniques with a focus on knowledge capturing and knowledge modeling is provided (Section 2.1) and in the latter part, DITA, DocBook and S1000D are briefly discussed (Section 2.2).

2.1. Documentation Approaches

2.1.1. Active Document System

Active Document System (ADS) [3] is an approach devised for capturing, indexing and managing engineering design knowledge (e.g., engineering drawings, prototype evaluations, analysis reports, engineering formulae and informal notes on specific design problems) in a way that makes search, identification and extraction of such knowledge easier. ADS is developed to capture the design knowledge that usually comes in unstructured format (e.g., free text) and that originates from the formal and informal design descriptions. It follows a technique called part or function relation for adding structure to the unstructured design knowledge of a product. Through this relational perspective (or aspect), structural design knowledge as well as part or function relational knowledge can be swiftly browsed, navigated and located. It allows the creation, maintenance and exploitation of customized facets on-demand to cover other aspects of design knowledge such as cost analysis.

ADS is applied to the creation of a knowledge management system for managing information about refrigerator design process. ADS engineering design knowledge management infrastructure consists of three different but correlated modules which are knowledge acquisition module, knowledge structuring module and knowledge retrieval module.

The knowledge acquisition module has provided the means for capturing both informal and formal design knowledge. An individual design engineer or a group of design engineers can write an informal design memo while product design activity is running in an engineering work environment. To capture this informal design knowledge, ADS has deployed an authoring web application with the necessary editing tools. Capturing formal knowledge including engineering drawings and mathematical formulae is on the other hand done through the use of templates, which are specific to design task and process users have at hand.

When the unstructured design knowledge flows from the acquisition module to the structuring module, it is passed through the knowledge perspective (KP) filter to link the slices of design knowledge to produce structured knowledge that is finally represented in the ADS knowledge base. A KP can be defined by following either top-down or bottom-up approach. In the top-down approach, a KP is defined first and then the ADS knowledge base is processed to produce relevant link between the pieces of relevant knowledge. In the bottom-up approach, the knowledge base is processed first to recognize the potential perspectives according to the relations between the pieces of knowledge.

The knowledge retrieval module works with the help of KPs created in the knowledge-structuring module on top of the design knowledge available in the ADS knowledge base. As like as the authoring application of the knowledge acquisition model, the KPs are also deployed on the Web to allow users browsing the design knowledge. By choosing the right KP, a user can browse to find knowledge about a specific design.

2.1.2. High Value Document Generation

Customized high-value document generation [4] requires synchronization of the inputs given by different experts to the process planning in the engineering product development. The goal of this work is to generate a value added document, e.g., experts supported semi-automatic generation of a CAM (Computer-Aided Manufacturing) document from the original CAD

(Computer-Aided Design) document. The execution of the following macro-steps -- project infrastructure definition, project architecture construction and document generation -- can assist to achieve this goal.

The infrastructure definition includes the identification of standard concepts about processes, products, resources and external effects needed for the specification of the functional, behavioral and structural characteristics of the engineered objects. The architecture construction includes the analysis of the concepts to understand and build the relationships between them. At this step, duplicate and unnecessary concepts and instances are removed. Document generation relies on the refined and matured architecture of the project and as a result, becomes more optimized.

2.1.3. *Concurrent Execution of Manufacturing and Documentation*

3DVIA Composer [1] is an authoring tool that can render 3D CAD models created in the engineering design phase and that can generate product documentation using these models for downstream phases such as manufacturing, assembly, operating and maintenance. As this composer can be integrated with the 3D CAD tool design environment, it offers the crucial advantage of executing the design and the documentation processes concurrently. Other advantages are automatic updating of the documentation when some changes are made in the CAD models, the facility to create animated visuals from the models to clearly show how to assemble and disassemble the product parts and capability to produce documentation in MS Word, MS PowerPoint and HTML formats.

2.1.4. *Model-Based Document and Report Generation*

Model-based document and report generation [5] is an approach relying on MBSE (Model-Based Systems Engineering) platform to produce documents from product or system models. It is one of the document generation initiatives taken at JPL (Jet Propulsion Laboratory). SysML (Systems Modeling Language) provides a firm underpinning for all of these initiatives.

This document generation approach leverages SysML's concepts Viewpoint and View to meet the requirements of different stakeholders, which might look at a product or a system from their own perspectives. With the help of these concepts, multiple views of the same (product or system) model can be created. A viewpoint captures the point of view of a stakeholder about a model using certain properties (e.g., purpose and stakeholder), constraints (e.g., view format and modeling language) and method. A view keeps it aligned with the constraints defined in the viewpoint to generate a view covering an aspect of the model. The method is responsible for running the process to create the view from the model. A viewpoint can consist of a number of subordinate viewpoints. For example, a document viewpoint might contain section and subsection viewpoints.

2.1.5. *Model-Based Virtual Document Generation*

DME (Device Modeling Environment) [6] provides a modeling and simulation environment for engineered systems and supports the generation of virtual documents, which are answers to the questions about various aspects of such systems. The document generation relies on the inference performed over the mathematical and symbolic model of the systems.

DME made available a Web-based model library so that users can build their customized systems online combining various components. It allows users to predict the behavior of the systems being built by performing simulations. Users can pose questions about the physical structure and dynamic behavior of the systems and the answers are published in text and graphics in the form of explanations.

2.1.6. *Model-Based Requirement Generation*

Model-based textual requirement generation [7] is an automatic documentation approach based on SysML structure and behavior diagrams. In the requirement generation, information from SysML's constructs such as blocks, states and activities are extracted and then represented in requirement constructs such as allocations, control flows and object flows.

2.1.7. *Rule-Centric Document Generation*

Rule-centric object oriented approach [8] uses data and document template as the basic components for generating technical document. In this approach, the data component consists of a set of data objects. Similarly, the document template component is built with a set of document template objects. The data component and the document component are independent of each other. There are objects, which have many-to-many relations with the document template objects. The data objects are connected to the document template objects via a mapping. The objects have properties and methods. The rule that is defined

by the user selects the document template object, which contains input parameters that help decide the right object and method and the corresponding data objects to populate the document template object.

2.2. Documentation Specifications

2.2.1. DITA

DITA (Darwin Information Type Architecture) [9] is a documentation architecture, based on XML, created to provide an end-to-end support, from semi-automatic authoring all the way through to automatic delivery of technical documentation. It was developed in IBM[§] and is currently managed by OASIS^{**}. It follows a set of principles for creating documentation in a specific way that can make the retrieval of information easier and that can meet the needs of product consumers. The principles are topic orientation, information typing, specialization and process inheritance. Topic orientation is about creating the topic as the smallest possible unit of information that can be reused and that can cover a subject or answer a question. Information typing deals with the creation of information of a specific type such as concept, task and reference topic. Specialization covers creating more specific information type depending upon the need. Process inheritance focuses on propagating processes of more generic information type to the more specific information type.

2.2.2. DocBook

DocBook [10] is a documentation system developed to facilitate the production of computer hardware and software related publications such as books and articles. Currently, it can be used for publishing almost everything that requires the structure of a set of books, chapters or papers. DocBook started as a collaborative project between partners HaL Computer Systems and O'Reilly. Since 1998, OASIS is responsible for its subsequent development and maintenance. XML is used for representing DocBook content. DocBook DTDs and schema languages such as XML Schema are employed to provide structure to the resulting documentation.

2.2.3. S1000D

S1000D [11] is a specification originally deigned for aerospace and defense industry for the preparation and production of technical documentation. Its scope has subsequently been expanded to the domains of land and water vehicles and various equipment of both military and civilian markets. After S1000D's development by the AeroSpace and Defence Industries Association of Europe (ASD), a steering committee, formed with the ASD, the Air Industries Association (AIA) and the Air Transportation Association (ATA) of the USA, together with the industry and defense partners from the countries, which have user communities, has been maintaining the specification. At its core, there is the principle of information reuse that is facilitated by Data Modules. The Data Modules in version 4.0 of S1000D are represented in XML. S1000D is suitable for publishing in both printed and electronic form. The S1000D document generation architecture includes a data storage component, which is called CSDB (Common Source Database).

2.3. Documentation Summary

In this subsection, we have summarized the documentation approaches and specifications from the design, modeling and representation technology perspectives. In Table 1, DITA, DocBook and S1000D are documentation specifications and the rest are documentation approaches. CAD is a design technology used both in Concurrent Execution of Manufacturing and Documentation and Model-Based Virtual Document Generation. SysML is a modeling technology (language) developed for systems engineering applications applied to Model-Based Document and Report Generation and Model-Based Requirement Generation. All three documentation specifications included XML as the data representation technology (language). XML is also included in the infrastructure of Model-Based Document and Report Generation and Rule-Centric Document Generation. Database and Knowledge Base are employed for representing (storing) data and knowledge, respectively. Ontology is used as a knowledge modeling and representation technology.

[§] <https://www.ibm.com>

^{**} <https://www.oasis-open.org>

Table 1. Knowledge representation technologies used by the documentation approaches and standards

Documentation Approach/Specification	Design/Modeling/Representation Technology	Semantic Technology
Active document system	Knowledge Base	Knowledge Base
High value document generation	Knowledge Base, Ontology	Knowledge Base, Ontology
Concurrent execution of manufacturing and documentation	CAD	Not used
Model-based document and report generation	SysML, XML	Not used
Model-based virtual document generation	CAD	Not used
Model-based requirement generation	SysML	Not used
Rule-centric document generation	XML, Database	Not used
DITA	XML	Not used
DocBook	XML	Not used
S1000D	XML, Database	Not used

As shown in Table 1, among the modeling and representation technologies, Knowledge Base and Ontology are semantic technologies. In Active Document System and High Value Document Generation documentation approaches, semantic technologies are applied.

2.4. Solution Approach

Research and development attempts have been taken both in the industry and academia to cover various aspects of technical documentation including ease of identification, machine readability and publishing format of information stored in the document databases of organizations.

Model-based documentation (MBD) is a documentation approach that uses models of different kinds such as ontological model and template-based model to provide structure to the data of the documents. It can be thought of as a way to better organize knowledge thereby knowledge identification and retrieval become easier, faster and efficient. One size does not fit all. Similarly, one model does not capture all kinds of data. In MBD, therefore, the model needs to be created or updated according to the data.

For achieving interoperability, we follow the best practices of the engineering community. The International Council on Systems Engineering (INCOSE) fosters MBSE approach along with SysML for specification, design, analysis, verification and validation of engineering systems. The aerospace and defense industries association recommended the use of S1000D for technical documentation. The documentation approaches other than the MBSE ones and the documentation specification other than S1000D out of the ones described in Section 2, to the best of our knowledge, lack connection with the recommendations of the engineering community. For interoperability and to avoid reinventing the wheel, our solution approaches include the latest documentation developments that include MBSE and S1000D. MBSE and S1000D-based MBD is described in the following subsections with a potential direction on how the problems identified in this paper can be addressed.

2.5. MBSE-based MBD

MBSE uses models to capture the system requirements and enables system design and documentation process [16]. In MBSE, the system description can be given in a data model, which then can be used to generate documentation of different granularity and needs [17]. As the integration of MBD with MBSE ensures alongside existence of both documentation and system models, data consistency can be maintained, document production and reproduction can be automated [18, 19] and in turn documentation as well as production time and cost can be reduced.

2.6. S1000D-based MBD

As S1000D is a specification for technical documentation and offers advantages including data management, it is currently in use in several organizations for the production of operating and maintenance manual [12]. In fact, many companies and organizations have converted their documentation into S1000D [12]. The key to S1000D's data management, search and retrieval is the CSDB database. S1000D allows data representation in various publishing format like HTML and PDF. By creating business rules (BRs), the intended piece of knowledge from the documents can automatically be identified. As a result, knowledge reuse can be enabled and manual efforts and man-months in generating documentation can be reduced.

2.7. MBSE vs. S1000D

With the capabilities described in Sections 3.1 and 3.2, MBSE and S1000D-based MBD can partially address Problems 1, 2 and 4. With respect to Problem 1, both the MBSE implementation using SysML and the S1000D implementation lack the support for creating engineering (product) design models with CAD design tools. Without the inclusion of engineering design models, documentation often remains incomprehensible and incomplete. To overcome this issue, a CAD design tool can be integrated with these implementations. In connection with Problems 2 and 4, search on the content of these implementations is limited to the keyword matching only without taking into account the meaning of the data. In addition to this, inference capability is missing in these implementations. To enable meaning and inference dependent search, domain ontology can be used [20].

As S1000D by design allows the use of Simplified Technical English (or Controlled Natural Language) [15], it is capable of overcoming the issue pointed out in Problem 3. According to the problem solving capability of both the implementations, S1000D can offer more benefits. Our solution proposal for various types of documentation described in the following subsections, therefore, includes S1000D.

2.7.1. Creating Requirements Document Using S1000D

Requirements of different stakeholders such as client, regulatory authorities and corporate standards are compiled in the product requirements document [13]. S1000D has been expanding its scope to produce various types of technical documentation. With the inclusion of business rules, it is now possible not only to create any types of documentation (e.g., requirements document) but also to identify and retrieve any types of information (e.g., weight and length). As shown in Figure 1, some data modules are created in S1000D to represent physical requirements such as weight and length of an aero-engine. In the creation of the data modules, a business rule is used that includes component name (e.g., engine), component id, attribute name (e.g., weight) and attribute code value for the codification and automatic identification and retrieval of physical requirements. By using XML as the means of representation and unique id both for the component and attribute, the requirements are made machine-readable.

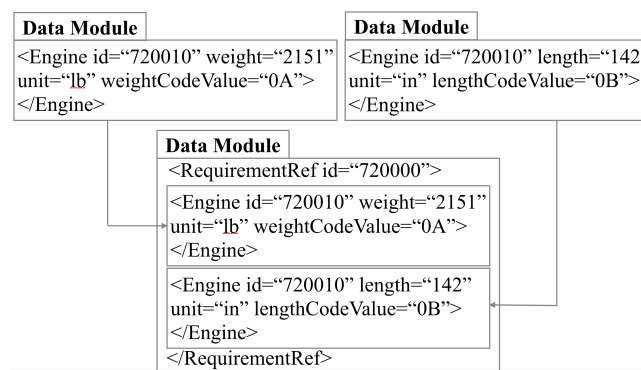


Figure 1: Requirements codified in S1000D.

2.7.2. Creating Maintenance Manual Using S1000D and 3DVIA Composer

A maintenance manual depicts the procedures for fixing issues that might arise with a product in the course of use. Integration of the 3DVIA Composer authoring tool with the S1000D documentation environment will enable the creation of animated visuals that can be put together with textual instructions to produce better understandable time saving maintenance manual.

It can be foreseen that the future will be more and more automated and the engineering tasks that are done today by humans alone or semi-automatically will be better supported by (software and physical) artifacts. Complete automation offers advantages such as getting rid of human errors. Product engineering world is heading towards complete automation. However, some areas are still behind, for example, the generation and exploitation of documentation, especially requirements document and product definition document. Improving machine readability of such documents is challenging, but has the potential to offer the crucial advantage of achieving better automation.

2.7.3. Semi-automatic Generation of Product Definition Document from Requirements Document

Requirements document is discussed with the client as many times as it necessitates eliciting a concrete description of the product, which goes to the product definition document [14]. As the product definition document contains all possible details about a product, it is also called as detailed specification. To the best of our knowledge to date the product definition document is created manually. Industries, which make domain specific products, could use domain ontology guided template for requirements elicitation as well as representation to allow semi-automatic generation of the production definition document from the product requirements document.

2.7.4. Automatic Generation of Design Model from Product Definition Document

Controlled natural languages, also supported by S1000D, offer the potential to produce unambiguous and machine-readable technical documentation [15]. It could be presumed that ontology guided semi-automatically generated product definition document could be given a structure that would be suitable for machines to convert into a controlled natural language developed for domain specific engineering design. The controlled natural language representation of a product definition document could be parsed to automatically generate the design model of the product. It can be used to identify and extract rules and also to codify them automatically in the design automation tools thanks to its unambiguous nature.

References

- ¹ Streamlining Product Documentation across the Manufacturing Enterprise with 3DVIA Composer. White Paper. http://www.solidworks.com/sw/docs/3dvia_2010_eng_final.pdf.
- ² Avers, K., Johnson, B., Banks, J. and Wenzel, B. Technical Documentation Challenges in Aviation Maintenance Workshop. *A Proceedings Report*, (2012).
- ³ S. Ha, G. Pahng, M. Chang, S. Park, H.M. Rho. Managing design knowledge: Active Document System. *Annals of the CIRP*, 48/1 (1999), pp. 89–92.
- ⁴ N. du Preez, N. Perry, A. Candlot, A. Bernard, W. Uys, L. Lou. Customised high-value document generation, *CIRP Annals*, 54 (1) (2005), pp. 123–126.
- ⁵ C. Delp, D. Lam, E. Fosse, C. Y. Lee. Model based document and report generation for systems engineering. *IEEE Aerospace Conference*, (2013).
- ⁶ T. R. Gruber, S. Vemuri and J. Rice. Model-based virtual document generation. *Int. J. Human Computer Studies*, 46, 687-706, 1997.
- ⁷ B. London, P. Miotto. Model-based requirement generation. *Aerospace Conference, 2014 IEEE*, 2014.
- ⁸ K. Rajbabu and S. Sudha. A novel rule-centric object oriented approach for document generation. *Computers in Industry*, 65 (2), 235–246, 2014.
- ⁹ M. Priestley. DITA XML: a reuse by reference architecture for technical documentation. *In Annual ACM Conference on Systems Documentation*, pages152–156. ACM Press, 2001.
- ¹⁰ N. Walsh, L. Muellner and B. Stayton. DocBook: The Definitive Guide. *O'Reilly & Associates*, 1999.
- ¹¹ Inmedius Understanding and Implementing S1000D Issue 4.0, Now and in the Future. *White Paper*.
- ¹² Business Rules in S1000D™. All You Need to Know. *Produced by CDG, A Boeing Company*.
- ¹³ R. Roy, C. Kerr, C. Makri and D. Kritsilis. Documenting Technical Specifications During The Conceptualisation Stages of Aeroengine Product Development.
- ¹⁴ G. Cabral, A. Sampaio: Automated Formal Specification Generation and Refinement from Requirement Documents. *Journal of the Brazilian Computer Society* 14(1), 87–106 (2008).
- ¹⁵ T. Kuhn. A survey and classification of controlled natural languages. *Computational Linguistics*, 40(1):121–170, March 2014.
- ¹⁶ J. S. Topper and N. C. Horner. Model-Based Systems Engineering in Support of Complex Systems Development. *Johns Hopkins Apl Technical Digest*, vol. 32, no. 1, 2013.
- ¹⁷ P. Logan, D. Harvey, and D. Spencer. Documents are an Essential Part of Model Based Systems Engineering. *INCOSE International Symposium*. Vol. 22. No. 1. 2012.
- ¹⁸ R. Karban, M. Zamparelli, B. Bauvier, G. Chiozzi. Three years of MBSE for a large scientific programme: Report from the Trenches of Telescope Modelling. *In the Proc. of 22nd Annual INCOSE International Symposium*, 2012.
- ¹⁹ R. Karban et al. Model based systems engineering for astronomical projects. *In SPIE Astronomical Telescopes + Instrumentation*, pp. 91500L-91500L., 2014.
- ²⁰ F. Farazi, C. Chapman, P. Raju, L. Melville. WordNet Powered Faceted Semantic Search With Automatic Sense Disambiguation For Bioenergy Domain. *IEEE Tenth International Conference on Semantic Computing*, pp. 112—115, 2016.