NordDesign 2020
August 11-14, 2020
Kgs. Lyngby, Denmark

A Design Specification Chain: A Concept for Improving Revision Management and Change Tracking

Christian Alexander Bertram[1], Jeppe Bredahl Rasmussen[1],
Georg Otto Mueller[1], Niels Henrik Mortensen[1]

[1]The Technical University of Denmark
chalbe@dtu.dk
jbrras@dtu.dk
gmul@dtu.dk
nhmo@dtu.dk

Abstract

Design-specifications play an important role in aligning organizational work and ensure everyone works towards the same solution. With many separate specifications and the fast revision pace of modern global business processes, there is a need to manage those revisions and changes and to make sure the information they hold is accessible for the whole organization. In this paper, a concept is introduced for a chain-like data structure that allows merging design specifications and tracking all changes. The functionality behind it is inspired from state of the art software development methods.

Keywords: concurrent engineering (CE), data driven design, engineering change, product data management (PDM)

1 Introduction

The world of engineering is increasingly global, fast-paced and complex. Businesses are attempting to work smoothly across borders and continents while designing more and more elaborate products. This development puts focus on the need to have relevant documentation readily available anywhere in the world. For many project-based businesses, this translates into an abundance of engineering design specifications done for each project and often in several revisions as the projects are detailed. The engineering design specifications exist to properly convey information and requirements between business units or stakeholders. Their purpose is to ensure that everyone agrees on the task at hand and the intended solution. They are the recipes for the solution and subsequent design tasks, guiding development towards the best solution and avoiding changes and re-work (Hvam et al., 2008; Simpson, 2010).

As product related businesses include many expert knowledge areas and function-specific work groups, design specifications are often tailored to individual aspects of the product. Process specifications are done for the process engineering, physical specifications are done for the drawers and electrical specifications are done for the electrical department to design a fitting control solution. This results in a multitude of domain-specific design specifications throughout the organization. An abundance of design specifications are done for a single product, often scattered across many different locations and in several analog and digital formats. These are all sources of information about a given product. If the revisions of the specifications are counted as well, the amount of information sources multiply.

Collecting project information is important for conducting project analyses and tracking of changes that are carried out throughout project execution. Businesses often employ a suite of management systems to cope with the need for collecting information and managing changes to that information (Bruun et al., 2015; Hameri & Nikkola, 1995; Lämmer & Theiss, 2015; Muli & Allen, 2013; Siddiqui et al., 2004). These systems are focused on project documentation and structured storage of files for specific purposes e.g. CAD files or technical drawings. The structured storage of project documentation ensures that it is readily available for project stakeholders. However, it does not ensure that project information is readily available for analyses such as portfolio management analyses, trend analyses and business intelligence inquiry, resulting in data and information that is notoriously difficult to gather (Bohm et al., 2011; Brunoe & Nielsen, 2012; Duchi et al., 2014; Foehr et al., 2015; Hvam et al., 2006; Robertson & Ulrich, 1998). These analyses often require access to project data – typically included in specifications. Product categories, sizes, types, functionalities, key performance indicators, etc., which are often defined before the actual solution is created. The primary use of this data is to determine the design the product without prioritizing the implications of the data format and structure on future data management and analyses.

There is a need to structure engineering design specification information in a system that allows proper storage and use, subsequent analyses and project continuity. To meet this need, this paper proposes a concept for merging domain-specific product specifications into a combined chain of information in order to support the structured storage of specification information and the accessibility to fundamental project data. This chain serves as the single source of specification information for the project stakeholders, while simultaneously tracking changes and supplying data for subsequent portfolio analyses. The chain also allows product development projects to re-use previous project structures and data.

2 Methodology

In order to develop a concept for an improved change management and revision tracking, a literature study is conducted to identify existing research in the areas of product design specifications and product data management (see section 3). Additionally, a case study including two case companies was carried out to determine industry requirements for the development of a specification management system (see section 4). The revealed insights are consolidated to form the requirements for the concept development and the proposed concept is presented in section 5, followed by a discussion of benefits and limitations in section 6.

3 Background

In order to describe the state of the art research in this area, the literature review has been focused on two aspects: (1) The design specification and engineering change management and (2) project- and product data management systems.

3.1 Design Specifications

Throughout the design process in an engineering company, the product is designed in an iterative process with an increasing level of detail over time. Therefore, customer requirements are translated into product specifications that provide the functional and physical details of the product (Cho et al., 2016; Darlington & Culley, 2002). Specifications can be created in a multitude of ways, formats and styles conveying everything from lists of functionalities, specific performance targets, solution outline sketches and preliminary drawings to name just a few (Halevi, 2012). In case of complex products, a collaborative and interdisciplinary design process is carried out. When developing customized products, as is the case in the Engineer-To-Order and Configure-To-Order business models, changes usually occur even in the late design and engineering stages due to design errors, customer-induced modifications or improvements (Bhuiyan et al., 2006). Given the high interconnectivity of the different product domains in complex products, a change in one domain often has knock-on effects in other domains as well. This phenomenon is also known as change propagation (Clarkson et al., 2001; Hamraz & Clarkson, 2015).

3.2 Project and Product Data Management

Past research efforts showed the need for the development of approaches to support the complex engineering process and the interaction between the responsible parties (Jarratt et al., 2005; Wasmer et al., 2011). Researchers have already proposed solutions to tackle engineering revision control with inspiration from the software domain. This is often referred to as Product Data Management.

Product Data Management is an umbrella term describing different systems for managing the information in a project or product development process (Forza & Salvador, 2006). Most of these solutions are aimed at the structured storage of documentation to ensure that involved parties can store, gather and find the latest versions of drawings, bill of materials and calculations.

A proposed concept suggests file-based revision control system using *Git* (Spinellis, 2005) to handle project documentation versions (Anzengruber & Hehenberger, 2017). The concept utilizes a distributed network of peers to enable version tracking of project documentation and mimics the functions of a traditional PDM system. However, the concept is file-based and thus relies on the applicability of existing project documents into this new system. Shortcomings of this approach include the handling of proprietary formats for CAD programs and alike.

Project data can also be stored as non-relational databases with a web interface (F. Rocha & Varela, 2013). They propose a concept where product data is stored in a document-oriented database called *CouchDB*. Document-oriented databases (Kuznetsov & Poskonin, 2014) are characterized by their main structure not being tables related to each other but rather independent collections of variables and data. It is proposed to use this data structure to collect information in new product development and to have a single location where stakeholders can collect their information. There are many types of databases using this structure or a similar approach (*NoSQL*, 2020).

Configurators and configurator systems are popular tools for implementing customization options and variance control in engineering sales activities. The main principle is the introduction of programmed logic to control combinations and instances of classified solutions and sub-solutions. This functionality has proven useful and powerful in many business contexts. However, the use of configurators is not limited to sales activities, hence there is an increasing effort to extend the functionality of configuration into subsequent project phases. (Myrodia et al., 2018a, 2018b). Configuration Lifecycle Management (CLM) systems does exactly this. It serves as a single basis for product/solution combinatory logic and the idea is that “all configuration-related business processes access CLM for configuration definitions and queries.” (www.cio-wiki.org, n.d.) ultimately being a hub for pre-defined product information.

4 Case Studies

Insights from two industry case companies are used for identifying the needs for specification revision management. The needs are used to match the developed concept with reality and to discuss improvement potentials. The two case companies are of different business size (small-medium and large) and cover different approaches to the product development of customized products (Configure-To-Order and Engineer-To-Order). Other approaches such as partly configurable or customer co-created products would form an interesting extension to this study but are not part of this research. The presented insights are based on semi-structured expert interviews with company employees and an analysis of currently used tools and applications for product design specification.

4.1 Case Company 1: Small-Medium Configure-To-Order Business

This company is a Scandinavian project based company that sources construction components and provides system deliveries such as installations. The company retrofits construction parts on existing buildings by following a project-to-project approach resulting in similar but never identical projects. The company uses a product configuration for sales of the initial expected design. However, the delivered product is rarely exactly the same as what was sold initially due to changes throughout the product development process.

The design specification constitutes four phases: (1) sales phase, (2) initial design specification, (3) detailed design specification and (4) installation on site. The sales phase generally consists of an initial design specifying the expected performance of the product, while the initial design specifications include principal permissions from authorities. The detailed design specification state measurements, static calculations, technical drawings and installation guidelines. Lastly, some properties of the specifications might be changed during on-site installation.

The design specifications are created in four different departments by employees with different competences and various ways of documenting the undertaken changes. This decentralized approach results in challenges when looking for correct data and identifying the latest specifications. Most specifications are updated to an Electronic Document and Records management system, while other specifications would be stored in spreadsheets, text documents, 3D drawing programs or as notes on paper. The result is a need for double checking and confirming that the data presented in the various systems is the latest, valid version. If wrong specifications are used the experienced consequences are quality problems as well as increased lead time and costs and an overall decreased profitability.

The company experiences problems with the accuracy of design specifications from sales to order and identified a need for systematic control of design specifications throughout all phases. The following requirements for a solution to handle product specifications in the particular project based company are identified:

- A single source for correct product documentation in all project phases needs to be established.
- A revision control mechanism has to be implemented, containing a changelog of the design history that cannot be changed or manipulated.
- The ability to handle unforeseen specifications that are not within standard must be ensured.

- Data needs to be well-structured in order to be used for analysis of project success related to design decision.

4.2 Case Company 2: Large Engineer-To-Order Business

The case company designs and installs highly customized processing plants. The developed solutions are offered on a global market. The case company is responsible for the entire product development from customer requirements to final start-up of the built plant. The product development is project-based and according to Engineer-To-Order (ETO) characteristics of low volume and high levels of customization, including stakeholders in different locations with different professional backgrounds.

In the beginning of a new product development project, many product requirements and design characteristics are unknown and need to be clarified and changed throughout the product development process. Communicating changes throughout the supply chain and ensuring that everyone is working with the latest versions of specifications is critical for the project success. Furthermore, the unique nature of the products results in unique product information and project data. In order to overcome this challenge, the company uses several Project/Product Documentation Management (PDM) systems and Project Lifecycle Management (PLM) systems. However, most fixed templates are modified to fit unique parameters of new customers and new projects. Thus, analyses of products across the organization and across projects are often very difficult, given that data and information is saved in different formats and locations and therefore cannot be directly compared.

In the company, specification sheets are created, updated and archived in separate routines in the lifetime of a project by the different domains, resulting in unique products. Even if all equipment would match that of earlier deliveries, placements and operational parameters would need to be modified to match the specific customer requirements. Therefore, it is important that a potential specification system can accommodate varied information and unforeseen data inputs. The ability to allow different stakeholders to read, write and change information while limiting these abilities for other stakeholders is important. New projects are commenced with a low level of detail compared to the available information in the later project phases. A specification revision system must be able to handle this gradual detailing. In contrast to project documentation management systems, the information has to be fully digitalized and ready for manipulation. This is not the same as digital versions of analogue information, e.g. scanned documents or electronic documents, but the ability of a new solution to be integrated in existing IT systems and digital tools is important. Business processes and key operations should not be repeated or overloaded by introducing a new specification revision system. Process calculations, equipment specifications, electrical requirements, operational parameters are examples of processes that must not be overcomplicated by the introduction of the specification management system.

5 Proposal for a Specification Chain

This section presents the concept of utilizing a chain structure to combine domain specific design specifications and track revisions. First, the identified requirements for a specification management system described in sections 3 and 4 are consolidated in section 5.1. Based on those requirements, the concept is developed and detailed from section 5.2 and onwards.

5.1 Concept Requirements

The high interconnectivity of different domains in the design of complex products requires the alignment of design specifications. Therefore, a new concept for a specification management system that *combines the specifications of different domains into one management system* is required in order to mitigate the change propagation effect. At the same time, the system needs to provide the *ability to differentiate read/write user rights* to prevent users from (accidentally or willfully) interfering with specifications belonging to a different domain. Ensuring that the specification data is *stored and kept up to date in one dedicated location* reduces the effort of collecting data for follow-up analyses. *Building the system on entirely digital data* will further improve the ability to conduct subsequent analyses. Additionally, the characteristics of customized product development, such as late engineering changes and a low level of detail in the beginning of the design process, need to be accounted for by ensuring that the specification management system is *able to handle unforeseen variables*. Another practical requirement is the ability of the new system to be *integrated with existing design tools*.

5.2 A Single Open-Ended Specification

The specification is born considering the available information when the project is started, e.g. with a customer name and a project label. As the project progresses and designers, engineering teams, controllers and managers add more information, the specification grows. Users (e.g. experts, engineers, stakeholders, etc.) can either edit the existing information or add new data to the specification. As work is detailed and the product is realized, the joint specification will represent the latest revision of all specifications at any given point of time. Figure 1 visualizes the difference between specifications that are created and stored by each

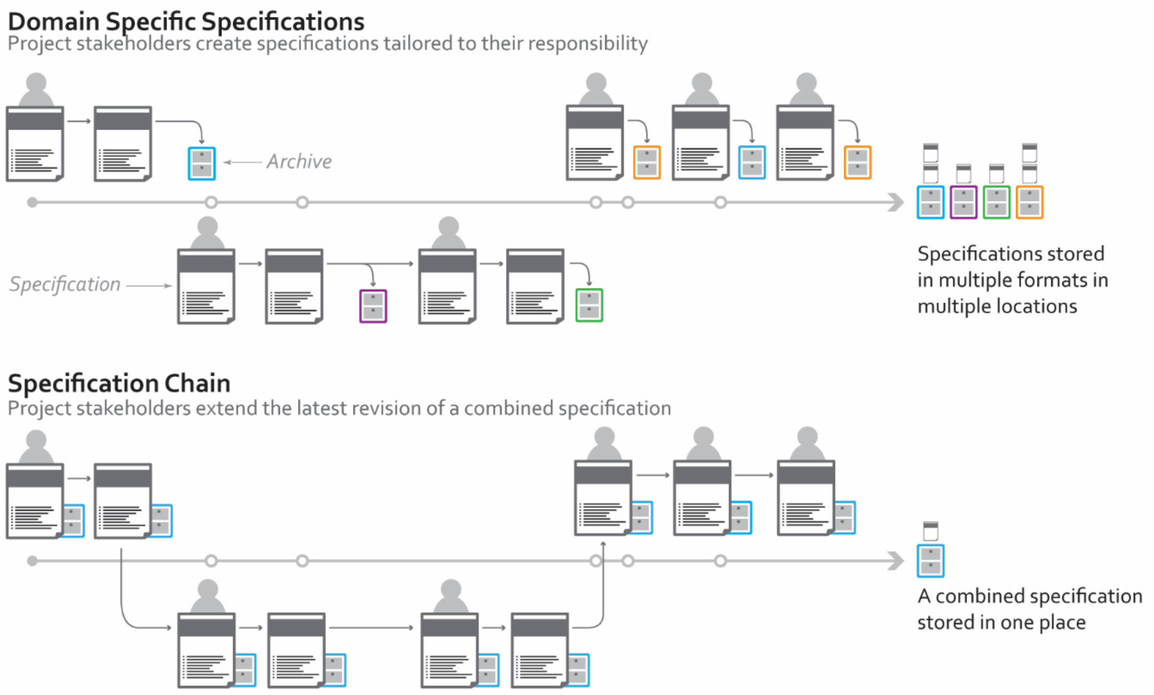
Domain Specific Specifications
Project stakeholders create specifications tailored to their responsibility
Archive
Specification
Specifications stored in multiple formats in multiple locations
Specification Chain
Project stakeholders extend the latest revision of a combined specification
A combined specification stored in one place

Figure 1: A visualization of the difference between (top) specific specifications for each domain or organizational function and (bottom) a single continuous specification that is extended for each project step.

domain or stakeholder individually and specifications that are stored and updated in one defined format. The top part of Figure 1 shows how different project stages and domains create, fill and archive their own specifications. In the end, multiple specifications in several revisions are stored in various locations. The bottom half of Figure 1 shows how each stage and project domain continues the specification of the last and every revision is archived. Lastly, a single specification is stored in one place.

Following the specification chain approach visualized in Figure 1 (bottom), the domain specific specifications are combined into one joint project specification. Picture them as chapters in a book or tabs on a website, belonging to the joint specification with the option of being collected into categories. To secure specification information to be trustworthy, different editing rights can be granted to various user groups. An example could be to limit the writing capabilities of the electrical department to the specifications that deal with electrical wiring and electrical component specifications only. Some of the specified information might also be confidential and require special reading rights, like letting certain suppliers see only a fraction of the specifications e.g. the component design information and not sensitive financial data.

Figure 2 presents the concept in more detail, visualizing the chain structure and chain interaction between users and projects. The details are further described in the following sections.

5.3 Chain Structure

Each company project has its own chain of specification data. This chain captures the project specification information from its creation (start of a new project) to the end (a completed project that is not updated further). The specifications chain consists of single blocks. These blocks have the same structure and core elements, although the data they contain may differ. They all include a unique identifier, i.e. an identification index and a reference to the preceding block. Whenever a user changes any information, a new revision of the specification is created. The user experience does not change, but in the background a new block is added to the chain, including the changed or added information and a reference to the former block. In order to ensure a sufficient quality of the information that is stored in the specification chain, each block must be designed to avoid misleading data. This could be realized by, amongst others, limiting the type of information that can be put in, forcing the use of certain units and using a common nomenclature. The responsibility for monitoring this process can either lie with the stakeholders who are responsible for the overall execution of the product development process or experts in each stakeholder group.

In project-based engineering businesses, many projects contain much of the same information. The actual details and values of variables might not be identical, but the categories of information needed and the specific fields to fill will most likely be the same across the majority of product development projects. In order to increase work efficiency the system must therefore allow this work to be re-used and not re-done in every project. To do this, the specification chain can be created using a previous project chain. By copying the state of another previous project at a specific time, all the defined variables and values are used as basis for the new project. For the project manager it functions as a template, at the back-end it functions as a fork to the chain. A fork is a common term in computer science and refers to the point where a continuous chain of revisions branches off into two separate new chains. The original chain is unaltered, while the new chain has its origin in the targeted point of the original chain by creating a new block that refers back to an existing block in the

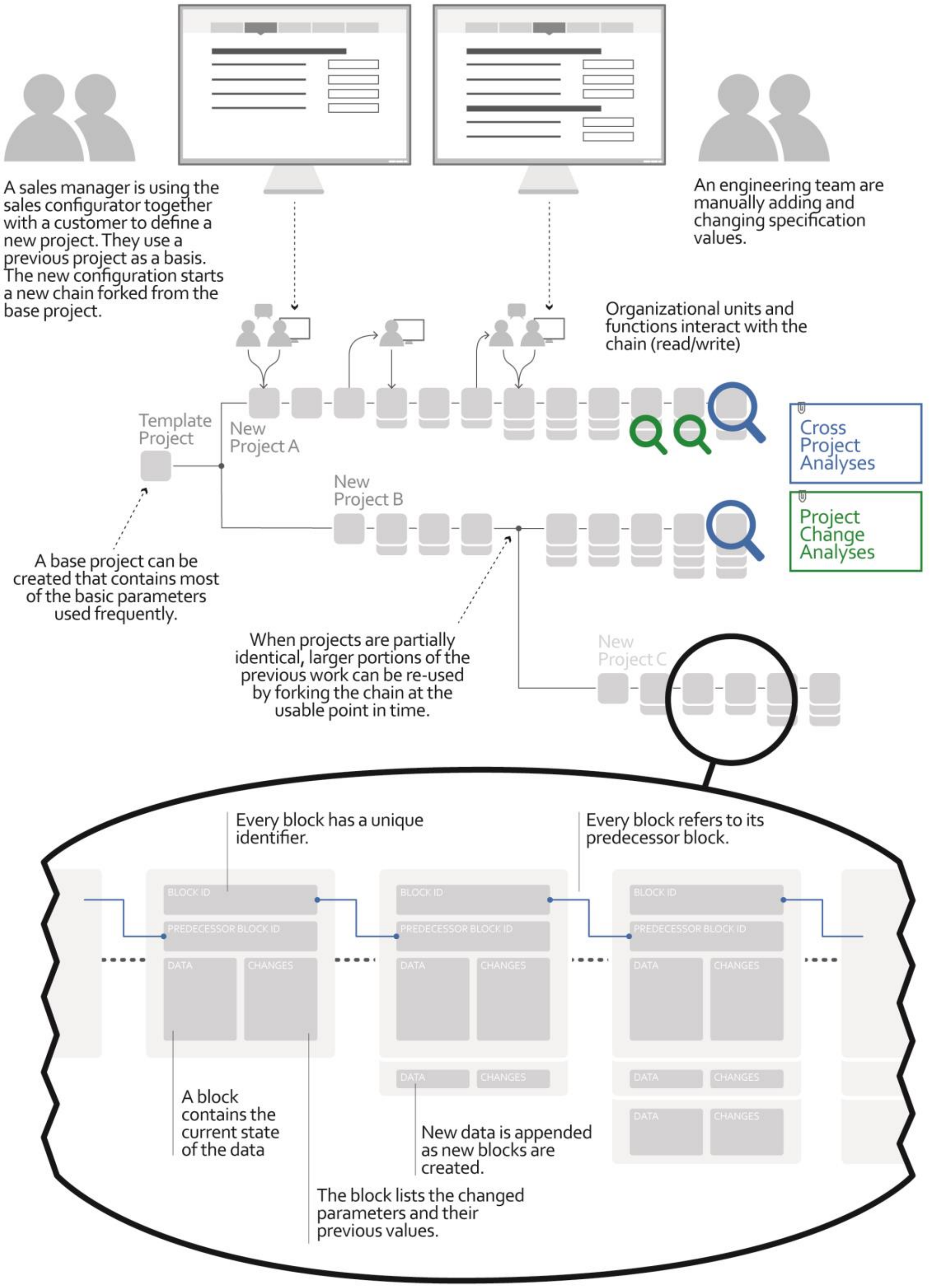
A sales manager is using the sales configurator together with a customer to define a new project. They use a previous project as a basis. The new configuration starts a new chain forked from the base project.
An engineering team are manually adding and changing specification values.
Organizational units and functions interact with the chain (read/write)
Template Project
New Project A
New Project B
New Project C
Cross Project Analyses
Project Change Analyses
A base project can be created that contains most of the basic parameters used frequently.
When projects are partially identical, larger portions of the previous work can be re-used by forking the chain at the usable point in time.
Every block has a unique identifier.
Every block refers to its predecessor block.
BLOCK ID
PREDECESSOR BLOCK ID
DATA
CHANGES
A block contains the current state of the data
New data is appended as new blocks are created.
The block lists the changed parameters and their previous values.

Figure 2: Overview of the concept of a combined specification chain. Top: Project stakeholders interacting with the chain through a web interface. Middle: Several product development projects emerging from the same root specification. Bottom: Chain-like file structure where every specification revision refers to the last one.

specifications and start as a mirror of a previous project. If a new project is very similar to a previous project, a later state of that previous project can be used as a basis. Figure 2 illustrates this with a *Template Project* serving as the base for a new project chain.

5.4 Interfaces

Accessing the specification chain as a user could be realized similar to accessing a website. Several tabs contain information regarding different business units or functional groups of the project team. In these tabs, sections or blocks of specifications group related information. The site shows lists of variables with three attributes each: An index (used for coded storage and reference), a label (for explaining the use of this variable) and a value (the actual specification). A user can then edit existing information or add new information as needed, as illustrated on Figure 2 (top). IT systems and business tools can interact as users on the chain. If simulation programs are used for specifying component dimensions, those dimensions can be written directly from the software to the specification chain.

5.5 Analyses

The structured data-storage also enables inter-project comparison and analysis of the change of variables along a chain i.e. as the project evolves. It furthermore enables cross-project comparison and analysis, to reveal how projects differ. Figure 2 visualizes both these analytical possibilities. This ability to systematically use the project specification data is important for conducting portfolio management and setting out portfolio strategies.

6 Discussion

This section will discuss the benefits and limitations of the proposed concept along with notes on related technology and the feasibility of realization.

6.1 Fulfilment of Identified Requirements

The developed concept for a specification management system meets the majority of the requirements identified in the literature and case study. The consolidation of all relevant design specifications into one management system is taken care of, thereby mitigating the change propagation effect. Changes are automatically broadcasted, ensuring that every stakeholder is working with the latest version of the design specifications. Furthermore, the effort of conducting follow-up analyses is reduced by storing the entire specification data in one location. In contrast to domain-specific specifications, where the information has to be retrieved from multiple locations and in multiple formats, the data is now readily available from the specification chain. Additionally, by governing the data and formats that make up the specifications, it can be ensured that all specifications exist in digital form.

Nevertheless, more research and development effort has to go into ensuring that the specification chain can handle unforeseen changes and variables throughout the product development process. Furthermore, the integration of the specification system into the existing IT infrastructure remains a challenge. The small-medium Configure-To-Order company started to develop and implement a specification management system similar to the proposed concept. The alignment with existing tools in the large Engineer-To-Order case company on the other hand proposes a challenge. Future research should focus on the identification of interfaces in the existing IT landscape in a company and propose measures to ensure the integration of a specification management system.

6.2 Practical Realization of the Concept

In the following subsection, potential tools for the implementation are presented. The proposed list of tools is not meant to be exhaustive, but rather to give an impression on how to practically realize the specification chain in an industrial context.

Git (Spinellis, 2005; Tsitoara, 2020) is a tool for managing revisions and versions of software. It allows developers to branch code bases and work on local mirrors of the code. The strict version controlling of *Git* would significantly improve change tracking for engineering projects. However, it is focused on code-bases. Hence, graphical data and more elaborate information is difficult to control with *Git*. If the specification information can be condensed into codified information (e.g. variables and data), then *Git* would be a good match.

In order to improve data integrity, the concept of structuring specifications in chain data structures can be combined with the possibilities of decentralized storage. A popular topic in recent years in this subject has been the implementation of blockchain data structures for handling and managing transactions across the internet. Transactional data is stored in blocks of data, strung together in a continuous series of tamper proof blocks. In the most popular implementations, the blockchain is combined with a decentralized storage scheme. Blockchain is especially known for the utilization within crypto currency systems and the anticipated revolution of the traditional financial banking systems (Antipova, 2019; Golosova & Romanovs, 2018; Jain et al., 2019; Nofer et al., 2017). Expanding beyond the initial use for currencies, the blockchain technology is being widely tested for so-called ”smart contracts” and other automated transactional interactions (Antipova, 2019). The idea is to let events and reactions be programmed into the data chain, so if certain transactions happen, other pre-determined transactions will be executed. This reduces the human interference when dealing with exchanges that are dependent on certain preconditions or prerequisites to be fulfilled. Furthermore, blockchains are getting attention as a medium for more widespread data retrieval and storage (Akbari et al., 2018; Rakovic et al., 2019) or backbones for large-scale data management systems and Internet-of-Things (IoT) networks (Jain et al., 2019).

All information in the specification chain is stored as simple variables, so there are no formats of data that require excessive storage capacities. However, a large amount of blocks (i.e. revisions) of the data needs to be created and stored. If they are stored in multiple redundant locations (for data integrity purposes), it requires even more data storage capacity. Yet storage capacities are increasing fast and by ensuring a simple datatype the development effort should be manageable.

6.3 Architectural Preparation

The concept of chain-based specifications requires a fundamental coherency of the system that defines the product. The definition of product sub-subdivision, interfaces and commonalities is often referred to as a *Product Architecture*. This architecture defines the basic blueprint form which products are created. This should form the base for the specifications in the specification chains, to ensure continuity and coherence. Without a properly defined architecture, specifications would be difficult to compare and add information upon. Agreement of nomenclature between departments, hierarchical systems decompositions and product/project interfaces is a prerequisite for the implementation of this system.

7 Conclusion

There have been plenty focus on the structured storage of product documentation in past research activities. However, the design specifications leading that function as the foundation of the product are usually left in domain-specific documents in various formats and locations. This leads to difficulties in sharing specification changes and retrieving project data for portfolio analyses. The reviewed literature combined with two industry cases suggest specific needs for engineering specification revision management: (1) To have updated and reliable specifications collected in one place, (2) to handle unknown characteristics of unique projects, (3) to be able to manage revisions and changes in project specifications and (4) to have data available for analyses and portfolio management. To accommodate these needs, a set of solution requirements are condensed from literature and industry. The paper presents a concept of combining specifications built on a chain-like data structure. It allows detailed tracking of changes and the reuse of previous work by forking (e.g. branching) of specification chains.

Project management tools are often designed to support execution and used to optimize the creation and documentation of the projects and products. They are seldom designed to support design re-use, data collection, portfolio analyses or enable project continuity. A specification management system based around chain data structures has these possibilities. By having new specifications refer to previous specifications, the structure and data within them can be re-used, enabling each project to serve as a template for the next. This connectivity between specifications and between projects also eases the analytical tasks of comparing product development projects and creating an overview of the portfolio of products. Essentially this concept showcases a management system designed for continuity and analyses instead of documentation and archival.

References

Akbari, E., Wu, Q., Zhao, W., Arabnia, H. R., & Yang, M. Q. (2018). From Blockchain to Internet-Based Voting. *Proceedings - 2017 International Conference on Computational Science and Computational Intelligence, CSCI 2017*, 218–221. https://doi.org/10.1109/CSCI.2017.34

Antipova, T. (2019). *Digital Science - Advances in Intelligent Systems and Computing* (A. Rocha (Ed.)). Springer. https://doi.org/https://doi.org/10.1007/978-3-030-02351-5

Anzengruber, K., & Hehenberger, P. (2017). *Distributed data management in product development using Git Andr e Klaus Anzengruber and Peter Hehenberger*. *10*(1).

Bhuiyan, N., Gatard, G., & Thomson, V. (2006). Engineering Change Request Management in a New Product Development Process. *European Journal of Innovation Management*, *9*(1), 5–19. https://doi.org/10.1108/14601060610639999

Bohm, B., Gewald, N., Kohlein, A., & Elger, J. (2011). Mechatronic models as a driver for digital plant engineering. *ETFA2011*, 1–8. https://doi.org/10.1109/ETFA.2011.6059100

Brunoe, T. D., & Nielsen, P. (2012). A case of cost estimation in an engineer-to-order company moving towards mass customisation. *International Journal of Mass Customisation*, *4*(3/4), 239. https://doi.org/10.1504/ijmassc.2012.047400

Bruun, H. P. L., Mortensen, N. H., Harlou, U., Wörösch, M., & Proschowsky, M. (2015). PLM system support for modular product development. *Computers in Industry*, *67*. https://doi.org/10.1016/j.compind.2014.10.010

Cho, D., Han, J., & Yang, J. (2016). Sandglass-type Product Specification Management Method for Supporting Modular Design of Semiconductor Manufacturing Equipment. *Journal of Mechanical Science and Technology*, *30*(5), 2243–2256.

https://doi.org/10.1007/s12206-016-0434-9

Clarkson, P. J., Simons, C., & Eckert, C. (2001). Predicting Change Propagation in Complex Design. *Proceedings of DETC'01 ASME 2001 Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, 1–10.

Darlington, M. J., & Culley, S. J. (2002). Current Research in the Engineering Design Requirement. *Proc Instn Mech Engrs Vol 216 Part B: J Engineering Manufacture*, *216*, 375–388.

Duchi, A., Pourabdollahian, G., Sili, D., Cioffi, M., Taisch, M., & Schönsleben, P. (2014). *Motivations and Challeges for Engineer-to-Order Companies Moving toward Mass Customization*.

Foehr, M., Gepp, M., & Vollmar, J. (2015). Challenges of system integration in the engineer-to-order business. *IECON 2015 - 41st Annual Conference of the IEEE Industrial Electronics Society*, 73–79. https://doi.org/10.1109/IECON.2015.7392078

Forza, C., & Salvador, F. (2006). Product Information Management for Mass Customization. In *Product Information Management for Mass Customization*. https://doi.org/10.1057/9780230800922

Golosova, J., & Romanovs, A. (2018). Overview of the Blockchain Technology Cases. *59th International Scientific Conference on Information Technology and Management Science of Riga Technical University, ITMS 2018 - Proceedings*, 1–6. https://doi.org/10.1109/ITMS.2018.8552978

Halevi, G. (2012). Product Specification and Design. In S. G. Tzafestas (Ed.), *Intelligent Systems, Control and Automation: Science and Engineering* (pp. 101–132). Springer.

Hameri, A. P., & Nikkola, J. (1995). How engineering data management and system support the main process-oriented functions of a large-scale project. *Production Planning and Control*, *10*(5), 404–413. https://doi.org/10.1080/095372899232920

Hamraz, B., & Clarkson, P. J. (2015). Industrial Evaluation of FBS Linkage – A Method to Support Engineering Change Management. *Journal of Engineering Design*, *26*(September 2015). https://doi.org/10.1080/09544828.2015.1015783

Hvam, L., Mortensen, N. H., & Riis, J. (2008). *Product Customization*. Springer. https://doi.org/10.1007/978-3-540-71449-1

Hvam, L., Pape, S., & Nielsen, M. K. (2006). Improving The Quotation Process With Product Configuration. *Computers in Industry*, *57*(7), 607–621. https://doi.org/10.1016/j.compind.2005.10.001

Jain, S., Rastogi, U., Bansal, N., & Kaur, G. (2019). Blockchain Based Cryptocurrency for IOT. *2019 6th International Conference on Signal Processing and Integrated Networks, SPIN 2019*, 744–749. https://doi.org/10.1109/SPIN.2019.8711727

Jarratt, T., Clarkson, J., & Eckert, C. (2005). Engineering change. In *Design Process Improvement* (pp. 263–285). Springer. https://doi.org/https://doi-org.proxy.findit.dtu.dk/10.1007/978-1-84628-061-0

Kuznetsov, S. D., & Poskonin, A. V. (2014). *NoSQL Data Management Systems*. *40*(6), 323–332. https://doi.org/10.1134/S0361768814060152

Lämmer, L., & Theiss, M. (2015). Product lifecycle management. In *Concurrent Engineering in the 21st Century: Foundations, Developments and Challenges*. https://doi.org/10.1007/978-3-319-13776-6_16

Muli, M., & Allen, J. (2013). Data management in the model-based design paradigm. *SAE Technical Papers*, *9*. https://doi.org/10.1093/qjmam/hbu021

Myrodia, A., Randrup, T., & Hvam, L. (2018a). Configuration Lifecycle Management – An Assessment of the Benefits Based on Maturity. *Proceedings of the 20th International Configuration Workshop*, 119–124.

Myrodia, A., Randrup, T., & Hvam, L. (2018b). Configuration Lifecycle Management –

Future of Product Configurators. *Proceedings of the 2018 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*, 1315–1319. https://doi.org/10.1109/IEEM.2018.8607471

Nofer, M., Gomber, P., Hinz, O., & Schiereck, D. (2017). Blockchain. *Business and Information Systems Engineering*, *59*(3), 183–187. https://doi.org/10.1007/s12599-017-0467-3

NoSQL. (2020). https://nosql-database.org/

Rakovic, V., Karamachoski, J., Atanasovski, V., & Gavrilovska, L. (2019). Blockchain Paradigm and Internet of Things. *Wireless Personal Communications*, *106*(1), 219–235. https://doi.org/10.1007/s11277-019-06270-9

Robertson, D., & Ulrich, K. (1998). Planning for Product Platforms. *Sloan Management Review*, *39*(4), 19–31. https://doi.org/Article

Rocha, F., & Varela, M. L. R. (2013). *Product Documentation Management Through REST-Based Web Service*. 179–189. https://doi.org/10.1007/978-94-007-4722-7

Siddiqui, Q. A., Burns, N. D., & Backhouse, C. J. (2004). Implementing product data management the first time. *International Journal of Computer Integrated Manufacturing*, *17*(6), 520–533. https://doi.org/10.1080/09511920410001674596

Simpson, P. (2010). Design Specification. In *FPGA Design* (pp. 9–13). Springer. https://doi.org/10.1007/978-1-4419-6339-0

Spinellis, D. (2005). *Git*. 100–101.

Tsitoara, M. (2020). *Beginning Git and GitHub*. APRESS. https://doi.org/doi.org/10.1007/978-1-4842-5313-7

Wasmer, A., Staub, G., & Vroom, R. W. (2011). An Industry Approach to Shared, Cross-Organisational Engineering Change Handling - The Road Towards Standards for Product Data Processing. *Computer-Aided Design*, *43*(5), 533–545. https://doi.org/10.1016/j.cad.2010.10.002

www.cio-wiki.org. (n.d.). *Configuration Lifecycle Management (CLM)*. Retrieved May 28, 2020, from https://cio-wiki.org/wiki/Configuration_Lifecycle_Management_(CLM)