

How to scope configuration projects and manage the knowledge they require

Abstract

Purpose – This study explores the use of the knowledge management (KM) perspective for configuration projects. Configuration projects implement configurators as IT systems that help companies manage the specification process of customised products. An effective method of retrieving and formalising knowledge for configurators is essential, because it can reduce the risk of unsuccessful implementation and the time and effort required for development. Unfortunately, no standard KM frameworks are available specifically for configuration projects. This study identifies the knowledge necessary for different phases of a configuration project (which knowledge, for what purpose, from what sources), examines how it is transformed during a configuration project (what KM activities and tools are used) and establishes how the knowledge can be documented for future maintenance and updates.

Design/methodology – This paper proposes a four-step framework for making the KM process more efficient in configuration projects. The framework is based on the literature and, developed in collaboration with industrial partners, and tested on four configuration projects in two engineering companies. The framework is a structured KM approach designed to save time for both domain experts and the configuration team. We use a qualitative exploratory design based on multiple data sources: documentation, workshops and participant observation.

Findings – The proposed framework comprises four steps: (1) determination of the system's scope, to establish the project's goal based on stakeholders' requirements and prioritise the required products and processes, (2) knowledge acquisition, to classify the knowledge according to the desired output and identify different knowledge sources, (3) modelling and knowledge validation, and (4) documentation and maintenance, to ensure the KM system can be maintained and updated in the future.

Research limitations/implications – Because the framework is tested on a limited number of cases, its generalisability may be limited. However, focusing on a few case applications allows us to assess the effectiveness of the framework in detail and in depth in order to identify the practical challenges of applying it. The results of the tests support the framework's validity. Although the framework is designed mainly for engineering companies, other industries could benefit from employing it as well.

Practical implications – The individual steps of the framework create a structured approach for the KM process. Thus, the approach can save both time and resources for companies, without the need for additional investment.

Originality/value – A standard framework is lacking in the literature on KM for configuration projects. This study fills that gap by developing a KM framework for configuration projects, based on KM frameworks developed for IT projects, and KM tools.

Keywords: Configuration projects, Knowledge management, Scoping, Knowledge acquisition, Product modelling, Product documentation

Paper type: Research paper

1. Introduction

Configurators have entered a new stage of maturity and have recently received increased attention from both researchers and practitioners. Configurators support decision-making processes in the sales and engineering phases of a product, which can determine the most important decisions regarding product features and cost (Hvam et al., 2008). Configurators enable companies to develop product alternatives to facilitate sales and production processes (Felfernig, Reiterer, et al., 2014) by incorporating information about product features, product structure, production processes, costs and prices (Forza and Salvador, 2007). This information is modelled into the configurators during their implementation (Forza and Salvador, 2007). Because configurators are implemented in the context of a project, we refer to this process as a 'configuration project'.

The variety and complexity of knowledge in configuration projects is discussed in the literature on configurators. Increased complexity of products increases the number of product features to be modelled and maintained in a configurator (Ardissono et al., 2003). The configuration knowledge for different parts of a product is often spread among various experts in a company (Hvam et al., 2008). Other valuable sources of knowledge are available in internal software systems, such as enterprise resource planning (ERP) systems, calculation systems and spreadsheets (Friedrich et al., 2014). Therefore, a knowledge acquisition and cleansing stage is required early in a configuration project's development phase (Friedrich et al., 2014). Once the configurator is up and running, further knowledge may be necessary to keep it up-to-date. Therefore, knowledge is required throughout a configurator's life.

Knowledge management (KM) in configuration projects is one of the most time-consuming tasks for domain experts¹ and the configuration team². KM is an integrated process incorporating a set of activities to create, store, transfer and apply knowledge to a knowledge business value chain (Aurum et al., 2008). In configuration projects, KM is challenging, because it involves the entire life cycle of knowledge, from acquisition (Hvam et al., 2008; Tiihonen et al., 1996) to modelling³, validating, and testing (Hansen et al., 2012; Magro and Torasso, 2003; Tseng, H.E. et al., 2005; Yang et al., 2009) and finally to documenting and updating. Project teams must have access to the knowledge required for configuration projects (Turley, 2007) in order to maintain awareness of the following issues when managing and modelling the configurator's knowledge (Studer et al., 1998): (1) models are only approximations of the reality, because the modelling process is open-ended, (2) the modelling process is cyclical, and because new observations may lead to a refinement, modification or completion of the already built-up model, the model may guide further knowledge acquisition, and (3) the modelling process is dependent on the subjective interpretations of the knowledge engineer⁴.

KM is critical to the development of centralised configurators (Fleischanderl et al., 1998), because (1) knowledge must be shared among supply-chain participants and different reasoning mechanisms and tools must be integrated, and (2) the adaptive user interface must be dynamically generated by the application of business rules and personalisation strategies based on the product knowledge stored in the knowledge base (Ardissono et al., 2003). Effective KM facilitates the creation and integration of knowledge, minimises knowledge losses and fills knowledge gaps throughout the project (Lech, 2014). The primary recommendation for achieving effective KM is to adopt a KM framework designed for the system context under consideration (Rubenstein-Montano et al., 2001).

Unfortunately, no systematic framework currently exists for KM in configuration projects (Forza and Salvador, 2002a; Friedrich et al., 2014). Based on the discussions above, the lack of such a framework leads to faulty KM processes for the following reasons: (1) the modelling process depends on new observations in a cyclic course, (2) the model requires continuous revision at all stages of the modelling process, so the

configuration team lacks confidence in the adequacy of the knowledge, (3) the acquired knowledge, which involves the project's entire life cycle, is influenced by knowledge engineers' interpretations, so its objectivity is questionable, (4) the project does not have a defined scope and extension, (5) the configuration team experiences difficulties when they have to manage knowledge for areas outside of their expertise (Haug and Hvam, 2008; Nonaka, 1994; Studer et al., 1998). Some studies have suggested a general KM framework (e.g. Heisig, 2009), others have proposed individual steps for the KM of configuration projects (e.g. Felfernig, Hotz, et al., 2014) and still others have highlighted tools or frameworks available for KM in information technology (IT) projects in general (e.g. McGinnis and Huang, 2007), but not specifically for configuration projects. These indications and tools for generic KM and KM in IT projects can serve as the basis for a systematic methodological framework for KM in configuration projects.

This study proposes a framework that identifies the most important KM steps in configuration projects. This framework is based on generic KM frameworks and KM frameworks designed for IT projects. By using this framework, companies can improve the quality of acquired knowledge and reduce its complexity by limiting the knowledge to that needed for the stakeholder's requirements. The framework also offers methods for keeping knowledge up-to-date.

The remainder of the paper is structured as follows. Section 2 explains the study's methodology (which is presented before the literature review because the literature review is an essential part of the framework development). Section 3 discusses the relevant literature, and Section 4 explains the framework development. Section 5 presents the results of the framework validation, and Section 6 discusses the results. Section 7 concludes the paper.

2. Research Method

This study was performed in three main phases: literature review, framework development and framework validation. Hereafter, for each phase we present the method that was adopted.

2.1. Literature review

The relevant literature is reviewed to clarify the present study's position in relation to existing research. This allowed us not only to ascertain whether our research has the potentials to add something new but also to identify which parts of the available knowledge are relevant to our purpose.

First, we take a generalised view of KM frameworks in order to explain key KM concepts and discuss literature reviews on numbers of KM frameworks. Studies comparing different KM frameworks are resulted in decisions regarding the development of a general KM framework (see, e.g., European Committee for Standardization, 2004; Heisig, 2009). This general KM framework supports decisions about the steps required for the framework proposed in the present article.

Second, we discuss the challenges of KM in configuration projects and identify existing methods that could be applied to overcome these challenges (e.g. McGinnis and Huang, 2007; Rodriguez and Al-Ashaab, 2005). This step reveals some KM specific challenges and also sheds light on the available KM literature in configuration projects as well as the lack of suitable KM framework for configuration projects.

Third, we consider KM frameworks developed for IT projects in general to determine whether any of them are applicable to configuration projects (e.g., Lech, 2014; Reich et al., 2012). Moreover, comparing different studies led to conclusions about a generic KM framework for IT projects.

Fourth, the differences between the required KM framework for IT projects in general and configuration projects are identified. This comparison highlighted the need for KM frameworks tailored to configuration projects (see, e.g., Basili and Weiss, 1984; Forza and Salvador, 2002; Friedrich et al., 2014; Tiisonen et al.,

1996). Nevertheless, KM frameworks for IT projects could serve as a foundation for developing KM frameworks for configuration projects and for proposing ad hoc frameworks.

2.2. Framework development

The framework development was based on (1) the literature review, (2) analytical thinking and (3) interactions with industrial partners (case company).

From the literature review, we obtained the following tools and methods for the individual phases of KM in configuration projects: (1) identifying stakeholder requirements, (2) prioritising products and product features to include in the configurators, (3) identifying knowledge resources, (4) modelling and validating knowledge, and (5) documenting configurators and maintaining knowledge.

Analytical thinking was used to break the problem into smaller sections. Categorising different aspects of KM in the literature supported the organisation of concepts into hierarchical phases. Afterwards, we used these phases to investigate the keywords for each section.

Based on the relevant information in the literature, the team then identified the key issues. Iterative design method, which blends the activities of designer and user, creator and player, is based on a cyclical process of prototyping, testing, analysing and refining a work in progress (Zimmerman, 2003); thus, we developed the framework through an ongoing dialogue between the researchers and configuration teams. In other words, the framework was developed and validated in an iterative process in one company which allows us to benefit from the strength of using case study method (McCutcheon and Meredith, 1993; Van de Ven, 1989). The industrial partner for developing the framework has experienced both successes and failures in configuration projects.

2.3. Framework validation

Finally, after making the final decisions regarding the sequences of steps and proposed tools for the framework, we conducted multiple case studies to validate the framework's usability in different circumstances. Our units of analysis were the individual configuration projects. We analysed four projects in two companies. Eisenhardt (1989) recommendation to conduct the analysis in two steps was followed: firstly we performed the analysis within each case (project) and subsequently we searched for cross-case patterns. Having four projects in two companies, cross case comparisons were conducted both across projects in the same company and across projects in different companies. The use of multiple cases in different settings and multiple sources of information for each case improved the validity of our findings. To facilitate the reading of the cases, the methodological details are reported in Section 5.

3. Literature Review

This section reviews, first, the general literature on KM frameworks, second, KM challenges in configuration projects, third, KM frameworks for IT projects, and fourth, the use of KM in configuration projects. With a general understanding of KM frameworks and how IT systems frame the KM process, and based on the available literature for KM in configuration projects, the paper introduces the proposed framework in Section 4.

3.1. Knowledge model and KM frameworks

To address KM in configuration projects, it is necessary to understand two fundamental KM notions, namely knowledge models and KM frameworks. A knowledge model comprises different categories of

knowledge organised in three main levels (Liebowitz and Megbolugbe, 2003). The bottom-level category is domain knowledge, which specifies domain-specific knowledge and information. The middle level is inference knowledge, which refers to the basic inference steps made on the basis of domain knowledge. The upper-level category is task knowledge, which refers to application goals and how they can be realised through decomposition into subtasks and inferences.

Knowledge has to be carefully exploited for its potential usefulness. Doing so constitutes a challenge. Heisig (2009) emphasised that addressing the challenge of knowledge handling with only one activity, like ‘sharing knowledge’, is insufficient. Several interconnected activities are needed for successful knowledge handling (Heisig, 2009). It is therefore necessary to have frameworks that present both the activities to be performed and to properly manage knowledge and the relationships among these activities.

A KM framework represents the relation and dependency among the various KM components (processes, activities and enablers) (Liebowitz and Megbolugbe, 2003). KM frameworks support the determination and positioning of KM activities (European Committee for Standardization, 2004). During the planning and implementation of projects with KM requirements, frameworks can provide useful assistance for holistic KM solutions (Liebowitz and Megbolugbe, 2003).

Several KM frameworks have been reported in the literature. Heisig (2009) outlined the similarities and differences between 160 KM frameworks (proposed in studies such as Bose and Sugumaran, (2003); BSI – British Standards Institute, (2001); Kelleher and Levene, (2001)). Heisig found that the most frequently mentioned categories of KM activities are, in decreasing order of frequency of appearance in KM frameworks, creation, application, storing and identification of knowledge. To share knowledge effectively, or use existing knowledge, tools are often necessary, although this does not always mean technical tools (European Committee for Standardization, 2004). The European CEN workshop introduced the KM framework for practitioners in terms of identification, creation, storage, sharing and use (European Committee for Standardization, 2004), in line with Heisig (2009). Comparing the various studies makes it possible to identify a general consensus (although with some terminological differences) regarding a general KM framework that includes the following activities: identification, creation, storage, sharing and use.

3.2. Configuration projects: a KM challenge

Academics and practitioners recognise that KM has a crucial influence on the success or failure of configuration projects (Lech, 2014). Configurators involve a great deal of knowledge that represents the complex relations among components or modules, such as configuration rules and assembly constraints (Jinsong et al., 2005). The knowledge complication of configurators is due to (1) volatility problems – they are a dynamic subject domain, (2) scope expansion – because configurators are successful business tools, users will ask for more, and (3) the large size and information complexity of the knowledge in the systems (Barker et al., 1989).

Because the mapping between functional roles and the set of components available is typically many to many, the configuration task is dynamic in nature (Sabin and Weigel, 1998). Today, companies integrate their configurators with company-wide data-modelling systems to facilitate the management of frequently changing product knowledge (Sabin and Weigel, 1998).

A simple medical device can illustrate the challenges of KM for configurators. Developing a system for configuring a hearing aid involves the following challenges: (1) the configuration engineer must learn about the details to be able to consider and model all the rules and attributes of the hearing aid, even though it is beyond his expertise, (2) because the knowledge covers millions of selections when configuring the simplest

hearing aids, the project will cover all the knowledge and will become complex, (3) the systems need integration with other systems, such as calculation systems, for accurate dimensioning in order to automate the whole process, and the scope of the project will change and expand, because the configuration engineer needs to become familiar with the other IT systems to map the systems, (4) based on recent research and developments in medical science, this product is dynamic and the configurator must be updated and aligned with all recent developments.

The tools for KM in configuration projects can be grouped, as shown in Table 1, based on commonly proposed steps for KM in configuration projects in general. These steps are summarised and explained in the following paragraphs.

Determining the scope of configurators is a KM-related challenge for industry. This step clarifies the knowledge requirements for the entire project and gives the team the opportunity to make intelligent decisions from the early phases of the project. Furthermore, in the early phases of the configuration project, the scope of the products sheds light on project goals and outputs, objectives and requirements from the stakeholders, IT architecture, etc. (Shafiee et al., 2014).

Knowledge acquisition is also frequently considered a challenge (Table 1). In the early phases of a configuration project, it is often difficult to identify and retrieve the right product knowledge to implement in the system (Shafiee et al., 2014). Knowledge acquisition entails categorising the knowledge based on the relevant stakeholders' needs, recognising all the possible sources and resources of knowledge, collecting the knowledge and categorising it based on previous analyses of the product/process. The processes by which the products are developed usually do not create the configuration-related knowledge as a part of the development effort. Instead, this additional knowledge acquisition task is performed by persons that are not product experts, which might lead to loss of data and erroneous configuration of the knowledge being used in the configuration process (Tiihonen et al., 1996).

Although gathering and representing relevant information is one of the most difficult tasks in configuration projects, modelling and validation are the challenges most frequently reported in the literature (Hvam et al., 2008; Sabin and Weigel, 1998). A considerable amount of research is therefore devoted to product modelling and communicating with domain experts to validate the knowledge.

Researchers have highlighted documentation and maintenance as a critical phase of KM for configurators (Forza and Salvador, 2002a; Shafiee et al., 2017). A primary motive for building a support system for product configuration is to support the transfer of up-to-date product configuration knowledge to the sales units and to enforce its proper use (Tiihonen et al., 1996). Studies of companies using configurators have shown that without proper documentation, they often become unable to utilise the configurators and have had to abandon or rebuild them (Haug et al., 2009). It is therefore important to have a reliable configuration model for the products implemented inside the configurator, i.e., one that has no technical errors and mirrors exactly the product design's updates (Forza and Salvador, 2002a).

Table 1. Literature base for the main steps in KM for configuration projects.

Author (year)	Determining the scope of the configurator	Knowledge acquisition	Modelling and knowledge validation	Documentation and maintenance
Forsythe and Buchanan (1998)	✓			
Tiihonen et al. (1996)		✓		
Sabin and Weigel (1998)			✓	
Aldanondo et al. (2000)			✓	
Chao and Chen (2001)			✓	
Forza and Salvador (2002a)				✓
Ardissono et al. (2003)		✓		
Margo and Torasso (2003)			✓	
Tseng et al. (2005)			✓	
Jinsong et al. (2005)		✓	✓	
Forza and Salvador (2007)		✓	✓	✓
Hvam et al. (2008)	✓	✓	✓	
Mortensen et al. (2008)	✓			
Haug et al. (2009)			✓	✓
Yang et al. (2009)			✓	
Hansen et al. (2012)			✓	
Felfernig et al. (2014)	✓		✓	
Shafiee et al. (2014)	✓			

As Table 1 shows, many authors have discussed different steps, but none have proposed a framework that incorporates the steps in sequence. The various steps explained in the literature on configurators can be connected to the overall KM framework presented in Section 3.1 (identify, create, store, share and use). Scoping the project means identifying the needed knowledge; knowledge acquisition is equivalent to creating knowledge; and the modelling and validation step is equivalent to the step of using and sharing; and finally, documentation is equivalent to storing.

3.3. KM framework in IT projects

Efficient creation, distribution, and reuse of up-to-date knowledge are critical success factors in IT projects, but unachievable in practice (Compton and Jansen, 1990; Komi-Sirvio et al., 2002). As noted, the literature has suggested a number of frameworks for KM in IT projects (Table 2). According to the level of abstraction, the frameworks range from three phases/actions to six phases/actions, and some of the frameworks focus more on acquisition (e.g. Basili and Weiss, 1984), whereas others consider the entire KM life cycle, including maintenance (e.g. Kucza and Komi-Sirviö, 2001). Some frameworks use identical terms, such as ‘knowledge identification’ and ‘knowledge scoping’, whereas other frameworks use different terms, even for similar activities/phases (e.g. ‘knowledge stock’, ‘scope’ and ‘socialisation’, all of which refer identifying the needs and goals).

Even though the frameworks use different terms for the various phases of KM in IT projects, they exhibit a number of similarities (Rubenstein-Montano et al., 2001). Almost all the frameworks start by determining the scope of the project to establish the goals, requirements and deliverables of the system. After these first phases, the frameworks typically aim to collect and categorise the knowledge and ascertain the knowledge sources and resources. Subsequently, knowledge acquisition is discussed in terms of communicating,

modelling and clarifying the knowledge. All authors consider the collection, validation and documentation of the knowledge as a separate step, and the majority of the frameworks end with a step for maintaining the knowledge.

Table 2. KM frameworks for IT projects.

Authors	Actions/phases included in KM framework
Basili and Weiss (1984)	1. Establish the goals of knowledge selection, 2. develop a list of questions of interest, 3. establish knowledge categories, 4. design and test knowledge collection form, 5. collect and validate gathered knowledge, 6. analyse the knowledge.
Kuczka and Komi-Sirviö (2001)	1. Identify need for knowledge, 2. share existing knowledge, 3. create new knowledge, 4. collect and store knowledge, 5. update knowledge.
Komi-Sirvio et al. (2002)	1. Define scope and requirements for knowledge capturing, 2. acquire knowledge, 3. package knowledge.
Rodríguez et al. (2004)	1. Identify knowledge sources and resources, 2. identify kind of knowledge, 3. identify knowledge flows (graphical modelling techniques), 4. identify faults in the knowledge flow (analyse knowledge).
Reich et al. (2012)	1. Knowledge stock (relevant domain knowledge of the IT team, the business team and the governance team), 2. enable the environment (combination of the technological and social aspects of a project that facilitate knowledge practices), 3. knowledge practices (actions taken to map and share knowledge within and between the IT, business and governance teams in an IT-enabled business project).
Lech (2014)	1. Identification (determine knowledge sources and resources), 2. acquisition/creation, 3. transfer/dissemination, 4. storage/capture, 5. use/application.
McGinnis and Huang (2007)	1. Socialisation (scoping and deliverables), 2. Externalisation (formalise the knowledge to be explicit), 3. Combination (knowledge clarification and team communication), 4. Internalisation (new deliverables, improved documentation, improved training, and process refinements).

However, these frameworks have a limited ability to support the KM process in configuration projects because of the differences between IT and configuration projects, which are explained in detail in the following section.

3.4. Configuration and IT projects: similarities and differences

Configurators are considered to be among the IT systems that are important for mass customisation (Blecker et al., 2004; Forza and Salvador, 2007; Hvam et al., 2008). However, there are several differences between IT projects and configuration projects. The first difference relates to the knowledge complexity and extensions of configurators, which make it critical to determine the scope of the project in the early phases in order to predict the level of the complexity and potential extensions. This is done by identifying the requirements, evaluating the time and budget and prioritising the different products and functions according to the variety and complexity of the knowledge, the required tasks and the resources for the project development (Männistö et al., 2001; Shafiee et al., 2014). In configuration projects, knowledge acquisition bottlenecks often occur because of the large and complex knowledge bases. In such scenarios, knowledge engineers get overwhelmed by the increasing amount, size and complexity of knowledge bases (Ulz et al., 2016). There are two types of IT projects (Whitney and Daniels, 2013). The first has a well understood, clear scope and few unknowns. The second is complex, with many unknowns; such projects often have planning and scoping issues. Therefore, IT projects can vary greatly, because they also have different natures and usually require less extensive product knowledge.

The second difference relates to the details of the communication level for configuration projects as compared to IT projects. The knowledge required for configuration projects is normally very specialised product knowledge that lies beyond the configuration team's expertise (Haug and Hvam, 2008; Studer et al., 1998). The consequence is that, for example, a knowledge engineer needs to learn the different domain aspects

from the experts to model medical equipment. Knowledge formalisation and communication in configuration projects correspond to product modelling, which is a method of representing the structure and knowledge of the product on a relatively visual, abstract level to ensure that they are understandable to all persons concerned (Shafiee et al., 2017). In IT projects, each project team declares its priorities as well as its communication and validation requirements. The team can orient itself to the amount of face-to-face communication it can manage and the extra methodology weight it should appropriately set in place (Cockburn, 2002). Because of the differences in the nature of the received knowledge, configuration projects are formalised and communicated differently than other IT projects. Consequently, the knowledge modelled in configuration projects is extensive and must be continually validated by domain experts (Basili and Weiss, 1984). Strong communication between the configuration team and domain experts in configuration projects is vital, and specific modelling techniques tend to meet this challenge in configuration projects (Forza and Salvador, 2002a). Additionally, without proper validation, very minor misunderstandings in the knowledge can lead to big errors in calculations and outputs.

The third difference relates to the need for specific comprehensive documentation and maintenance of the knowledge in configuration projects (Haug and Hvam, 2007). The knowledge has to be clear and understandable to all stakeholders and expressed in non-IT language. There is a high level of integration with other IT systems, and the knowledge must be shared among participants in the supply value chain. In addition, the frequent changes in product knowledge necessitate continual updating and maintenance of the knowledge (Friedrich et al., 2014; Tiihonen et al., 1996). By contrast, the documentation in IT projects is normally a summarised explanation of the codes and a set of user stories that are passed on to another IT specialist (Coram and Bohner, 2005). Most IT projects are not required to work with complicated products or process knowledge, and IT specialists do not have to communicate with people outside the IT field to verify the knowledge contained in the system. Furthermore, the knowledge required for IT projects does not require constant updates (Coram and Bohner, 2005).

In summary, the knowledge, and thus the KM, for IT projects differs from that for configuration projects. Owing to the complex nature of KM in configuration projects, the frameworks designed for IT projects are unsuitable for configuration projects, because they fail to incorporate sufficient steps to cover all KM needs. Table 3 presents a summary of this section.

Table 3. Differences between IT and configuration projects.

Differences	1. Knowledge complexity and project extensions	2. Level of communication	3. Documentation and maintenance of knowledge
Configuration projects	Highly complexity and varied knowledge; continuous requests for project extensions because of updated product portfolios; further development due to usage frequency; requests for more outputs (Hvam et al., 2008; Shafiee et al., 2014)	Requires very strong communication that covers all stakeholders; requires continuous validation from domain experts (Basili and Weiss, 1984; Forza and Salvador, 2002a)	Documentation of different ranges of knowledge, from integrations to product knowledge, by using modelling techniques; frequent updating of the documents due to frequent changes in product portfolio (Friedrich et al., 2014; Tiihonen et al., 1996)
IT projects	Different levels of complexity based on the type; minor or major extensions in some of the IT projects (Whitney and Daniels, 2013)	Requires minimal communication; requires final testing for each version of the project (Cockburn, 2002)	Documentation of codes; documentation updates in the event of code updates (Coram and Bohner, 2005)

4. The Proposed Framework

The framework proposed here is based on the literature on configurators and the literature on KM frameworks for general and IT projects, integrating the main phases of general and IT projects KM frameworks and including specific tools and methods. However, owing to the similarities and differences between configuration and IT systems, the framework includes the different steps available for configurator KM and incorporates experiences from the development, implementation and maintenance of existing configuration projects. The framework was improved in an iterative process using a case company and benefited from the experiences and knowledge of practitioners and academics.

As a configurator becomes more successful and popular among users, users' expectations and requirements for the system increase (Barker et al., 1989). The framework therefore needs to include the possibility of iterations in the KM. As illustrated in Figure 1, the first step involves determining the scope of the system; in the second, knowledge acquisition is carried out; in the third, the knowledge is structured (using special modelling techniques) and validated; and the final step is concerned with documentation and maintenance. Figure 1 illustrates the individual steps of the framework, showing the relations and iterations between the steps.

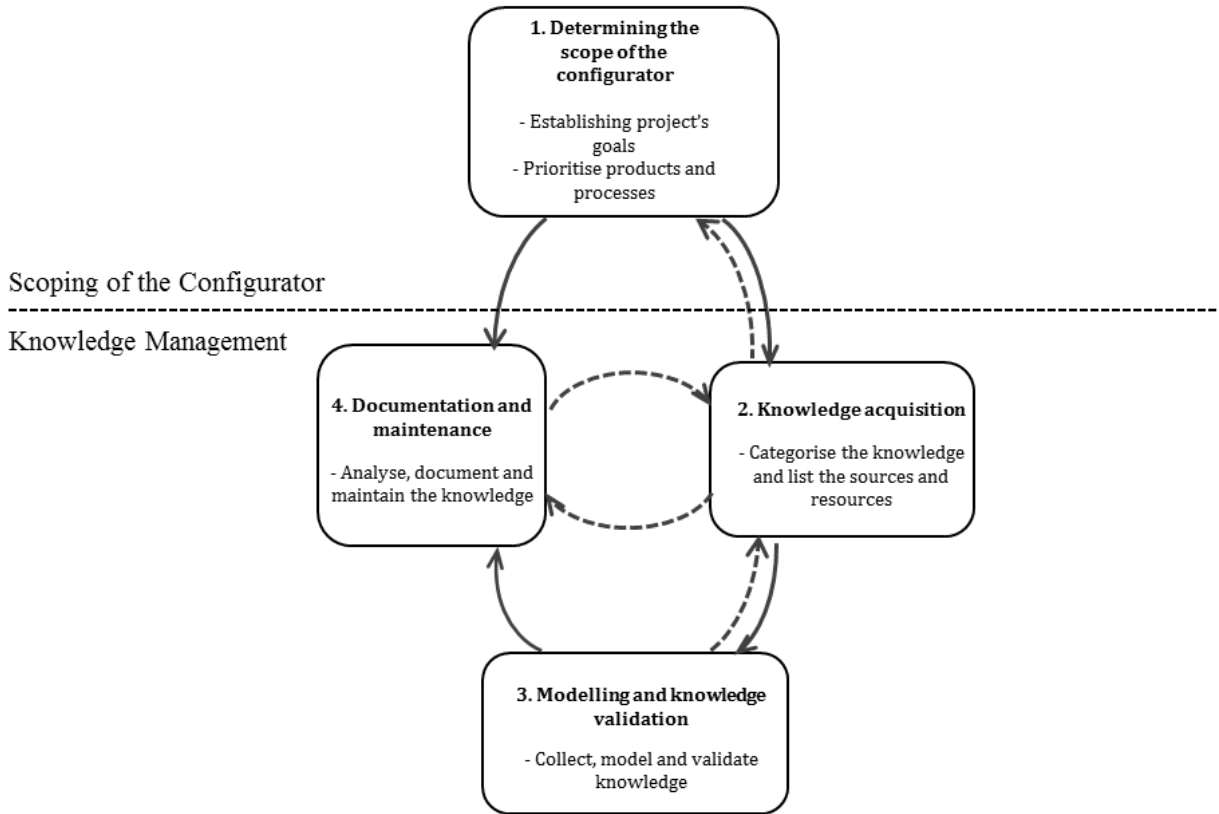


Figure 1. Proposed framework for knowledge management in configuration projects.

The following sections explain the four steps of the framework in greater detail and introduce the tools and the method used.

4.1. Step 1: determining the scope of the configurator

4.1.1. Establishing project goals

Project goals are determined by identifying stakeholders' functional and non-functional requirements. This step aims to improve the understanding of the project by identifying the main stakeholder requirements (Basili and Weiss, 1984). Non-functional requirements are general quality attributes that emphasise quality and compliance with requirements. A non-functional requirement describes not what the software will do but how it will work (Ebert, 1997), such as the reliability, consistency and maintainability of configurators. A functional requirement specifies each of the functions that a system must be capable of performing (Ebert, 1997), such as all the features of the user interface.

The stakeholders and their requirements can be drawn up using process flowcharts based on rational unified process (RUP) methods (Compton and Jansen, 1990) as well as use-case diagrams. Process flowcharts can be used to describe the current situation and different scenarios for future work (Hvam et al., 2008), whereas use-case diagrams (Figure 2) can illustrate the requirements and the actors involved in the project (Kruchten, 2007).

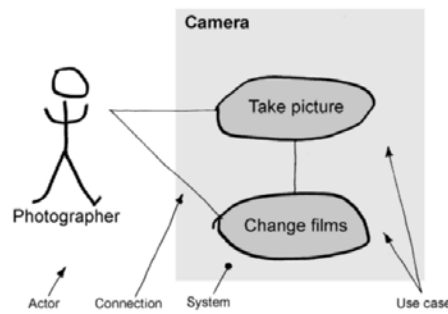


Figure 2. Example of a use-case diagram (Hvam et al., 2008).

The MoSCoW rules are commonly used when prioritising stakeholder requirements. MoSCoW is derived from the first letters of the following criteria: Must have (Mo), Should have (S), Could have (Co) and Want to have (W) (Bittner, 2002). Further details of stakeholder analysis are available in Ebert et al. (1997), Jiao et al. (2006), Lim et al. (2011) and Bittner (2002) for generic IT projects, and in Forsythe and Buchanan (1989), Hvam et al. (2008), Nellore et al. (1999), Felfernig et al. (2014) and Mortensen et al. (2011) for configuration projects.

4.1.2. *Prioritising products and processes to be included in the system*

In this step, the products or product features and functionalities to be included in the system are prioritised. The purpose of using a component-based structure, based on RUP methods, is to break a large and complex project into smaller pieces (Briand, 2003). This makes the development process easier, especially in complicated and highly engineered projects (Felfernig et al., 2014). After breaking down the project, the team should start developing one of the components or products, depending on the size of the project. To make prioritisation more systematic, a supporting tool is needed.

The recommended tool for this step is a weighting table, in which each of the components is rated against several specific weighted project success criteria and a score is computed to rank the priority of the components (Wiegers, 1999). In Weigers's (1999) approach, prioritisation is calculated based on (1) the benefits and penalties of including a feature in the system (the feature could cover both functional and non-functional requirements), (2) the cost of implementing the feature, and (3) the time and technical issues associated with the feature. This method seems to be applicable to prioritising products and processes in configuration projects.

4.2. *Step 2: knowledge acquisition*

Data clustering is a multivariate analysis technique that assigns observations (objects) of a population to clusters (groups) so that observations within the same cluster have a high degree of similarity; while observations from different clusters have a high degree of dissimilarity (Anzanello and Fogliatto, 2011; Kaufman and Rousseeuw, 2009; Tsai et al., 2009).

Walz (1993) observed a software design team sharing knowledge with customers, and he recommends (1) increasing the amount of application domain knowledge, (2) promoting knowledge acquisition by facilitation techniques and formally recognising these activities by allocating time to them, and (3) recognising that much of the information that needs to become part of the team's memory is not captured formally, particularly in standard documentation. According to Waltz, experienced designers recognised that customers may not understand the true nature of the requirements and the expectations from the results at the beginning of a project (Walz et al., 1993).

Some knowledge acquisition tools are intended for a wide variety of contexts. For example, a card sorting tool should in theory be of value in any domain where objects, concepts or even processes can be named, shuffled about and sorted (Shadbolt et al., 1999). Some knowledge acquisition tools belong to specific domains. For example, Compton et al. (1990) rejected the need for modelling and focused instead on the evaluation of prototypes developed on the basis of increasing numbers of test cases. The questions about knowledge are designed to reveal the expert's recommendations and hence strategies for how to deal with a variety of conditions, such as how to identify current conditions and which conditions warrant which actions (Woodward, 1990).

The process of knowledge acquisition in configuration projects includes the following activities: the knowledge engineer (1) communicates techniques for eliciting knowledge from relevant experts, (2) interprets this knowledge in order to draw conclusions about the reasoning process of the product experts and what may be the underlying knowledge, and (3) uses his conclusions to direct the construction of the product model (Byrd, 1992). However, these activities, which are common in configuration teams, can lower the quality of acquired knowledge and consume time and resources that could be devoted to validation (Shafiee et al., 2017).

One method of clustering in configurators is to determine output knowledge according to stakeholder requirements and subcategorise them step by step. Table 4 shows a categorisation table in which all the needed inputs and resources are determined.

Table 4. Example of a categorisation table.

Categorised phase	Needed input	Needed resources
Configuration requirements	The product data should configure the product according to the stakeholders' requirements in the execution of the system.	Stakeholders from mechanical and chemical departments, external vendors
Calculation pre-requirements	The data need to be used for the calculations and simulations that could not be contained inside the configurator and need to be integrated with the simulations software.	Stakeholders from the sales department
Document generation requirements	The data need to be used in the document generation part for the price calculation sheets (PCSs), bills of materials (BOMs), scope of supply (SOP), etc.	Stakeholders from all related departments
Integration requirements	The data need to be used for the integration section. ✓ For calculations ✓ For flow diagrams	Stakeholders from the process and mechanical departments

Listing the sources and resources of the knowledge creates value in categorising the knowledge and helps to delegate the tasks to different resources (Tiihonen et al., 1996). Organisations have two types of knowledge: explicit and tacit. Explicit knowledge is formal and systemic, whereas tacit knowledge is highly personal and difficult to formalise. Depending on the resources, the knowledge might be explicit, and come from the company's internal documentation systems, or tacit, and come from domain experts (Nonaka, 1994).

4.3. Step 3: modelling and knowledge validation

One of the steps of KM in configuration projects relates to modelling the knowledge inside the system, which normally requires validation from domain experts. Communication between IT personnel (software developers and modellers) and domain experts is an important factor for configuration projects (Stelzer and Mellis, 1998).

The knowledge modelling of configurators, known as the product (phenomenon) model structure, is one of the greatest challenges in configuration projects (Hansen et al., 2012; Sabin and Weigel, 1998). Product models

are also used for communicating with people outside the IT field, which is required to validate the knowledge (Duffy and Andreasen, 1995).

Many researchers have developed product modelling techniques to meet this challenge (Aldanondo et al., 2000; Chao and Chen, 2001; Hvam, 2001; Hvam et al., 2008; Jinsong et al., 2005; Magro and Torasso, 2003; Tseng, H.E. et al., 2005; Yang et al., 2009). This paper recommends using product variant master (PVM) along with class, responsibilities and collaboration (CRC) cards, which are based on unified modelling language (UML) notation (Hvam, 2001).

4.3.1. PVM and CRC cards

The PVM presents product knowledge in a structured format from three different perspectives: the customer's view, the engineering view and the production/part view (Hvam et al., 2008). The PVM comprises two structures: 'part-of-structure' and 'kind-of-structure', which are analogous to the structures of aggregation and specialisation within object-oriented modelling. Beck and Cunningham (1989) first proposed using CRC cards to teach object oriented thinking. Hvam et al. (2008) later presented revised versions of the CRC cards for use in configuration projects. Figure 3 shows the PVM and CRC card structure. For example, a car consists of a chassis, motor, brake system, etc. Each module/part of the product range is marked with a circle. The individual modules/parts are also modelled with a series of attributes that describe their properties and characteristics.

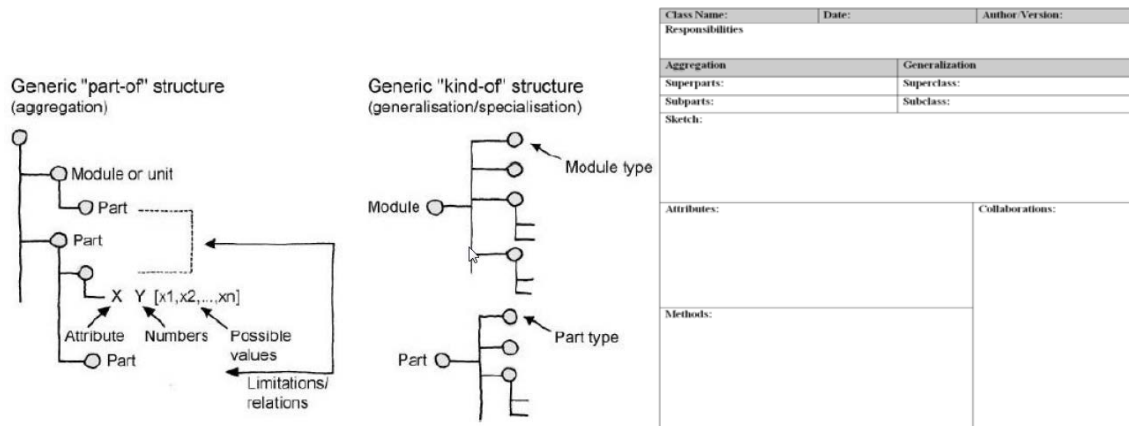


Figure 3. Structure of PVM and CRC cards (Hvam et al., 2008).

4.4. Step 4: documentation and maintenance of knowledge

This step addresses how to document and maintain the knowledge to ensure that the configurator remains stable and up-to-date. Studies of companies using configurators have revealed that without a documentation system, companies are unable to develop and maintain their configurators (Haug et al., 2009). The iterative process of testing enables feedback in the early phases of a project (Kruchten, 2007). To reach the feedbacks require a proper communication and maintenance tool. Numerous methods exist for conducting iterative project testing and validation, which eliminate unnecessary debugging processes at the end of the project (Hirsch, 2002). Modelling techniques are used as documentation tools alongside the task of communication and validation. Research supports the modelling process by adding software support and integrating these different modelling techniques (PVM and CRC) (Haug and Hvam, 2007; Shafiee et al., 2017). Selic (2009) explained agile documentation by elaborating different steps for design and development. Avoidance of duplicate knowledge is critical in documenting IT systems (Selic, 2009). The automatic agile IT system, proposed by Shafiee et al., involves two steps. The first concerns building the initial product model (PVM or

any modelling technique), which is used for the programming of the configurators. In the second step, the product model is generated directly from the configurator and is based on the structure, attributes and constraints inside the configurators. The configuration engineer can control the models, such as showing/hiding different parts or providing users with descriptions. Therefore, the product model does not need to be maintained outside the configurators. This approach meets the demand for agile documentation and efficient communication with domain experts and uses the fewest resources possible (Shafiee et al., 2017).

5. Framework validation

5.1. Method setting

Having developed the framework, we needed to assess whether and where it works. One case project was used to develop and improve the framework iteratively, and the analytical thinking and literature base used in the development of this framework should ensure that the framework works logically in different situations. However, the framework's actual practical performance can be proved only by applying it to real settings. For this reason, we decided to use our framework in case companies. However, because applying a framework requires not only a company's availability, but also considerable time and resources in the organisation, we were able to apply the frameworks in only four projects at two companies. The study of a limited number of case applications allowed us to conduct a detailed assessment of how the framework works and to understand why it may present challenges in application. Case study research seeks to find logical connections among observed events, relying on knowledge of how systems, organisations and individuals work (Kaplan and Duchon, 1988; McCutcheon and Meredith, 1993). Understanding the 'how' and 'why' is one of the main reasons for using multiple case studies in several disciplines, such as explanatory studies in operations management and technology management (McCutcheon and Meredith, 1993; Yin, 2013).

When conducting multiple case studies in this type of research, attention should be given to data triangulation as well as observer triangulation (Creswell and Clark, 2011; Johnson et al., 2007; Yin, 2013). Multiple benefits can be gained from triangulation, such as complementary insights, which enhance the richness and convergence of observations, which in turn enhance confidence in the findings. For example, interviews can be conducted by two persons, with one researcher handling the interview questions and the other taking notes and recording observations (Eisenhardt, 1989). The research team observed the participants and document them during the projects by focusing on KM process.

5.2. Selection of cases

A key concern of this study is the application of the proposed framework. Because the framework applies to different configuration projects, the unit of analysis is defined as a configuration project. Four projects at two case companies, which provide highly engineered products and were currently in the process of developing and utilising configurators, were used for the case studies. Both companies were engineer-to-order (ETO) and in the development phase of a configuration project, and both understood to benefit from a better KM framework. In the selected cases the products are physical goods with stable product architecture. A lot of configuration projects regards physical products, where the basic product architecture (or at least its core part) is stable over long periods of time, thus the configurators are also stable over years (Haug et al., 2011). The survey of Haug et al. (2011) shows that the average life time for the configurators handling the complicated ETO products can exceed 11 years.

As Figure 4 illustrates, both companies had launched the first version of their configurators and had begun to develop the second version. Thus, we were able to compare the KM processes between the first version, where the company did not have a structured framework, and the second version, which applied the proposed KM framework. The second version of a configuration project extends the project on the basis of version 1. An example of this is a plant configurator whose first version includes one plant type and whose second version introduces another plant type. The two versions are strongly related in terms of both the product or process domain and the organisation, which includes stakeholders and management principles, even though they are completely separate projects. Figure 4 shows the complexity of the projects (white boxes = less complexity, grey = more complexity), which is calculated based on the configurator's parameters (number of rules and attributes) and the number of integrations required to complete the configuration task (Brown et al., 2007).

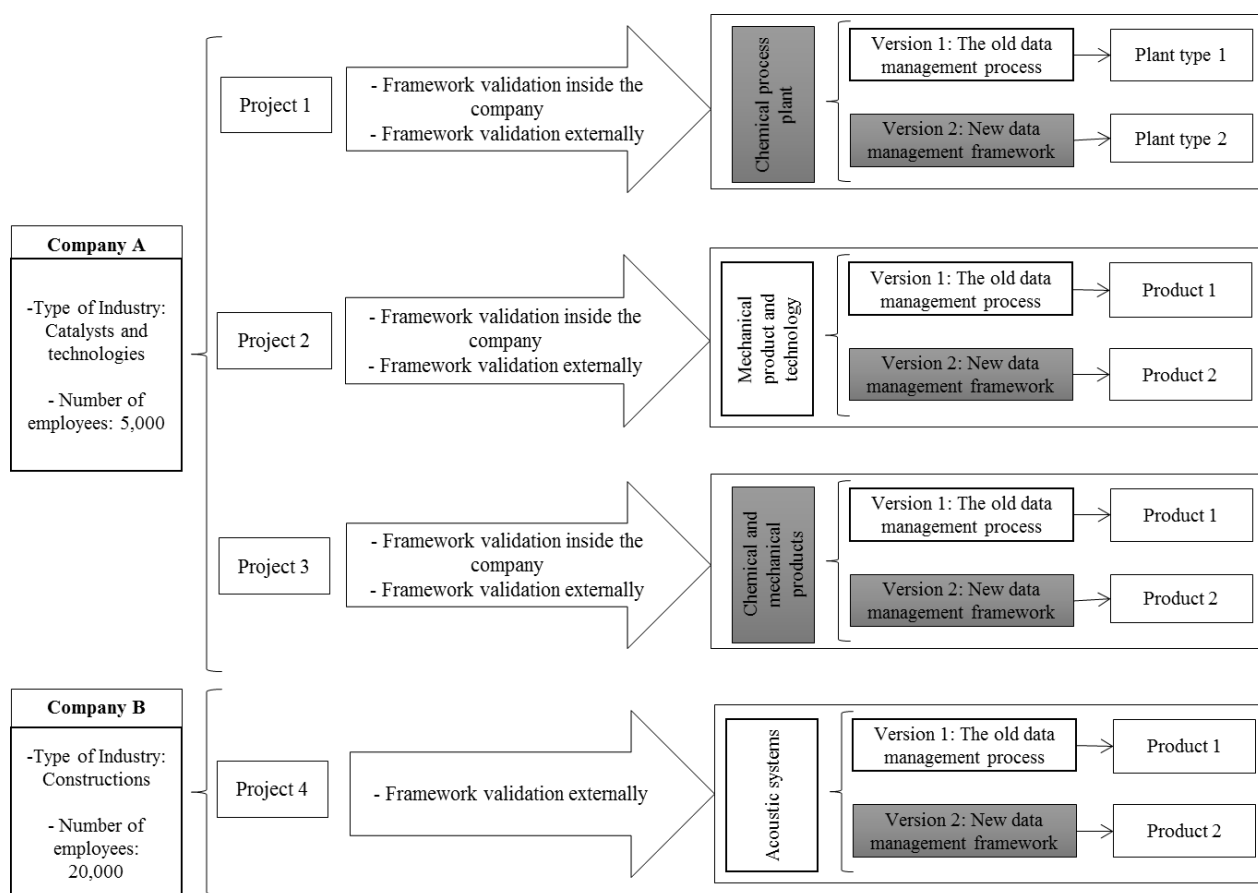


Figure 4. Selection of case studies.

5.3. Framework testing

A research team was formed in two industrial companies (Table 5). In Company A, two researchers and two configuration engineers from the company spent 50% of their time testing the framework for almost one year. In Company B, a research team comprising two researchers from the university and one employee from the company tested the framework for four months. In version 1, proper documentation was not done; however, the researchers used documents (such as Excel sheets that contained engineering calculations). For version 2, the team made documents for the undocumented knowledge (such as the knowledge elicited directly from the employees). Workshops were conducted for each project and for all the stakeholders to introduce the proposed framework and the required tools. Some of the researchers were also the practitioners at the case companies

who observed the KM process for version 1 of the project. Thus, this triangulation observation leads to valuable data, related especially to the organisational challenges of implementing the new KM framework. Finally, feedback meetings were held as semi-structured interviews to collect knowledge about the team's satisfaction with the new framework. Each meeting lasted 30 minutes and included members of the configuration teams, which included project managers, developers, end users and top managers. The purpose was to obtain an assessment of the framework from all involved stakeholders. The questions aimed to elicit a general evaluation of the framework, the benefits and challenges regarding the framework's performance and the organisational and management influences on the framework's applicability.

Table 5. Background information for modelling and implementing the configurators used in the four case studies.

Projects	Company A			Company B
	Case 1	Case 2	Case 3	Case 4
Time frame (months)	6	24	12	4
Complexity of the project	Medium	High	Medium/High	Medium
No. of employees involved	4	10	6	4
No. of workshops	3	6	4	3
No. of feedback meetings	4	15	4	5

The testing phase of this study aimed to validate the framework (1) within different projects and (2) across different companies. In order to validate the framework, (1) the application of the framework and different steps were clarified, and (2) the organisational situation and cultural influences on the applicability of the framework were analysed. The findings from the case studies are described in terms of the main benefits and obstacles that resulted from applying the suggested approach.

The following sections present the results of the framework tests within the different projects and across companies for each of the proposed steps. Analyses of the steps show the benefits and challenges associated with the framework testing and compare the two companies in terms of the new changes and use of the new techniques and tools. To demonstrate the individual steps, the application of the different steps is shown using examples from case 2.

5.4. Step 1: determining the scope of the configurators

5.4.1. Establishing the project's goal

Framework application

The recommended tools for this step include use-case diagrams, process flowcharts and MoSCoW categorisation of requirements. Use-case diagrams are used for the visualisation of requirements and goals and to communicate with domain experts (Figure 5). Flowcharts are also used to identify the current work processes (AS-IS) and determine the future processes (TO-BE) (Figure 6). A long list of functional and non-functional requirements for individuals is recognised and prioritised according to MoSCoW principles (Table 6).

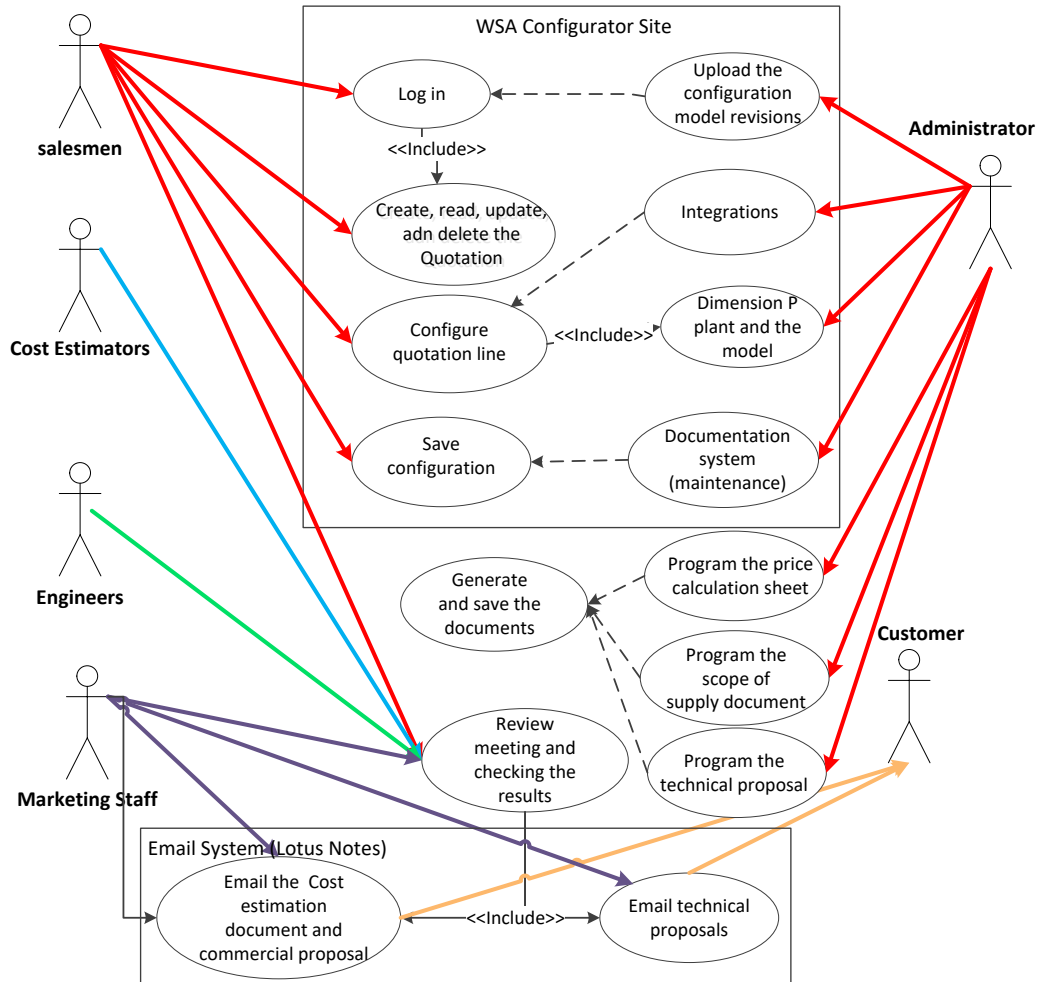


Figure 5. Example of use-case diagram for case 2.

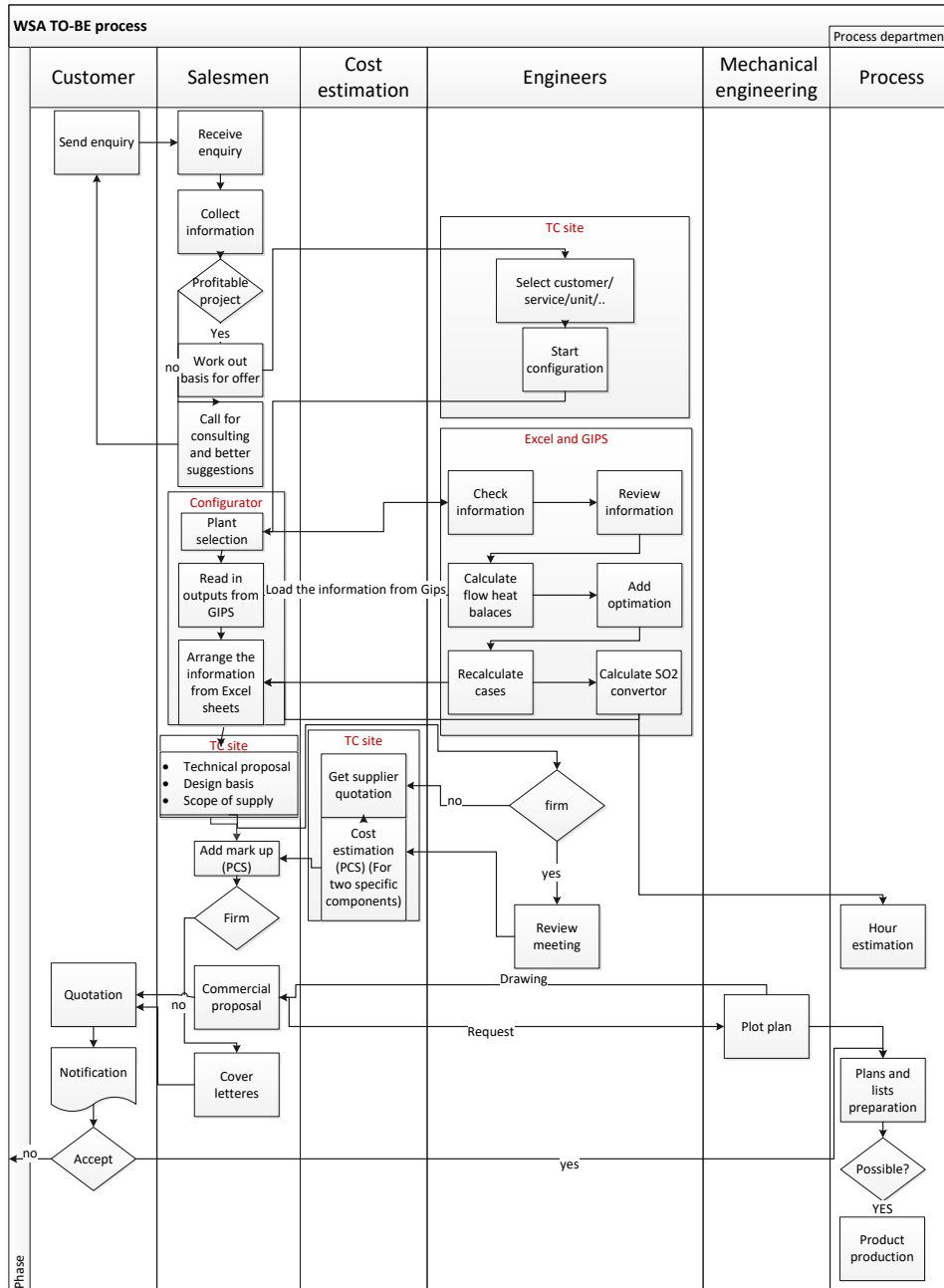


Figure 6. Example of TO-BE process flowchart for case 2.

Table 6. Examples of stakeholder-requirement prioritisation for case 2.

Interviews with the domain experts revealed that the goals of version 1 of the project were usually determined in unstructured meetings with the main stakeholders. However, the various requirements of stakeholders were not identified and clarified before starting the projects. Some of the requirements were ignored due to a lack of communication and tools, such as requests for outputs, user interfaces and additional IT automation. The configuration teams and domain experts at the case companies described the problems with the current situation, such as reworks in the configurator during development, late debugging, time consumed for development and excess or lack of knowledge in the development phase.

Cross-case comparison

Table 7 lists the tools applied in the cases before (version 1) and after applying the framework (version 2).

Table 7. Cross-case comparison of the configuration project goals.

List of requests	Must have	Should have	Could have	Want to have
Combining document snippets into full technical or commercial proposals (salespeople and cost estimators)		✓		
Loading data from the configurator into tables in the technical and commercial (sales, cost estimators and marketing group)			✓	
Price calculation, bills of material and scope of supply (all stakeholders)	✓			
Having colours for different components in user interface				✓

Awareness of project goals and of the importance of stakeholder requirements before starting the project proved to be helpful for the project team. The benefits of using the methods in version 2, as opposed to those of version 1, for the stakeholder analysis are listed below:

- Improved understanding of the stakeholders’ requirements for the system; visualising their needs established a common understanding
- Reduced time needed for the meeting with experts as a result of clear goal setting in the first step.
- Improved communication and task delegation between the resources, which reduced the consumption of time and resources.

Obstacles involved in applying this step included unfamiliarity with the tools, amount of time needed to change the current way of working and the needed time and resources for workshop preparations. In Company A, it was difficult for the team to use and see the purpose of the use-case diagrams at first, because it is difficult to change their habits and enable them to see the value of using illustration tools. However, the workshops proved to be helpful, because they provided step-by-step training for the configuration team and domain experts. Company B had already been using flowcharts in version 1 and applying the MoSCoW principles in version 2. However, Company B refused to incorporate use-case diagrams, because the managers considered it time-consuming and preferred to use flowcharts when communicating with stakeholders. The configuration team recognised the benefits as a result of discussions with different stakeholders about how to prioritise the requirements.

5.4.2. Prioritising products and processes to be included in the system

Framework application

Weighting tables are recommended for prioritising the products and functionalities to be included in a configurator (Wiegiers, 1999). In Table 8, the calculations of weighting table is assessed using the explanations from Weigers (1999).

Table 8. Example of a priority table (case 2).

Feature	Relative benefit	Relative penalty	Total value	Value %	Relative cost	Cost %	Relative risk	Risk %	Priority
Product 1	5	5	15	21.1	2	12.5	5	31.3	0.116
Product 2	9	9	27	38.0	5	31.3	2	12.5	0.209
Product 3	5	2	12	17.0	3	18.7	3	18.7	0.151
Product 4	4	1	9	12.6	4	25	5	31.3	0.049
Product 5	2	4	8	11.3	2	12.5	1	6.25	0.361
Totals	25	21	71	100	16	100	16	100	--

Based on interviews with configuration teams and domain experts from the case companies, prioritising products and functionalities becomes more critical as the complexity of the project increases. However, the significance of this step varies greatly across the cases. In Company A's version 1 projects, the products and functionalities were prioritised according to interviews with the domain experts; however, the participation of only a few domain experts meant they could ignore important parts of the process because it did not directly touch on their daily work. In version 1, Company B broke down the overall design processes for complex construction into smaller subprojects, which are mostly prioritised based on coincidences. However, the configuration engineers in both cases companies pointed out problems that arose because important features and functionalities were not recognised in the early phases of the project in version 1, and the configuration model had to be restructured.

Cross-case comparison

Table 9 lists the tools used for product and process prioritisation in different cases before (version 1) and after development of the framework (version 2).

Table 9. Cross-case comparison of the methods used for prioritising of products and functionalities to be included in the configurator.

Projects	Company A			Company B
	Case 1	Case 2	Case 3	Case 4
Version 1 (Work procedure before applying the framework)	Interviews with 1–2 domain experts	Interviews with the resources listed in the organisational chart	Interviews with department manager and one domain expert	Prioritisation based on coincidences on high level of abstraction
Version 2 (Methods used in the testing period of the framework)	Weighting tables (different modules and functionalities)			Weighting tables (overall configurator concept)

The initial resistance to adding a new tool to current work routines stemmed from the requirements for managers' time and energy. Using weighting tables and formulas to calculate the priorities of the components and functionalities required some training, and debates arose with regard to setting the values and deciding which parts and functions should be included. However, when the weighting tables were applied, a difference was noticed in the domain experts' prioritisations of products. In case 2, this was especially important because of the complexity of the overall project.

5.5. Step 2: knowledge acquisition

Framework application

This stage of the project was concerned with categorising the required knowledge and identifying knowledge sources and resources. Because neither companies had a structured way to identify knowledge

sources and resources, they were typically identified during project development, as needed; the result was many meetings and much wasted time. However, the new framework required the companies to apply categorisation tables in version 2 based on the needed configurator outputs. The categorisation table significantly increased the speed of knowledge collection, because the source of the knowledge and the person responsible for delivering the knowledge to the configuration team was identified. In addition, the management of the knowledge was improved in both companies such that various actors involved in the project could access the shared knowledge.

Cross-case comparison

Table 10 lists the work procedures applied in all cases before (version 1) and after implementation of the new framework (version 2).

Table 10. Cross-case comparison of knowledge categorising and knowledge sources.

Projects	Company A			Company B
	Case 1	Case 2	Case 3	Case 4
Version 1 (Work procedure before applying the framework)	Difficulties identifying knowledge sources and resources. Experienced employees asked to provide required knowledge. Lack of knowledge led to delays.	Configuration team was responsible for identifying experts who could provide knowledge. Lack of responsibility and access was reported.	Access to knowledge was challenging because resources were not identified beforehand.	Categorisation for some of the required knowledge was unstructured. Lack of KM.
Version 2 (Methods used in the testing period of the framework)	Categorisation tables. Knowledge sources such as ERP system, regular meetings and shared folders.	Categorisation tables. KM systems like ERP. Drawings and explanatory documents for this product were stored in the documentation system.	Categorisation tables. Knowledge sources such as ERP system, regular meetings and shared folders.	Categorisation tables. Knowledge managed in shared folders.

The categorisation tables were easily generated from the stakeholder requirements, based on the expected configurator outputs. The tables were used for categorising the required knowledge, sources and defining the resources.

In version 1, a misunderstanding that could be traced to insufficient categorisation led to unnecessary meetings. Conversely, in version 2, all the required knowledge was determined before starting the project, and resources were aware of their tasks.

5.6. *Step 3: modelling and knowledge validation*

Framework application

In this step, the knowledge was modelled and validated by domain experts to improve the system’s quality and accuracy. The suggested modelling method was PVM along with CRC cards (Figure 3). Figure 7 presents an example of PVM structure. In this step, the following achievements were fundamental for the project success:

- *Logical consistency*: the attributes, variables and constraints should be consistent when entered into the configurator.
- *Validation of the model with domain experts*: an efficient communication method was established between the configuration group and domain experts so the domain experts could validate the critical knowledge modelled in the configurator.

Because the tree structure, hierarchy, rules and attributes of the configurator model, which are written in an IT language, are not easily understandable for people outside the IT field, other methods were required for

communicating with domain experts. The PVM–CRC method was used for documentation and maintenance and for communication with domain experts. However, in case 4, manually updating and maintaining all the product models proved to be a significant task.

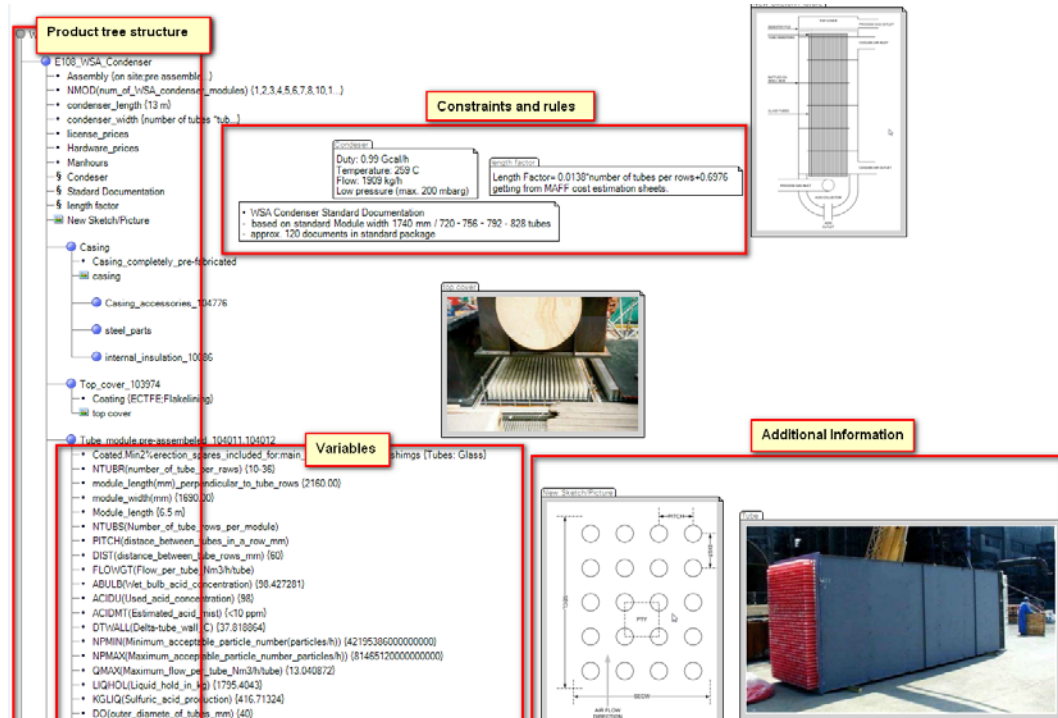


Figure 7. Example of initial PVM structure (case 2).

Cross-case comparison

Table 11 shows the methods used for this step before (version 1) and after applying the framework (version 2).

Table 11. Cross-case comparison of modelling and knowledge validation.

Projects	Company A			Company B
	Case 1	Case 2	Case 3	Case 4
Version 1 (Work procedure before applying the framework)	PVMs the only modelling tool; challenges regarding knowledge validation because the PVM did not include all the knowledge.	PVM–CRC method used but not updated; therefore, validation was performed by testing the system and using Excel spreadsheets.	PVMs the only modelling tool; challenges regarding knowledge validation because the PVM did not include all the knowledge.	No standard modelling technique; knowledge stored in various Excel spreadsheets; validation was performed by testing and using the system.
Version 2 (Methods used in the testing period of the framework)	PVM and CRC cards, additional tables and an agile and efficient documentation system.	Agile documentation system automatically generated PVM and CRC and allowed domain experts to validate the knowledge.		PVM for system validation and communication with stakeholders.

A comparison of the two versions reveals a significant difference in system quality that resulted from the validation by domain experts due to the visual representation. In version 1, quality was reported as an issue, errors resulted in infeasible configurations, and the configuration team faced difficulties finding the source of the errors. This highlights the importance of incorporating knowledge validation into different steps of configuration projects. In Company A, an agile system was developed to structure the knowledge included in

the configurator and generate a PVM structure, which allowed domain experts to validate the knowledge. Two main benefits were gained from this phase:

- This validation phase saved time and resources for future testing to find possible minor and major errors.
- The quality and reliability of and confidence in the system improved as stakeholders took control of knowledge validation.

5.7. Step 4: documentation and maintenance of the knowledge

Framework application

This step involved the documentation and maintenance of the knowledge to ensure that the configurator was up-to-date and could be maintained. Both companies neglected documentation because of heavy workloads. In Company B, documents representing the knowledge contained in the system were spread across the firm. Company A implemented PVM in all cases and used CRC cards in case 2. CRC cards were used only in case 2 due to the complexity of the project. Even though Company A used a formal modelling technique, it is difficult to know whether the PVM and CRC cards were up-to-date and aligned with the knowledge inside the configurator and with product changes, because updates were neglected. As noted, Company A developed an agile documentation system in version 2 to represent the knowledge inside the configurator in the form of a PVM and class diagrams, as shown in Figure 8 (Shafiee et al., 2017). Conversely, Company B delegated the responsibility for performing updates to the different classes of the PVM.

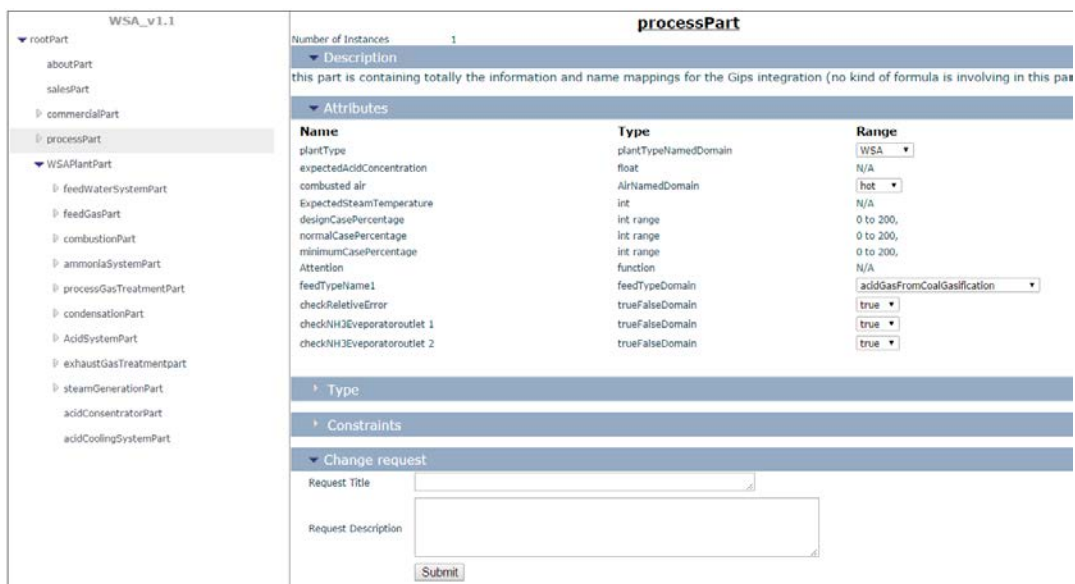


Figure 8. Automatic generation of PVM and CRC structure in the developed documentation system.

Cross-case comparison

Table 12 lists the work procedures applied to the cases before (version 1) and after using the new framework (version 2).

Table 12. Cross-case comparison of knowledge documentation and maintenance.

Projects	Company A			Company B
	Case 1	Case 2	Case 3	Case 4
Version 1 (Work procedure before applying the framework)	PVM, spreadsheet documents gathered from stakeholders.	PVM, spreadsheet documents, documentation in internal team sites.	PVM, spreadsheet documents, documentation in team sites.	Spreadsheet documents.
Version 2 (Methods used in the testing period of the framework)	An agile documentation system, updated automatically based on the configuration model.			PVMs and CRC cards, a manual but structured system for updates.

A comparison of the cases using automatic documentation with those using completely manual processes highlights the importance of using an automatic documentation system. Company B began to benefit even from manual documentation, because in version 1, their configurator was out of use since they were not updating and maintaining knowledge inside the configurator.

5.8 Summary of the cross-case analysis

The cross-case comparisons show that the framework affects the companies differently. Company A took the risk of experimenting with new tools and techniques, and the employees reported that many benefits and challenges resulted from employing them. Company B's management board achieved efficiency by keeping up with routine work while making minor changes. In comparison, Company A's management board aimed to improve the current work flow by accepting the changes and modifications recommended by researchers.

The differences in the way the framework has effected companies A and B may derive from the different cultures of Companies A and B. Studies have found that results- and job-oriented company cultures have positive effects on employee decisions during the KM process, whereas a tightly controlled culture has negative effects (Chang and Lin, 2015). Although every organisation has its own identity and language, the aim is to find a common basis and help companies define their own KM framework with minor changes (European Committee for Standardization, 2004). KM is a difficult task because knowledge sharing and transfer, and the consequent realisation of the full value of the organisation's knowledge resources, require changes in organisational culture (Firestone and McElroy, 2003). The perception of the involved people about KM process in the configuration projects changed as the project related information was precisely communicated, they were trained and experimented the benefits of the new approach. Thus, the application of the framework could have initiated a small change in the organizational culture. The companies considered this framework as the efficient KM process for the future projects and decided to continue to use it. However, we don't know whether organizational culture has changed enough to continue to keep the new approach.

6. Discussion

In the first part of the discussion section, we discuss the framework development, which was based on the literature and experiences in one case company. In the second, we discuss the framework validation, which took place in the two case companies.

6.1 Integration of the proposed framework with the existing literature

By analysing the extant literature, our study has shown that implementing the KM process in configuration projects is challenging because of the amount and complexity of the knowledge involved (Jinsong et al., 2005).

The available frameworks for KM in IT projects do not meet the KM challenges for configuration projects, because the knowledge in configuration projects is even more complex and vast and often lies outside the configuration team's expertise (Basili and Weiss, 1984). There is also a strong need for communication between domain experts and the configuration team to validate the knowledge in configuration projects (Shafiee et al., 2017). Additionally, the KM tools and techniques needed for managing the knowledge in configuration projects are specific to configurators (Hvam et al., 2008). Without a clear KM framework, configurators become complicated and unstructured (Forza and Salvador, 2007).

There is a paucity of research on developing a comprehensive KM framework for configuration projects even while many of the recognised critical challenges in configuration projects are related to KM (Jinsong et al., 2005; Lech, 2014; Sabin and Weigel, 1998). Moreover, we maintain that modern configurators are IT based and that some IT-oriented KM frameworks can inspire KM frameworks tailored to configuration projects (Basili and Weiss, 1984; Reich et al., 2012; Rodriguez and Al-Ashaab, 2005). The research on configurators has investigated KM steps during different phases of configurator development and the specific tools and methods (Forza and Salvador, 2002b; Haug, 2010; Hvam et al., 2008). The present study therefore took the opportunity to exploit these potentially useful integrations of different, though closely related, research streams.

The framework proposed here integrates the literature on configurators with the literature on general KM frameworks and the literature on KM frameworks for IT projects. The suggested framework contains the same main phases as suggested in literature on general and IT projects KM frameworks but it also adds specific tools and methods needed for knowledge management in configuration projects such as the critical need for modelling of the products knowledge. In building this bridge, the research on configurators takes advantage of what scholars have discovered about KM in general and IT-oriented KM in particular. On the other hand, the literature on KM has a new case to address, namely KM in projects aimed at implementing configurators. This case is particularly intriguing, because it suffers from serious problems of knowledge validation resulting from issues related to communication and the complexity, specificity and vastness of the knowledge involved.

The proposed framework includes four KM steps for configurator projects based on KM frameworks presented in the extant literature on IT projects (e.g. Lech, 2014) and on KM steps outlined in the literature on configuration projects (e.g. Forza and Salvador, 2007). The first step involves determining the scope of the project while suggesting tools for analysing stakeholders and prioritising different products and processes (Basili and Weiss, 1984; Shafiee et al., 2014). The knowledge acquisition step discusses how to manage all the sources and resources in order to categorise inputs and outputs (Nonaka, 2008; Tiihonen et al., 1996). The third step analyses different product modelling techniques for better communication across the supply chain and for knowledge validation (Aldanondo et al., 2000; Chao and Chen, 2001; Hvam, 2001; Hvam et al., 2008; Jinsong et al., 2005; Magro and Torasso, 2003; Tseng, H.E. et al., 2005; Yang et al., 2009). The last step considers documentation and maintenance, which help configurators remain stable and up-to-date (Haug and Hvam, 2007; Shafiee et al., 2017). Departing from the existing literature on configurators, the proposed framework integrates the proposed steps into a specific sequence in order to fulfil the need for a standard process in managing knowledge. It also makes knowledge validation possible by establishing communication between domain experts and the configuration team.

6.2 Applicability of the framework

The suggested framework was tested on four configuration projects in two industrial ETO companies. The configuration projects were engineering projects in which vast, complicated knowledge had to be managed. The proposed framework helped the companies address the main challenges of KM in configuration projects.

The scope of the projects was kept limited (whereas before they were continuously expanded); this limitation supported collaboration with domain experts, thus reducing the difficulties associated with accessing their knowledge. Continuous validation of the knowledge was enabled by modelling the knowledge. Consequently, the companies witnessed a reduction of the time and resources needed for scoping, developing, implementing and documenting their configuration projects. The proposed KM framework aligned all members of the configuration project team, from the IT team to domain experts, thus leading to a better configurator. In the end, the framework standardised the knowledge acquisition process, using simple tools to align the entire configuration team.

The configuration teams involved in the development and testing of the framework expressed a willingness to use the framework in future projects to save both time and resources. Domain experts at the company also appreciated their involvement in knowledge verification. These results indicate both the effectiveness of the framework and its positive involvement effects on the people engaged in the configuration project.

The main obstacle for the configuration team's use of the framework was their lack of familiarity with the suggested tools. An introduction of the tools in workshops significantly reduced their resistance to the framework. Using the framework and suggested tools did not introduce additional burdens or costs, and the training for configuration engineers and domain experts was carried out in a short amount of time (two weeks maximum).

7. Conclusion

The challenges of KM and the ability of the organisations to handle knowledge have been thoroughly considered in both research and practice. The present study proposes a KM framework for projects aimed at the implementation of configurators. The framework includes four steps: (1) determining the scope of the project, (2) acquiring knowledge, (3) modelling and validating knowledge and (4) documenting and maintaining the system. The execution of each step is supported with relevant tools. The proposed framework was tested on multiple projects and companies. These tests demonstrate both its applicability in different industrial settings and its potential to enhance the quality and speed of the implementation of configurators.

This paper contributes to the existing literature by proposing a KM framework for configuration projects, developed on the basis of the literature on general KM frameworks (e.g. European Committee for Standardisation, 2004; Liebowitz and Megbolugbe, 2003), KM frameworks in IT projects (e.g. Lech, 2014; McGinnis and Huang, 2007) and available KM tools and steps for configuration projects (e.g. Forza and Salvador, 2007; Tiihonen et al., 2014; Yang et al., 2009). The steps and tools are drawn from the general KM frameworks and the frameworks for IT projects as well as the proposed solutions for configuration projects. The results fill the research gap by connecting and sequencing these available tools and methods in the proposed framework and validating the framework in multiple case studies.

The proposed framework is a powerful tool for reducing the scope and complexity of configuration projects, making the KM in such projects more manageable. In this way, managers can more easily collaborate with the people involved, reducing the time and resources required. Configuration projects can be less risky and the small deliverables have a shorter payback time.

To validate the framework, versions 1 and 2 of the same project were compared to eliminate the threat of uncontrolled influencing factors. The companies used the proposed framework in version 2 of the project and compared the results with version 1, when no KM framework was available. However, some threats remained while the configurator experts gained knowledge about the product in version 1 and while the product experts became familiar with the configurator. Because the framework compared two versions of the same project, the

team were familiar with the stakeholders and with the product and process in general, and the team was aware of all the available sources and resources. Although such familiarity could be considered a risk in testing the framework by clarifying parts of the KM process, it was still beneficial for the researchers to compare the same situations and observe the benefits. Further assessment of the framework with a low level of initial knowledge by the configurator team and domain experts could therefore strengthen our confidence in the framework's capabilities that limit the threat of uncontrolled influencing factors.

The use of cases allowed us to assess – in depth, in detail and in real-world contexts – the proposed framework's effectiveness. However, we were able to apply the framework only in a limited number of projects and companies, and this limits the generalisability of our results. Furthermore, we have studied configuration projects in companies making physical products with a stable product architecture. The study does not include KM on configurators for non-physical products or services. The study also does not consider KM on configurators for the new physical products, where the product architecture is not stable. Moreover, one of the limitations in testing the framework is to compare two versions of the same project, which was decided in order to control all the dependent factors in terms of both the product or process domain and the organisation, even though they are completely separate projects. Another limitation of the test is that the observation did not continue for long time to check the continued utilisation of the framework in the future. Even though the companies announced that they will continue to use the framework, only a longitudinal observation could inform us about the role of organizational culture in absorbing the proposed KM framework in configuration projects. The further testing is needed to verify the successful changes in the organizational culture. Therefore, the successful implementation of the framework during the present research does not mean the full adaptation and change at the companies. The ability of the framework to cope with highly engineered, complex products in ETO companies indicates that it could also be used in configuration projects of less complexity. However, the necessity of applying such a structured framework in smaller projects is questionable and needs further testing. Future research should test the framework in various industrial settings and identify more efficient and simpler tools and techniques for use in each step of the framework.

References

- Aldanondo, M., Rouge, S. and Ve'ron, M. (2000), "Expert configurator for concurrent engineering : Caméléon on software and model", *Journal of Intelligent Manufacturing*, Vol. 11 No. 2, pp. 127–134.
- Anzanello, M.J. and Fogliatto, F.S. (2011), "Selecting the best clustering variables for grouping mass-customized products involving workers learning", *International Journal of Production Economics*, Vol. 130 No. 2, pp. 268–276.
- Ardissono, L., Felfernig, A., Friedrich, G., Goy, A., Jannach, D., Petrone, G., Schafer, R., et al. (2003), "A Framework for the Development of Personalized, Distributed Web-Based Configuration Systems", *AI Magazine*, Vol. 24 No. 3, p. 93.
- Aurum, A., Daneshgar, F. and Ward, J. (2008), "Investigating Knowledge Management practices in software development organisations - An Australian experience", *Information and Software Technology*, Vol. 50 No. 6, pp. 511–533.
- Barker, V.E., O'Connor, D.E., Bachant, J. and Soloway, E. (1989), "Expert systems for configuration at Digital: XCON and beyond", *Communications of the ACM*, Vol. 32 No. 3, pp. 298–318.
- Basili, V.R. and Weiss, D.M. (1984), "A Methodology for Collecting Valid Software Engineering Data", *IEEE Transactions on Software Engineering*, No. 6, pp. 728–738.

- Beck, K. and Cunningham, W. (1989), "A laboratory for teaching object oriented thinking", *ACM SIGPLAN Notices*, Vol. 24 No. 10, pp. 1–6.
- Bittner, K. (2002), *Use Case Modeling*, Addison-Wesley Longman Publishing Co., Inc.
- Blecker, T., Abdelkafi, N., Kreutler, G. and Friedrich, G. (2004), "Product configuration systems: state of the art, conceptualization and extensions", *Proceedings of the Eight Maghrebian Conference on Software Engineering (MCSEAI 2004)*, pp. 25–36.
- Bose, R. and Sugumaran, V. (2003), "Application of knowledge management technology in customer relationship management", *Knowledge and Process Management*, Wiley Online Library, Vol. 10 No. 1, pp. 3–17.
- Briand, L.C. (2003), "Software documentation: how much is enough?", *Seventh European Conference on Software Maintenance and Reengineering, 2003. Proceedings.*, pp. 3–5.
- Brown, A.B., Keller, A. and Hellerstein, J.L. (2007), "A configuration complexity model and its application to a change management system", *IEEE Transactions on Network and Service Management*, Vol. 4 No. 1, pp. 13–27.
- BSI – British Standards Institute. (2001), *PAS 2001- Knowledge Management*.
- Byrd, T. (1992), "Implementation and use of expert systems in organizations: perceptions of knowledge engineers", *Journal of Management Information Systems*, Vol. 8 No. 4, pp. 97–116.
- Chang, C.L. and Lin, T.-C. (2015), "The role of organizational culture in the knowledge management process", *Journal of Knowledge Management*, Vol. 19 No. 3, pp. 433–455.
- Chao, P.Y. and Chen, T. Te. (2001), "Analysis of assembly through product configuration", *Computers in Industry*, Vol. 44 No. 2, pp. 189–203.
- Cockburn, A. (2002), "Learning from agile software development–part one", *CrossTalk*, pp. 10–14.
- Compton, P. and Jansen, R. (1990), "A philosophical basis for knowledge acquisition", *Knowledge Acquisition*, Vol. 2 No. 3, pp. 241–258.
- Coram, M. and Bohner, S. (2005), "The Impact of Agile Methods on Software Project Management", *12th IEEE International Conference and Workshops on the Engineering of Computer-Based Systems (ECBS'05)*, pp. 363–370.
- Creswell, J.W. and Clark, V.L.P. (2011), *Designing and Conducting Mixed Methods Research*, Sage Publications, Inc., Thousand Oaks, CA.
- Duffy, A. and Andreasen, M. (1995), "Enhancing the evolution of design science", *Proceedings of ICED 95*, Praha.
- Ebert, C. (1997), "Dealing with nonfunctional requirements in large software systems", *Annals of Software Engineering*, Vol. 3 No. 1, pp. 367–395.
- Eisenhardt, K. (1989), "Building theories from case study", *Academy of Management Review*, Vol. 14 No. 4, pp. 532–550.
- European Committee for Standardization. (2004), *European Guide to Good Practice in Knowledge Management - Part 1: Knowledge Management Framework, CEN Workshop Agreement*.

- Felfernig, A., Hotz, L., Bagley, C. and Tiihonen, J. (2014), *Knowledge-Based Configuration From Research to Business Cases*, Morgan Kaufman, available at:<https://doi.org/10.1016/B978-0-12-415817-7.00029-3>.
- Felfernig, A., Reiterer, S., Reinfrank, F., Ninaus, G. and Jeran, M. (2014), “Conflict Detection and Diagnosis in Configuration”, in Felfernig, A., Hotz, L., Bagley, C. and Tiihonen, J. (Eds.), *Knowledge-Based Configuration: From Research to Business Cases*, Morgan Kaufman, pp. 73–87.
- Firestone, J.M. and McElroy, M.W. (2003), *Key Issues in Knowledge Management*, Butterworth-Heinemann, Amsterdam.
- Fleischanderl, G., Friedrich, G.E., Haselböck, A., Schreiner, H. and Stumptner, M. (1998), “Configuring large systems using generative constraint satisfaction”, *IEEE Intelligent Systems*, Vol. 13 No. 4, pp. 59–68.
- Forsythe, D.E. and Buchanan, B.G. (1989), “Knowledge acquisition for expert systems: some pitfalls and suggestions”, *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 19 No. 3, pp. 435–442.
- Forza, C. and Salvador, F. (2002a), “Managing for variety in the order acquisition and fulfilment process: The contribution of product configuration systems”, *International Journal of Production Economics*, Vol. 76 No. 1, pp. 87–98.
- Forza, C. and Salvador, F. (2002b), “Product configuration and inter-firm coordination: An innovative solution from a small manufacturing enterprise”, *Computers in Industry*, Vol. 49 No. 1, pp. 37–46.
- Forza, C. and Salvador, F. (2007), *Product Information Management for Mass Customization: Connecting Customer, Front-Office and Back-Office for Fast and Efficient Customization*, Palgrave Macmillan, New York.
- Friedrich, G., Jannach, D., Stumptner, M. and Zanker, M. (2014), “Knowledge Engineering for Configuration Systems”, in Felfernig, A., Hotz, L., Bagley, C. and Tiihonen, J. (Eds.), *Knowledge-Based Configuration: From Research to Business Cases*, Morgan Kaufmann, pp. 139–155.
- Hansen, C.L., Mortensen, N.H., Hvam, L. and Harlou, U. (2012), “Calculation of Complexity Costs – An Approach for Rationalizing a Product Program”, *Proceedings of NordDesign Conference 2012*, p. 8.
- Haug, A. (2010), “Managing diagrammatic models with different perspectives on product information”, *Journal of Intelligent Manufacturing*, Vol. 21 No. 6, pp. 811–822.
- Haug, A. and Hvam, L. (2007), “The modelling techniques of a documentation system that supports the development and maintenance of product configuration systems”, *International Journal of Mass Customisation*, Vol. 2 No. 1–2, pp. 1–18.
- Haug, A. and Hvam, L. (2008), *Representation of Industrial Knowledge-as a Basis for Developing and Maintaining Product Configurators*, Technical University of Denmark.
- Haug, A., Hvam, L. and Mortensen, N. (2009), “Implementation of conceptual product models into configurators: From months to minutes”, *5th World Conference on Mass Customization and Personalization (Mcpc)*, pp. 1–23.
- Heisig, P. (2009), “Harmonisation of knowledge management – comparing 160 KM frameworks around the globe”, *Journal of Knowledge Management*, Vol. 13 No. 4, pp. 4–31.
- Hirsch, M. (2002), *Making RUP Agile, OOPSLA 2002 Practitioners Reports on - OOPSLA '02*, ACM Press,

New York, New York, USA.

- Hvam, L. (2001), "A procedure for the application of product modelling", *International Journal of Production Research*, Vol. 39 No. 5, pp. 873–885.
- Hvam, L., Mortensen, N.H. and Riis, J. (2008), *Product Customization*, Springer, Berlin Heidelberg.
- Jiao, J. and Chen, C. (2006), "Customer requirement management in product development: a review of research issues", *Concurrent Engineering*, Vol. 14 No. 3, pp. 173–185.
- Jinsong, Z., Qifu, W., Li, W. and Yifang, Z. (2005), "Configuration-oriented product modelling and knowledge management for made-to-order manufacturing enterprises", *International Journal of Advanced Manufacturing Technology*, Vol. 25 No. 1–2, pp. 41–52.
- Johnson, R.B., Onwuegbuzie, A.J. and Turner, L.A. (2007), "Toward a Definition of Mixed Methods Research", *Journal of Mixed Methods Research*, Vol. 1 No. 2, pp. 112–133.
- Kaplan, B.B.B. and Duchon, D. (1988), "Combining Qualitative and Quantitative Methods in Information Systems Research: A Case Study.", *MIS Quarterly*, Vol. 12 No. 4, pp. 571–586.
- Kaufman, L. and Rousseeuw, P.J. (2009), *Finding Groups in Data: An Introduction to Cluster Analysis*, Vol. 344, John Wiley & Sons.
- Kelleher, D. and Levene, S. (2001), *Knowledge Management: A Guide to Good Practice*, British Standards Institution London.
- Komi-Sirvio, S., Mantyniemi, A. and Seppanen, V. (2002), "Toward a practical solution for capturing knowledge for software projects", *IEEE Software*, IEEE, Vol. 19 No. 3, pp. 60–62.
- Kruchten, P. (2007), *The Rational Unified Process, an Introduction, Physician Sportsmedicine*, 3rd ed., Addison-Wesley Professional.
- Kuczaj, T. and Komi-Sirviö, S. (2001), "Utilising knowledge management in software process improvement-The creation of a knowledge management process model", *International Conference of Concurrent Enterprising*, pp. 241–250.
- Lech, P. (2014), "Managing knowledge in IT projects: a framework for enterprise system implementation", *Journal of Knowledge Management*, Emerald Group Publishing Limited, Vol. 18 No. 3, pp. 551–573.
- Liebowitz, J. and Megbolugbe, I. (2003), "A set of frameworks to aid the project manager in conceptualizing and implementing knowledge management initiatives", *International Journal of Project Management*, Vol. 21 No. 3, pp. 189–198.
- Lim, S., Damian, D. and Finkelstein, A. (2011), "StakeSource2. 0: using social networks of stakeholders to identify and prioritise requirements", *Proceedings of the 33rd International Conference on Software Engineering. ACM*, pp. 1022–1024.
- Magro, D. and Torasso, P. (2003), "Decomposition strategies for configuration problems", *Ai Edam*, Vol. 17 No. 1, pp. 51–73.
- McCutcheon, D.M. and Meredith, J.R. (1993), "Conducting case study research in operations management", *Journal of Operations Management*, Vol. 11 No. 3, pp. 239–256.
- McGinnis, T.C. and Huang, Z. (2007), "Rethinking ERP success: A new perspective from knowledge

- management and continuous improvement”, *Information and Management*, Vol. 44 No. 7, pp. 626–634.
- Mortensen, N.H., Harlou, U., Hvam, L. and Haug, A. (2008), “Improving decision making in the early phases of configuration projects”, *International Journal of Industrial Engineering*, Vol. 15 No. 2, pp. 185–19.
- Männistö, T., Soininen, T. and Sulonen, R. (2001), “Product Configuration View to Software Product Families”, *10th International Workshop on Software Configuration Management (SCM-10)*, Toronto, pp. 14–15.
- Nellore, R., Söderquist, K. and Eriksson, K. (1999), “A specification model for product development”, *European Management Journal*, Vol. 17 No. 1, pp. 50–63.
- Nonaka, I. (1994), “A dynamic theory of organizational knowledge creation”, *Organization Science*, Vol. 5 No. 1, pp. 14–37.
- Nonaka, I. (1991). (2008), *The Knowledge Creating Company*, Harvard Business Review Press.
- Reich, B.H., Gemino, A. and Sauer, C. (2012), “Knowledge management and project-based knowledge in it projects: A model and preliminary empirical results”, *International Journal of Project Management*, Elsevier Ltd, Vol. 30 No. 6, pp. 663–674.
- Rodriguez, K. and Al-Ashaab, A. (2005), “Knowledge web-based system architecture for collaborative product development”, *Computers in Industry*, Vol. 56 No. 1, pp. 125–140.
- Rubenstein-Montano, B., Liebowitz, J., Buchwalter, J., McCaw, D., Newman, B., Rebeck, K. and The Knowledge Management Methodology Team. (2001), “A systems thinking framework for knowledge management”, *Decision Support Systems*, Vol. 31 No. 1, pp. 5–16.
- Sabin, D. and Weigel, R. (1998), “Product configuration frameworks-a survey”, *IEEE Intelligent Systems and Their Applications*, Vol. 13 No. 4, pp. 42–49.
- Selic, B. (2009), “Agile Documentation, Anyone?”, *IEEE Software*, Vol. 26 No. 6.
- Shadbolt, N., O’hara, K. and Crow, L. (1999), “The experimental evaluation of knowledge acquisition techniques and methods: history, problems and new directions”, *International Journal of Human-Computer Studies*, Vol. 51 No. 4, pp. 729–755.
- Shafiee, S., Hvam, L. and Bonev, M. (2014), “Scoping a product configuration project for engineer-to-order companies”, *International Journal of Industrial Engineering and Management*, Vol. 5 No. 4, pp. 207–220.
- Shafiee, S., Hvam, L., Haug, A., Dam, M. and Kristjansdottir, K. (2017), “The documentation of product configuration systems: A framework and an IT solution”, *Advanced Engineering Informatics*, Vol. 32, pp. 163–175.
- Stelzer, D. and Mellis, W. (1998), “Success factors of organizational change in software process improvement”, *Software Process: Improvement and Practice*, Vol. 4 No. 4, pp. 227–250.
- Studer, R., Benjaminsc, R. and Fensela, D. (1998), “Knowledge engineering: Principles and methods”, *Data & Knowledge Engineering*, Vol. 25 No. 1–2, pp. 161–197.
- Tiihonen, J., Felfernig, A. and Mandl, M. (2014), “Knowledge-Based Configuration From Research to Business Cases”, *Knowledge-Based Configuration*, Morgan Kaufman, available

at:<https://doi.org/10.1016/B978-0-12-415817-7.00029-3>.

- Tiihonen, J., Soininen, T., Männistö, T. and Sulonen, R. (1996), “State-of-the-Practice in Product Configuration — A Survey of 10 Cases in the Finnish Industry”, *Knowledge Intensive CAD*, Vol. 1, pp. 95–114.
- Tsai, C.Y., Tsai, C.Y. and Huang, P.W. (2009), “An association clustering algorithm for can-order policies in the joint replenishment problem”, *International Journal of Production Economics*, Vol. 117 No. 1, pp. 30–41.
- Tseng, H.E., Chang, C.C. and Chang, S.H. (2005), “Applying case-based reasoning for product configuration in mass customization environments”, *Expert Systems with Applications*, Vol. 29 No. 4, pp. 913–925.
- Turley, F. (2007), “An Introduction to PRINCE2”, *JISC Infonet - Project Management Methodologies*, Management Plaza.
- Ulz, T., Schwarz, M., Felfernig, A., Haas, S., Shehadeh, A., Reiterer, S. and Stettinger, M. (2016), “Human computation for constraint-based recommenders”, *Journal of Intelligent Information Systems*, Vol. 49 No. 1, pp. 37–57.
- Van de Ven, A.H.A.H. (1989), “Nothing is quite so practical as a good theory”, *Academy of Management Review*, Vol. 14 No. 4, pp. 486–489.
- Walz, D.B., Elam, J.J. and Curtis, B. (1993), “Inside a software design team: knowledge acquisition, sharing, and integration”, *Communications of the ACM*, Vol. 36 No. 10, pp. 63–77.
- Whitney, K.M. and Daniels, C.B. (2013), “The root cause of failure in complex IT projects: Complexity itself”, *Procedia Computer Science*, Vol. 20, pp. 325–330.
- Wieggers, K. (1999), “First things first: prioritizing requirements”, *Software Development*, Vol. 7 No. 9, pp. 48–53.
- Woodward, B. (1990), “Knowledge acquisition at the front end: defining the domain”, *Knowledge Acquisition*, Vol. 2 No. 1, pp. 73–94.
- Yang, D., Miao, R., Wu, H. and Zhou, Y. (2009), “Product configuration knowledge modeling using ontology web language”, *Expert Systems with Applications*, Elsevier Ltd, Vol. 36 No. 3, pp. 4399–4411.
- Yin, R.K. (2013), *Case Study Research: Design and Methods*, Sage publications.
- Zimmerman, E. (2003), *Design Research: Methods and Perspectives*, The MIT press.

¹ The experts who provide domain knowledge of the process of performing the task and the data content, as well as quality assurance, verification support (Barker et al., 1989).

² The team working on configuration projects include knowledge engineers, modellers, developers and project managers (Hvam et al., 2008).

³ By building a configurator, engineers design the engineering model the rules to construct the product and define the methods of work (a so-called product model). Hence, the knowledge can be expressed explicitly and incorporated into a configurator, which can subsequently be used by the company’s sales staff to configure a product in collaboration with the customer (Hvam et al., 2008).

⁴ Knowledge engineers interpret and organise knowledge from domain experts. The expert system technologist performs the knowledge acquisition, knowledge representation, and knowledge base development and testing tasks (Barker et al., 1989).