K-Model – Structured Design of Configuration Models

Dr. Axel Brinkop[1] and **Dr. Thorsten Krebs**[2] and **Hartmut Schlee**[3]

Abstract. The purpose of this paper is to introduce the novel knowledge acquisition methodology K-Model. We describe the methodology itself and how it was applied within a project for creating a prototype configuration application at J. Schmalz GmbH. K-Model is supporting both the formalism of designing configuration models on a conceptual level as well as the method to actually implement these models. Based on the experience that configuration knowledge is tacit and distributed within the heads of several product experts', the methodology is focusing on cross-department communication about future goals of the configuration application. The visualization facilities of standard mind maps help them to achieve a common agreement and to focus on the product domain rather than on knowledge representation formalisms. The methodology was successfully used in the project to set up a configuration prototype for complex products in the area of vacuum technology.

1 MOTIVATION

A major challenge in realizing knowledge-based configuration systems is the acquisition and formalization of configuration knowledge. But knowledge acquisition is notoriously a very expensive process. Actually, most of the complexity of solving a configuration problem is said to lie in representing the domain knowledge [2].

One of the main reasons for the complexity of knowledge acquisition is that two types of expertise are required: knowledge about the product domain and dealing with the representation language that is used for modeling the product domain. But very few persons are both domain expert and knowledge modeling expert. Thus, in practice the modeling task is carried out by one of the two engineers, probably being assisted by the other one.

In this paper we introduce the knowledge acquisition methodology K-Model. This methodology helps the knowledge engineer to focus on the product domain rather than on knowledge representation formalisms. K-Model consists of formalism for designing the contents of a configuration model and a method for acquiring configuration knowledge and actually creating the contents. The formalism describes the types of knowledge required for creating a configuration system in a way that is well-founded on semantics but at the same time easily understandable for domain experts like product managers or sales engineers. We use mind map structures to visualize the relevant types of content, i.e. classification data, sales questions and the sales bill of materials together with their interdependencies. We further use MS Excel to define data about available components according to the definition of classification data as well as tabular dependencies, i.e. variant tables. The method describes a process consisting of workshops, reviews and "offline" refinement steps in which the relevant configuration knowledge for the product domain is acquired and actually implemented within a configuration model.

J. Schmalz GmbH is a family-run company situated in Glatten, Germany. Schmalz is a leading global supplier of vacuum technology in the fields of automation, handling and clamping technology with an export quota of 50%, 15 subsidiaries abroad, and sales partners within 40 countries all over the world. When it comes to automated production processes, Schmalz offers a wide range of individual vacuum components and related services. Different vacuum systems can be operated in different environments, e.g. vacuum gripper systems are ready-to-connect modular systems for usage in robotic applications, vacuum handling systems are operated manually and ease the handling of work pieces and vacuum clamping systems offer short set-up times for CNC machining centers.

Schmalz is a very innovative company with permanent readiness to implement and accept changes. A current change of the company is driven by investing in a quote generation process including configuration of vacuum products. The goal of this change is to ease generating technically correct solutions for complex configuration problems together with high quality quote documents. The K-Model methodology was used to set up a prototype quote generation application for complex configurable products from the families of vacuum handling systems and vacuum clamping systems.

The remainder of this paper is organized as follows. In Chapter 2 we describe the knowledge acquisition methodology K-Model, i.e. both the formalism and the method, in more detail and give mind map representation examples. Chapter 3 describes the application of K-Model within a real-life customer project, i.e. both applying the formalism and method of K-Model for acquiring a configuration model as well as implementing the acquired contents. Chapter 4 concludes this paper with the major findings and in Chapter 5 we present related work.

[1] Brinkop Consulting, Oberschlettenbach, Germany, email: Brinkop@brinkop-consulting.com
[2] encoway GmbH, Bremen, Germany, email: Krebs@encoway.de
[3] J. Schmalz GmbH, Glatten, Germany, email: Hartmut.Schlee@schmalz.de

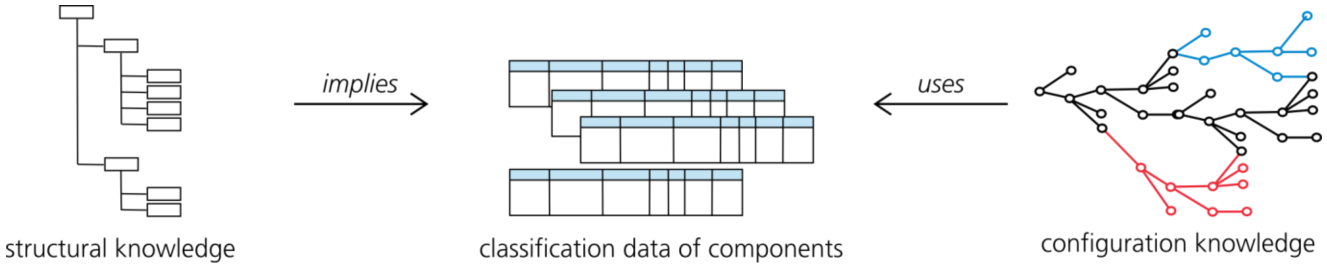

Figure 1: The relations between structural knowledge, classification data and configuration knowledge.

2 K-MODEL

K-Model is a methodology developed by Brinkop Consulting supporting both the formalism of designing configuration models as well as the method to actually develop these models ("K" = "Konfiguration", German for configuration). It is not designed for any specific software but it is based on the approach to separately represent structural knowledge, configuration knowledge and available components.

The structural knowledge is a conceptual-level representation of the internal structure of the product to be configured; i.e. the product itself together with the parts from which it is assembled. The options for each part are defined in the available components themselves. Structural knowledge and available components are strongly related, though. The structural knowledge is expressed as a hierarchy of classes, each class defined by a set of attributes. Inheritance of attributes is assumed. Every available component is an instance of a class with given attribute values. The configuration knowledge represents knowledge about dependencies and methods to determine components. Figure 1 illustrates the corresponding relations. The result of the configuration process is a sales bill of material consisting of well-defined instances from these classes. In short, the structural knowledge defines the classes; the available components are defining the instances.

K-Model assumes that the configuration model and the underlying configuration engine are separated. There is no need (and no possibility) to express specific solution strategies. It is assumed that the configuration engine can interpret the dependencies specified. No specific configuration software is targeted; several commercial configuration engines can handle configuration problems designed with the K-Model methodology.

K-Model is evolved by Brinkop Consulting in a multitude of projects. It was learned that configuration knowledge is distributed on several persons, each focusing on a different perspective of the configuration task. The challenge is not to acquire the configuration knowledge but to achieve a shared commitment of the way how to solve the configuration task at hand. Therefore K-Model concentrates on cross-department communication. The methodology addresses product experts with no specific IT skills. The formalism allows informal descriptions of configuration details as well as formal specifications. The description of the configuration model is based on a mind map with special keywords and structure. The tool of choice is Freeplane[4], which is open source and easy to use.

[4] http://freeplane.sourceforge.net/

Experience shows that methodology is very well suited for workshops from several departments such as product management, research & development, and sales. By applying the methodology to a known domain, the participants are learning the formalism very easily. In early phases the discussion is focusing on domain specific configuration problems. There is no need for deep IT background; the content of the mind map is understood by anybody easily. It is a good basis to discuss alternative ways for solving the configuration problem and to achieve a shared commitment.

2.1 The Formalism

The formalism distinguishes between the items (i.e. classification data of available components), the questions (i.e. sales-relevant configuration questions) and the resulting sales bill of material (i.e. proposal items).

2.1.1 The Items

The tag ITEMS introduces the class hierarchy of available components. Below the tag ATTRIBUTES introducing the classifying attributes with their data type, possible values and translations and are listed. The optional tag SUBTYPES marks the classes of the next hierarchy level. The attributes are inherited along the hierarchy, i.e. all attributes of higher levels are known as well. Structural sub-components might be defined using the tag HAS-PARTS.

The data about available components is defined in so-called selection lists in MS Excel format. The structure of the selection lists must be consistent with the structure defined herein.

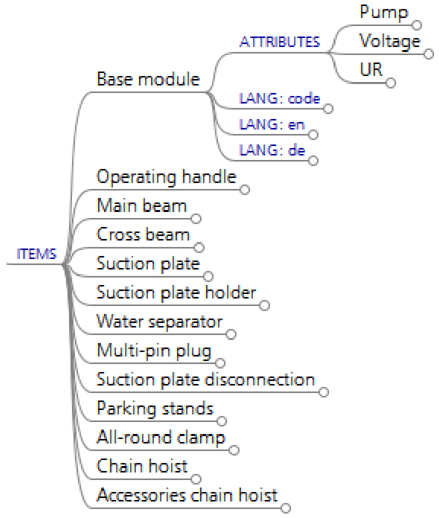

Figure 2: The ITEMS.

2.1.2 *The Catalog*

The catalog is the starting point for the user in the quote generation process. A user can do both, select completely defined (standard) products or configure an individual product that consists of a set of items. Both types of products can be included in a quote. The catalog is structured by categories; each category contains either item classes or other categories. An item may be assigned to several categories; i.e. the assignment must not be unique. The user can find such an item on several paths.

The tags DISPLAY and SEARCH are used to define the attributes to be shown or searched respectively.

2.1.3 *The Questions*

Variables are specifying the object to be configured. They are organized in classes below the tag QUESTIONS. In fact, variables are grouped in classes defining the user interaction. For easy handling variables of a class might be organized additionally in topics. This organization results in a three level hierarchy "class-topic-attribute" which can be found again in the formal names of variables. The use of just three levels is a simplification which was not perceived as a restriction in past projects.

K-Model assumes that there is no additional specification for the user interface; the variables are presented to the user "as they are". Input variables are tagged as EDIT, SELECT, CHECKBOX etc. and output variables as OUTPUT or HIDDEN. The organization in classes and topics is assumed to be used for organizing the questions, for instance in tabs.

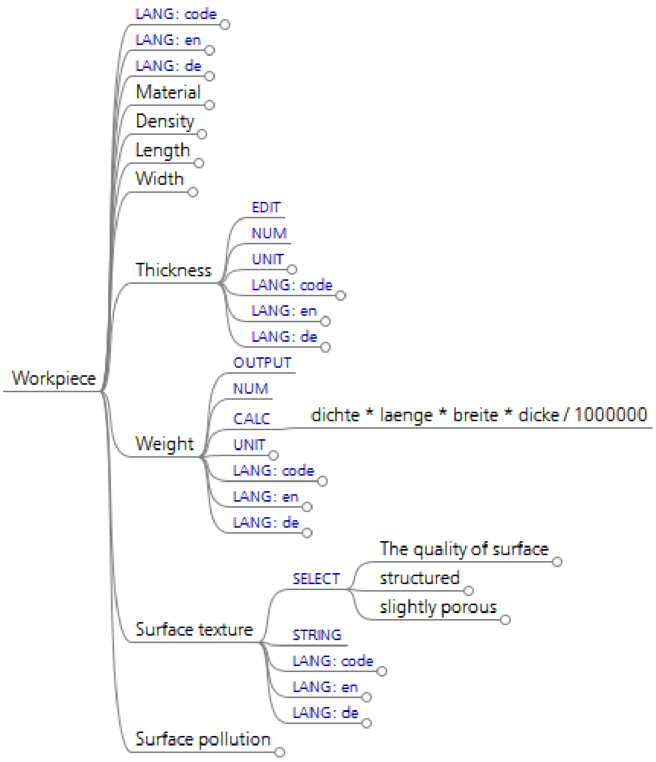

Figure 3: The QUESTIONS.

Tags for language specific translations of the variables are included as well (LANG: en, LANG: de, etc.).

2.1.4 *The Sales Bill of Material*

The result of the configuration process is a sales bill of material tagged as BOM. The bill of material can be structured to any level desired, the "leaves" are instances of the classes below ITEMS. Hereafter the "leaves" are called in short "positions".

Every position is defined by a query. A query to select a position consists of the specification of the class to be searched and conditions to be met by the attributes of the position. It is required that the assignment is unique.

To express relations like "select the drive with the lowest power which is higher as required by x, queries can be specified using the combination of the tags ORDER-BY with FIRST or LAST with the same meaning as in SQL.

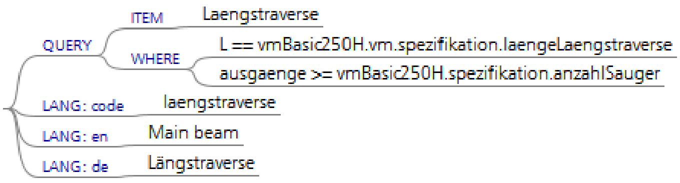

Figure 4: QUERY in the BOM.

2.2 The Method

The method describes the steps that are necessary to set up a configuration model using the formalism presented in the previous section. The following sections each describe a step of the process that are carried out during workshops or reviews, according to figure 5.

2.2.1 *Capture variables*

The model design process starts with a kick-off workshop to define the scope of the model and to get an idea about the configuration problem. In a kind of brainstorming the relevant characteristics are collected and captured in the mind map. These characteristics are called variables in the following. The objective is not to describe the configuration problem formally but to gain the key variables for the problem.

After the first phase the variables are discussed more in detail, e.g. whether a variable is an input or an output. In case of an input, does the value come from a fixed set of values or is a user free to enter any value. In case of an output it is discussed how the value can be computed.

2.2.2 *Organize variables*

Variables describing the same object should be placed as attributes of the same class. K-Model has the concept of a "topic" to organize attributes of a class in another level. This allows easily handling classes with a large number of attributes.

As already stated, it is assumed that the organization of the variables directly influences the user interface. Variables belonging to the same topic are represented to the user at the same time: e.g. classes in tabs and topics as groups of decisions.

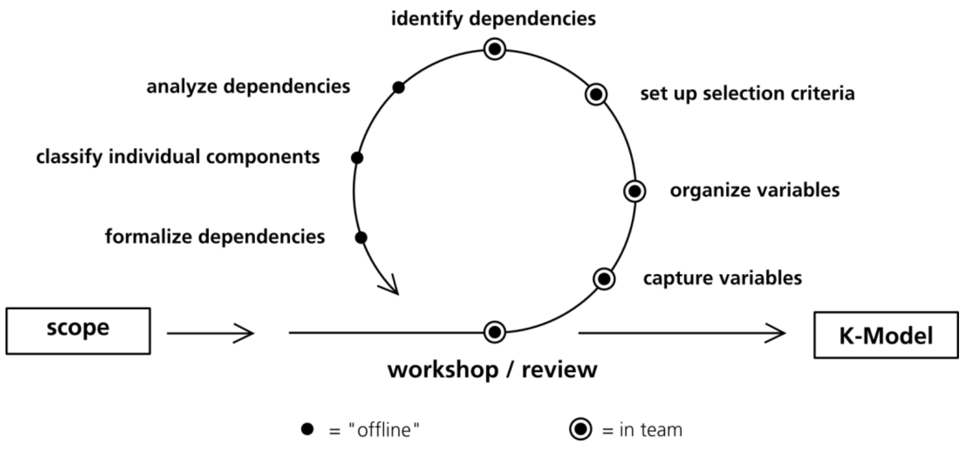

Figure 5: The cycle of workshops and reviews for designing configuration models.

2.2.3 Set up component selection criteria

Available components are organized in classes according to the defined ITEMS. Individual components as parts of the configuration solution are selected by a set of specific criteria. These criteria make up the characteristic attributes of the components' class.

For every component class these characteristic attributes have to be listed and their domains specified. Especially for discrete value domains, every possible value has to be specified.

2.2.4 Identify dependencies

It is the dependencies between variables that turn a configuration problem into a hard problem. There are several ways in which variables can influence one another. The calculation of the following variables' properties may be based on other variables:

- Value
- Default value
- Existence condition
- Selection of component type (i.e. the type of class)

It is assumed that the configuration engine selected for the implementation exposes default values to the user and does not assign defaults directly to the variable.

2.2.5 Analyze dependencies

After the informal definition dependencies are analyzed. As already stated, the variables ("questions") are describing classes of user interaction. Variables which have a strong relationship should be placed in the same class. This reduces complexity for the model as well as complexity of user interaction.

A good tool for analysis is a dependency matrix containing the configuration variables as header for rows and columns. A field (x, y) contains a cross when variable x influences variable y. The distribution of the crosses visualizes the dependencies.

2.2.6 Classify available components

Available components are organized in classes with characteristic attributes; every individual component is classified by assigning values to its attributes. Available components are selected by their attribute values; i.e. they represent the providing "function" within the attributes.

In case of automatic selection, the components must have mutually exclusive sets of attribute values. Each query should have exactly one hit. This requirement can be relaxed when there is a scenario of interactive selection by the user. In that case there might be multiple hits of a query, but user must have the possibility to distinguish between the components. Ideally, there is either a text or a picture describing the components.

2.2.7 Formalize dependencies

Finally the specifications of the variables and the dependencies are written down formally using formulas, algorithms and query statements. This step is required for ensuring that the model can be implemented with the configuration software.

It is important to keep the informal description as well for documentation purposes and to control the formalization and to keep the ability for an easy discussion.

3 APPLICATION OF K-MODEL IN THE CASE OF SCHMALZ

This section describes how K-Model was applied at J. Schmalz GmbH to acquire the relevant configuration knowledge and for realizing a prototype configuration application.

3.1 Procedure in the Workshops

According to the K-Model method the relevant configuration knowledge for realizing a configuration application was acquired during a kick-off workshop and follow-up review workshops.

The configuration team was set up from product manager, sales manager, product data management and K-Model expert. The knowledge acquisition process is driven by the K-Model expert and supported by all other team members. During all workshops the K-Model expert takes notes visible for every participant using the K-Model mind map.

The kick-off workshop started with specifying the scope of the configuration model. Two distinct product families were chosen for realizing the prototype application in order for being able to assess the results independent from a single product domain. After this the K-Model methodology was applied by capturing variables, organizing variables, setting up components' selection criteria, and identifying dependencies. This work is done in a kind of "brainstorming" style with documenting every statement informally in the K-Model mind map.

After the kick-off workshop the mind map was refined by adding formalized definitions according to the informal notes that were taken within the workshop. The individual components are classified and the specifications and the dependencies are formalized. This is typically done "offline"; the K-Model expert is extending the mind maps and dependencies accordingly and the product managers or other persons at the customer's site define the available components within MS Excel sheets.

The resulting mind map and data in Excel sheets were reviewed in some follow-up review workshops by the same team. Just a few cycles of the design process were required to extend the mind map and MS Excel documents for reaching a level that satisfies all participants. After that the model was released for realization.

The main point of discussion in the workshops at J. Schmalz GmbH was about the targeted user group. Should the product configurator be designed for the product novice with only little knowledge about Schmalz' products and enrich the application with product details or should it rather address the expert and thus focus on few decisions without explicit marketing information? At the end it was decided to assist both of them. The system should guide the novice and should not stop the expert from realizing the configuration he has in mind.

3.2 Realization at encoway

encoway received the mind maps and Excel sheets that are the result of applying the K-Model methodology. The documents contain a formal description of the product structure and dependencies together with the available component for two product families. Our modeling experts were directly able to use this structured information for modeling the products within encoway's modeling environment K-Build.

K-Build is web-based application for formalizing configuration knowledge consisting of structure-based modeling facilities, i.e. concepts together with their attributes arranged in taxonomy and partonomy, as well as constraint definitions. This modeling tool contains a test environment which uses the inference engine engcon for interpreting the configuration knowledge. For detailed information about structure-based configuration and engcon the interested reader is referred to [4] and [3], respectively.

The structure within a K-Model mind map can be mapped directly to concepts representing separate branches within the taxonomy: one each represents the sales questions, the classification data and the sales bill of materials:

- A group of sales questions is mapped to a single concept; the questions themselves are mapped to attributes of that concept.
- The classification data is mapped to concepts and the available components defined in MS Excel sheets are imported into lower levels of the specialization hierarchy; i.e. as specializations of those concepts.
- The sales bill of materials (also called bom) can be structured into groups. Each group is mapped to a concept. The configuration solution consists of instances of the available components which are modeled as parts of the bom group structure.

The dependencies within a K-Model mind map can be mapped to so-called rules, each being equipped with a condition and possibly multiple constraints. A condition describes a situation of the configuration solution that must be given for the constraints to be evaluated. engcon offers a wide variety of pre-defined constraints that restrict a given set of concept attributes, including formulae and tables. Simple dependencies (such as greater, less, equals, and so on) and formulae can easily be created using K-Build. Tabular dependencies from K-Model can also be mapped to K-Build's Excel representation with little effort.

The configuration application for Schmalz was set up in two distinct steps. In a first step we created a proof-of-concept for which the least effort should be used. This proof-of-concept was the configuration model running in K-Build's test environment K-Test. In a second step we realized the configuration application full-scale: with stable data exchange interfaces and full graphical user interface. Hence, the data about available components was received in two different ways within the respective steps.

1. In the proof-of-concept step the product data was transferred from the K-Model Excel format to the K-Build Excel format.
2. In the full-scale step the configuration application was set up using encoway's standard architecture. The product data contained in the Excel sheets was converted into encat, which is encoway's standard format for realizing media-neutral master data exchange, based on a well-defined xml structure.

The K-Model Excel sheets containing product data can be transformed into a corresponding K-Build Excel sheet with little manual effort. This way, the available components are imported into the configuration model as specializations of concepts that stem from the classification data. This first step was carried out for testing purposes.

The encat xml document containing product data was imported into the so-called catalogue. encat documents also contain all relevant translations and pricing information, which is relevant for the application user interface and for quote generation, i.e. during run-time, not for creating the configuration model during build-time. Instead, encoway configuration models are typically language-neutral and do not contain the available components or pricing information. The catalogue is a single place for all this information. Technically, it is a database that comes with an advanced API for querying the different types of data during run-time.

While product information, including the translations and pricing information change over time, the physics, on which the product configuration is based, typically stays stable. The physics is represented within the configuration model while the actual components are not. The major benefit of using encat as stable data exchange interface is thus that the configuration model need not be changed when importing new product data.

For realizing the Schmalz configuration application we use encoway's quoting process-supporting tool QuoteAssistant. The QuoteAssistant is a web-based application for browsing catalogue content, configuring products, creating quoting structures together with pricing and generating high quality quote documents; all in one place. The QuoteAssistant contains a standard user interface design for displaying concepts and their attributes within a tab structure using a widget collection containing checkboxes, select boxes or text input fields. This means that, when treating all concepts that are modeled as parts of the K-Model questions as tabs, the placement of sales questions is determined by their attributes and no extra definition for user interface is required.

The user is free in structuring configurable products and available components from the catalogue within folders of a quoting structure. The result of the quoting process is such a structure together with pricing and conditions. This quote result can be exported to a MS Word or PDF documents via the tool K-Document. This tool allows using pre-defined MS Word templates and enriching them with the configuration results, content from a CRM system (such as address data) and from the catalogue (product information or images) automatically during run-time.

4 CONCLUSION

In this paper we have shown how the knowledge acquisition methodology K-Model helps a knowledge engineer to focus on the product domain rather than on knowledge representation formalisms while creating a configuration model. The visualization facilities of standard mind maps ease the creation of configuration models for product managers and sales personnel who are typically not experts in the area of knowledge representation. Especially within workshops where persons with different backgrounds together acquire the relevant knowledge for a configuration applications this informal mind map representation is a valuable tool.

The results which are produced by the analysis steps described in Section 2.2 may seem rather tentative at first sight. However, the results remain stable once the process of designing a configuration model has gone through a small number of design cycles (see also Figure 5). K-Model was already used to analyze and design configuration models for roughly 20 domains, mostly of very different nature and size. The largest domains consist of up to 2000 variables that are relevant for product configuration within this domain. We thus see this as a significant number of cases to call K-Model a success for supporting the process of analyzing a configuration domain and designing a respective configuration model. For encoway, however, the Schmalz configurator is the first application of a K-Model. But nonetheless, the input in form of well-designed mind maps and Excel sheets significantly improved setting up a configuration model from scratch.

An extension of K-Model that is currently under development is modularizing the mind map in multiple sub-maps. With this approach it is possible to describe smaller parts of a configuration that can be reused (multiple times) within larger configuration contexts. The modularization also enhances keeping an overview of large configuration domains.

For J. Schmalz GmbH, K-Model was applied while creating a working prototype configuration application. It took just a few workshops with product managers and sales personnel to set up the K-Model mind maps and MS Excel. This input data was of high quality and could be directly used by encoway modeling experts for creating a configuration model of the product domain.

Schmalz is now able to fully benefit from the configuration application that was set-up using the K-Model knowledge acquisition methodology. Applying K-Model within this project was successful in that all relevant persons – including product managers, sales personnel and technicians – were able to focus on the specific characteristics of the desired configuration application without extra effort for learning representation facilities. The methodology significantly increased the efficiency of cross-department communication and reduced the time-to-prototype during realization.

5 RELATED WORK

Because knowledge acquisition in the environment of knowledge-based configuration systems is notoriously a very expensive process, there is other work concentrating on this task. Support for knowledge acquisition tasks ranges from propose-and-revise techniques that help users in deciding on correctness to graphical representation in form of UML class diagrams or mind maps.

The work described in [7] explicitly targets to support the task of knowledge acquisition for configuration knowledge bases with a propose-and-revise strategy. It is implemented in the knowledge acquisition tool EXPECT, which uses LOOM, a knowledge representation system based on description logics. The focus of this work is on correctness of the underlying knowledge and does not take graphical representation into account.

In [6] a UML representation for configuration knowledge bases is introduced for the purpose of enhancing sharing, distribution and cooperation within the use configuration knowledge. UML stereotypes are defined to represent the specifics of configuration such as concepts, attributes, taxonomy and partonomy. Constraints are defined using OCL. In [8] the authors bring the idea one step further by introducing a set of rules for transforming UML models into configuration knowledge based on description logics such as OIL or DAML+OIL. This work explicitly aims at supporting the knowledge acquisition bottleneck with graphical representation as a frontend and can thus be seen similar to the K-Model approach, although K-Model prefers mind maps over UML diagrams.

The authors of [5] also use mind map structures to support knowledge engineers. However, their work focuses on formalizing, sharing and reusing experiences of past projects in order to help avoiding mistakes that these projects have already encountered. Their work differs from ours in the sense that they use mind maps to capture and represent project experience while we use mind maps to capture and represent configuration knowledge.

The methodology K-Model is novel in the way that is explicitly targets to support non-experts during the acquisition of configuration knowledge by using mind maps as a graphical frontend. Furthermore, the K-Model explicitly distinguishes master data and product structure, configuration decisions and the configuration solution. It defines the syntax and semantics of usable mind map structures as well as the modeling process, i.e. how to use the mind maps in workshop situations together with non-experts such as product managers or sales personnel.

REFERENCES

[1] A. Brinkop. *Variantenkonstruktion durch Auswertung der Abhängigkeiten zwischen den Konstruktionsbauteilen*, Dissertationen zur Künstlichen Intelligenz, Band 204, Infix, St. Augustin, 1999.

[2] D. Sabin and R. Weigel. Product Configuration Frameworks – a Survey. *IEEE Intelligent Systems*, 13(4):42–49, 1998.

[3] O. Hollmann et al. EngCon: A Flexible Domain-independent Configuration Engine. In: *Proceedings of Configuration (ECAI 2000-Workshop)*:94–96, 2000.

[4] C. Ranze et al. A Structure-based Configuration Tool: Drive Solution Designer (DSD), In: *Proceedings of AAAI02 / IAAI02*: 845–852, 2002.

[5] C.-S. Chen and Y.-C. Lin. Enhancing Knowledge Management for Engineers Using Mind Mapping in Construction, *New Research on Knowledge Management Technology*, Dr. Huei Tse Hou (Ed.), ISBN: 978-953-51-0074-4, 2012.

[6] A. Felfernig et al. Configuration Knowledge Representation Using UML/OCL, *Lecture Notes in Computer Science*, Volume 2460, 91-108, 2002.

[7] S. Ramachandran, Y. Gil. Knowledge Acquisition for Configuration Tasks: The EXPECT Approach. In: *Proceedings of Configuration (AAAI 1999-Workshop)*:29–34, 1999.

[8] A. Felfernig at al. UML As Knowledge Acquisition Frontend for Semantic Web Configuration Knowledge Bases, In: *RuleML 2002 – Proceedings of the International Workshop on Rule Markup Languages for Business Rules on the Semantic Web*, Michael Schröder and Gerd Wagner (Eds.), Volume 60, 2002.