How to Analyze and Quantify Similarities between Configured Engineer-To-Order Products by Comparing the Highlighted Features Utilizing the Configuration System Abilities

Sara Shafiee[1], Lars Hvam[2], Katrin Kristjansdottir[3]

Abstract. Engineering-To-Order (ETO) companies making complex and highly engineered products, face the challenge of delivering highly customized and engineered products with high quality and short delivery time. In order to respond to those challenges ETO companies strive to increase commonality between different projects and to reuse product related information. For that purpose companies need to be able to retrieve previously designed products and identify which parts of the design can be reused and which parts to redesign. This allows companies to reduce complexity in the product range, to decrease the engineering hours and to improve the accuracy of the product specifications. In this article we suggest a framework where product features from the company's configuration system are listed up in order to compare with previously made products by retrieving information from internal ERP/PLM systems. The list of features consists of defining features with potential sets of values e.g. capacity, dimensions, quality of material, energy consumptions, etc. When identifying a specific previously designed product, it allows access to all of the specifications of the existing product along with the engineering hours used, materials used, and hours used in the workshop. The aim of this paper is to make a framework for setting up a database before starting the comparison.

1 INTRODUCTION AND PROBLEM STATEMENT

A configurator supports the user in specifying different features of a product by defining how predefined entities (physical or non-physical) and their properties (fixed or variable) can be combined [1]. Improving the quotation process with the help of configuration systems is a great opportunity for enhancing the presale and production process efficiency in the companies [2]. There are several benefits that can be gained from utilizing product configuration systems, such as a shorter lead-time for generating quotation and fewer errors, increased ability to meet customer requirements with regards to functionality and quality of the products, increased customer satisfaction, etc. [3]. Theoretical elaboration of the empirical evidence suggests that, in order to reach all the advantages that can be gained from utilizing product configuration systems, changes in the organization and the supporting systems in the order acquisition and fulfillment process are needed [4]. These issues can be solved by double checking all the outputs generated by the configurator through an automated IT solution. "All designs are redesigns" has long been a popular cliché in design research [5]. More generally it has been observed that in many firms the reuse and generalization of past experiences (often called "lessons learned") is becoming a key factor for the improvement, in time and in quality, of operational processes [6]. It is rational to say that all the attributes of the products and all their relations are available in the configuration system; and for every received order from the customer, changes and specifications for the product are entered into the configuration system. The idea is to make a connection between ERP and the configuration system, when generating quotations in the product configuration systems and compare it with the previous done projects saved in the ERP system from different perspectives. ETO companies producing complex highly engineered products have a significant problem when calculating the prices for the presale and sale processes. That is especially the case when domain experts cannot determine accurate price curves or when vendors are not providing sufficient information to be modeled inside the configurator. Therefore estimates are used or markup factors are added. When underestimating costs in projects the company will lose profit and when overestimating the cost the customer might go elsewhere where he can buy the product at a reasonable price. The accuracy of calculations is affected, as previous projects are not easily accessible and it requires significant work to compare potential new projects with previous projects manually in order to find the relevant information.

Hvam et al. [1] explains this problem by using an example from F.L. Smidth, which is an ETO company selling cement plants. In this example, the company strives to reuse information from previously made projects to calculate the most accurate price based on weight and capacity. According to Hvam et al. [1], the price and weight curves are made by inserting the capacity, price and weight based on information from 3-5 previously produced machines. A curve is then drawn through the points as is demonstrated in

Figure 4. This allows identification of prices and weights for machines that have not previously been produced.

[1] Industrial PhD Student, Management Engineering department, Technical University of Denmark, 2800 Kgs.Lyngby, Denmark, sashaf@dtu.dk

[2] Professor, Centre for Product Modelling (CPM), Department of Management Engineering, Technical University of Denmark, 2800 Kgs.Lyngby, Denmark,lahv@dtu.dk

[3] PhD Student, Management Engineering department, Technical University of Denmark, 2800 Kgs.Lyngby, Denmark, katkr@dtu.dk

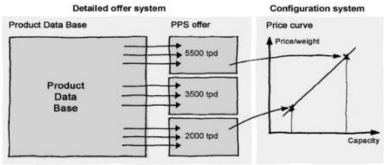

Figure 1. Price and weight curve for main machines in F.L Smith

However, with regards to highly complex products, the price curves are not thought to be the most accurate method as there are several dependent features and great numbers of neighbors on the curve. Another important drawback from the price curves is that the user is only provided access to some of the previously made projects. Therefore the most similar previous projects might be missed.

The first benefits of using an automated IT process, where an integration between the configuration system and the company's internal ERP in order to get access to previously saved project information, is to avoid time consuming redesigning activities in the production phase. This means that it will be possible to produce the same component or product while spending the least possible time and resources.

Salvador and Forza [7] offer much anecdotal evidence of the issues related to product configuration systems. These are listed in terms of: excessive errors, too long time between sales and installation due to inadequate product information supply to the sales office, an excess of repetitive activities within the technical office, and a high rate of configuration errors in production. Even if there are often concerns regarding product configuration projects and the possible errors in the early phases of deploying the systems, the confirmation of the configuration system is not the only benefit from the mentioned solution.

Salvador and Forza [7] describe product configuration systems as aid systems for the end users or customers for creating communication value. Comparing the new project with previous ones could also turn into a recommendation system in the companies. Felfernig [8] discusses different recommendation systems that are divided into Collaborative Filtering (CF), Content Based Filtering (CBF) and Knowledge Based Recommendations (KBR). The available recommendation technologies in e-commerce are potentially useful in helping customers to choose the optimal products configuration [9]. It seems that the mentioned idea is similar to the values that come from recommendation systems. This means that if a 95% similarity between the current project and a previous project is found, the previous project can be re-used and thereby cost related to making the product specification significantly reduced. This includes costs in the sales phase, engineering and production. Furthermore, this is likely to improve the quality and the accuracy of the cost estimations. It also makes it easy to reach an agreement with the customer, and to recommend to them a consultancy to confirm the success of the project by small changes in the order.

Furthermore, this approach enables companies to analyze the products statistically for future product development. Using the configuration systems and comparing different orders can provide valuable information to managers, as it helps them to keep track of product features and to get an overview of market demands. This helps companies to be more in control over the product assortment and eliminates the complexity related to the diversity of product features offered in the production line.

Modular architecture is a term that usually refers to the construction of a building from different instances of standardized components, and in manufacturing it is used for interchangeable units that are used to create the product variants [10]. Dahmus et al. [11] defines a Modularity Matrix to find the similarities between product platforms across columns for a single function in the matrix. Thereafter, architecting of the product portfolio is recommended to take advantage of possible commonalities through the reuse of modules across different product families. If an existing product has standardized and decoupled interfaces, the design of the next product can re-use heavily from the components of the previous product. Holmqvist [12] identifies existing modularization methods and analyses them with regards to their ability to deal with different degrees of product complexity. Based on that he proves that modularization methods are really useful for a simple product architecture but for higher degrees of product complexity, when several functions are allocated to several physical modules, or large variation of variants, these methods seem inefficient [12]. Zamirowski et al. [13] presents three additional heuristics to find common modules across products in a product family. By knowing the previously ordered products, there will be the opportunity of decoupling of design and production tasks.

The potential benefits that can be gained from using the comparison capabilities between configuration systems and other databases at the companies are summarized in Table 1.

Table 1. Benefits from reusing the previous projects

Area	Benefits
Management	1. Lean management by avoiding all the presales, production and sales activity that have been performed before.
Configuration system development	2. Reducing errors and increasing reliability of the configuration system.
	3. Facilitating the testing process for the configuration systems development.
Standardization, Product planning, Configuration system	4. Recommending previously successful projects to the end users.
	5. Basis for product standardization.
Product planning, management	6. Statistical approach to the information and market requirements of the product.
Product planning, Configuration system	7. Improve the quality of the configuration system, lead time, manufacturing, sales engineering.

Inakoshi et al. [14] propose a framework for product configuration that integrates a constraint satisfaction problem with a Case-based Reasoning tool (CBR), where the framework is applied to an on-line sales system. This framework contains the following steps:

1. **Case retrieval:** similar cases are retrieved from the case base in accordance with the similarities between the current query and the cases.
2. **Requirement formalization:** a well-defined requirement consists of the current query and the object function, and it is supplied to **a** Constraint Satisfaction Problem (CSP) solver.
3. **Requirement modification:** The well-defined requirement is modified only if there is no configuration and the CSP solver returns no solution back to the CBR Wrapper.

4. **Parts database:** a parts database that contains the definition of a product family. It defines the types of parts, the constraints on parts connectivity, and other kinds of restrictions on the products.
5. **CSP solver:** The CSP solver receives a well-defined user requirement and solves the problem.

The physical structure of the configuration system is illustrated in Figure 2.

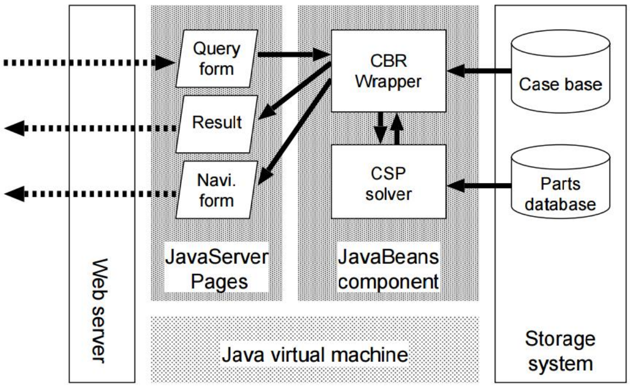

Figure 2. Physical architecture of configuration system [14]

This research work has been used as an inspiration for creating a database development framework and then doing a comparison and integrating it with the product configuration system. There is no discussion in detail on how to make a database from the ERP system, where all the previous projects are stored.

2 RESEARCH METHOD

In accordance with the overall objective, the first phase is focused on the development of the framework, devoted to selecting a framework for product configuration, which integrates a constraint satisfaction problem with Case-Based Reasoning tool (CBR) from previous literature.
The framework development is an ongoing research project to be developed further and tested by a group of researchers and practitioners with an applied research background in modelling products, product architecture, knowledge engineering and product configuration, software development, combining traditional domains of mechanical engineering with product configuration and software development. The framework will be tested in an ETO company specializing in production of catalysts.

3 SUGGESTED METHOD FOR IDENTIFICATIO AND COMPARISON BETWEEN PRODUCT FEATURES

Previous researchers define different tools and methods to measure the similarities between product features. Using configuration systems and techniques for comparing products, it is possible to compare different product features that have been ordered with the new coming orders. One of the prerequirements for using the automatic comparison is to have product configuration system in the sales process. The scenario is to use product features in the configuration system to compare with all the previously generated quotations, which are documented in a desired database. In Figure 3 the process needed for the comparison accomplishment is illustrated.

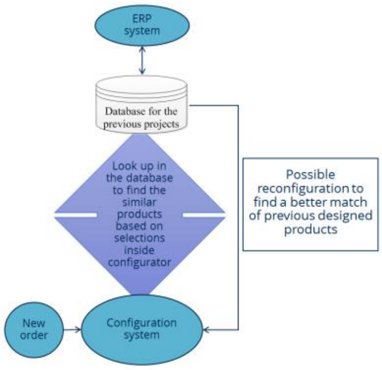

Figure 3. The process of comparing and find similar products

3.1 Set up of the database with previous projects, comparing the configured products with the previously designed products

Inakoshi et al. [14] introduce a framework for comparing a product configuration that integrates a constraint satisfaction problem with a Case-Based Reasoning tool (CBR) for a specific case and with specific tool. In this paper the aim is to make a framework in order to create a database for the comparison, which allows the comparison to be done in a standardized way where the currently available tools and methods can be utilized. Based on literature a seven step framework has been developed, the individual steps are illustrated in Figure 4. The process is not a complete waterfall process, as it is necessary to iterate some of the steps depending on the product.

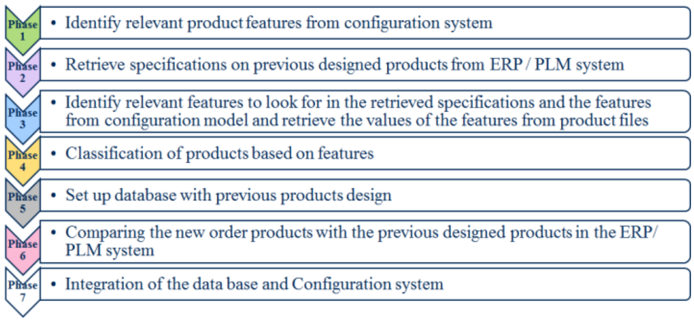

Figure 4. Database set up process in 7 phases

3.1.1 Identify relevant features according to features from configuration model

Previous research that describes how to use modules across different products [13] [15] will be used in order to compare different products. Commonality is best obtained by minimizing the non-value added variations across the products within the same product family without limiting the choices for the customers [16]. According to Ulrich [10], if an existing product has standardized and decoupled interfaces, the design of the next product can borrow heavily from the components of the previous product. Thevenot and Simpson [16] discuss a framework where commonality indices are used for redesigning the product families to align with cost reductions in the product development process aligned with the standardized and modularized product structure incorporated to the configuration system, makes it easier to pick the relevant features or add them to the configurator.

E. Lopez-Herrejon et al. [17] introduce Software Product Line Engineering (SPLE) to represent the combinations of features that distinguish the system variants using feature models.

3.1.2 Retrieve specifications on previous designed products from ERP / PLM system

The current generation of database systems is designed mainly to support business applications and most of them offer discovery features using tree inducers, neural nets, and rule discovery algorithms [18]. One of the fundamental problems of information extraction from ERP systems is that the formats of available data sources are often incompatible, requiring extensive conversion efforts [19]. Knowledge discovery in databases represent the process for transformation of available data into strategic information, which is characterized by issues related to the nature of data and desired features [20] [21]. Brachman et al. [22] define Knowledge Discovery (KD) process elements to be in three steps:

1. Task discovery, data discovery, data cleansing, data segmentation
2. Model selection, parameter selection, model specification, model fitting
3. Model evaluation, model refinement, output evaluation

KD has a variety of meanings. It includes, at one end the derivation of useful information from a database like “which products are needed for the specific amount of engineering hours for installation?” [23].

3.1.3 Retrieve features from product files and determining the values

Most companies use the old technique called “British classification” when naming different components according to the product variants. However as the products get more complicated this technique becomes impractical. In this technique, as shown in Figure 5, there is a “surname” of five digits it is the general class of an item and the “Christian name” of three digits for an exact identity of for the particular item [24].

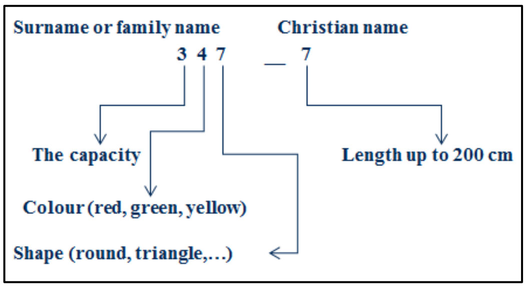

Figure 5. Expansion of a major class [24]

This technique could be used for finding the projects or products with the same specification but for a high level of similarities. This could help us to identify some of the product features and then search for their range of values.

3.1.4 Classifying the products based on features

For identifying and classifying relevant features in order to make a database, classification techniques are required. Burbidge describes how to classify the needs for the product components and coding them by introducing the Group Technology (GT) method [24]. Martinez et al. [25] then use the GT technique as a base for developing a new GT method [25] they provide an example where the GT technique is used in manufacturing plant where it help in the processes of minimizing unnecessary variety by making designers aware of existing components [24]. The aim of classification and coding is to provide an efficient method of information retrieval for decision making. To be efficient enough a code must be designed for the particular purpose for which it will be used [24]. Leukel et al. [26] discuss the design and components of product classification systems in B2B e-commerce and suggested a data model based on XML. Fairchild [27] discuss the application of classification systems and the requirements on them.

Simpson [28] uses GT for adding, removing, or substituting one or more modules to the product platform for product platform design and customization. Sousa et al. [29] suggest an automated classification system for specialization of life cycle assessment. First of all they manage to have a conceptual framework for environmental performance of product concepts. Then, the hierarchical clustering has been used in several applications to show useful ways of grouping objects according to their similarities and product descriptors data. Finally, it is used to develop an automated classification system based on decision trees algorithms. Sousa et al. [29] also use Matlab and C4.5 decision tree algorithm, which seems to be applicable in all classification cases. C4.5 is an algorithm used to generate a classification in form of a decision tree that is either a leaf indicating a class or a decision node that specifies some test to be carried out on a single attribute value. This algorithm has a few base cases as below [30]:

1. All the samples in the list belong to the same class. When this happens, it simply creates a leaf node for the decision tree saying to choose that class.
2. None of the features provide any information gain. In this case, C4.5 creates a decision node higher up the tree using the expected value of the class.
3. Instance of previously unseen class encountered. Again, C4.5 creates a decision node higher up the tree using the expected value.

Ho [21] introduces OSHAM system generated in hierarchical graphical browser which is competing with C4.5.

Magali and Geneste [6] propose object oriented modeling language, Unified Modeling Language (UML) as a standard modelling of domain knowledge for their research work to represent field data. The exploitation of the object modeling as an indexing base is suggested to allow a fast selection of potentially interesting objects during the similar case search [6].

Guillaume et al. [31] developed six heuristics for clustering and weighting the logical, syntactical and semantical relationships between feature names. The other representation introduced as the so-called Product Comparison Matrices (PCMs), can help to make a choice, where the aim is to visualize all the products characteristics through a metrical representation, [32].

3.1.5 Set up database with previous products design

Ramakrishnan et al. [33] give an overview of database design in the following three steps:

1. Requirement analysis: Understanding of what data is to be stored in the database, what applications must be built on top of it, what operations are most frequent and subject to performance requirements.
2. Conceptual database design: The information gathered in the requirements analysis step is used to develop a high-level description of the data along with the constraints to be stored in the database.
3. Logical database design: Database Management System (DBMS) has to be chosen to implement the database design, and convert the conceptual database design into a database schema in the data model of the chosen DBMS.

3.1.6 Comparing the new order products with the previous designed products in the ERP/ PLM system

There are extensive research works in the field of IT illustrating different methods to do the comparison in an automated way. Classical Case-based Reasoning tool (CBR) methodologies [34] [35] are based on four tasks, which are: Retrieve, Reuse, Revise and Retain are highly used for this purpose.

Navinchandra's [36] developed CYCLOPS, which was the first system to explore CBR in interactive design. Vareilles et al. [37] proposed an approach to use 'contextual knowledge corresponding to past cases' and 'general knowledge corresponding to relations, rules or constraints that link design variables'. In this research, Constraint Satisfaction Problem (CSP) is used regarding general knowledge and CBR operates with conceptual knowledge.

Magali and Geneste [6] propose a method to define the neighborhood of the retrieved case to propagate domain constraints. In this method they use Fuzzy Search is divided in two steps that are: rough filtering process and similarity measuring.

Coudert et al. [38] suggest an integrated case-based approach by using ontology of concepts for guiding project planning and system design processes.

3.1.7 Integration of the database with the product configuration system

According to Inakoshi [14], there is the possibility to integrate a constraint satisfaction problem with CBR for a product configuration system.

4. PLAN FOR THE CASE STUDY

The case study is planned based on a group of researchers from the Technical University of Denmark in collaboration with Haldor Topsoe. The aim is to test and make further developments to the proposed framework. The case study should aim to find the major and minor drawbacks in the current framework and refines it based on experiment. The main things that will be tested in the case study are listed below:

1. Can we retrieve the products' features out of the ERP system?
2. Can we classify the products?
3. Can we make a data base according to the product features?
4. How to do the comparison between the new product and the previous designed products?
5. How to integrate the data base and configuration systems? How to make the user interface in the configuration system?

5. CONCLUSION

In this paper we suggest an approach for comparing a new order that is being configured with previous made configurations, which are usually stored in various internal systems at the companies. This will lead to some advantages such as increased commonality across different products and reuse of modules across the family of products. To achieve the goal of comparing different products a database for the necessary features is needed. The proposed approach includes 7 separate phases. Finally after the database setup, the comparison method based on literature will be accomplished and the integration between the configuration system and database will be performed. The paper is just mentioning a problem realized as one of the configuration system drawbacks and suggests a framework for using comparison method to solve this problem. To have a generic framework to retrieve data from ERP/ PLM systems and compare them in configuration projects further research work is required as listed below:

1. Framework testing for a case study and test the available tools for retrieving and comparing the features.
2. Development of the possible ways to integrate database with product configuration system.

6. REFERENCES

[1] L. Hvam, N.H. Mortensen and J. Riis, *Product Customization*, Springer, Berlin, 2008.

[2] A. Felfernig, L. Hotz, C. Bagley and J. Tiihonen, *Knowledge-based Configuration: From Research to Business Cases*, Morgan Kaufman, 2014.

[3] L. Hvam, S. Pape and M.K. Nielsen, "Improving the quotation process with product configuration," *Computers in Induustry*, 607-621, (2006).

[4] D.L. McGuinness and J.R. Wright, "Conceptual modelling for configuration: A description logic-based approach," *AI EDAM*, 12, 04, 333-344, (1998).

[5] A.K. Goel and S. Craw, "Design, innovation and case-based reasoning," *The Knowledge Engineering Review*, 271-276, (2005).

[6] R. Magali and L. Geneste, "Search and adaptation in a fuzzy object

oriented case base," *Advances in Case-Based Reasoning*, Springer, Berlin, 350-364, 2002.

[7] F. Salvador and C. Forza, *Product Information Management for Mass Customization: Connecting Customer, Front-office and Back-office for Fast and Efficient Customization*, New York: Palgrave Macmillan, 2007.

[8] A. Felfernig, L. Hotz, J. Tiihonen and C. Bagley, "Configuration-Related Topics.", in *Knowledge-based configuration: From research to business cases*, Morgan Kaufmann",. 21-28, 2014.

[9] L.L. Zhang, "Product configuration: a review of the state-of-the-art and future research," *International Journal of Production Research,* 52, 21, 6381-6398, (2014).

[10] H. Ulrich, "Fundamentals of product modularity", in "*Management of Design: Engineering and Management Perspectives*, Atlanta, GA, Springer, 219-231, (1994).

[11] J.B. Dahmus, J. P. Gonzalez-Zugasti and K. N. & Otto, "Modular product architecture," *Design studies,* 22, 5, 409-424, (2001).

[12] T.K. Holmqvist and M. L. Persson, "Analysis and improvement of product modularization methods: Their ability to deal with complex products.," *Systems Engineering,* 6,. 3, 195-209, (2003)

[13] E.J. Zamirowski and K. N. Otto, "Identifying product family architecture modularity using function and variety heuristics," in *11th International Conference on Design Theory and Methodology, ASME*, Las Vegas, (1999).

[14] H. Inakoshi, S. Okamoto, Y. Ohta and N. Yugami, "Effective decision support for product configuration by using CBR," in *Fourth International Conference on Case-Based Reasoning (ICCBR), Workshop Casebased Reasoning in Electronic Commerce*, Vancouver, Canada, 2001.

[15] A. Ericsson and G. Erixon, *Controlling design variants: modular product platform*, Society of Manufacturing Engineers, 1999.

[16] H. J. Thevenot and T. W. Simpson, "Commonality indices for product family design: a detailed comparison.," *Journal of Engineering Design,* 17, 2, 99-119, (2006).

[17] R.E. Lopez-Herrejon, L. Linsbauer, J.A. Galindo, J.A. Parejo, D. Benavides, S. Segura and A. Egyed, "An assessment of search-based techniques for reverse engineering feature models," *Journal of Systems and Software,* 103, 353-369, (2015).

[18] T. Imielinski and H. Mannila, "A database perspective on knowledge discovery," *Communications of the ACM,* 39, 11, 58-64, (1996).

[19] E. Bendoly, "Theory and support for process frameworks of knowledge discovery and data mining from ERP systems," *Information & Management,*. 40, 7, 639-647, (2003).

[20] R. Davies, "The creation of new knowledge by information retrieval and classification," *Journal of Documentation,* 45, 273–301, (1989).

[21] T. B. Ho, "Discovering and using knowledge from unsupervised data" *Decision Support Systems,* 21, 29–42, (1997).

[22] R. J. Brachman and T. Anand, "The process of knowledge discovery in databases," in *In Advances in knowledge discovery and data mining*, CA, (1996).

[23] F. H. Grupe, "Using domain knowledge to guide database knowledge discovery," *Expert Systems With Applications,* 10, 2, 173-180, (1996).

[24] J. L. Burbidge, *The introduction of group technology*, London: Heinemann, 1975.

[25] M. Martinez, J. Favrel and P. Ghodous, "Product Family Manufacturing Plan Generation and Classification," *Concurrent Engineering: Research & Applications,* 8,. 1, 12–23, (2000).

[26] J. Leukel, V. Schmitz and F. D. Dorloff, "A modeling approach for product classification systems," in *13th International Workshop on Database and Expert Systems Applications*, (2002).

[27] A. M. Fairchild and B. de Vuyst, "Coding standards benefiting product and service information in e-commerce," in *35th Annual Hawaii International Conference on System Sciences* , (2002).

[28] T. W. Simpson, "Product platform design and customization: Status and promise," *AI EDAM: Artificial Intelligence for Engineering Design, Analysis and Manufacturing,* 18, no. 01, pp. 3-20, 2004.

[29] I. Sousa and D. Wallace, "Product classification to support approximate life-cycle assessment of design concepts," *Technological Forecasting and Social Change,* 73, 3, 186-189, (1980)..

[30] J. R. Quinlan, *C4. 5: programs for machine learning*, Morgan Kaufmann Publishers, 1993.

[31] G. Bécan, M. Acher, B. Baudry and S.B. Nasr, "Breathing ontological knowledge into feature model synthesis: an empirical study," *Empirical Software Engineering,* 1-48, (2015).

[32] G. Bécan, N. Sannier, M. Acher, O. Barais, A. Blouin and B. Baudry, "Automating the formalization of product comparison matrices," in *29th ACM/IEEE international conference on Automated software engineering*, 433-444, 2014.

[33] R. Ramakrishnan and J. Gehrke, *Database management systems*, Osborne: McGraw-Hill, 2000.

[34] Y. Avramenko and A. Kraslawski, "Similarity concept for case-based design in process engineering," *Computers & chemical engineering,* 30, 548-557, (2006).

[35] F. Grupe, R. Urwiler, N. Ramarapu and M. Owrang, "The application of case-based reasoning to the software development process," *Information and Software Technology,*. 40, 493–499, (1998).

[36] D. Navinchandra, "Exploration and innovation in design: towards a computational model", Springer Science & Business Media, New York, 2012.

[37] E. Vareilles, M. Aldanondo, A. C. De Boisse, T. Coudert, P. Gaborit and L. Geneste, "How to take into account general and contextual knowledge for interactive aiding design: Towards the coupling of CSP and CBR approaches," *Engineering Applications of Artificial Intelligence,* 25, 31-47, (2012).

[38] T. Coudert, E. Vareilles, L. Geneste, M. Aldanondo and J. Abeille, "Proposal for an integrated case based project planning and system design process," in *2nd International Conference en Complex Systems Design and Management, CSDM,* 2011.

[39] H. Fargier, J. Lang and T. Schiex, "Mixed constraint satisfaction: A framework for decision problems under incomplete knowledge," *AAAI*, 1, (1996).

[40] A. K. Goel and S. Craw, "Design, innovation and case-based reasoning," *The Knowledge Engineering Review,* 20, 271-276, (2006).

[41] V. R. Basili and D. M. Weiss, "A Methodology for Collecting Valid Software Engineering Data," *IEE transactions of software engineering,* 10, 728-738, (1984).

[42] A. Haug, "Representation of Industrial Knowledge – As a Basis for Developing and Maintaining Product Configurators," Technical University of Denmark, Lyngby, 2008.

[43] S. Shafiee and L. Hvam, "An agile documentation system for highly

engineered, complex product configuration systems," 22nd EurOMA Conference, NEUCHÂTEL, SWITZERLAND, 2015.

[44] P. Kruchten, The Rational Unified Process: An Introduction, New York: Addison-Wesley, 1998.