20TH INTERNATIONAL DEPENDENCY AND STRUCTURE MODELING CONFERENCE, DSM 2018
TRIESTE, ITALY, OCTOBER 15 – 17, 2018

DSM Modeling and Requirement Specification in Developing a Product Platform for Locks

T. Wilschut[1], L.F.P. Etman[1], J.E. Rooda[1], J.A. Vogel[2]

[1]Eindhoven University of Technology, 5600 MB, Eindhoven, The Netherlands
[2]Rijkswaterstaat, 3526 LA, Utrecht, The Netherlands

Abstract: This paper presents dependency structure matrix (DSM) modeling and requirement specification methods that support the development of a lock product platform. The study concerns methods for similarity, modularity, and commonality analysis of a navigation lock portfolio and the creation of a language for writing concise and unambiguous function- and design-specifications. In this paper, we present the methods that we have developed and show how we have used them for the development of a navigation lock product platform. The study bridges DSM modeling with requirements specification in an engineering systems design context.

Keywords: product platform, product family, similarity, modularity, commonality, specification language, requirement specification

1. Introduction

Navigation locks are vital assets in the Dutch infrastructure, which regulate the flow of water through the waterways and enable ships to cross differences in water levels between waterways. In the Netherlands, a considerable number of navigation locks were built during the first half of the previous century. In the coming decades, approximately fifty navigation locks have to be thoroughly renovated or replaced, since they have reached their end-of-life, no longer meet modern-day safety standards, or have insufficient capacity to keep up with growing waterborne transportation.

Historically, locks have been built using an Engineer-to-Order (EtO) production strategy. Each lock has been uniquely designed to meet location specific requirements and constraints. As a consequence, a great variety of lock designs currently exists in the Netherlands.

Lock asset managers have observed that due to the design variety, specialized knowledge, equipment, and spare parts are required to operate and to maintain the locks. The asset managers consider this to be inefficient and expensive. What is more, an EtO strategy requires excessive (human-) resources to renovate and to replace fifty locks within a few decades. Therefore, Rijkswaterstaat (RWS), the executive branch of the Dutch Ministry of Infrastructure and Water Management, founded the Multi-Water-Werk (MWW) project, which is dedicated to the modularization of locks, and the standardization of selected lock modules. By doing so, RWS aims to increase lock reliability and availability (RA), to decrease life-cycle-costs (LCC), and to decrease uncertainty in construction costs and time.

Design and realization of a series of locks using a modularized architecture and standardized solutions for selected modules, resembles a mixture of a Make-to-Order (MtO) and a Configure-to-Order (CtO) production strategy. An MtO strategy requires a basic product structure (design) to be present at the moment a customer order is received, i.e., in the case of RWS at the moment a lock is due for renovation or replacement. This basic product structure is subsequently modified to specific customer needs. A CtO strategy requires standard module and component designs to be present at the moment a customer order is received. A selection of standard modules and component designs is subsequently combined and configured to customer specific needs. A CtO approach allows for mass customization while still benefiting from economies of scale (Jiao et al., 1999).

A challenge in implementing MtO and CtO production strategies is to balance the product variety that is offered to the customer with the internal complexity of managing the design of many product variants (Jiao et al., 2007). To do so effectively, companies often resort to the creation of a product platform, which is defined by Meyer (1997) as: '*a set of subsystems and interfaces developed to form a common structure from which a stream of derivative products can be efficiently developed and produced.'*

The level of standardization of a product platform may differ. Alblas et al., (2012, 2014), for example, advocate the usage of function - technology platforms in traditional EtO industries. Such a platform contains a standard set of functions, working-principles, and technologies from which engineers can choose during the conceptual and embodiment design phases of a design project. It does not contain detailed designs of standard components.

This study contributes to the development of a lock product platform composed of fully-, semi-, and non-standardized component modules and the interfaces between them. The platform distinguishes between basic modules and optional modules. Basic modules are groups of components that are always present in all locks. Optional modules are groups of components that are only occasionally present in a lock. The level of standardization of each module may range from a functional level to a full detailed design level. RWS can use this platform for the efficient development of (semi)-standardized locks that meet location-specific requirements and constraints while reducing the design variety in their lock portfolio.

2. Research objectives

The objectives of this research are two-fold. Firstly, methods are sought to study the similarity, modularity, and commonality of existing locks in the portfolio of RWS. Secondly, methods are sought for to create design specifications for future locks.

The first objective provides insight on how to shape the lock product platform based on the current lock portfolio. In particular, analysis methods are sought:

1. To find groups of similar locks in the lock portfolio of RWS, i.e., groups of locks that share many functions and design characteristics. It is argued that locks within a group can be renovated or replaced using the same set of (semi-)standardized component modules. Hence, the number of groups provides an indication of how many conceptual lock variants one should be able to derive from the lock platform. This number may decrease if RWS decides to no longer build a certain variant in the future or this number may increase if RWS decides to add a new variant.

2. To find modules of lock components within locks based on the system architecture. The basic building blocks of the lock platform are modules of components and their interfaces. System architecture is described as the mapping of a system's functions to the physical components within the system, and the dependencies between those components (Ulrich, 1995). In designing a product platform it is desirable to create modules of components that are as independent as possible (Simpson, 2004).
3. To determine which modules of component and interfaces of components are part of the basic lock structure, and which are part of the optional lock structure. Modules and interfaces that are part of the basic lock structure are the primary candidates for full standardization.
4. To determine which component modules are candidates for full-, semi-, or non-standardization, given the desire of RWS to increase lock reliability and availability (RA), to decrease lock life-cycle-costs (LCC), and to decrease uncertainty in construction costs and time.

The second objective contributes to the actual implementation and usage of the platform. In particular, methods are sought:

5. To create structured and consistent design specifications. RWS outsources the design and construction of locks. To ensure that future locks will meet the predefined standards and interfaces dictated by the lock platform, detailed specifications need to be created for each of the component modules. The consistency of such specifications is essential to ensure the compatibility of the different modules, and to prevent costly and lengthy design iterations.
6. To derive a model of the system architecture directly from design specifications. A visual model of the system architecture helps engineers to increase their understanding of the system, to identify dependencies between components, and to promote communication between engineers (Sosa et al., 2007). For each renovation and replacement project, RWS has to write a public tender. As such, RWS has to work with many different subcontractors. A graphical model of the system architecture aids in the communication and in the transfer of knowledge.

In the next section we summarize the methods we have used and developed to reach the objectives presented above. The Design Structure Matrix is the fundamental modeling concept.

3. Methods

Objective 1 - To identify groups of similar locks a similarity matrix, as presented by Chen (2005), is used. A similarity matrix is a square numerical matrix in which entries have a value of a least 0 and at most 1. A value of 0 at position i, j indicates that elements i and j are 0% similar, while a value of 1 indicates that element i and j are 100% similar. We obtained this matrix by first manually building a characteristic matrix in which the rows are labeled with lock characteristics, such as the type of doors, type of leveling systems, and door-actuators, and the columns are labeled with locks. A non-zero entry within the characteristic matrix at position i, j indicates that lock j possesses characteristic i.

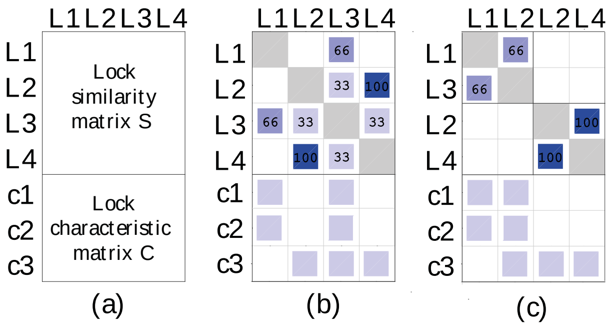

Figure 1: Schematic similarity matrix **S** and characteristic matrix **C**

Subsequently, Jaccard's resemblance coefficient (Jaccard, 1908) is used to calculate the similarity values for all lock pairs based on the characteristics they possess. These values are placed within the similarity matrix. Next, the similarity matrix is pruned to 0.40, i.e., all values below 0.40 are set to zero to increase the sparsity of the matrix as most locks share at least a few characteristics. The pruned matrix is subsequently clustered using the algorithm of Wilschut (2017).

For example, Figure 1a schematically shows characteristic matrix $\boldsymbol{C}$, in which the rows are labelled with characteristics c1, c2, and c3 and the columns are labelled with locks L1, L2, L3, and L4. Figure 1b shows that, for example, lock L1 possesses characteristics c1 and c2 and that lock L3 possesses characteristics c1, c2, and c3. Lock L1 and L3 share two out of the three characteristics they mutually possess, as such they have a similarity of 66% as shown in similarity matrix S. By pruning and clustering similarity matrix $\boldsymbol{S}$, we obtain Figure 1c, in which locks L1 and L3, and locks L2 and L4 are clustered together.

Objectives 2 and 3 – To find modules of components within locks and to determine which modules are common and which are optional, n DSMs are built which are subsequently combined into a ΣDSM $\boldsymbol{F}$ (Gorbea, 2007), schematically depicted in Figure 2. The higher a value within ΣDSM $\boldsymbol{F}$ at position i, j, the more likely that the dependency between component i and component j is present within all locks in the portfolio. $\boldsymbol{F}$ is analyzed using a clustering algorithm to find modules of components that have relatively many mutual dependencies and relatively few external dependencies. For this purpose, we developed a multi-level Markov Clustering algorithm (Wilschut, 2017) that can handle bus structures within the DSM. Modules that have many high valued dependencies are likely to be common. Modules that have many low valued dependencies are not likely to be common.

To fully represent the design variety within the lock portfolio, n should be equal to 127, i.e., the number of locks in the portfolio. However, building 127 DSMs is not feasible within a reasonable amount of time. Therefore, we assume that locks that share many characteristics show little to none variation in system architecture. This enables us to reduce n to the number of lock groups which result from Objective 1, i.e., for each group a single representative lock is chosen.

To ensure that the DSMs, that represent the different groups of locks, can be merged into a single DSM, a single general lock decomposition is made that contains all possible components that a lock may contain. Each representative lock contains a subset of components of the general lock decomposition.

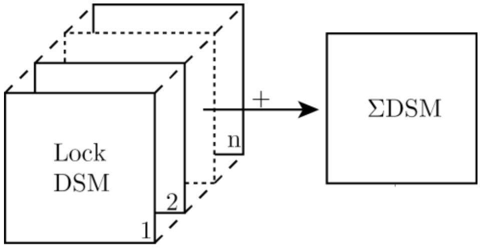

Figure 2. Schematic ∑DSM $\boldsymbol{F}$.

Objective 4 – The method of Brady (2002) is used to determine which component modules are candidates for full-, semi-, or non-standardization, given the desire of RWS to increase lock reliability and availability (RA), to decrease lock life-cycle-costs (LCC), and to decrease uncertainty in construction costs and time. Brady presented a method to identify development risks in the NASA pathfinder architecture. That is, each component in the lock decomposition is given an impact score of 0, 1, 3, or 9 with respect to performance indicators reliability, availability, construction cost, maintenance cost, renovation cost, and life-cycle cost, respectively. The individual component scores are determined using expert interviews as no field-data was readily available. The values are subsequently projected upon ΣDSM $\boldsymbol{F}$ resulting in six projection matrices, i.e., one matrix for each performance measure. For example, reliability projection matrix $\boldsymbol{P}_{\mathrm{R}}(i,j) = \boldsymbol{F}(i,j) \cdot (I_{\mathrm{R},i} + I_{\mathrm{R},j})$, in which $I_{\mathrm{R},i}$ and $I_{\mathrm{R},j}$ are the reliability impact scores for component i and component j, respectively. Thus, the impact score assigned to each component dependency depends on how common that dependency is within the lock variants and the scores assigned to each component.

Objectives 5 and 6 - RWS outsources the design and the construction of locks. To ensure that future locks will meet the predefined standards and interfaces dictated by the lock platform, detailed specifications need to be created for each of the component modules. What is more, the to be renewed locks may have to fulfill additional (function) requirements that may require a change in system architecture. Therefore, we decided to develop a language for the specification of concise and consistent multi-level function- and design specifications from which multi-domain matrix (MDM) models can automatically be generated.

In Wilschut (2018a), we showed that by writing function requirements in terms of goal-functions and transformation-functions following a fixed grammar, a component – function – parameter MDM can be automatically derived. A goal-function denotes the purpose of a component with respect to another component, e.g., to provide power. A transformation-function denotes the internal conversion of flow within a component, e.g., the conversion of power to torque.

The automated generation of MDMs directly from function requirements is the bridge between requirement specification and DSM modeling. Such a bridge is essential in ensuring the compatibility of the various modules of components within the lock product platform. That is, the derived MDMs provide clear insight into the dependencies between the various modules. As such, we continued research into this bridge, which resulted into the Elephant Specification Language (ESL) (Wilschut, 2018c). ESL allows for the creation of function- and design-specifications in terms of needs, requirements, and constraints at multiple granularity levels, following the systems engineering V-model. The system decomposition tree forms the central structure of an ESL specification. The function- and design-needs, requirements, and constraints are specified within the body of component definitions. ESL has a fixed syntax and semantics and supports the formal derivation of dependencies between components, needs, requirements, constraints, variables, and combinations thereof throughout the branches and layers of the system decomposition tree. These dependencies are visualized using DSMs, MDMs, and are analyzed using clustering algorithms.

4. Results

Objective 1 – The clustered similarity matrix revealed that the 127 locks in RWS's lock portfolio can be clustered into seven groups (details presented in Wilschut et al., 2018b). Each group of locks has a distinct combination of characteristics and, therefore, represents a distinct lock variant. Four clusters have a high mutual similarity. Interestingly, most locks that have been built after the year 2000 are a member of the same cluster. As such, this seems the preferred variant in modern-day lock engineering in the Netherlands. Most locks that are due for renewal before 2030 are a member of two distinct clusters. The locks that are due for renewal before 2050 are distributed over four clusters. These results enabled us to categorize the seemingly diverse lock portfolio of RWS into seven lock variants and gain insight in scope of the upcoming renovation and replacement task.

Objectives 2, 3 and 4 – The results of Objective 1 indicate that seven representative locks, i.e., one four each group, can represent the architectural variety in the lock portfolio. In a previous study, Dijkstra (2015) had manually built and analyzed four DSMs of four distinct as-built locks by reviewing design documentation on spatial, information, and energy dependencies. These locks were selected based on expert opinions such that they represent the lock portfolio variety. Not surprisingly, these locks are a member of four different lock groups. Two lock groups do not possess characteristics, different from the other groups, that cause variations in system architecture. These locks primarily differ in geometrical dimensions, which are important from a civil engineering point of view. Therefore, building DSMs for those groups would not yield any additional insight. This left only one group for which an additional DSM had to be built. This DSM has been built during a student project in which the general decomposition of Dijkstra was used as a starting point to allow for easy comparison with Dijkstra's DSMs.

The five DSMs are summed into ΣDSM $\boldsymbol{F}$, which has been subsequently clustered. Component dependencies with a value of at least four are marked as being likely to be common, component dependencies with a value of at least 2 and at most 3 are marked as being semi-likely, and component dependencies with a value of at most one are marked as not likely to be common. Next, the various component impact factors regarding R, A, and LCC are projected upon ΣDSM $\boldsymbol{F}$ yielding six projection matrices. Each of these matrices is separately discussed in Wilschut et al. (2018b).

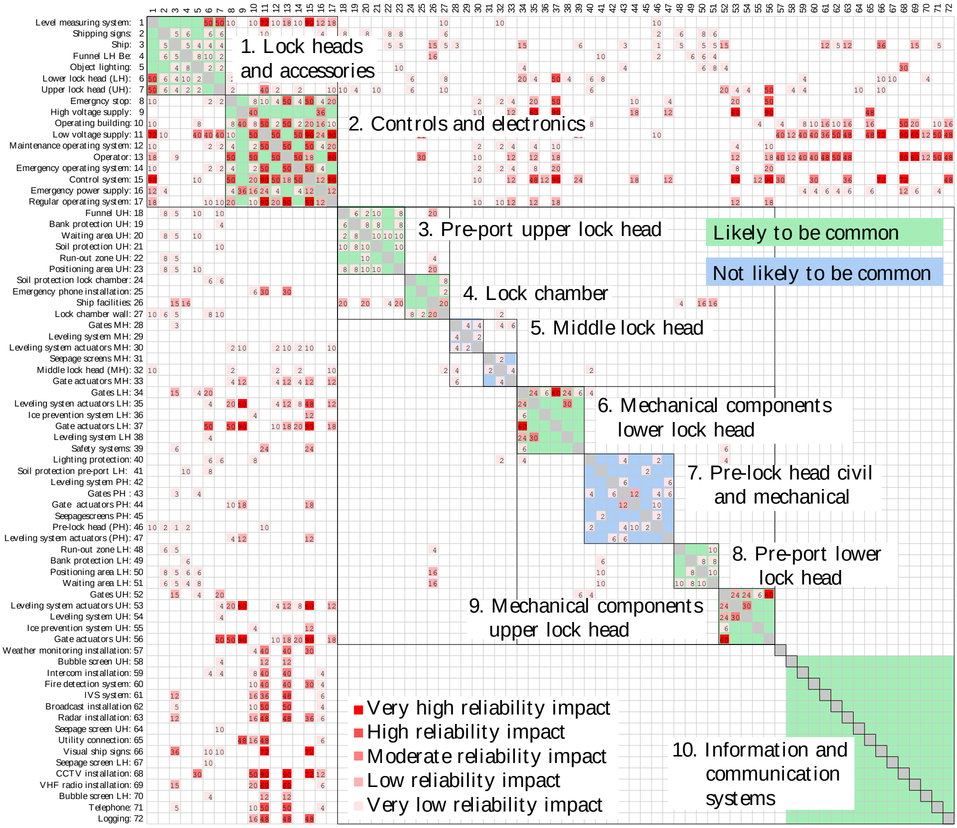

Figure 3. Reliability projection matrix $\mathbf{P}_{\mathrm{R}}$.

Figure 3 combines the results of ΣDSM $\boldsymbol{F}$ with the result of reliability projection matrix $\boldsymbol{P}_{\mathrm{R}}$. That is, clusters are shaded green if they contain primarily component dependencies that are marked as likely to be common and clusters are shaded blue if they primarily contain component dependencies that are marked as not likely to be common. The dependency values indicate the reliability impact score.

Note that eight out of the ten component clusters are likely to be common. However, internally these contain component dependencies that are semi-likely to be common (not visible here). These variations are often due to variations in working principle or in embodiment of components, and not due to variations in desired functionality. Thus, the next step in RWS' standardization efforts should focus on selecting preferred working principles and embodiment of components to reduce architectural design variety in their lock portfolio.

The reliability dependency impact scores are particularly useful to draw conclusions on a cluster level. Figure 3, for example, clearly shows that Clusters 2, 6, and 9 have the highest impact on the reliability of the lock portfolio. As such, in selecting preferred working principles and embodiment of components in these clusters, RWS should carefully evaluate the reliability of each option.

```
goal-requirement
  gr-wf-01: lock-complex-x must regulate water-out-flow-1 to water-way-1 ...
    with subclauses
      c-02: reliability must be at least 'TBD' [%/year]
      c-03: availability must be at least 'TBD' [%/year]
      c-04: upper-flow-range must be at least 'TBD' [m3/s]
      c-05: lower-flow-range must be at most 'TBD' [m3/s]
    end

  gr-wf-02: lock-complex-x must regulate water-out-flow-2 into water-way-2 ...
    with subclauses
      c-02: reliability must be at least 'TBD' [%/year]
      c-03: availability must be at least 'TBD' [%/year]
      c-04: upper-flow-range must be at least 'TBD' [m3/s]
      c-05: lower-flow-range must be at most 'TBD' [m3/s]
    end
```

Figure 4. An example ESL specification.

Overall, the six projection matrices enabled us to determine which component modules should be targeted if one wants to improve a certain performance indicator and which performance indicators are most important while selecting preferred working principles and embodiment of components within modules.

Objective 5 and 6 – A dedicated language, referred to as Elephant Specification Language (ESL), has been developed to support the creation of concise and unambiguous function- and design-specifications for (semi-) standardized modules of components (Wilschut et al., 2018c). In Wilschut et al. (2018d), we present the first proof of principle of ESL in a pilot study concerning a lock renovation project. In this pilot study, we converted natural language requirement statements, such as:

> "*SYS-0194: The navigation lock must retain high water without any unacceptable leakage flow*" (Nieman, 2016, translated from Dutch).

into ESL statements, as shown in Figure 4. Each goal-requirement consists of a main-clause, stating the function that must be fulfilled, and zero or more sub-clauses that state additional conditions that must be fulfilled.

ESL distinguishes between requirements and constraints. Requirements denote what is desired, while constraints denote limitations on what is desired. We used this feature to visualize the impact of a renovation project. For example, Figure 5 shows the component DSM at decomposition level 2 that has been automatically generated from an ESL specification, in which the functions and design of components that are due for renewal are specified in terms of requirements and the functions and design of components that are not due for renewal are specified in terms of constraints. This DSM is part of a larger component – goal-function MDM presented in Wilschut (2018d, 2018e). In Figure 5, the component DSM shows the various types of dependencies between components and several clusters. All dependencies and components that are marked with a red circle are affected by the renovation, i.e., those dependencies are derived from requirements. For example, all electrical-energy-flow dependencies are marked as the locks power-supply is due for replacement.

The results show that ESL has sufficient expressiveness and flexibility to capture the content of natural language requirement documents. Additionally, the generated DSMs and MDMs provide direct insight into the system architecture, and in particular, into which component interfaces area affected by the renovation. This enables engineers to quickly focus their efforts in designing the replacement parts.

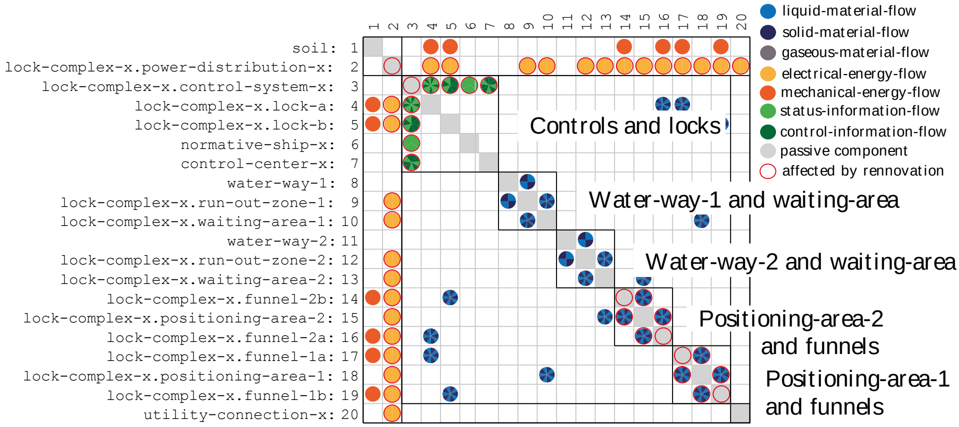

Figure 5. Generated component DSM.

5. Closing remarks

The presented DSM based methods enabled us: (a) to bring structure to the seemingly diverse lock portfolio of RWS; (b) to find the similarity, modularity, and commonality of locks; and (c) to identify component modules with a significant impact on RA and LCC. That is, the clustered similarity matrix revealed that the 127 locks in the portfolio can be grouped into seven groups which possess a distinct combination of characteristics. As such, the future lock platform should support the development of seven locks variants (if one decides to maintain all variants). The comparison of the system architecture of five locks that are part of different groups using a ΣDSM, revealed that most variety in system architecture designs results from differences in working principle and embodiment of components, not from differences in provided functionality. As such, RWS can reduce the design variety by selecting preferred (standard) working principles and embodiments for components. The projection matrices revealed which component modules should be targeted if one wants to improve a certain performance indicator, and thus, are candidates for full standardization.

The developed specification language ESL, enables one to concisely and unambiguously create function- and design-specifications for modules of components in the lock platform. Additionally, when ESL is used to describe an existing lock, one can quickly gain insight into the system architecture and identify those component interfaces that are affected by, for example, the implementation of a new (standardized) module.

Acknowledgements

We would like to thank Rijkswaterstaat, part of the Dutch Ministry of Infrastructure and Environment, which has generously provided funding for this research. In particular we would like to thank, Maria Angenent, Erik-Jan Houwing and Robert de Roos for their enthusiastic involvement in this project.

References

Alblas A. A. and Wortmann. J. C., 2012 Impact of product platforms on lean production systems: evidence from industrial machinery manufacturing. *International Journal of Technology Management*, 57(1):110-131.

Alblas, A. A. and Wortmann J. C., 2014. Function-technology platforms improve efficiency in high-tech equipment manufacturing: A case study in complex products and systems (CoPS*). International Journal of Operations & Production Management*, 34(4):447-476.

Brady, T. K., 2002. Utilization of dependency structure matrix analysis to assess complex project designs. *Proceedings of ASME 2002 IDETC/CIE Conference*, Montreal, Canada, 231-240.

Chen, L. and Li, S., 2005. Analysis of Decomposability and Complexity for Design Problems in the Context of Decomposition. *Journal of Mechanical Design*, 127(4).

Dijkstra, M., 2015. Reliability and availability risk identification within a lock's dependency structure. M.S. thesis, Eindhoven University of Technology, Eindhoven, The Netherlands.

Gorbea, C., 2007. Fricke, E., and Lindemann, U., 2007. Pre-selection of hybrid electric vehicle architectures during the initial design phase. *DSM 2007: Proceedings of the 9th International DSM Conference,* Munich, Germany.

Jaccard, P., 1908. Nouvelles recherches sur la distribution orale. *Bulletin de la Societe vaudoise des sciences naturelles*, 44.

Jiao, J., Tseng, M. M., 1999. A methodology of developing product family architecture for mass customization*. Journal of Intelligent Manufacturing*, 10(1):3-20.

Jiao, J., Simpson, T. W., Siddique, Z.. 2007. Product family design and platform-based product development: a state-of-the-art review. *Journal of Intelligent Manufacturing,* 18(1):5-29

Meyer, M., Lehnerd, A. P., 1997. *The power of product platform - building value and cost leadership*. Free Press, New York, USA.

Nieman, W., 2016. Basisspecificatie Schutsluis. Technical report, Ministry of Infrastructure and Water Management, The Netherlands. In Dutch.

Simpson, T. W., 2004, Product platform design and customization: Status and promise. *AI EDAM*, 18(01):3-20

Sosa, M. E., S. D. Eppinger, S. D., Rowles, C. M., 2007. Are your engineers talking to one another when they should*? Harvard Business Review*, 85(11):133, 2007.

Ulrich, K., 1995. The role of product architecture in the manufacturing firm. *Research Policy*, 24(3):419-440.

Wilschut, T., Etman, L. F. P., Rooda, J. E., Adan, I. J. B. F., 2017. Multi-level flow-based Markov clustering for design structure matrices. *Journal of Mechanical Design*, 139(12):1-10.

Wilschut, T., Etman, L. F. P., Rooda, J. E., Vogel. J. A., 2018a. Automated generation of a function-component-parameter multi-domain matrix from textual function specifications. *Research in Engineering Design,* doi: https://doi.org/10.1007/s00163-018-0284-9.

Wilschut, T., Etman, L.F.P., Rooda, J.E., Vogel. J.A., 2018b. Similarity, modularity, and commonality analysis of navigation locks in The Netherlands. Working Paper

Wilschut, T., Etman, L.F.P., Rooda, J.E., Vogel. J.A., 2018c. A language for multi-level system specification and automated derivation of component, function, and variable dependencies. Working Paper.

Wilschut, T., Etman, L. F. P., Rooda, J. E., Vogel. J. A., 2018d. Multi-level function specification and architecture analysis using ESL: a lock renovation project pilot study. *In Proceedings of the ASME 2018 IDETC/CIE.* Quebec City, Quebec, Candada.

Contact: T Wilschut, Eindhoven University of Technology, Department of Mechanical Engineering, PO BOX 513, Eindhoven, The Netherlands, t.wilschut@tue.nl