22nd International Configuration Workshop

Proceedings of the
22nd International Configuration Workshop

Edited by
Cipriano Forza, Lars Hvam, and Alexander Felfernig

September 10 - 11,
2020 Vicenza, Italy

22nd International Configuration Workshop

Proceedings of the
22nd International Configuration Workshop

Edited by
Cipriano Forza, Lars Hvam, and Alexander Felfernig

September 10 - 11,
2020 Vicenza, Italy

Organized by

Università degli Studi di Padova

TU Graz
Graz University of Technology

DTU
Technical University of Denmark

Sponsored by

SIEMENS
Ingenuity for life

Università degli Studi di Padova
Dip. di Tecnica e Gestione dei Sistemi Industriali
Stradella San Nicola, 3
36100 Vicenza
ITALY

Cipriano FORZA, Lars HVAM, Alexander FELFERNIG, Editors
Proceedings of the 22nd International Configuration Workshop
September 10-11, 2020, Vicenza, Italy

Chairs

Cipriano Forza, University of Padova, Vicenza, Italy
Lars Hvam, Technical University of Denmark, Lyngby, Denmark
Alexander Felfernig, Graz University of Technology, Graz, Austria

Program Committee

Michel Aldanondo, Toulouse University, Mines Albi, France
Tomas Axling, Tacton Systems, Denmark
David Benavides, University of Seville, Spain
Andreas Falkner, Siemens AG, Austria
Gerhard Friedrich, University of Klagenfurt, Austria
Jose Galindo, University of Seville, Spain
Chiara Grosso, Università Ca' Foscari Venezia, Italy
Albert Haag, Product Management GmbH, Germany
Alois Haselböck, Siemens AG, Austria
Birgit Hofer, Graz University of Technology, Austria
Lothar Hotz, University of Hamburg, HITeC, Germany
Dietmar Jannach, University of Klagenfurt, Austria
Gerhard Leitner, University of Klagenfurt, Austria
Thorsten Krebs, encoway GmbH, Bremen, Germany
Tomi Männistö, University of Helsinki, Finland
Seda Polat-Erdeniz, Graz University of Technology, Austria
Mikko Raatikainen, Aalto University, Finland
Rick Rabiser, Johannes Kepler University Linz, Austria
Enrico Sandrin, University of Padova, Italy
Sara Shafiee, Technical University of Denmark, Denmark
Markus Stumptner, University of South Australia, Australia
Nikola Suzic, University of Padova, Italy
Svetlana Suzic, University of Padova, Italy
Richard Taupe, Siemens AG, Austria
Juha Tiihonen, University of Helsinki, Finland
Elise Vareilles, ISAE SUPAERO, Toulouse, France
Franz Wotawa, Graz University of Technology, Austria
Markus Zanker, Free University of Bolzano, Italy
Linda Zhang, IESEG Business School of Management Paris, France

Local Arrangements

Enrico Sandrin, University of Padova, Italy
Nikola Suzic, University of Padova, Italy
Svetlana Suzic, University of Padova, Italy

Preface

Configure a product is the task of composing a product variant from an established product architecture using parametrized components. Products may be both physical goods and services. Products may even be very complex systems. Configurable products and software applications to support the configuration task play a crucial role in obtaining the mass customization capability, i.e. in providing customized products with costs, delivery time and quality close to those of mass produced ones.

Configuration problems are among the most fruitful domains for applying and developing advanced artificial intelligence (AI) techniques. Powerful knowledge-representation formalisms are required to capture the great variety and complexity of configuration problems. Efficient reasoning is required to provide intelligent interactive behavior in contexts such as solution search, satisfaction of user preferences, personalization, optimization, and diagnosis.

The main goal of the workshop is to promote high-quality research in all technical areas related to configuration. The workshop brings together industry representatives and researchers from various areas of AI and management. It provides a forum to identify important configuration scenarios found in practice, exchange ideas and experiences and present original methods developed to solve configuration problems. It promotes discussion on new technologies that can support the automatized solution of configuration problems as well as the investigation on the relationship between configuration technology and the business problems behind configuration and mass customization.

The 2020 Workshop on Configuration continues the series of successful workshops organized within IJCAI, AAAI, and ECAI since 1999. Starting from 2013, the workshop was held independently from major conferences. Even in this 22nd edition, beside researchers from a variety of different fields, it attracted a significant number of industrial participants from major configurator vendors as well as from end-users. The 2020 Workshop on Configuration is a standalone two-day event. It was planned to takes place in Vicenza, Italy at the Department of Management and Engineering of Padova University. Due to COVID19 pandemic, it has been moved online.

A total of 18 papers were selected for presentation on the Configuration Workshop. All papers underwent to full paper blind review with a minimum of two independent reviewers per paper. All papers have been substantially changed to comply with the reviewers' observations.

The themes of the technical sessions are knowledge representation & reasoning, peculiar technologies for configuration (machine learning, conversational agents (chatboats and voiceboats), social software, Microsoft excel), configuration of products in use (reconfiguration, adaptation, renovation, maintenance, repair), business applications with a special focus on the provision of empirical data to depict the state of the art on configuration practices.

Cipriano Forza, Lars Hvam, and Alexander Felfernig August 2020

Contents

KNOWLEDGE REPRESENTATION & REASONING

Structure oriented sales configuration of precast concrete production factories 1
Juha Tiihonen, Ville-Valtteri Korppila, Jorma Heimonen and Andreas Anderson

Bi-objectives configuration optimization for smart product service with hybrid uncertain instance attributes 9
Zhihua Chen, Elise Vareilles, Olga Battaia and Xinguo Ming

Functional Testing of Conflict Detection and Diagnosis Tools in Feature Model Configuration: A Test Suite Design 17
Cristian Vidal-Silva, José A. Galindo and David Benavides

KNOWLEDGE REPRESENTATION & REASONING WITH PECULIAR TECHNOLOGIES

Towards machine learning based configuration 25
Mathias Uta and Alexander Felfernig

Configuration assisted through conversational agents (chatbots and voicebots) 29
Nicolás Afonso, José A. Galindo and David Benavides

Supporting Feature Model-Based Configuration in Microsoft Excel 35
Alexander Felfernig, Viet-Man Le and Thi Ngoc Trang Tran

RECONFIGURATION AND ADAPTATION OF PRODUCTION SYSTEMS

Production Reconfiguration with ASP 39
Richard Taupe and Andreas Falkner

A development approach towards user-centered front-ends for knowledge-based engineering configurators: a study within planning of robot-based automation solutions 47
Eike Schäffer, Sara Shafiee, Tobias Frühwald and Jörg Franke

Towards a Modular Distributed Configuration Model for Autonomous Machines 53
Lothar Hotz, Stephanie Von Riegen, Rainer Herzog and Raoul Pascal Pein

CONFIGURATION IN/FOR RENOVATION, MAINTENACE AND REPAIR

Configuration Models Used for Design Review of Façade Systems in Building Renovation 57
Irene Campo Gay and Lars Hvam

Configuration and Mass Customization of Domotics to support SMEs and their Customers 63
Gerhard Leitner, Martin Stettinger and Anton Josef Fercher

Including Maintenance Services in the Solution Space — Considering Life Cycle Costs in Product Configuration 68
Nicola Ganter, Daniel Kloock- Schreiber, Paul Christoph Gembarski and Roland Lachmayer

BUSINESS APPLICATIONS: SPECIFIC CHALLANGES AND OPPORTUNITIES

Toward Data-Driven Modeling of Configurable Products 76
Albert Haag

Managerial Challenges in Designing an IT Service Configuration System 81
Franziska Schorr, Amartya Ghosh and Lars Hvam

Exploring features for digital social interaction between configurator users and their friends 89
Chiara Grosso and Cipriano Forza

BUSINESS APPLICATIONS: EMPIRICALLY-BASED DESCRIPTION OF THE STATE OF THE ART

Complexity of Configurators Relative to Types of Outputs 95
Sara Shafiee, Eike Schäffer and Lars Hvam

Do product configurators comply with HCI guidelines? A preliminary study 101
Tony Leclercq, Claire Deventer and Patrick Heymans

Product Configuration Activities in SMEs and their Digitalization: Preliminary Results of a Survey Study 106
Svetlana Suzic, Enrico Sandrin, Nikola Suzic, Cipriano Forza and Alessio Trentin

Structure oriented sales configuration of precast concrete production factories

Juha Tiihonen[1] and **Ville-Valtteri Korppila**[2] and **Jorma Heimonen**[1] and **Andreas Anderson**[1]

Abstract. Literature on sales configuration shows surprisingly few examples of mechanisms for configuring product structures. The company Elematic supplies factories for precast concrete element production. They respond to hundreds of requests for quotes annually. The lead time for answering may be weeks and preparing a quote manually takes hours or even days per quote. We describe concepts and mechanisms that allow a sales configurator engine to model configurable product structures, perform dimensioning, determine dynamic component defaults and even automate component selection based on locally optimal fitness. We apply the concepts and mechanisms to support sales configuration of such factories. We characterize the resulting configuration model that has 4 levels of depth in product structure, over 200 unique component items and almost 800 configurable attributes, and about $2 * 10^{1411}$ possible configurations. We also describe experienced effects at Elematic.

1 INTRODUCTION

The authors have often found configure-price-quote (CPQ) applications to require product structure configurations for accurate quotes. This is especially true when the price is determined from production cost + margin. Often, delivery time depends on the configured product structure. Configure-price-quote applications also increasingly include 3D visualizations (e.g. [8]), which often require configured product structures in addition to dimensions and other parameters.

The company Elematic[3] delivers factories for precast concrete element production. Precast concrete element factories for hollow core slab production are relatively complex with lots of variation. Elematic intends to cover 80 - 90% of this variation with a sales configurator. Elematic prepares, without a configurator, hundreds of quotes per year, and each quote takes from a couple of hours up to a couple of days. The lead time to answer a request for quote is longer, the goal is to answer within two weeks. Special requests or exact freight cost calculation may further increase the lead time and effort. Structures are important because the quotes are based on identifying, pricing and describing the equipment for a factory that meets the customer requirements.

We have not found much literature on sales configuration with significantly varying product structures in industrial contexts. The aim of this paper is to contribute to filling that gap.

[1] Variantum oy, Finland. Email: {first.last@variantum.com}
[2] Elematic oyj, Finland. Email: {Ville-Valtteri.Korppila@elematic.com}
[3] www.elematic.com

1.1 Previous work

Literature identifies *components* as building blocks of products in the sense that products (product individuals) consist of components (component individuals). Often a component is a physical, usually separable part of another component. Pre-designed components that form the basis of configuration tasks may have specification variables a.k.a. *parameters*, such as physical dimensions, surface material, color, or capacity [1, 5, 20, 23]. To manufacture or configure a component, one must specify values for its parameters. Many components are non-parametric – without any specification variables. Some authors only recognize non-parametric components (e.g., [12]). All components – parametric or not – may be characterized with properties such as mass, material, capacity or power. Some properties have constant values whereas the values of some properties (e.g. mass) may depend on parameter values. *Attributes* are variables that can be used both to represent parameters of components, and to represent derived or fixed properties.

A central phenomenon in the context of configuration is determining which components (can) become part(s) of a whole. Early work on configuration, such as R1 [11] and Mittal & Frayman [12], already identify component selection as a central aspect of the configuration task. An example in [18] indicates that configurable products often have rules of the following type: A `Car` should have exactly one "`Motor`" that can be selected out of available motor types e.g. "`Motor_1.8_120HP_Petrol`" or "`2.0_100HP_Diesel`". Here, several aspects are noteworthy. First, it is desirable to be able to refer to parts by their *generic part name* [3], for example "`Motor`". Second, *cardinality* indicates a valid number of parts with a generic part name. For example, a "`Car`" might have exactly one "`Motor`", exactly four "`Wheels`", and an *optional* (0 or 1) "`Sunroof`". Third, only some types of component (allowed types, possible part types, key component assumption of [12]) are eligible for a generic part name; in this example "`Motor_1.8_120HP_Petrol`" or "`2.0_100HP_Diesel`" could be eligible as "`Motor`" while "`Motor_1.8_180HP_Turbo_D`" might not be.

Structure-based configuration modeling emphasizes the modeling of the compositional structure of products (e.g., [4]). In the configuration domain, this whole-part relationship is usually called *has-part relationship*. Generic product structures were introduced as a mechanism for representing configurable product structures [22]. One conceptualization of configuration knowledge unifying previous approaches [15] has *part-definitions* as a central configuration modeling concept for defining configurable product structures.

To support inference based on varying product structures, the selection of a component in generative constraint satisfaction [17] instantiates a component with variables (attributes) and constraints that represent the component.

The set of variables and constraints is dynamic instead of fixed as in traditional CSPs. This method of defining instantiable components, parameters and constraints is a major step towards modeling concepts that directly reflect configuration domain phenomena [10]. Sometimes this kind of approach is also known as component-based or component-oriented configuration [16, 10].

Aldanondo et al. [2] describe a model and a tool where functional view of a product and bill-of-material are presented in the same system for quoting purposes. The idea is to provide "external data (for customer) or product detailed functions and quotation" and "internal data (for supplier) or bill of materials, routings and cost.". Tiihonen et al. [21, 19] describe a conceptual model and system versions that support configuration of structures during sales phase. Despite these contributions, research literature seems to have a gap in actually describing the mechanisms used for product structure configuration in the context of sales configuration.

1.2 Research questions and methodology

This paper addresses the following research questions:

- RQ1: What concepts and mechanisms support structure-based sales configuration?
- RQ2: What concepts and mechanisms support engineering calculations for sales configuration?
- RQ3: How concepts of RQ1 and RQ2 apply to industrial sales configuration?

We follow the Design Science approach [9, 7]: Our aim is to provide innovative tools that provide concrete benefits in their environment of use. We develop and build methods and instantiations that are iteratively improved based on evaluation at use. We aim to utilize existing knowledge base.

The rest of this paper is structured as follows. Section 2 introduces concepts and mechanisms that provide one possible answer to RQ1 and RQ2. Section 3 describes and quantifies the Elematic configuration challenge and a configuration model aiming to solve that challenge. We discuss results and experiences in Section 4. Finally, Section 5 concludes.

2 VARISALES CONFIGURATION MODELS AND CONFIGURATIONS

VariSales is the sales configurator of Variantum. A configuration model consists of one or more *sales items*. One sales item acts as the *top-level item* a.k.a. *root* of a configuration model. A sales item can contain a number of *children* that have other items as *possible items*, see Figure 1. If the top-level item defines children, the possible items of the children (and possible items of their children) are included in the configuration model.

In the rest of this section, we rely on figures of the next section that provide concrete examples. Children are objects attached to a parent object with *has_child* relationships, see Figure 8. A *relationship* is a directed connection between two objects and has a *position code*, which is an arbitrary string. The child objects with same position code represent the possible items of a child definition. A *child definition* consists of

- *child id* (shown as "relationship Position" in Figure 8),
- *possible items* - the items whose individuals can realize this child, sharing the same "relationship Position",

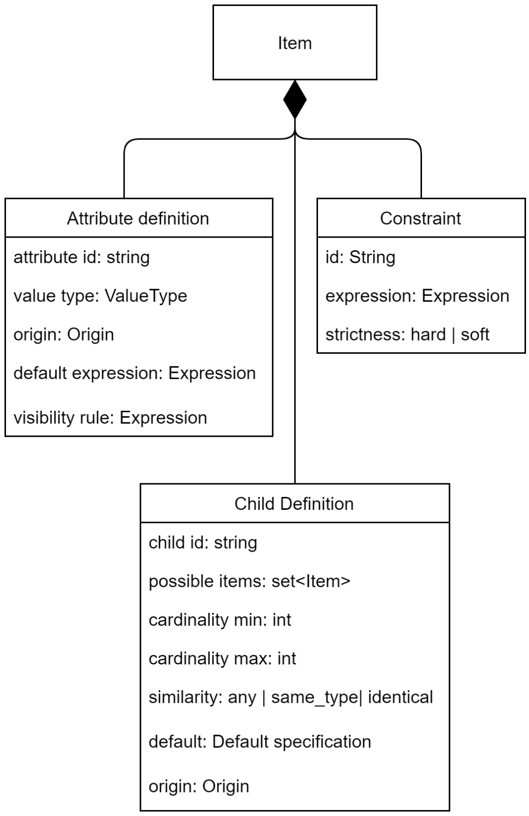


Figure 1. Simplified conceptualization of configuration model. An item consists of a number of attribute declarations, a number of constraints and a number of child definitions.

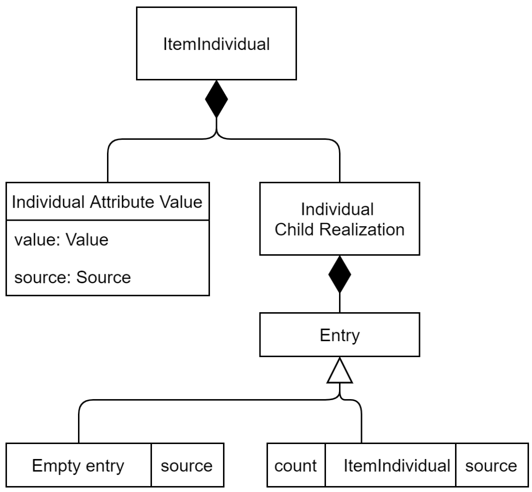


Figure 2. Simplified view of item individuals in a configuration. Each individual consists of a number of individual attribute values, and a number of individual child realizations. An individual child realization can be either an empty entry or an entry with a count and an ItemIndividual. "Source" refers to the Configuration Source of the entry.

- *Cardinality*: minimum and maximum count of item individuals in a consistent and complete child realization. Specified with Child properties, see Figure 9.

- *Similarity*: “any”, “same item”, “identical”. Specified with Child properties, see Figure 9.

Similarity affects valid child realizations as follows:

- “any” allows all child item individuals of the child realization to be individuals of different possible items. Attribute values and child realizations of item individuals can be set individually.
- “same item” requires that all individuals of the child realization are individuals of same item. Attribute values of and child realizations of item individuals can be set individually.
- “identical” means that all individuals of a child realization are individuals of the same item and have identical attribute values and child realizations. This is represented by a child realization that has (at most) one item individual and a count.

The end user configures a product by specifying values for the root object’s (parameter) attributes and by configuring its children. An ItemIndividual of type of the root item is created using a data structure shown in Figure 2. Configuring a child implies instantiating zero or more of the possible items of the child into the configuration as entries. For each child definition of an Item, the corresponding (complete) ItemIndividual will have a corresponding Individual Child Realization. These selected Sales items “bring in” their children, effectively forming a product structure tree, resembling a bill-of-materials. The configured structure varies with selection of possible items. Each ItemIndividual must respect the constraints of the type. Each Individual Child Realization must respect the corresponding child definition. The configuration is incomplete if there is no Individual Attribute Value corresponding to an Attribute Definition or if there is no Individual Child Realization corresponding to a Child Definition.

2.1 VariSales Engineering Calculation System

VariSales Engineering Calculation System (VECS) provides functionality for engineering calculations such as component dimensioning and automated component selection within sales configuration tasks.

Origin of an attribute or a child specifies in the configuration model the source from which the element will get its value. Origin also determines how sales configurator processes the element. Origins are summarized in Table 1.

For attributes, origin “user” means a “normal” configuration parameter with a value entered by the user. Sales configurator can calculate a dynamic default with Default expression. Static defaults can be specified simply by specifying a value in the item, see Figure 3. If default values (static or dynamic) of origin “user” cause inconsistency (broken constraints), sales configurator selectively removes defaults.

Origin “parent” provides an important mechanism for passing attribute values (or dynamically calculated values) both to selected and possible child Items. One can think that child Items are stand-alone objects that “get to know” the current context via interface parameter values given by their parent. Based on these, the child Item knows how to configure itself or if it can be configured according to requirements of the current context.

Origin expression supports calculations that the user cannot modify. For example, attribute `AREA` of a panel could have Origin “expression” and a default expression specifying `WIDTH * HEIGHT`.

Closely related to Origin defined in a configuration model is the concept of *Configuration Source* of configurations. The idea is that

Table 1. “`Origin`” of element in a configuration model defines how an instance of that element in a configuration gets it’s value

Origin	Meaning
“user”	User can give the value Sales configurator may infer a value or provide a static or dynamic default. ’Normal’ configuration parameter or child
“expression”	Sales configurator determines the value of the attribute or child. The value is updated as necessary. The user cannot overwrite the value A.k.a. “*calculated value*”
“item”	The value of the attribute comes directly from the Item. The value is fixed, it cannot be changed. Non-parametric technical properties of components are often naturally of Origin “item”.
“parent”	Attribute of an Item used as child gets its value through an attribute assignment from the parent item and cannot be changed.
“external”	The element is set with means external to sales configurator engine, e.g., Sales Configuration Scripts. Sales configurator can set a default if the external mechanisms have not provided a value.

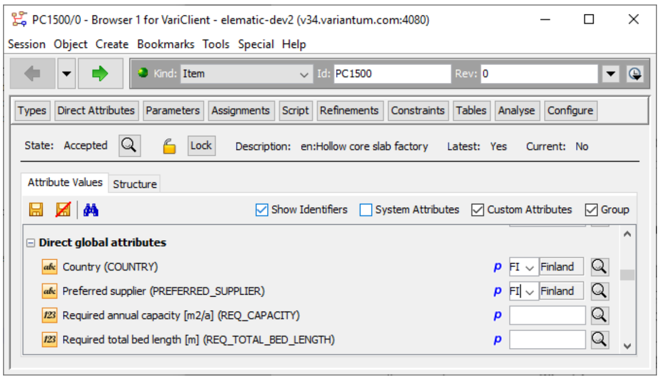


Figure 3. Some parameter attributes of Item “PC1500”, revision “0” (`item:PC1500/0`). Setting a value in Item sets a static default (Origin “user” or a fixed value (Origin “item”).

sales configurator records for each attribute value and child realization entry of a configuration where it came from. For example, suppose that an attribute with Origin “user” got its value with a static or dynamic default, hence Configuration Source is `DEFAULT`. Sales configurator can overwrite a value with Configuration Source `DEFAULT`. But if the same attribute got its value from user, the Configuration Source would be `CONFIRMED`. Sales configurator never overwrites without permission an element that has Configuration Source `CONFIRMED`. The other Origins have a corresponding Configuration Source, e.g. origin “item” has source `ITEM`.

2.2 Product structure configuration via automated and manual configuration of children

Children enable structurally varying sales configurations. Depending on selected items to realize a child, a different set of attributes (and further children) are subject to configuration. A number of mechanisms support child modeling.

Constraints in parent Inference via constraints in parent item constrains the allowed total count of the Individual Child Realization entries or the count of ItemIndividuals of a specific item. This can lead to selection of children or determining that some children are impossible.

Child default Child default expressions(s) determine dynamically the count and type of Individual Child Realization entries. These expressions can be conditional - depending the current context, different entries are created. If the calculated count is 0, the corresponding entry is not created. See Figures 9 and 10 for examples. The entries are created with Configuration Source `DEFAULT` or `EXPRESSSION` depending on the Origin of the child. Individual ChildRealizations containing `CONFIRMED` entries are not affected by child defaults.

Child filtering Each possible child item is tentatively instantiated and the tentative instance examines via constraints if it is compatible with respect to the current context. Figure 12 provides a concrete example. The current context is specified via attributes that get their value via assignments passed from parent ItemIndividual to a tentative instantiation of the possible item. These attributes have value origin "parent". In addition, an item has values for some attributes (origin "item") based on the item's static values. Constraints used for child filtering examine these attributes. If any constraint is broken, the possible item is not possible in this context.

Child filtering may include *eligibility calculations*. Eligibility calculations calculate (dynamic) defaults for origin "user" attributes, or calculated values of origin "expression" attributes, typically based on the item's static value, values received via assignments, or via other eligibility calculations. Constraints used for child filtering may refer also to these attributes.

Incompatible possible items can be grayed out or even hidden. The last possible child item will be selected automatically.

Child ranking Child ranking is based on eligibility calculations and child filtering. One attribute of a possible item represents the "*fitness*" or "suitability" or "goodness" of a possible item. Typically fitness is calculated with eligibility calculations, but it can also be a fixed attribute value. Sales configurator ranks (orders) the possible items by their fitness. The modeler can specify if a larger or a smaller attribute value is better. Child filtering is applied to filter out incompatible items.

Child ranking can be used to support local optimization, e.g., support the selection of the smallest compatible component that has enough capacity and whose conditions of usage are satisfied. Child ranking can be applied to provide recommendations. For example, fitness can be calculated using the principles of Multi Attribute Utility Theory (MAUT).

Child select is similar to child ranking. The best ranking item is automatically selected as the child realization. The count can be specified with an expression. Depending on the child properties, selection can be default (origin "user") or forced ("expression")

Child access functions allow a parent item to examine attribute values of children (the unique value (if max cardinality = 1), set of values, minimum, maximum, sum). These values can be passed to other children. The system internally creates dependency graphs that allow performing non-cyclic child calculations in correct order.

Model of computation The underlying model of computation is a hybrid of constraint programming and processing of object-oriented data structures including those described above. For each Item, corresponding Choco-based [14] CSP model(s) are generated automatically. Supported constructs and data types of the constraint language and child filtering results are taken into account. These CSP models are re-used by all individuals of the type. The CSP models only consider one ItemIndividual and child access functions examining the count(s) of an individual child realization. The configuration model and configurations are also represented with traditional data structures. All constraints and expressions of items can be evaluated in corresponding individuals with "traditional" means (e.g. evaluation of expressions and constraints, checking a configuration w.r.t. the configuration model). Completeness and consistency are determined this way. Dead-end detection, "graying", and selection of the only remaining attribute/child rely on the CSP models.

Put together, the mechanisms of VariSales Engineering Calculation System outlined above provide a clean and efficient combination of "normal" engineering calculations and constraint based AI technologies that work seamlessly together. They facilitate efficient and manageable configuration of complex and structurally significantly varying systems.

3 ELEMATIC'S CONFIGURATION CHALLENGE AND STRUCTURE-BASED SALES CONFIGURATION MODEL

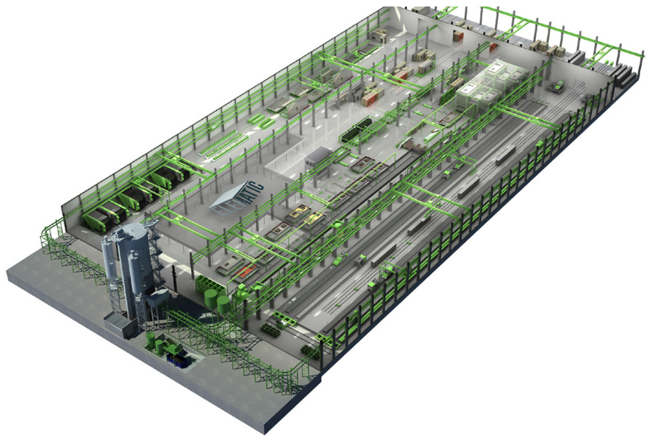


Figure 4. Elematic hollow core slab production model factory "Edge". Source and © Elematic oyj

The Elematic sales configuration approach is based on so called model factories designated as SEMI, PRO and EDGE. SEMI is the most affordable and relatively labor-intensive factory with the smallest production capacity. EDGE is highly automated and offers the largest production capacity, see illustration in Figure 4. Attribute `PRODUCT` of the root item `PC1500/0` represents the selection of the model factory.

Statistics of the configuration model are presented in Table 2.

One of the model factories is recommended based on required output capacity and desired output product parameters (maximum height of produced concrete elements). Almost all attributes and children get default values. Most often, the default values are based on

Table 2. Model statistics, starting from root item PC1500/0

Element	Number of elements
Items	213
Total attributes	793
Integer range attributes	406
Enumerated integer attributes	26
Total integer attributes	442
Total float attributes	25
Boolean attributes	42
String attributes (non-enumerated)	133
Enumerated string attributes	151
Total string attributes	284
Single-valued attributes	792
Set-valued attributes	1
Attributes of origin user	393
Attributes of origin expression	46
Attributes of origin parent	143
Attributes of origin item	201
Attributes of origin external	10
Child definitions	73
Possible Items in child definitions	306
Child definitions with child filters	6
Child definitions with child selects	0
Child definitions with child defaults	69
Entries in child defaults	120
Constraints	119
Attributes with default	632
Origin user attributes with defaults	381
Default expressions, total	189
Dynamic default expressions (origin user)	143
Expressions for Origin expression attributes	46
Visibility rules	223
Items with scripts	9
Potential configurations	$2 * 10^{1411}$

the type of the selected model factory. Many defaults are determined with more complex calculations. For example, *beds* are structures on which concrete elements are cast and left to cure. The number of beds (BEDS) and their length (BED_LENGTH) get defaults based on the selected PRODUCT. However, these can be configured otherwise than the model factory suggests. Component dimensions, calculated CAPACITY and numerous other aspects depend on these selections.

Figure 5 shows some capacity planning parameters in the configurator user interface. Figure 6 shows the configuration after PRODUCT = SEMI was selected. Almost all attributes have a value and almost all children have a realization.

Figure 7 shows a subset of attribute properties of the main item PC1500/0. For example, attributes BEDS and BED_LENGTH get dynamically calculated default values based on the value of attribute PRODUCT. Attribute TOTAL_BED_LENGTH is calculated (defined to have origin “expression”). This implies that the user cannot directly modify the value. This is reflected in Figure 6 so that TOTAL_BED_LENGTH does not have an input widget, rather the value is just displayed. Similarly, the CAPACITY (shown with English display name “Annual capacity”) cannot be edited. The recommended product was also calculated with origin “expression”[4].

Figure 8 shows a part of the sales product structure of the main item PC1500/0 defined with “has-child” relationships. The first three lines define the three possible items for child Site lifting, other properties of the child are shown in the fourth group of Figure 9. The total count of the corresponding child realization must be in

[4] the rest of the expression is intentionally hidden

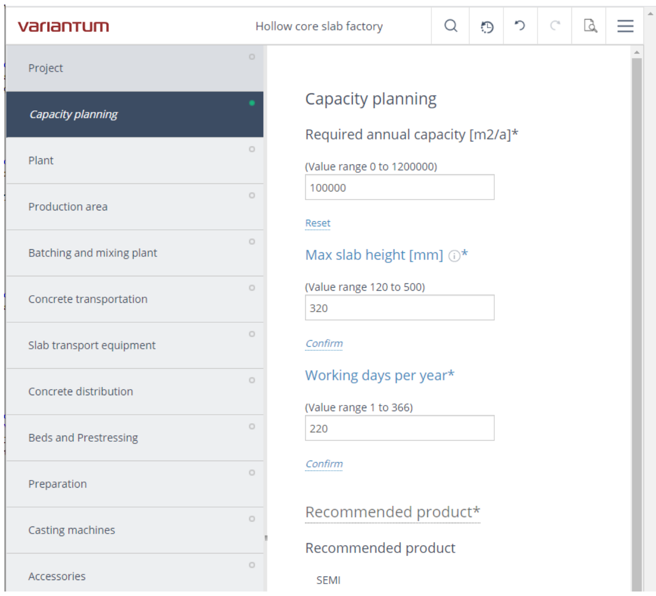


Figure 5. Capacity planning (cropped, some parameters excluded)

range 0 to 5. If product PRODUCT = SEMI as in the example, an empty realization is set as the default. This is reflected in Figure 6 so that the group “Lifting Equipment” does not show Site-lifting. After a few configuration steps, the defaults for STRAND_COMBS can be evaluated and they become selected, see Figure 10.

The deepest structure of the sales configuration model has 4 levels of structure (Figure 11). Nine items have scripts. However, all scripts could be converted into calculations with origin “expression”.

The reported number of possible configurations is based on permutations, the order of Individual Child Realizations is significant. E.g., 1* itemA followed 1 * itemB is different from 1* itemB followed by itemA. How many configurations a floating point attribute with range 0 .. 10 generates? We count (conservatively) 1000 values per one unit of integer, in this case 10000. In this configuration model a smallish number is semantically more justified. Constraints are not taken into account.

3.1 Experiences in use

The system went to production at Elematic in January 2020. Over 40 offers have been created and sent to the customers. The most significant benefits are the reduction of human errors, faster quoting and the beginner-friendly user interface that allows new employees to easily learn the products and make quotes.

Manual quoting is error-prone, because one has to remember what quote lines (items) are required in the plant. The configurator ensures the completeness of the quote. The quality of quotes has increased. A user interface with more guidance reduces the number of mistakes.

The effort of creating a quote has reduced from typical 2 - 3 hours to 15 - 30 minutes. Thus, the quote can be prepared at site when a sales director visits a customer. This reduces the chance for miscommunication during the quoting phase. Off-line functionality makes it possible to create quotations even without a network connection. Furthermore, the system assembles a substantial attachment document of technical specifications: Products have attached PDF docu-

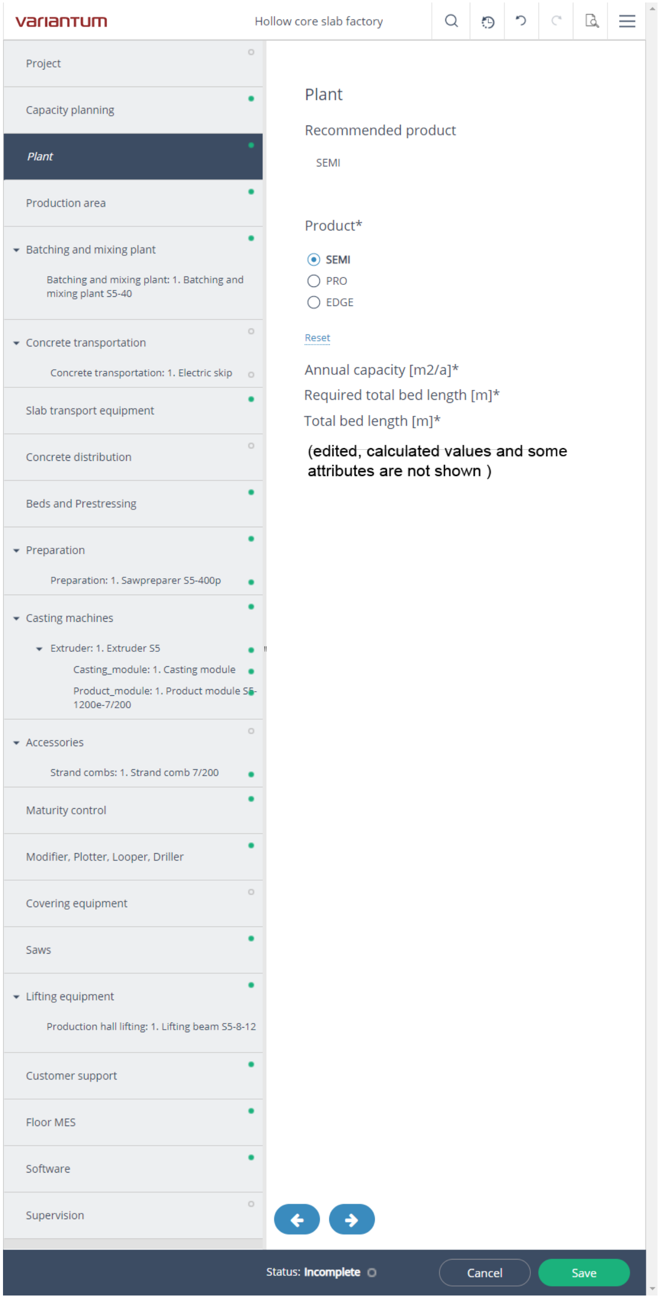


Figure 6. Once the PRODUCT has been selected, most attributes have a value and most children have a realization. The green dots on the left pane indicate consistent and complete "groups".

ment(s) for technical specifications, often in various languages. The system collects the technical specifications in the selected language from each product in the quote and assembles them into a single file in the same order as the products are shown in the quote. This makes it easy for the customer to navigate technical specifications – essential because the document can easily be over 100 pages long and include technical specifications of over 100 different products.

There are no performance challenges. Initializing a configuration session takes some seconds. Afterwards, the system responds with a feeling of instant response.

Attribute	Optionality	Origin	Default expression
COUNTRY	required	user	
PRODUCT	required	user	
BED_LENGTH	required	user	if PRODUCT = "SEMI" then 120 else 150
BEDS	required	user	if PRODUCT = "SEMI" then 4 else if PRODUCT = "PRO" then 6 else 8
TOTAL_BED_LENGTH	required	expression	BEDS * BED_LENGTH
CAPACITY	required	expression	round(BEDS * BED_LENGTH * (rest not disclosed)
SHUTTLE_TRACK_SUPPORT_DISTANCE	required	user	6
PRESTRESSER_TYPE	required	user	if PRODUCT = "EDGE" then "E9-200" else "P7-16"
BED_TYPE	required	user	if PRODUCT = "SEMI" then "S5" else "E9"
BED_HEATING_AND_INSULATION	required	user	if BED_TYPE = "S5" then false else true
BED_SUPPLIER	required	user	PREFERRED_SUPPLIER
DISTRIBUTOR_TYPE	required	external	
DISTRIBUTOR_LIFTING	required	user	if PRODUCT = "EDGE" and DISTRIBUTOR_TYPE = "P7" then true else false
PREFERRED_SUPPLIER	required	user	if null(COUNTRY) then "FI" else if COUNTRY = "IN" then "IN" else "FI"
WORKING_DAYS	required	user	220
CASTINGS_PER_DAY	required	user	if RECOMMENDED_PRODUCT = "EDGE" then 2.0 else 1.0
RECOMMENDED_PRODUCT	optional	expression	(not disclosed)
PA_HEIGHT_MIN	required	expression	if childcount("Concrete transportation","PC1711") > 0 then 8000 else 7000 (line moved with Photoshop)

Figure 7. Attribute properties significantly affect the behavior of an attribute. Many attributes of Origin "user" have dynamic defaults computed from other attributes. Some attributes such as TOTAL_BED_LENGTH or CAPACITY have Origin "expression". The default expression provides a value that the user cannot modify. In addition, DISTRIBUTOR_TYPE is determined with a script, Origin "external".

		revision Latest	relationship Position	.. ..	revision Description	revi… Uni…	revision Sales price	revision Currency
PC5712410/A		✔	Site lifting	0	en:Site lifting beam P7-8-15	pc(s)	33	EUR
PC5712500/A		✔	Site lifting	0	en:Site lifting beam P7-10-18	pc(s)	.28	EUR
PC5712600/A		✔	Site lifting	0	en:Site lifting beam P7-12-15	pc(s)	.44	EUR
PC1527/0	p a+b	✔	Slipformer	3	en:Slipformer P7			EUR
PC5701410/A		✔	Storage lifting	0	en:Clamp S5-4	pair	15	EUR
PC5701600/A		✔	Storage lifting	0	en:Storage clamps E9-16	pair	.27	EUR
PC5611401/B	p	✔	Strand combs	0	en:Strand comb 8/100	pair	9	EUR
PC5611402/B	p	✔	Strand combs	0	en:Strand comb 8/120	pair	9	EUR
PC5611405/B	p	✔	Strand combs	0	en:Strand comb 8/150	pair	9	EUR
PC5611410/B	p	✔	Strand combs	0	en:Strand comb 8/160 (165)	pair	9	EUR
PC5611410I/A	p	✔	Strand combs	0	en:Strand comb 8/160	pair	.6	INR
PC5611415/D	p	✔	Strand combs	0	en:Strand comb 7/150	pair		EUR
PC5611416/C	p	✔	Strand combs	0	en:Strand comb 7/160	pair	3	EUR
PC5611417/B	p	✔	Strand combs	0	en:Strand comb 7/200	pair	56	EUR
PC5611420/G	p	✔	Strand combs	0	en:Strand comb 6/200	pair	56	EUR
PC5611420I/A	p	✔	Strand combs	0	en:Strand comb 6/200	pair	.53	INR
PC5611422/B	p	✔	Strand combs	0	en:Strand comb 6/220	pair	56	EUR
PC5611423/A	p	✔	Strand combs	0	en:Strand comb 7/220-1500	pair	55	EUR

Figure 8. The child structure defines possible items (the leftmost column). Column "relationship Position" defines the childID. Some items used have a price, here in EUR or INR. Some items such as for child Slipformer do not have a price: the price of a configurable item is determined by configuration decisions. The actual prices are not disclosed.

3.2 Modeling Experiences

As can be inferred from Figure 7, some components (e.g., BED_TYPE) are represented as enumerated attribute values. This works well if there is little to configure in the components. However, genuine structure-based modeling soon becomes a much better alternative if the selected components are configurable with their own "business rules" or constraints. For genuine objects (Items), it is also easy to add information such as documents, images and other attributes that can be made available for the person configuring or as outputs of a configuration process.

According to our experiences, most benefit for "basic" level of modeling with genuine structures instead of attributes comes from configurable, repetitive components. The example case supports 4 types of saws: PC1706, PC1707, PC1708, and PC1709). There can be up to 6 saws (Figure 9). The available saws support different length of blades (enumerated integer SAW_BLADE) and some support automation (Boolean SAW_AUTOMATION) There are additionally simple constraints. In this case they are the same for all saws, but this is not always the case. With "flat" attributes only, we would have attributes like SAW_TYPE_1 ... SAW_TYPE_6, SAW_BLADE_1 ... SAW_BLADE_6 and SAW_AUTOMATION_1

Child	Min count	Max count	Similarity	Default
Slipformer	0	1	any	none
Saw	0	6	any	PC1709 if (PRODUCT = "EDGE") then 2 else 0 PC1708 if (PRODUCT = "PRO") then 1 else 0 none if (PRODUCT = "SEMI") then 1 else 0
Storage lifting	0	5	any	PC5701600 if (PRODUCT = "EDGE") or (PRODUCT = "PRO") then 1 else 0 none if (PRODUCT = "SEMI") then 1 else 0
Site lifting	0	5	any	PC5712300 if (PRODUCT = "EDGE") then 2 else 0 PC5712300 if (PRODUCT = "PRO") then 1 else 0 none if (PRODUCT = "SEMI") then 1 else 0
Other lifting	0	5	any	none
Heating plant	0	4	any	none
Strand combs	0	100	any	PC5611401 if "8/100" in (PRODUCT_COMBINATIONS) then BEDS else 0 PC5611402 if "8/120" in (PRODUCT_COMBINATIONS) then BEDS else 0 PC5611405 if "8/150" in (PRODUCT_COMBINATIONS) then BEDS else 0 PC5611410 if "8/160" in (PRODUCT_COMBINATIONS) and STRAND_COMBS_SUPPLIER <> "IN" then BEDS else 0 PC5611410I if "8/160" in (PRODUCT_COMBINATIONS) and STRAND_COMBS_SUPPLIER = "IN" then BEDS else 0

Figure 9. Child properties. Columns "Min Count" and "Max Count" define the cardinality of the child specified in column "Child". All these have "Similarity" "any". Here, all shown children have child defaults. Most are conditional, e.g. based on the selected `PRODUCT` attribute. For example, selecting `PRODUCT = EDGE` implies creating for child `Saw` a child realization with an entry that has count 2 of item individual of type `PC1709`. `Strand combs` child defaults set the count of the default to the number of `BEDS` as one pair is required for each bed. There are 28 strand comb child defaults

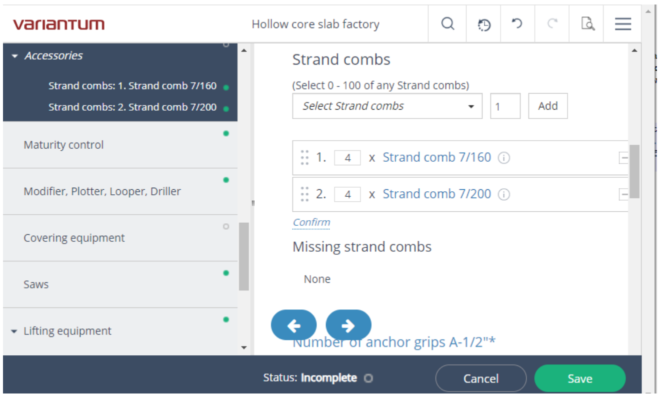


Figure 10. An example of applying `Strand combs` child defaults of Figure 9 in a configuration with four `BEDS`.

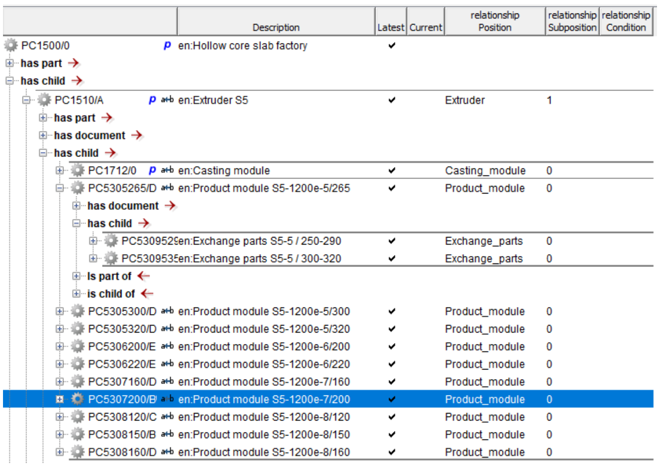


Figure 11. The deepest product structure of the factory: The top-level item is connected with three "has-child" relationships with the lowest level possible items.

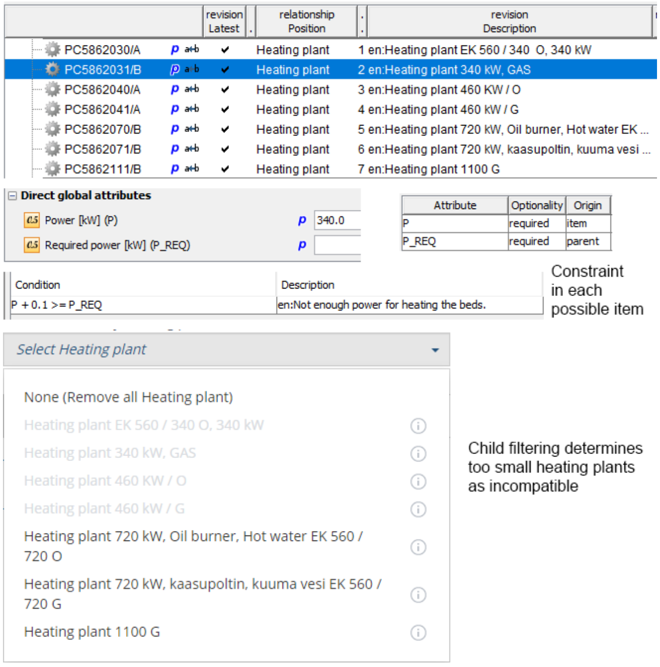


Figure 12. Child filtering for heating plants. Each possible heating plant defines its fixed power output `P` (Origin "item") and gets the currently required power `P_REQ` from its parent (Origin "parent"). A constraint specifies that $P \geq P\_REQ + 0.1$. In the user interface, too small heating plants are detected as incompatible.

...`SAW_AUTOMATION_6`. We would have to create constraints that ensure that for each saw that allowed values of each saw type are respected, like `SAW_TYPE_1 = "PC1706" implies SAW_BLADE_1 in (800, 1100)` and `SAW_TYPE_1 = "PC1706" implies not (SAW_AUTOMATION_1)`. Because different saws are Items with their own allowed attribute values and constraints, each Item has this information "built-in"; there is no need to repeat the "rules" or possible values each instance. Furthermore, to help the user in the case of repetitive attributes, we would probably create visibility rules to reveal attributes representing the second saw only if the first saw is selected. Furthermore, price calculations based on such attributes quickly become very complex.

The Elematic model has many structurally configurable elements, and many of them can occur repetitively. Managing that kind of complexity would not have been feasible, or at least maintainable, without genuine structure based configuration. The number of attributes and required rules would have grown to unmanageable proportions.

The calculation mechanism with origin "expression" proved valuable for determining required technical properties for dimensioning, e.g., required production area height or total bed length. Component selection (type, count) is supported with conditional child defaults. For some child definitions, such as the `Heating plant`, calculated values are used as input values for child filtering.

4 DISCUSSION

Comparison of our technical solution with state-of-the-art is limited due to scarce detailed and peer-reviewed literature on product structure configuration. Our conceptual model w.r.t structure modeling is the same except for naming as that of [19, 21]. Child defaults are now dynamic (conditional) instead of static. Child filtering, child ranking and child select are new mechanisms. Child access functions are applied instead of a direct mechanism for accessing child attributes.

This approach makes it possible to create multi-level configuration models that are configured as a whole, with corresponding multi-level configurations. In some cases, such as Elematic, detailed enough configurations of factory, system and equipment level can be created in sales configuration tools. In other words, it is not necessary to have separate plant, system and equipment configurator layers as has been applied in some contexts [6]. However, Elematic still has a separate production configuration task: the sales configurations from quotes do not go directly to production.

Child ranking and child filtering were not available during modeling the major parts of the Elematic model. We expect that other configuration models would use them more extensively. Other cases indicate that they can be highly useful, e.g. for selecting the smallest pump that meets computed technical requirements.

As was mentioned in Section 2, the CSP models of different items are separate. Attribute assignments via Origin "parent" attributes and child access functions provide mechanisms for their communication. As a result, universal quantification or aggregation is not possible at the CSP level. Aggregation can be performed with VariSales Engineering Calculation System child access functions in a way that is practical, e.g. it is possible to collect the number of different units, power consumption etc. from different parts of the configuration. The result can be used for inference (by setting it to an attribute), but it cannot automatically "feed back" information to achieve the desired result with respect to some criteria. Global optimization would require separately developed optimization models connected as configurator extensions.

A similar business case of cement plants [13] provides limited technical details. The reported benefits include effort and lead-time savings and improved quality of configurations, similar to ours. Their solution includes CAD integration while our solution does not.

5 CONCLUSIONS

We presented the VariSales Engineering Calculation System with mechanisms for calculations and sales configuration with genuinely varying product structures using the child concept. This is a contribution for scientific literature. However, some configuration system vendors may have similar functionality. The approach combines CSP as a problem solving mechanism with traditional data models. The approach is modular and seems to scale well. The huge potential number of configurations of the case ($2 * 10^{1411}$ potential configurations) causes no performance or memory challenges.

We applied many of the modeling mechanisms to the Elematic case. We also regard this as a contribution. First, it gives real industrial validation for the applicability of the approach. Despite author bias, we can claim that the approach works well and feels like a practical way for modeling. Elematic has, independently from Variantum, modeled another product family. Second, there are not too many real industrial cases described even at this level of detail.

ACKNOWLEDGEMENTS

We thank the reviewers and Hannu Peltonen for their comments that helped to improve this paper considerably. The authors affiliated with Variantum thank Elematic for allowing to describe their case and for sharing their experiences.

REFERENCES

[1] Michel Aldanondo, Khaled Hadj-Hamou, Guillaume Moynard, and Jacques Lamothe, 'Mass customization and configuration: Requirement analysis and constraint based modeling propositions', *Integrated Computer-Aided Engineering*, **10**(2), 177–189, (2003).

[2] Michel Aldanondo, Sylvie Rouge, and Mathieu Véron, 'Expert configurator for concurrent engineering: Cameleon software and model', *Journal of Intelligent Manufacturing*, **11**(2), 127–134, (2000).

[3] Alessandro Artale, Enrico Franconi, Nicola Guarino, and Luca Pazzi, 'Part-whole relations in object-centered systems: An overview', *Data & Knowledge Engineering*, **20**(3), 347–383, (1996).

[4] R Cunis, A Günter, I Syska, H Peters, and H Bode, 'PLAKON - an approach to domain-independent construction', in *Proc. of the Second International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems (IEA/AIE - 89)*, ed., A Monnis, volume 2, pp. 866–874, New York, NY, USA, (1989). ACM.

[5] Gerhard Fleischanderl, Gerhard E Friedrich, Alois Haselböck, Herwig Schreiner, and Markus Stumptner, 'Configuring large systems using generative constraint satisfaction', *IEEE Intelligent Systems*, **13**(4), 59–68, (1998).

[6] Amartya Ghosh, Anna Myrodia, Lars Hvam, and Niels Henrik Mortensen, 'Reusing modules across configuration systems', in *21st Configuration Workshop, Hamburg, Germany, 2019*, volume 2467 of *CEUR Workshop Proceedings*, pp. 53–60. CEUR-WS.org, (2019).

[7] S. Gregor, 'The nature of theory in information systems', *MIS Quarterly*, **30**(3), 611–642, (2006).

[8] Petri Helo, Sami Kyllönen, and Samuli Pylkkönen, 'Features of 3d graphics in sales configuration', in *19 th International Configuration Workshop*, pp. 33–37, (2017).

[9] A. R. Hevner, S. T. March, J. Park, and S. Ram, 'Design science in information systems research', *MIS Quarterly*, **28**(1), 75–105, (2004).

[10] L. Hotz, A. Felfernig, M. Stumptner, A. Ryabokon, C. Bagley, and K. Wolter, 'Configuration Knowledge Representation and Reasoning', in *Knowledge-based Configuration*, ed., A. Felfernig et al., chapter 6, 41–72, Morgan Kaufmann, Waltham, MA, USA, (2014).

[11] John McDermott, 'R1: A Rule-based configurer of computer systems', *Artificial Intelligence*, **19**(1), 39–88, (1982).

[12] Sanjay Mittal and Felix Frayman, 'Towards a generic model of configuration tasks', in *11th International Joint Conference on Artificial Intelligence (IJCAI-89)*, volume 2, pp. 1395–1401, (1989).

[13] Klas Orsvärn and Morten H. Bennick, 'Chapter 17 - Tacton: Use of Tacton configurator at FLSmidth', in *Knowledge-Based Configuration*, ed., A. Felfernig et al., 211 – 218, Morgan Kaufmann, Boston, (2014).

[14] Charles Prud'homme, Jean-Guillaume Fages, and Xavier Lorca, *Choco Documentation*, TASC, INRIA Rennes, LINA CNRS UMR 6241, COSLING S.A.S. www.choco-solver.org, 2016.

[15] Timo Soininen, Juha Tiihonen, Tomi Männistö, and Reijo Sulonen, 'Towards a general ontology of configuration', *AI EDAM*, **12**(4), 357–372, (1998).

[16] Markus Stumptner, 'An overview of knowledge-based configuration', *AI Communications*, **10**(2), 111–125, (1997).

[17] Markus Stumptner, Gerhard E Friedrich, and Alois Haselböck, 'Generative constraint-based configuration of large technical systems', *AI EDAM*, **12**(4), 307–320, (1998).

[18] Juha Tiihonen, *Support for configuration of physical products and services*, Ph.D. dissertation, Aalto University, School of Science, Helsinki, Finland, 2014.

[19] Juha Tiihonen and Andreas Anderson, 'Chapter 26 - VariSales', in *Knowledge-Based Configuration*, eds., A. Felfernig, L. Hotz, C. Bagley, and J. Tiihonen, 309–318, Morgan Kaufmann, Boston, (2014).

[20] Juha Tiihonen and Timo Soininen, 'State-of-the-practice in product configuration - a survey of 10 cases in the Finnish industry', in *Knowledge Intensive CAD.*, eds., T. Tomiyama, M. Mäntylä, and S. Finger, volume 1 of IFIP, pp. 95–114, London, (1996). Chapman & Hall.

[21] Juha Tiihonen, Timo Soininen, Ilkka Niemelä, and Reijo Sulonen, 'A Practical Tool for Mass-Customising Configurable Products', in *Proceedings of the 14th International Conference on Engineering Design*, volume Stockholm,, pp. CDROM, paper number 1290, 10 pp., (2003).

[22] E A Van Veen, *Modelling product structures by generic bills-of-materials*, Elsevier Science Inc., New York, NY, USA, 1992.

[23] Bob Wielinga and Guus Schreiber, 'Configuration-design problem solving', *IEEE Expert*, **12**(2), 49–56, (1997).

Bi-objectives configuration optimization for smart product service with hybrid uncertain instance attributes

Zhihua Chen[1, 2] and **Elise Vareilles**[2*] and **Olga Battaia**[3] and **Xinguo Ming**[1]

Abstract. The application of advanced smart technologies enables the traditional product extension service transform toward smart product service (SPS) that has three emerging characteristics: smarter capabilities of service delivery, involvement of more stakeholders and higher implementation cost of smart technologies. These emerging characteristics have brought new challenges for SPS configuration, such as, definition of smart capability of SPS module instances, manipulation of multiple objectives of configuration optimization, hybrid uncertainties involved in determination of instance attributes and high speed solution of optimization model. However, the previous research rarely simultaneously handle these challenges of SPS configuration optimization. Therefore, this study proposes a bi-objectives optimization model for SPS configuration considering the smart capability and service cost. The attributes value (configuration parameter) of SPS module instances are obtained using rough-fuzzy matrix with objectives of handling the hybrid uncertainties involved in the group decision environment. Moreover, an adaptive non-dominated sorting genetic algorithm II (ANSGA-II) is applied to obtain a pareto-set of optimal configuration solutions with high convergence speed and accuracy. Finally, an illustrative example of smart vehicle service configuration optimization and some comparisons with other approaches are carried out to demonstrate the feasibility and potential of the proposed approach.

1 INTRODUCTION

Smart product service system (PSS) has emerged as the mainstream offering to achieve higher market competiveness, customer satisfaction and environmental sustainability [1, 2]. Smart PSS comprises a smart connected product (SCP) and various high-valued smart product service (SPS) modules that are provided to create added value profits to customers, such as optimization of product performance, reduction of product failure and extending of product useful life on the basis of application and implementation of smart technologies (e.g. smart sensing, Internet of Things, cyber-physical system, digital-twin (DT) and artificial intelligence (AI)) [3-5]. Specific examples of such SPS modules can be smart maintenance service module based on AI and DT [6], data-driven product's operational optimization module [7], smart technology-enabled proactive recovery service module [8], big data-driven energy saving module [9], smart service of spare parts supply [3] and product insurance module based on the historical operational data [10].

Compared with traditional product service that is mainly delivered in physical product use cycle without much supporting of advanced smart technologies [11, 12], SPS perform three emerging features: (1) smarter capabilities of service delivery (e.g. monitor capability, predictive capability) which enable the operation of product service more agile and dynamic so as to achieve higher customer satisfaction [7, 13]; (2) involvement of more stakeholders in the lifecycle of SPS, such as the service designers, operators, smart system providers and field service providers, which lead to a group decision environment for development, implementation and operation of SPS [14]; and (3) higher cost for manufacturers and customers caused by the application of smart technologies [15]. These emerging features lead to some new challenges for the configuration of SPS whose aim is to select the best portfolios of pre-defined smart service module instances (SMIs) from the module instances library with respect to certain objectives and constraints in the specific scenarios [16]. Firstly, it is necessary to define new attributes (configuration parameters) of SMIs for SPS configuration in the light of smart capability. The SPS configuration should take both the smart capability and cost into account to achieve balance between the customer experiences and use cost. Secondly, the smart attributes and cost of each SMIs are normally obtained through collecting linguistic responses from multiple stakeholders in the early stage of conceptual design due to the scant historical data and short time frame [17]. This group decision process inherently involve two types of uncertainties, i.e. intrapersonal uncertainty caused by the individual linguistic vagueness and interpersonal uncertainty led by the group preference randomness [18, 19]. Without feasible uncertainty handling method, the two uncertainties would have a hybrid influence on the accuracy of configuration parameters of SMIs and configuration results of final SPS scheme. Thirdly, higher speed and accuracy to obtain configuration solutions are required for agile customization requirement. However, most the previous research cannot simultaneously manipulate the three challenges of SPS configuration optimization.

Orienting to the multiple objectives of SPS configuration, among the previously used approaches for product service configuration, such as ontology-based knowledge-driven reasoning methods [20-23], data-driven mapping methods [24, 25] and combinatory optimization methods [16, 26], the combinatory optimization methods have been validated and proven more feasible and effective for manipulating multiple configuration objectives. Therefore, the combinatory optimization approach is used as the basic framework for SPS configuration optimization in

[1] Department of Industrial Engineering and Management, School of Mechanical Engineering, Shanghai Jiao Tong University, China, email: zh03.chen@sjtu.edu.cn
[2] ISAE-SUPAERO, Université de Toulouse, France, email: Elise.VAREILLES@isae-supaero.fr
[3] Kedge Business School (Talence), France, email: olga.battaia@kedgebs.com

the current study. However, the previously used attributes for describing service module instances are not adaptive for the SMIs for SPS. It is thus necessary to establish a novel attributes system to represent and define the configuration parameters of SMIs. Moreover, in the previous research [16, 21], the SMIs' configuration parameters are determined using deterministic methods (e.g. analytic hierarchy process (AHP), correlation matrix) without considering the hybrid uncertainties involved in the group decision process. Thus, it is necessary to develop a feasible method to handle these hybrid uncertainties in the determination of SMIs' parameters. In addition, the previously used algorithms (such as standard GA and NSGA-II) for solving multi-objectives optimization model of product service configuration often suffer local optimal and worse convergence performance, and they cannot provide fast configuration speed in the context of dynamic customization of SPS. It will be thus a meaningful attempt to adopt new algorithm to obtain higher solving speed.

Therefore, in this study, a systematic approach to bi-objectives optimization of SPS configuration is proposed, with considering the smart capability and service cost as two essential objectives. A new optimization problem of SPS configuration is formulated according to the emerging characteristics of SPS. Moreover, a rough-fuzzy matrix method is introduced to determine the comprehensive performance of each SMI with respect to the constructed attributes system under hybrid uncertain environments. Then, a bi-objectives optimization model for SPS configuration under combination constraints is proposed, and finally solved by integrating an adaptive generation operators [27] and NSGA-II [16] with objective of obtaining a Pareto set of optimal configuration solutions under higher convergence speed. Finally, an illustrative example of the proposed model's application in smart vehicle service is carried out to demonstrate the feasibility and potential.

2 PROBLEM FORMULATION OF SPS CONFIGURATION OPTIMIZATION

The basic concept of SPS configuration optimization is shown in Figure 1. The configuration optimization model includes three spaces: SPS space, instance attributes space and customer space. SPS space represents a library of various pre-defined SPS module instances, among whom exists knowledge-based combination rules. Instance attributes space incorporates the parameters used to define the features of each SMI. Customer space is applied to instore the customer expectation and requirements. The goal of configuration optimization is to select the most appropriate portfolio of SMIs from the SPS space whose performance are measured by the instance attributes space, with respect to the customer expectation in the customer space.

In the SPS space, i.e. the left part of Figure 1, a SPS concept is composed of various service modules (SMs) that are clustered into different service scopes. Five service scopes are identified as product use-oriented (PUO), product sharing-oriented (PSO), user activity-oriented (UAO), product feedback-oriented (PFO) and industry ecosystem-oriented (PIO) [3]. In each service scope, various SMs are predesignated and generated based on the customer requirements and the manufacturer marketing strategy. For example, in the PUO-SS, operational optimization module, smart repair module, smart maintenance module and smart recovery module are generated to reduce the product operating failure, improve performance, extend product useful life and increase product use rate, etc. Such SMs can be represented by various SMIs with different attributes value. For example, smart maintenance module can be represented by three SMIs: SMI1 {proactive maintenance module with good predictive and prescriptive capabilities for product fault and degradation}, SMI2 {predictive maintenance module with an excellent predictive capability for product fault and degradation}, SMI3 {reactive maintenance with good monitor and diagnostic capabilities for product fault and degradation}. In addition, some combination rules exist between each pair of SMIs. The combination rules mainly comprise two types: inclusive and exclusive. Inclusive rule means that if one instance is selected, the other one must also be selected. Exclusive rule means that if one instance is selected, the other one cannot be selected. These rules are identified based on the professional domain knowledge.

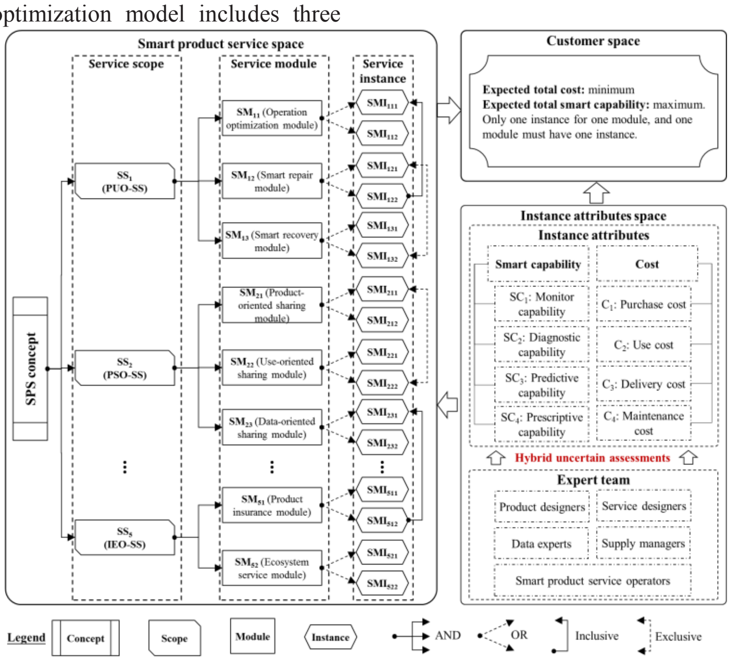


Figure 1. SPS configuration optimization model

In the instance attributes space, two types of attributes, i.e. smart capability and cost, are identified to measure the performance of SMIs. Smart capability and cost respectively consists of four sub-attributes as shown in Figure 1. The comprehensive performance of SMI are obtained by inviting the expert team which includes product designers, service designers, smart technique experts, mangers, etc. to provide linguistic judgements based on their own expertise and knowledge. The group evaluation process inherently involves two types of uncertainties [19]: intrapersonal uncertainty caused by the individual linguistic ambiguousness and interpersonal uncertainty led by the multiple preference randomness. These hybrid uncertainties will be manipulated by converting the group linguistic assessments into rough-fuzzy numbers [3]. In the customer space, as depicted in the right-up part of Figure 1, the customer expectation are the goal or constraints for SPS configuration optimization. In this study, the total smart capability and cost of the configured SPS scheme are considered as two objectives and the requirement "one instance must be selected for one module" is considered as one essential optimization constraint.

3 ROUGH-FUZZY BI-OBJECTIVES OPTIMIZATION MODEL FOR SPS CONFIGURATION

3.1 Evaluation of attributes value of SPS module instances

In this section, a rough-fuzzy method is proposed to determine the SMIs' configuration parameters under hybrid uncertain environments. The smart capability and cost of SMIs can be respectively obtained by the following procedures.

First, an expert team including r members with wide domain expertise is invited to make linguistic judgement of SMIs' performance with respect to each attribute. The linguistic score matrix A_p carried out by the p^{th} expert is established as follows:

$$A_p = \begin{array}{c} \\ SMI_{111} \\ \vdots \\ SMI_{ijk} \\ \vdots \\ SMI_{nm_n o_{mn}} \end{array} \begin{bmatrix} SC_1 & \cdots & SC_s & \cdots & SC_t \\ a^p_{1111} & \cdots & a^p_{111s} & \cdots & a^p_{111t} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ a^p_{ijk1} & \cdots & a^p_{ijks} & \cdots & a^p_{ijkt} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ a^p_{nm_n o_{mn}1} & \cdots & a^p_{nm_n o_{mn}s} & \cdots & a^p_{nm_n o_{mn}t} \end{bmatrix} \tag{1}$$

where SC_s represents the s^{th} smart capability attribute (s=1, 2, …, t), a^p_{ijks} refers to the p^{th} expert's linguistic score for the kth module instance of the j^{th} service module in the i^{th} service scope, and n_s, n_i and n_{ij} separately denotes the number of service scopes, number of service modules of the n^{th} service scope and number of instances of the m_n^{th} module in the n^{th} service scope, the numbers of total SPS module instances $N=\sum_{i=1}^{n_s}\sum_{j=1}^{n_i}\sum_{k=1}^{n_{ij}} ijk$.

By combining the r experts' linguistic matrices together, the group linguistic score matrices A_p can be obtained as $A=[a_{ijks}]_{N\times t}$, where $a_{ijks}=\{a^1_{ijks},\ldots,a^p_{ijks},\ldots,a^r_{ijks}\}$ is a set of r experts' assessments. According to the triangular fuzzy logic, a linguistic term a^p_{ijks} can be transformed to a triangular fuzzy number $\tilde{a}^p_{ijks}=(a^{pl}_{ijks},a^{pm}_{ijks},a^{pu}_{ijks})$ based on the following membership functions:

$$f_{\tilde{a}^p_{ijks}}(a^p_{ijks})=\begin{cases} 0, & a^p_{ijks}<a^{pl}_{ijks} \\ \dfrac{a^p_{ijks}-a^{pl}_{ijks}}{a^{pm}_{ijks}-a^{pl}_{ijks}}, & a^{pl}_{ijks}\le a^p_{ijks}\le a^{pm}_{ijks} \\ \dfrac{a^{pu}_{ijks}-a^p_{ijks}}{a^{pu}_{ijks}-a^{pm}_{ijks}}, & a^{pm}_{ijks}\le a^p_{ijks}\le a^{pu}_{ijks} \\ 0, & a^p_{ijks}>a^{pu}_{ijks} \end{cases} \tag{2}$$

By following Eq.(2), the group linguistic score matrices $A=[a_{ijks}]_{N\times t}$ can be converted into group fuzzy score matrices $\hat{A}=[\hat{a}_{ijks}]_{N\times t}$ where $\hat{a}_{ijk}=(\hat{a}^l_{ijks},\hat{a}^m_{ijks},\hat{a}^u_{ijks})$. By referring Chen, et al. [18], the three boundaries of the group fuzzy numbers $\hat{a}_{ijk}$ can be separately transferred to rough-fuzzy forms $RF(\hat{a}^l_{ijks})=[a^{lL}_{ijks},a^{lU}_{ijks}]$, $RF(\hat{a}^m_{ijks})=[a^{mL}_{ijks},a^{mU}_{ijks}]$ and $RF(\hat{a}^u_{ijks})=[a^{uL}_{ijks},a^{uU}_{ijks}]$. Thus, the rough-fuzzy form of $\hat{a}_{ijk}$ can be obtained as:

$$RF(\hat{a}_{ijks})=[(a^{lL}_{ijks},a^{mL}_{ijks},a^{uL}_{ijks}),(a^{lU}_{ijks},a^{mU}_{ijks},a^{uU}_{ijks})] \tag{3}$$

In this way, the group fuzzy score matrices $\hat{A}$ can be transited to a rough-fuzzy score matrix as $RF(\hat{A})=[RF(\hat{a}_{ijks})]_{N\times t}$. Considering the weight of each smart capability, the comprehensive smart capability $RF(\vec{a})=[RF(a_{ijk})]_{N\times 1}$, where $RF(a_{ijk})=\sum_{s=1}^{t}RF(\hat{a}_{ijks})w^a_s$ represents the ability of the instance to satisfy the smart attribute; w^a_s denotes the weight of the s^{th} smart capability and $\sum_{1\le s\le t}w^a_s=1$. $RF(a_{ijk})$ can be thus rewritten as follows:

$$\begin{aligned} RF(a_{ijk}) &=[(a^{lL}_{ijk},a^{mL}_{ijk},a^{uL}_{ijk}),(a^{lU}_{ijk},a^{mU}_{ijk},a^{uU}_{ijk})] \\ &=\begin{bmatrix} (\sum_{s=1}^{t}a^{lL}_{ijks}w^a_s,\sum_{s=1}^{t}a^{mL}_{ijks}w^a_s,\sum_{s=1}^{t}a^{uL}_{ijks}w^a_s), \\ (\sum_{s=1}^{t}a^{lU}_{ijks}w^a_s,\sum_{s=1}^{t}a^{mU}_{ijks}w^a_s,\sum_{s=1}^{t}a^{uU}_{ijks}w^a_s) \end{bmatrix} \end{aligned} \tag{4}$$

Similarly, SMIs' cost $RF(\vec{c})=[RF(c_{ijk})]_{N\times 1}$ can be obtained using the same calculation procedure as comprehensive smart capability.

3.2 Constructing rough-fuzzy bi-objectives configuration optimization model

3.2.1 *Bi-objectives of configuration optimization*

In order to obtain higher performance on the total smart capability, the first optimization objective is to maximize the total smart capability of the SPS scheme, expressed as follows:

$$\max RF(A)=\sum_{i=1}^{n_s}\sum_{j=1}^{n_i}\sum_{k=1}^{n_{ij}}x_{ijk}RF(a_{ijk}) \tag{5}$$

where $RF(A)$ is the rough-fuzzy total smart capability of the configured SPS scheme that consists of multiple SMIs, x_{ijk} denotes the binary decision variable that indicates if the kth SMI of the j^{th} SM in the i^{th} SS is selected or not. When x_{ijk} = 1, the corresponding SMI is selected, or x_{ijk} = 0, the corresponding one is not selected.

As expectation of customers, the total cost of the configured SPS scheme should be as low as possible. Thus, the second optimization objective is to minimize the total cost of the SPS scheme, expressed as follows:

$$\min RF(C)=\sum_{i=1}^{n_s}\sum_{j=1}^{n_i}\sum_{k=1}^{n_{ij}} x_{ijk} RF(c_{ijk}) \tag{6}$$

where $RF(C)$ represents the rough-fuzzy form of total cost of the configured SPS solution.

These two rough-fuzzy objective functions can be rewritten in the extended form as follows:

$$\min -RF(A)=-\sum_{i=1}^{n_s}\sum_{j=1}^{n_i}\sum_{k=1}^{n_{ij}}\left(x_{ijk}\sum_{s=1}^{t} RF(\hat{a}_{ijks})w_s^a\right)$$

i.e. (7)

$$\begin{cases} \min -A^{lL}=-\sum_{i=1}^{n_s}\sum_{j=1}^{n_i}\sum_{k=1}^{n_{ij}}\left(x_{ijk}a_{ijk}^{lL}\right); \min -A^{mL}=-\sum_{i=1}^{n_s}\sum_{j=1}^{n_i}\sum_{k=1}^{n_{ij}}\left(x_{ijk}a_{ijk}^{mL}\right);\\ \min -A^{uL}=-\sum_{i=1}^{n_s}\sum_{j=1}^{n_i}\sum_{k=1}^{n_{ij}}\left(x_{ijk}a_{ijk}^{uL}\right); \min -A^{lU}=-\sum_{i=1}^{n_s}\sum_{j=1}^{n_i}\sum_{k=1}^{n_{ij}}\left(x_{ijk}a_{ijk}^{lU}\right);\\ \min -A^{mU}=-\sum_{i=1}^{n_s}\sum_{j=1}^{n_i}\sum_{k=1}^{n_{ij}}\left(x_{ijk}a_{ijk}^{mU}\right); \min -A^{uU}=-\sum_{i=1}^{n_s}\sum_{j=1}^{n_i}\sum_{k=1}^{n_{ij}}\left(x_{ijk}a_{ijk}^{uU}\right); \end{cases}$$

$$\min RF(C)=\sum_{i=1}^{n_s}\sum_{j=1}^{n_i}\sum_{k=1}^{n_{ij}}\left(x_{ijk}\sum_{s=1}^{t} RF(\hat{c}_{ijks})w_s^c\right)$$

i.e. (8)

$$\begin{cases} \min C^{lL}=\sum_{i=1}^{n_s}\sum_{j=1}^{n_i}\sum_{k=1}^{n_{ij}}\left(x_{ijk}c_{ijk}^{lL}\right); \min C^{mL}=\sum_{i=1}^{n_s}\sum_{j=1}^{n_i}\sum_{k=1}^{n_{ij}}\left(x_{ijk}c_{ijk}^{mL}\right);\\ \min C^{uL}=\sum_{i=1}^{n_s}\sum_{j=1}^{n_i}\sum_{k=1}^{n_{ij}}\left(x_{ijk}c_{ijk}^{uL}\right); \min C^{lU}=\sum_{i=1}^{n_s}\sum_{j=1}^{n_i}\sum_{k=1}^{n_{ij}}\left(x_{ijk}c_{ijk}^{lU}\right);\\ \min C^{mU}=\sum_{i=1}^{n_s}\sum_{j=1}^{n_i}\sum_{k=1}^{n_{ij}}\left(x_{ijk}c_{ijk}^{mU}\right); \min C^{uU}=\sum_{i=1}^{n_s}\sum_{j=1}^{n_i}\sum_{k=1}^{n_{ij}}\left(x_{ijk}c_{ijk}^{uU}\right); \end{cases}$$

3.2.2 Constraints of configuration optimization

The SPS configuration should consider the resource limitation and combination rules. Under resource limitation, the selection process should comply with the following conditions:

$$\sum_{k=1}^{n_{ij}} x_{ijk}=1; \sum_{j=1}^{n_i} x_{ijk}=n_i. \tag{9}$$

The first equation of constraint (9) means that each service module must be selected to constitute an integral SPS scheme. The second equation of constraint (9) indicates that only one instance can be chosen from one service module.

Weight constraints of the instance attributes should be in line with the following conditions:

$$\sum_{s=1}^{t} w_s^a=1, w_s^a>0; \sum_{s=1}^{t} w_s^c=1; w_s^c>0. \tag{10}$$

where w_s^c denotes the weight of the s^{th} cost attribute.

The combination between pairs of SPS module instances should consider some knowledge-based rules as follows:

$$\sum_{i=1}^{n_s}\sum_{j=1}^{n_i}\sum_{k=1}^{n_{ij}}(x_{i'j'k'}R_{ijk-i'j'k'})=\begin{cases}0, & x_{ijk}=0, \quad R_{ijk-i'j'k'}=1;\\ 0, & x_{ijk}=1, \quad R_{ijk-i'j'k'}=-1.\end{cases} \tag{11}$$

where $R_{ijk-i'j'k'}$ is the combination variable between SMI_{ijk} and $SMI_{i'j'k'}$, defined as follows:

$$R_{ijk-i'j'k'}=\begin{cases}1, \text{ if } SMI_{ijk} \text{ and } SMI_{i'j'k'} \text{ are mutually inclusive;}\\ -1, \text{ if } SMI_{ijk} \text{ and } SMI_{i'j'k'} \text{ are mutually exclusive;}\\ 0, \text{ otherwise.}\end{cases} \tag{12}$$

where "mutually inclusive" means the two instances must be selected together in a configuration scheme, and "mutually exclusive" cannot be combined into one same configuration process.

4 SOLUTION FOR SPS CONFIGURATION OPTIMIZATION MODEL

4.1.1 Dimensionality reduction

Since the extended form of the proposed configuration optimization model incorporates twelve objective functions, it is difficult to directly solve the models with so many objectives. It is necessary to use effective and accurate approach to dimensionality reduction. This section proposes a graphics-based aggregating operation approach for transforming the rough-fuzzy bi-objectives to crisp bi-objectives. The transformation process is described as follows:

Define a points set $P=\{p_k=(x_k,y_k)|k=1,2,\ldots,6\}$, where:

$$(x_k,y_k)\in\begin{Bmatrix}(-A^{lL},C^{lL}),(-A^{mL},C^{mL}),(-A^{uL},C^{uL}),\\ (-A^{lU},C^{lU}),(-A^{mU},C^{mU}),(-A^{uU},C^{uU})\end{Bmatrix} \tag{13}$$

In order to establish a closed polygon in which the six points of *P* are used as the six convex, it is first to transform the points set *P* to a new points set $P^*=\{p_k^*=(x_k^*,y_k^*)|k=1,2,\ldots,6\}$. In the points set P^*, all the points $p_k=(x_k,y_k)$ are reorganized in clockwise order. The operating process is described as Operation (1):

Operation (1): Define $\bar{p}=(\bar{x},\bar{y})$ as the center point of the points set *P*, where $\bar{x}=\sum_{1\le k\le 6}x_k/6$, $\bar{y}=\sum_{1\le k\le 6}y_k/6$. The relative coordinates $(\Delta x_k,\Delta y_k)$ of each point p_k to center point $\bar{p}$ is calculated as: $\Delta x_k=x_k-\bar{x}$, $\Delta y_k=y_k-\bar{y}$. Then transform the relative coordinate to polar coordinate $(\Delta\rho_k,\Delta\theta_k)$, where $\Delta\theta_k=\arctan(\Delta y_k/\Delta x_k)$. Thus, the points set *P* is transformed to P^* by ranking the $\Delta\theta_k$ in descending order.

By connecting the points $p_k^*=(x_k^*,y_k^*)$ of P^* in a two-dimensional plane, a polygon *G* with the two rough-fuzzy numbers as convex is formed as Figure 2. The point C_G is the centroid of *G*, whose abscissa x_c refers to the negative value of crisp total smart capability $-A_c$ and ordinate y_c denotes the crisp total cost C_c. They are determined by follows:

$$x_c=\frac{\sum_{k=1}^{6}\left(\left(x_k^*+x_{k+1}^*\right)\times\left(x_k^*y_{k+1}^*-x_{k+1}^*y_k^*\right)\right)}{3\times\sum_{k=1}^{6}\left(x_k^*y_{k+1}^*-x_{k+1}^*y_k^*\right)} \tag{14}$$

$$y_c=\frac{\sum_{k=1}^{6}\left(\left(y_k^*+y_{k+1}^*\right)\times\left(x_k^*y_{k+1}^*-x_{k+1}^*y_k^*\right)\right)}{3\times\sum_{k=1}^{6}\left(x_k^*y_{k+1}^*-x_{k+1}^*y_k^*\right)} \tag{15}$$

where $x_7^*=x_1^*$, $y_7^*=y_1^*$, since the ending point is also the starting point.

Therefore, the rough-fuzzy bi-objectives configuration optimization model can be converted into the deterministic bi-objectives optimization model, expressed as Eq. (16).

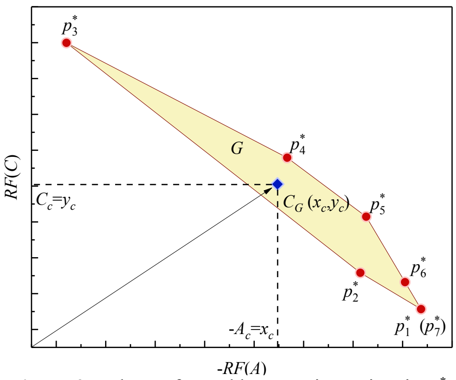


Figure 2. Polygon formed by mapping points in P^*

$$
\begin{cases}
F(X)=\begin{cases}\min -A_c\\ \min C_c\end{cases}\\
s.t.\begin{cases}
\sum_{k=1}^{n_{ij}} x_{ijk}=1;\sum_{j=1}^{n_i} x_{ijk}=n_i;\\
\sum_{s=1}^{t} w_s^a=1, w_s^a>0;\sum_{s=1}^{t} w_s^c=1; w_s^c>0;\\
\sum_{i=1}^{n_s}\sum_{j=1}^{n_i}\sum_{k=1}^{n_{ij}}(x_{i'j'k'}.R_{ijk-i'j'k'})=\begin{cases}0, & x_{ijk}=0,\quad R_{ijk-i'j'k'}=1;\\ 0, & x_{ijk}=1,\quad R_{ijk-i'j'k'}=-1.\end{cases}\\
R_{ijk-i'j'k'}=\begin{cases}1, & \text{mutually inclusive;}\\ -1, & \text{mutually exclusive;}\\ 0, & \text{otherwise.}\end{cases}
\end{cases}
\end{cases}
\quad (16)
$$

4.1.2 *Adaptive NSGA-II based solution*

In this section, the ANSGA-II is introduced by integrating the adaptive generation operator [27] and the original NSGA-II [16, 28 is used to solve the deterministic bi-objectives SPS configuration optimization model (16) due to its superiority in adaptive generation operation and faster search speed. Due to the paper length limitation, the detailed description of the ANSGA-II is not presented. With this solution, a pareto-set of relative optimal solutions for the bi-objectives can be obtained.

5 ILLUSTRATIVE EXAMPLE

In this section, the configuration optimization of smart vehicle service is taken as an example to illustrate the feasibility of the proposed optimization model. Company M, a world class vehicle manufacturer, is committed to provide customized smart vehicle services (SVSs) with objectives of agilely delivering more profits and better experience to customers. The SVS modules and module instances are shown in Table 1, which respectively have 12 modules and 32 instances. The rough-fuzzy attributes are obtained by inviting 5 experts to assess the performances of each instance on the corresponding criteria according to Eq. (1)~(4). The acquired rough-fuzzy smart capability and cost of SVS module instances are presented in Table 2. The goal of SVS configuration optimization is to search for an appropriate instance portfolio to achieve a balance between the total smart capability and total cost of the configured scheme. In addition to consider the customer requirement (i.e. best smartness and lowest cost), the configuration should also meet the following combination constraints: If SMI_{111} (Operational state optimization) is selected of the first SVS module SM_{11}, SMI_{132} (Predictive maintenance) of the third SVS module SM_{13} is must be also selected. If SMI_{113} (Performance improvement) of the third SVS module SM_{13} is chosen, SMI_{131} (Proactive maintenance) should be chosen as well. If SMI_{211} (Intermediary sharing) of the fifth SVS module SM_{21} is selected, SMI_{221} (Dynamic sharing) of the sixth SVS module SM_{22} cannot be selected.

Table 1. SVS modules and module instances

	Service scope	SVS modules	SVS module instances	Instance code
1	PUO-SPS	Operational optimization	Operational state optimization	SMI_{111}
2	(SS_1)	modules (SM_{11})	Energy saving management	SMI_{112}
3			Performance improvement	SMI_{113}
⋮		⋮	⋮	⋮
9		Smart recovery	Smart disassembly	SMI_{141}
10		(SM_{14})	Smart remanufacturing	SMI_{142}
11			Smart reverse logistics	SMI_{143}
12	PSO-SPS	Product-oriented sharing (SM_{21})	Intermediary sharing	SMI_{211}
13	(SS_2)		Product escrow	SMI_{212}
14		Use-oriented sharing (SM_{22})	Dynamic sharing	SMI_{221}
⋮	⋮	⋮	⋮	⋮
32	PIE-SPS (SS_4)	Product integrating service (SM_{42})	Control other products	SMI_{423}

Table 2. Rough-fuzzy attributes of SVS module instances

$RF(a_{ijk})$	a_{ijk}^{lL}	a_{ijk}^{mL}	a_{ijk}^{uL}	a_{ijk}^{lU}	a_{ijk}^{mU}	a_{ijk}^{uU}
SMI_{111}	0.257	0.507	0.755	0.571	0.821	0.952
SMI_{112}	0.138	0.388	0.638	0.407	0.657	0.853
SMI_{113}	0.150	0.400	0.646	0.301	0.551	0.765
SMI_{121}	0.108	0.358	0.608	0.337	0.587	0.837
SMI_{122}	0.137	0.387	0.615	0.454	0.704	0.851
SMI_{131}	0.138	0.388	0.638	0.418	0.668	0.864
SMI_{132}	0.147	0.397	0.646	0.362	0.612	0.844
⋮	⋮	⋮	⋮	⋮	⋮	⋮
SMI_{423}	0.126	0.376	0.626	0.328	0.578	0.828
$RF(c_{ijk})$	c_{ijk}^{lL}	c_{ijk}^{mL}	c_{ijk}^{uL}	c_{ijk}^{lU}	c_{ijk}^{mU}	c_{ijk}^{uU}
SMI_{111}	0.278	0.528	0.775	0.584	0.834	0.962
SMI_{112}	0.158	0.408	0.658	0.436	0.686	0.874
SMI_{113}	0.190	0.440	0.687	0.320	0.570	0.792
SMI_{121}	0.112	0.362	0.612	0.363	0.613	0.863
SMI_{122}	0.156	0.406	0.631	0.490	0.740	0.867
SMI_{131}	0.158	0.408	0.658	0.448	0.698	0.886
SMI_{132}	0.170	0.420	0.669	0.381	0.631	0.857
⋮	⋮	⋮	⋮	⋮	⋮	⋮
SMI_{423}	0.145	0.395	0.645	0.345	0.595	0.845

The parameters of the adaptive NSGA-II algorithm are assigned as follows: population size = 500, evolutionary generation=200, calculation runs = 5, minimum competition size = 5, maximum competition size = 20, minimum crossover probability = 0.3, maximum crossover probability = 0.9, minimum mutation probability = 0.02, and maximum mutation probability = 0.08. Firstly, the optimization results without considering the combination constraints are shown in Figure 3, where A_c and C_c separately denotes the total cost and total smart capability of the configured schemes. Since the objectives of smart capability and cost cannot be consistent with each other, the obtained solution of the SVS configuration is not a point but a set of solutions. The pareto-solution set in Figure 3 has a rational distribution, since the algorithm can keep elites in a population through the evolution process to reach the highest optimal solution rate. The solution set includes 288 points, and each point represents an optimized SVS configuration scheme. The initial population with size of 500 has a random distribution in Figure 3, and finally evolves to the optimal pareto-solution set without considering the combination rules. Finally, 65 pareto-solutions under combination constraints are targeted as shown in Figure 5(a) (marked as rough-fuzzy ANSGA-II). The SVS designers can then select suitable configured service schemes from these solutions to satisfy various customers. In this case study, five SVS configuration schemes are sorted out in the light of customer preference for the smart capability and cost (see Table 3).

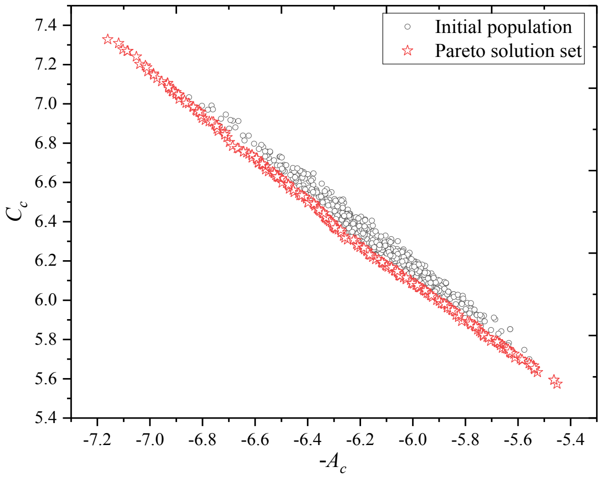


Figure 3. Initial population and Pareto-solution set of SVS configuration

In order to demonstrate the feasibility and merits of the proposed approach, two comparisons under the same linguistic matrix of SMIs' attributes. The first comparison is carried out to reveal the difference of the obtained SMIs' configuration parameters with using crisp number, fuzzy number, rough number and rough-fuzzy number to transform the linguistic assessment matrix [13], as shown in Figure 4. The second comparison is conducted to uncover the different solutions between different configuration parameters using the same ANSGA-II and different solving algorithms (ANSGA-II vs original NSGA-II [16]) with the same configuration parameters, as shown in Figure 5.

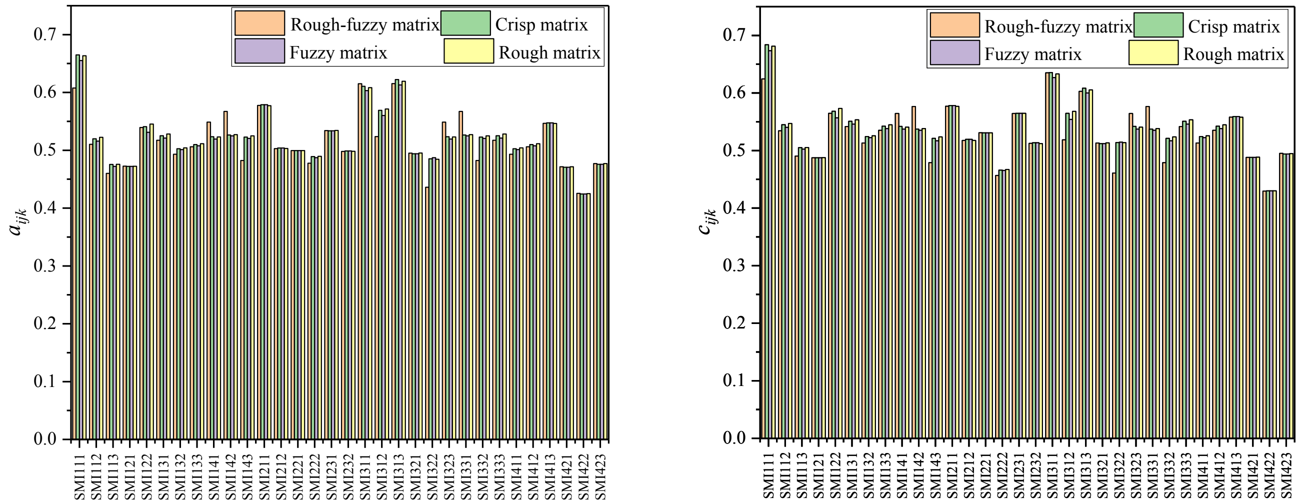
As shown in Figure 4, the obtained smart capability and service cost with different numbers have distinctive values with each other, due to the different handling mechanism of uncertainty between the rough-fuzzy number, crisp number, fuzzy number and rough number [18]. For example, a_{111}=0.608 with rough-fuzzy number, while a_{111}=0.665, 0.655 and 0.664 respectively using crisp number, fuzzy number and rough number. Such situations almost exist for every SMI. These differences can be derived from the four methods' working principle of manipulating the hybrid intrapersonal and interpersonal uncertainties. The rough-fuzzy number can fully handle the intrapersonal linguistic ambiguousness and interpersonal preference randomness at the same time, since it combines the superiority of fuzzy logic to model lexical vagueness and the merits of rough variable in capturing group preference diversity and subjectivity. However, neither the single fuzzy nor the rough method can simultaneously perceive and process the hybrid uncertainties. Therefore, the feasibility and effectiveness of rough-fuzzy matrix for obtaining SMIs' configuring parameters are validated and illustrated.

Figure 5(a) depicts that the proposed optimization model with configuration parameters from the proposed rough-fuzzy matrix have larger feasible solution space (sixty-five solutions) compared with crisp approach (seventeen solutions), fuzzy approach (fourteen solutions) and rough approach (seventeen solutions). Differing with traditional product configuration, the product service configuration is frequently and dynamically required. In this respect, the rough-fuzzy approach is advantageous in providing wider selection scope for different customers.

Figure 5(b) presents the average trends of optimizing objectives changing with generation numbers for adaptive and original NSGA-II. For ANSGA-II, the average A_c and C_c of populations in each generation increase remarkably in the first 50 generations, and then intend to be stable in the subsequent evolutions. This means that the search process after 50 generations have better convergence due to the application of adaptive generation operators. However, the average objectives of original NSGA-II keep changing throughout the evolution process and do not reach a stable status even at the ending of evolution. This is because that the original NSGA-II uses fixed crossover probability and mutation probability in binary-coded generation operation which make the evolution often suffer local optimal and worse convergence performance. Since the adaptive generation operators can guarantee higher global search capacity in the early evolution stage and increase local intensity in the later evolution stage, the ANSGA-II perform higher convergence speed and better search capacity.

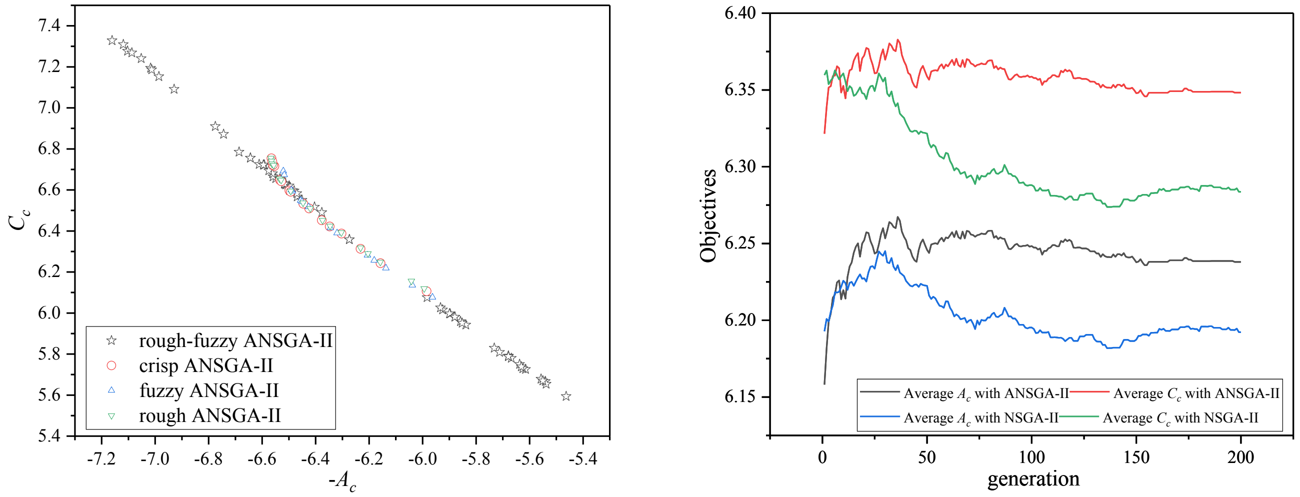
Table 3. The five recommended SVS configuration schemes

No.	SVS module instances	*RF*(*A*)	*RF*(*C*)	A_c	C_c
CS1	SMI_{111}, SMI_{121}, SMI_{131}, SMI_{142}, SMI_{211}, SMI_{221}, SMI_{231}, SMI_{313}, SMI_{323}, SMI_{331}, SMI_{413}, SMI_{421}	[(2.035, 5.035, 8.017), (4.945, 7.945, 10.589)]	[(2.209, 5.209, 8.194), (5.116, 8.116, 10.739)]	7.161	7.328
CS2	SMI_{112}, SMI_{121}, SMI_{133}, SMI_{143}, SMI_{212}, SMI_{222}, SMI_{232}, SMI_{312}, SMI_{322}, SMI_{332}, SMI_{411}, SMI_{421}	[(1.676, 4.557, 7.544), (4.598, 7.601, 10.221)]	[(1.764, 4.728, 7.709), (4.709, 7.712, 10.311)]	5.463	5.594
CS3	SMI_{111}, SMI_{122}, SMI_{133}, SMI_{143}, SMI_{211}, SMI_{222}, SMI_{232}, SMI_{313}, SMI_{323}, SMI_{332}, SMI_{412}, SMI_{423}	[(1.937, 5.091, 8.091), (4.843, 7.811, 10.479)]	[(2.016, 5.189, 8.191), (4.988, 7.954, 10.544)]	6.497	6.598
CS4	SMI_{112}, SMI_{122}, SMI_{132}, SMI_{142}, SMI_{211}, SMI_{221}, SMI_{232}, SMI_{313}, SMI_{323}, SMI_{331}, SMI_{413}, SMI_{423}	[(1.926, 4.926, 7.887), (4.809, 7.809, 10.451)]	[(2.104, 5.104, 8.066), (4.979, 7.979, 10.577)]	6.929	7.091
CS5	SMI_{112}, SMI_{121}, SMI_{133}, SMI_{143}, SMI_{211}, SMI_{222}, SMI_{232}, SMI_{312}, SMI_{323}, SMI_{332}, SMI_{411}, SMI_{422}	[(1.730, 4.635, 7.625), (4.654, 7.657, 10.224)]	[(1.805, 4.776, 7.760), (4.717, 7.720, 10.259)]	5.983	6.076



(a) SMIs' smart capability (b) SMIs' service cost

Figure 4. SMIs' attribute value with different uncertainty handling methods



(a) Pareto-solutions obtained with combination constraints under different forms of instance attributes and adaptive NSGA-II

(b) Comparison of average trends of optimizing objectives with generation number

Figure 5. Comparisons between the proposed approach and other approaches

6 CONCLUSIONS

This paper proposes a rough-fuzzy bi-objective optimization model for SPS configuration under hybrid uncertain environments. The group linguistic assessments of each instance on smart capability and service cost are converted into rough-fuzzy numbers. Then, a rough-fuzzy bi-objective optimization model is proposed for SPS configuration with respect to the total smart capability and cost with the rough-fuzzy attributes value. Moreover, a graphics-based aggregation approach is introduced for simplification of the rough-fuzzy optimization model by composing the twelve extended optimization objectives to two objectives. In addition, an adaptive NSGA-II algorithm is adopted to solve the simplified configuration optimization model. The application of the proposed model in smart vehicle service configuration has validated the model's feasibility and effectiveness. The contribution of this research can be summarized as follows. The proposed SPS configuration optimization model fully consider the smart features brought by the application of smart technologies and thus can help the smart service designers quickly and efficiently customize appropriate service schemes to satisfy various users' requirements and preferences. The proposed rough-fuzzy number is feasible to handle the hybrid uncertainty involved in the evaluation process of SPS module instance attributes compared with the traditional deterministic method. The rough-fuzzy approach can provide more accurate values of instance attributes and offer more realistic configuration optimization results. Moreover, the introduced ANSGA-II presents better convergence speed and evolutionary performance over the original NSGA-II, since it uses the adaptive generation operators to guarantee higher global search capacity in the early evolution stage and increase local intensity in the later evolution stage. Although the proposed optimization approach has presented some advantages, it still provides a set of solutions, but not a single one. Thus, the future work can be conducted to develop a recommendation model for selecting the most suitable one based on customer preference.

REFERENCES

[1] P. Zheng, Z. Wang, C.-H. Chen, L. Pheng Khoo, A survey of smart product-service systems: Key aspects, challenges and future perspectives, Adv. Eng. Inf., 42 (2019) 100973.

[2] M. Saunila, M. Nasiri, J. Ukko, T. Rantala, Smart technologies and corporate sustainability: The mediation effect of corporate sustainability strategy, Comput Ind, 108 (2019) 178-185.
[3] Z. Chen, M. Lu, X. Ming, X. Zhang, T. Zhou, Explore and evaluate innovative value propositions for smart product service system: A novel graphics-based rough-fuzzy DEMATEL method, Journal of Cleaner Production, 243 (2020) 118672.
[4] E. Siow, T. Tiropanis, W. Hall, Analytics for the internet of things: A survey, ACM Comput Surv, 51 (2018).
[5] A. Rymaszewska, P. Helo, A. Gunasekaran, IoT powered servitization of manufacturing–an exploratory case study, International Journal of Production Economics, 192 (2017) 92-105.
[6] J. Lee, H.-A. Kao, S. Yang, Service Innovation and Smart Analytics for Industry 4.0 and Big Data Environment, Procedia CIRP, 16 (2014) 3-8.
[7] C. Suppatvech, J. Godsell, S. Day, The roles of internet of things technology in enabling servitized business models: A systematic literature review, Ind. Mark. Manage., 82 (2019) 70-86.
[8] H. Schallehn, S. Seuring, J. Strähle, M. Freise, Customer experience creation for after-use products: A product–service systems-based review, Journal of Cleaner Production, 210 (2019) 929-944.
[9] M. Ardolino, M. Rapaccini, N. Saccani, P. Gaiardelli, G. Crespi, C. Ruggeri, The role of digital technologies for the service transformation of industrial companies, Int J Prod Res, 56 (2018) 2116-2132.
[10] X. Kuang, F. Zhao, H. Hao, Z. Liu, Intelligent connected vehicles: the industrial practices and impacts on automotive value-chains in China, Asia Pacific Business Review, 24 (2018) 1-21.
[11] A. Tukker, Eight types of product–service system: eight ways to sustainability? Experiences from SusProNet, Business Strategy and the Environment, 13 (2004) 246-260.
[12] O.K. Mont, Clarifying the concept of product–service system, Journal of Cleaner Production, 10 (2002) 237-245.
[13] Z. Chen, X. Ming, T. Zhou, Y. Chang, Z. Sun, A hybrid framework integrating rough-fuzzy best-worst method to identify and evaluate user activity-oriented service requirement for smart product service system, Journal of Cleaner Production, 253 (2020) 119954.
[14] Z. Chen, X. Ming, A rough–fuzzy approach integrating best–worst method and data envelopment analysis to multi-criteria selection of smart product service module, Applied Soft Computing, 94 (2020) 106479.
[15] Z. Chen, X. Ming, R. Wang, Y. Bao, Selection of design alternatives for smart product service system: A rough-fuzzy data envelopment analysis approach, Journal of Cleaner Production, (2020) 122931.
[16] W. Song, F.T.S. Chan, Multi-objective configuration optimization for product-extension service, Journal of Manufacturing Systems, 37 (2015) 113-125.
[17] Z. Liu, X. Ming, A framework with revised rough-DEMATEL to capture and evaluate requirements for smart industrial product-service system of systems, Int J Prod Res, (2019) 1-19.
[18] Z. Chen, X. Ming, X. Zhang, D. Yin, Z. Sun, A rough-fuzzy DEMATEL-ANP method for evaluating sustainable value requirement of product service system, Journal of Cleaner Production, 228 (2019) 485-508.
[19] D. Wu, J.M. Mendel, Computing With Words for Hierarchical Decision Making Applied to Evaluating a Weapon System, IEEE Transactions on Fuzzy Systems, 18 (2010) 441-460.
[20] J. Shen, J.A. Erkoyuncu, R. Roy, B. Wu, A framework for cost evaluation in product service system configuration, Int J Prod Res, 55 (2017) 6120-6144.
[21] P.P. Wang, X.G. Ming, Z.Y. Wu, M.K. Zheng, Z.T. Xu, Research on industrial product-service configuration driven by value demands based on ontology modeling, Comput Ind, 65 (2014) 247-257.
[22] J. Shen, L. Wang, Y. Sun, Configuration of product extension services in servitisation using an ontology-based approach, Int J Prod Res, 50 (2012) 6469-6488.
[23] D. Guillon, R. Ayachi, É. Vareilles, M. Aldanondo, É. Villeneuve, C. Merlo, Product˅service system configuration: a generic knowledge-based model for commercial offers, Int J Prod Res, (2020) 1-20.
[24] J. Shen, B. Wu, L. Yu, Personalized configuration rules extraction in product service systems by using local cluster neural network, Ind Manage Data Sys, 115 (2015) 1529-1546.
[25] H.J. Long, L.Y. Wang, J. Shen, M.X. Wu, Z.B. Jiang, Product service system configuration based on support vector machine considering customer perception, Int J Prod Res, 51 (2013) 5450-5468.
[26] H. Li, Y. Ji, L. Chen, R.J. Jiao, Bi-Level Coordinated Configuration Optimization for Product-Service System Modular Design, IEEE Trans. Syst. Man Cybern. Syst., 47 (2017) 537-554.
[27] L. He, W. Song, Z. Wu, Z. Xu, M. Zheng, X. Ming, Quantification and integration of an improved Kano model into QFD based on multi-population adaptive genetic algorithm, Comput Ind Eng, 114 (2017) 183-194.
[28] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, IEEE transactions on evolutionary computation, 6 (2002) 182-197.

Functional Testing of Conflict Detection and Diagnosis Tools in Feature Model Configuration: A Test Suite Design

Cristian Vidal-Silva[1] and **José A. Galindo**[2] and **David Benavides**[3]

Abstract. In configuration scenarios such as the product configuration of a Feature Model (FM), conflict detection and diagnosis operations are high-value configuration tasks for creating consistent products. Conflict detection and diagnosis are about identifying a set of (minimal) restrictions responsible for conflicts and which of those restrictions after their update would enable consistent configuration. These functionalities encompass the development of consistent configuration products. The lack of specific testing mechanisms is becoming a significant obstacle that affects the quality and reliability of new solutions for those tasks. In this paper, we present FAMAPRODCONF TESTSUITE, a set of implementation-independent test cases to validate the functionality of conflict detection and diagnosis solutions in FM product configuration. FAMAPRODCONF TESTSUITE is an efficient and handy mechanism to help develop new solutions and detect faults to solve the found issues and improve the quality of solutions. As an effectiveness proof, we evaluated the test suite using existing solutions, one for conflict detection and another for the diagnosis of product configuration: QUICKXPLAIN and FASTDIAG. Obtained results validate our approach for testing conflict detection and diagnosis of FM product configuration solutions.

1 INTRODUCTION

FMs are information models that describe all the features and constraints for the configuration of valid products in Software Product Lines (SPLs). In the configuration of FM, each feature is decided to be either present or absent in the resulting products. The configuration of FM is a time consuming and complex task due to the existence of constraints among features [26], and more expensive in large-scale FMs which can define an exponential number of products. Configurations can result in misconfigurations (i.e., non-valid configurations) impacting the system availability [49]. The development of automated solutions to assist the product configuration process represents a highly valuable task [46]. Conflict detection and diagnosis are two AUTOMATED ANALYSIS OF FEATURE MODEL (AAFM) operations for assisting the product configuration process. Two well-known solutions for conflict detection and diagnosis are QUICKXPLAIN [28] and FASTDIAG [15], respectively.

The development of AAFM configuration solutions such as the conflict detection and diagnosis of invalid products are hard task examples that involve complex data structures and algorithms [42]. Developing those solutions can involve errors that increase the development time and reduce the reliability of their analysis [43]. To guarantee the absence of defects in conflict and diagnosis detection tasks represents a valuable issue for developers and future users. However, the lack of specific testing mechanisms in this context appears to be a significant obstacle to assessing the functionality and quality of new solutions. Moreover, software testing accounts for about 50% of the total cost of software development [37, 45].

Software testing is a concurrent process to determine if a program behaves as expected [7, 36]. Software testing mainly applies test cases to verify the conformance between obtained and expected output results for defined input sets. As stated by [42], the main challenge of software testing is to achieve a balance between the number of test cases and their effectiveness for the detection of faults. A goal of software testing is neither to use a reduced number of test cases nor to increase that number to unmanageable levels. Hence, this article's main objective is to define FAMAPRODCONF TESTSUITE, a reliable set of implementation-independent test cases to validate the functionality of conflict detection and diagnosis solutions for assisting the FM product configuration process. Our test cases can help develop and improve the reliability and quality of products since each test case assists in the fault detection in feature model configurations. We create a representative set of input-output combinations using techniques from the software testing community to be applied either in isolation or as a complement to additional testing methods such as white-box testing. We use the QUICKXPLAIN and FASTDIAG solutions for the conflict detection and diagnosis in product configurations for developing the test suite. Next, we briefly describe the main characteristics of our suite:

- **Operations tested**. FAMAPRODCONF TESTSUITE addresses the minimal conflict detection and preferred diagnosis of conflicts in the product configuration of FMs. We identified particular situations for both actions.
- **Testing techniques**. As in the work of Segura et al. [42], we applied the equivalence partitioning, boundary-value analysis, pairwise testing, and error guessing black-box testing techniques in the test cases design.
- **Test cases**. The suite consists of 78 test cases each designed in terms of the inputs (i.e., FMs and product configurations) and the expected outputs of the analysis operations.

The remainder of the article is structured as follows: Section 2 describes FMs and product configurations, their analyses operations, namely conflict detection and diagnosis with the QUICKXPLAIN

[1] Catholic University of the North, Chile, email: cristian.vidal@ucn.cl
[2] University of Seville, Spain, email: jagalindo@us.es
[3] University of Seville, Spain, email: benavides@us.es

and FASTDIAG as example solutions, and black-box testing techniques in a nutshell. Section 3 details how we designed our test cases. Section 4 describes adequacy evaluation and refinement of the suite. Section 5 presents a set of related work. Finally, we summarize our main conclusions and future work in Section 6.

2 PRELIMINARIES

2.1 Feature Model & Configuration

An FM [30] represents the set of configuration choices and their constraints to describe a family of products within a system family. FMs are the "de facto" standard to describe commonalities and variabilities in the product configuration of SPLs [20]. Each feature in an FM (except the root) has one parent feature and can have a set of child features. Notice that a child feature can only be included in a feature combination of a valid product if that also includes its parent. A valid product configuration always includes the feature root [35]. The following types of relationships or constraints exist in traditional FMs [3]: *a)* parent-children or inclusion relationships, and *b)* Cross-Tree Constraints (CTC). Four kinds of parent-child relationships exist: *i) mandatory* (the parent requires its child, and vice versa), *ii) optional* (the parent does not require its child), *iii) inclusive*-OR (the parent requires at least one child of the set of children), and *iv) alternative*-XOR (the parent requires exactly one of child features of the set of children). Besides the parent-child relationships, features can also relate across different branches of the FM with Cross-Tree Constraints (CTCs). CTCs of traditional FMs are *requires* (a feature requires another), and *excludes* (two features cannot be in the same configuration).

Figure 1 illustrates an FM for an *eShop* (head of the tree). *eShop* presents the mandatory child features *Catalogue*, *Payment* and *GUI*, and the optional children *Security* and *Banners*. Feature *Catalogue* presents the optional child features *Offers* and *Search*, and the mandatory child feature *Info*. Feature *Payment* presents an inclusive set of two child features (*Bank* and *CreditCard*). Likewise, feature *GUI* presents an inclusive set of child features (*PC* and *Mobile*). Feature *Security* has an alternative set of child features (*High* and *Medium*). Feature *Banners* does not present children. Finally, feature *Info* has an inclusive set of child features (*Image*, *Description* and *Price*), feature *Search* presents an inclusive set of child features (*Basic* and *Advance*), feature *CreditCard* has an inclusive set of child features (*Visa* and *AmericanExpress*). This FM also presents two cross-tree constraints: i. Feature *CreditCard* requires feature *High*, i.e., if a product configuration contains feature *CreditCard*, that product must contain feature *High* to be consistent; ii. Feature *Mobile* excludes feature *Banners*, i.e., a product configuration that contains feature *Mobile* cannot contain feature *Banners* and vice-versa.

An FM configuration corresponds to the selection of features. A configuration is valid if it does not contradict any of the constraints in the FM. A valid configuration defines a product, and a configuration is like an FM instance. That is, it is similar to the relationship between a class and one of its objects [8]. The configuration process refers to the derivation of products from an FM. Dark grey feature in Figure 1 represent a product configuration. In that configuration there are two conflicts: *i)* the user selects $CreditCard$ as the $Payment$ option and $Medium$ as $Security$ option (the user does not select $High$ and $CreditCard$ requires $High$); and *ii)* the user selects both options of GUI, i.e., PC and $Mobile$, and also selects the feature $Banner$ that is exclusive to $Mobile$.

2.2 Automated Analysis of Feature Model

The AUTOMATED ANALYSIS OF FEATURE MODEL (AAFM) [5, 19] is a computer-aided process that *i)* translates feature model constraints into a logical representation (e.g., SATisfiability problem or Constraint Satisfaction Problem); *ii)* applies some defined operations by using off-the-shelf solvers or specific programming solutions; and *iii)* provides feedback on specific model properties as a process result. The conflict detection and diagnosis are two AUTOMATED ANALYSIS OF FEATURE MODEL examples: *a)* for detecting conflicts in FM or product configuration [47]; *b)* for obtaining a set of constraints which state update permits resolving conflicts and developing an error-free solution [10] . For instance, if a user wants to select and deselect a set of features but some selections and deselections are incompatible, a conflict detection solution would indicate the minimal set of constraints in conflict. For the same case, a diagnosis (conflict resolution) would suggest which features have to be selected/deselected from the original configuration to fix the problem and convert the original inconsistent configuration into a valid one [48]. A further application example of model-based diagnosis in the context of AUTOMATED ANALYSIS OF FEATURE MODEL is the identification of minimal sets of constraints responsible for the existence of dead features in an FM (i.e., features no part of any configuration [10]). Another application example is the identification of minimal sets of constraints responsible for a void FM (i.e., no configuration can be found [10]).

Conflict detection algorithms determine minimal sets of inconsistent constraints that do not allow the determination of a solution (configuration) [12]. Diagnosis algorithms determine minimal sets of constraints to delete or adapt so that the remaining set of constraints is consistent [12]. Conflict detection algorithms such as QUICKXPLAIN help determine preferred minimal conflicts in the underlying model [28, 41], and constitute major support for conflict detection and diagnosis operations. Examples of diagnosis solutions for getting a preferred minimal diagnosis of conflict models include FASTDIAG [10, 15] and FLEXDIAG [16]. A preferred set of configuration constraints sets the order of relevance of each constraint in the configuration. Hence, QUICKXPLAIN determines a preferred set of constraints in conflict to be solved with that set being minimal and composed of the less relevant constraints in the relevance order. FastDiag determines the set of less relevant constraints in the relevance hierarchy whose adaptation permits solving all the conflicts to get a configuration solution. Other diagnosis solutions exist, such as HS-DAG [40, 14], minimum cardinality candidate generation [9], and RC-Tree [39] in which outputs can differ from those of QUICKXPLAIN and FASTDIAG. This test suite can be adapted to work with those solutions for the diagnosis of FM product configurations.

For the product configuration $\{eShop$, $Catalogue$, $Info$, $Image$, $Description$, $Price$, $Payment$, $Bank$, $CreditCard$, $Catalogue$, $Visa$, $MasterCard$, $Security$, $Medium$, GUI, PC, $Mobile$, $Banners\}$ in Figure 1, taking into account the order of selection as a determinant factor, i.e., last selected features and the non-selected features are the less relevant of the selection, $\{Banners$, $Mobile\}$ represents the minimal conflict and $\{Banners$, $Medium\}$ corresponds to the preferred minimal diagnosis. For the configuration $\{Banner$, $Mobile$, PC, GUI, $Medium$, $Security$, $MasterCard$, $Visa$, $Catalogue$, $CreditCard$, $Bank$, $Payment$, $Price$, $Description$, $Image$, $Info$, $eShop\}$ that is the same configuration but in a different

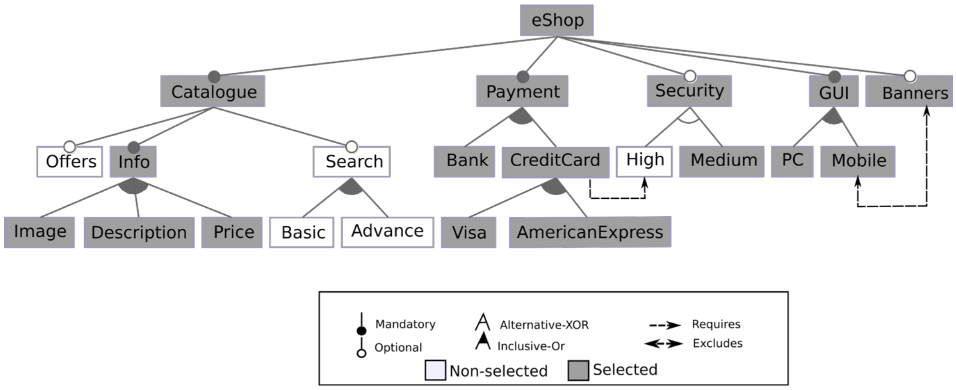


Figure 1: Example feature model of a configurable eShop. (An adapted version of FM example of [42])

order, the minimal conflict is $\{CreditCard, Medium\}$ and the preferred minimal diagnosis is $\{CreditCard, Visa, AmericanExpress, Mobile\}$. A preferred minimal conflict set contains a minimal set of constraints in conflict (not necessarily all the constraints in conflict) whereas a minimal diagnosis set includes all the constraints that after updating permit getting a solution. Hence, we can appreciate that QUICKXPLAIN enables obtaining a preferred minimal conflict set whereas FASTDIAG provides a preferred minimal diagnosis to get a consistent configuration. We use QUICKXPLAIN and FASTDIAG for getting minimal conflicts and preferred diagnosis, respectively.

QUICKXPLAIN and FASTDIAG solutions share an efficient divide-and-conquer algorithmic strategy with a different purpose. Both solutions are independent of consistency checking strategies; i.e., they can use different reasoning solvers for the consistency checking of rules [12]. Algorithms 1 and 2 are the QUICKXPLAIN solution. Algorithms 3 and 4 depict the FASTDIAG solution. In Algorithm 1, the knowledge-base B does not include the configuration C. The first base case of Algorithm 2 looks for identifying inconsistencies after adding a portion of C in B. Yet, the knowledge-base AC includes the configuration S, and the first base case of Algorithm 4 tries to identify a consistency after removing a portion of S from AC. The second base case of Algorithms2 and 4 are analogue, i.e., the minimal size of the configuration C and S in QUICKXPLAIN and FASTDIAG, respectively.

2.3 Black-Box Testing

A primary purpose of software testing is failure detection to discover defects and correct their source [29]. Black-box is a software testing approach that focuses on verifying whether the external behavior functions as expected concerning its requirements or design specification [38]; i.e., Black-box testing focuses on the output responses for given inputs and execution conditions without concern with the internal mechanisms of a system.

Algorithm 1 QUICKXPLAIN$(C, B) : CS$

if CONSISTENT$(B \cup C)$ **then**
  $return$('no conflict')
else if $C = \emptyset$ **then**
  $return(\emptyset)$
else
  $return$(QX$(C, B, \emptyset)$)
end if

Algorithm 2 QX$(C = \{c_1..c_m\}, B, B\delta) : CS$

if $B\delta \neq \emptyset$ **and** INCONSISTENT(B) **then**
  return$(\emptyset)$
end if
if $C = \{c_\alpha\}$ **then**
  return$(\{c_\alpha\})$
end if
$k = \lfloor \frac{m}{2} \rfloor$
$C_a \leftarrow c_1...c_k; C_b \leftarrow c_{k+1}...c_m$;
$\Delta_2 \leftarrow$ QX$(C_a, B \cup C_b, C_b)$;
$\Delta_1 \leftarrow$ QX$(C_b, B \cup \Delta_2, \Delta_2)$;
return$(\Delta_1 \cup \Delta_2)$

Algorithm 3 FASTDIAG$(C, AC) : diagnosis\ \Delta$

if $C = \emptyset$ **or** INCONSISTENT$(AC - C)$ **then**
  $return(\emptyset)$
else
  $return$(FD$(\emptyset, C, AC)$)
end if

Algorithm 4 FD$(D, C = \{c_1..c_q\}, AC) : diagnosis\ \Delta$

if $D \neq \emptyset$ **and** CONSISTENT(AC) **then**
  return$(\emptyset)$
end if
if $|C| = 1$ **then**
  return(C)
end if
$k = \lfloor \frac{q}{2} \rfloor$
$C_1 \leftarrow c_1...c_k; C_2 \leftarrow c_{k+1}...c_q$;
$D_1 \leftarrow$ FD$(C_2, C_1, AC - C_2)$;
$D_2 \leftarrow$ FD$(D_1, C_2, AC - D_1)$;
return$(D_1 \cup D_2)$

Black-box testing assumes no knowledge about the internal structure or implementation details of the software being tested. Black-box testing applies test cases (i.e., a set of known input and expected output) to verify requirements compliance [4]. A test suite represents a collection of test cases [4, 36]. Different black-box testing techniques exist, such as equivalence partitioning, boundary value analysis, pairwise testing, and error guessing [18]. Test Case Generation (TCG) for each black-box testing technique permits automatically obtaining main components such as test case and expected output. We use those techniques for their generic nature and effectiveness in the functional testing of AAFM operations [42]:

- **Equivalence partitioning**. This technique divides the input domain into partitions (also called equivalence classes), which should be processed similarly by the program and produce similar results. Afterward, only a few test cases of each partition are needed to evaluate the program behavior for the corresponding partition. Thus, a subset of test cases from each partition is enough to test the program effectively while keeping manageable the number of test cases.
- **Boundary value analysis**. This method guides the tester to select those inputs located at the "edges" of each equivalence partition.
- **Pairwise testing**. This technique focuses on testing discrete combinations of each pair of input parameters. Pairwise testing permits addressing error cases involving two parameters while keeping the number of test cases at a manageable level.
- **Error guessing**. The ability and experience of testers to predict the location of faults is the basis of this technique. Error-guessing test cases typically look for the occurrence of common error-prone aspects for testing the type of system.

3 TEST SUITE DESIGN

We designed the test cases as input-output combinations to reveal failures in the operational implementation of the conflict detection and diagnosis for FM product configurations. We took the following steps for the test suite design: i) identification of inputs and outputs for conflict detection and diagnosis operations of product configuration, ii) selection of representative input instances, iii) combination of previous instances for each operation, and iv) test case reports .

3.1 Identifying Inputs and Outputs

Our main focus is to identify instances of input and expected output for the minimal conflict detection and preferred diagnosis of FM product configurations. As Table 1 describes, both operations receive the same kind of input instances, that is, a consistent FM and a product configuration.

Each product of Table 1 is a collection of selected features of the model (non-selected features are not in that list). We assign an identification number to each operation to refer to them for the sake of simplicity. We use the feature modeling notation of Figure 1 for its simplicity and extended use in existing FM analysis tools and literature.

3.2 Inputs selection

We describe the techniques and their application for the definition of input instance of each test case (i.e., FM and product). We followed and adapted the approach of [42].

- FEATURE MODELS. We used the equivalence partitioning and error guessing testing techniques to select a suitable set of input FMs. We applied them as follows:
  - *Equivalence partitioning*. We divided input feature models into equivalence classes according to the different types of relationships and constraints among features, i.e., mandatory, optional, or, alternative, requires, and excludes. This approach is widely used in mapping processes from FM to a specific logic paradigm (e.g., SAT or CSP) [6]. Using this technique, if an operation correctly manages an FM with a single mandatory relationship, we could assume that FMs with more than one mandatory relationship would also be processed successfully. Therefore, we divide the FMs input domain into three partitions to keep equivalence classes at a manageable level like [42]. We consider only consistent FMs for the product configuration: i) FMs including a single type of relationship or constraint; ii) FMs combining two different types of relationships or constraints; and iii) FMs including three or more different types of relationships or constraints.By the application of this technique, we obtained sixteen feature models representing the whole input domain of basic feature models (see Figure 2) that we used as inputs for our suite operations. We only considered FMs that minimally permit one configuration.
  - *Error guessing*. Like [42], we used FMs with dead features as suitable inputs following the guidelines of the error guessing testing approach. We consider three more FMs following the inputs refinement process of [42].
- PRODUCTS. First, we divided input products into two equivalent partitions: valid and non-valid products. Input instances should be treated the same way by the solution being tested; i.e., solutions should give the same answer for each equivalence class. Then, we used boundary analysis from each partition for a systematic selection of input products. We quantified products according to their number of features to select their instances at the "edges" of each partition. We defined those "edges" for the FMs of Figure 2; i.e., we selected four input products for each input FM, two valid, and two non-valid configurations. i) VP_{min}: valid product with the minimum number of features; ii) VP_{max}: valid product with the maximum number of features; iii) NVP_{min}: non-valid product with the minimum number of features, usually with one fewer feature than VP_{min}; iv) NVP_{max}: non-valid product with the maximum number of features, usually with one more feature than VP_{max}. For VP_{max} with all the features, NVP_{max} usually has one fewer feature . We consider the following special case: a) There are cases where VP_min = VP_max or NVP_min = NVP_max. b) VP_{min} with only the root feature. For that case, NVP_{min} can contain one more feature, or no contain the root. We used the second option. c) Products without considering the root feature are non-valid per-default. We used this approach to define NVP_{max} products for FMs that contain only the root as a core feature. Products containing non-existing features in the FM are discarded.

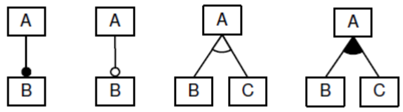
3.3 Input combination

. Input combination can be a challenge, as the number of combinations can be sizeable, and testing all combinations may be impractical if not impossible. Combination strategies define ways to combine input values in programs that receive more than one input parameter [22]. We used the pairwise testing technique to create effective test cases; i.e., we built a test case for each possible combination of the two input parameters. Hence, our test suite satisfies 2-wise coverage.

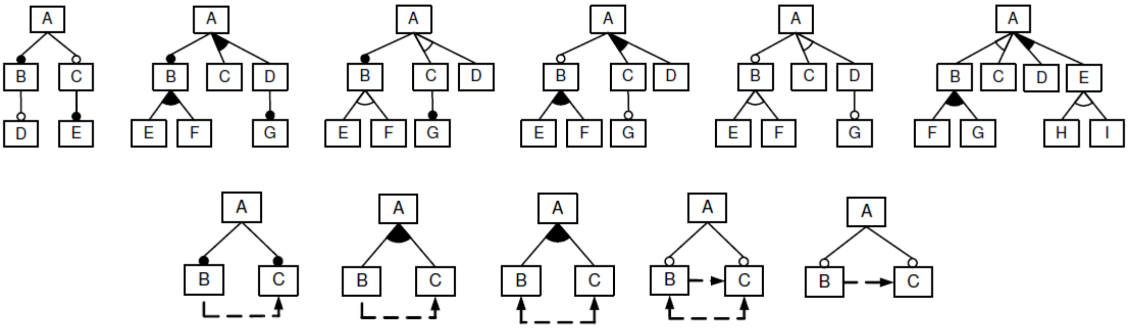
Our test suite considers CONFPROD and DIAGPROD operations

ID Operation	Operation	Input	Output
CONFPROD	Conflict detection in Product Configuration	Consistent FM Product	Collection of features in a minimal conflict
DIAGPROD	Diagnosis in Product Configuration	Consistent FM Product	Collection of features to get a consistent product after updating their selection state

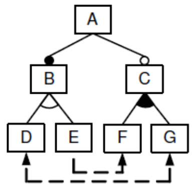
Table 1: Input and output of conflict detection and diagnosis operations in FM product configurations



(a) Feature models with a single relationship or constraint



(b) Feature models including two different types of relationships or constraints



(c) Feature models including all different types of relationships and constraints

Figure 2: Equivalence partitions on feature models (An adapted version of the input FMs of [42]) for the input of the test suite

that receive an FM and a product. For these inputs, we select sixteen FMs (see Figure 2) considering different types of relationships and three refined FMs of [42], along with four products (i.e. VP_{min}, VP_{max}, NVP_{min} and NVP_{max}). Using pairwise testing, we created a test case for each possible combination (i.e., 19*4 potential test cases).

3.4 Test Case Reports

We organized the selected pairs of input and expected output into test cases; 76 in total. We specify each test case using the IEEE Standard for Software Testing Documentation guideline [27] meaning each test case has its own ID, description, input, and expected output. Each input describes an FM along with four product configuration instances: P_1 = VP_{min}, P_2 = VP_{max}, P_3 = NVP_{min} and P_4 = NVP_{max}. Each expected conflict and diagnosis output C_i and D_i, respectively are for a P_i input.

Table 2 depicts five of the test cases included in the suite: i) Test case 6 tries to uncover failures in the product configuration of FMs with an alternative relationship. Each test case considers four products ($P_1 and$ P_2 for valid products, P_2, P_4 for non-valid products) each with an ascending order of preference in their constraints configuration, minimal conflict detection for each product (C_1, C_2, C_2, C_4), and preferred diagnosis for each product (D_1, D_2, D_2, D_4). For example, ii) Test case 7 exercises the product configuration of FMs with the interaction between mandatory and optional features. iii) Test case 8 exercises the product configuration of FMs with the interaction between mandatory and alternative set constraints on two levels. iv) Test case 9 exercises the product configuration of FMs with the interaction between mandatory and or-set constraints on two levels. Lastly, we design v) test 16 to reveal failures when checking the product configuration of an FM with the interaction of all the basic FM constraints. We can solve a minimal conflict of our report by selecting and updating one of the features in conflict. For example, we can solve the minimal conflict $\{A, B\}$ of P_3 of case ID 7 by updating the state of feature A or B. A preferred diagnosis for this case is $D_3 = \{B\}$. Table 2 does not consider intercase dependencies because the test cases do not require any previous execution of other test cases. Main source code and a complete list of the test cases in the suite are in `https://github.com/cvidalmsu/ProjectsTestingProdConfiguration`.

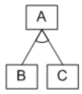
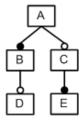
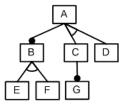
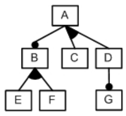
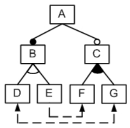
ID	Description	Input	Expected Output
6	Check product configuration for alternative relationship.	 $P_1 = \{A, B\}$ $P_2 = \{A, C\}$ $P_3 = \{A\}$ $P_4 = \{A, B, C\}$	*Conflicts* $C_1 = \{\}$ $C_2 = \{\}$ $C_3 = \{A, C, B\}$ $C_4 = \{C, B\}$ *Diagnosis* $D_1 = \{\}$ $D_2 = \{\}$ $D_3 = \{B\}$ $D_4 = \{C\}$
7	Check product configuration for the interaction between mandatory and optional relationships.	 $P_1 = \{A, B\}$ $P_2 = \{A, B, C, D, E\}$ $P_3 = \{A\}$ $P_4 = \{A, B, C, D\}$	*Conflicts* $C_1 = \{\}$ $C_2 = \{\}$ $C_3 = \{A, B\}$ $C_4 = \{C, E\}$ *Diagnosis* $D_1 = \{\}$ $D_2 = \{\}$ $D_3 = \{B\}$ $D_4 = \{E\}$
8	Check product configuration for the interaction between mandatory and alternative relationships.	 $P_1 = \{A, B, D, E\}$ $P_2 = \{A, B, C, E, G\}$ $P_3 = \{A, C\}$ $P_4 = \{A, B, C, D, E, G\}$	*Conflicts* $C_1 = \{\}$ $C_2 = \{\}$ $C_3 = \{C, G\}$ $C_4 = \{D, G\}$ *Diagnosis* $D_1 = \{\}$ $D_2 = \{\}$ $D_3 = \{B, G, F\}$ $D_4 = \{D\}$
9	Check product configuration for the interaction between mandatory and or relationships.	 $P_1 = \{A, B, C, E\}$ $P_2 = \{A, B, C, D, E, F, G\}$ $P_3 = \{A, B, D\}$ $P_4 = \{A, B, C, D, E, F\}$	*Conflicts* $C_1 = \{\}$ $C_2 = \{\}$ $C_3 = \{D, G\}$ $C_4 = \{D, G\}$ *Diagnosis* $D_1 = \{\}$ $D_2 = \{\}$ $D_3 = \{G, F\}$ $D_4 = \{G\}$
16	Check product configuration for the interaction of all the FMs relationships.	 $P_1 = \{A, B, D\}$ $P_2 = \{A, B, C, E, F, G\}$ $P_3 = \{A, C\}$ $P_4 = \{A, B, C, D, E, F, G\}$	*Conflicts* $C_1 = \{\}$ $C_2 = \{\}$ $C_3 = \{C, F\}$ $C_4 = \{D, G\}$ *Diagnosis* $D_1 = \{\}$ $D_2 = \{\}$ $D_3 = \{B, D, F\}$ $D_4 = \{E, G\}$

Table 2: ID, Description, input and expected output of a test case examples of our test suite

4 TEST SUITE EVALUATION

We evaluated the ability to detect faults in the software under test (i.e., fault-based adequacy criterion) of our test suite by applying mutation testing on QUICKXPLAIN and FASTDIAG AAFM solutions for the conflict detection and diagnosis of product configuration in the FaMa open-source framework [1], respectively. FaMa integrates different analysis components (so-called reasoners) for the AAFM process.

Mutation testing [2, 42] is a common fault-based testing technique to measure the effectiveness of fault detection in test cases. Briefly, the method works as follows: first, simple faults are introduced in a program creating a collection of faulty versions (mutants), i.e., mutants are created from the original program by applying syntactic changes to its source code (e.g., i++ for i–). A mutation operator determines each syntactic change. Test cases are then used to check whether the mutants and the original program produce the same responses. If differences exist between the original program and a mutant instance, then the test case finds faults in the mutant and kills that program. Otherwise, the mutant remains alive. Mutants that keep the program's semantics unchanged are referred to as equivalent. The detection of equivalent mutants is an undecidable problem in general. The percentage of killed mutants regarding their total number (discarding equivalent mutants) provides an adequacy measurement of the test suite called mutation score.

We selected the key classes of the QUICKXPLAIN and FASTDIAG solutions in the FaMa framework for the mutation testing. First, we use the PIT Eclipse plug-in v1.5.2 [2] for an automated mutation testing process. PIT permits generating and executing mutants in separate steps. We obtained a high-rate of alive mutants using PIT. Hence, we decided to apply increasing and equivalent variations for the mutants of both solutions. We followed three steps for the second test suite evaluation: $i)$ *Validating existing solutions*. We made sure that existing QUICKXPLAIN and FASTDIAG solutions passed all the test cases in our suite. $ii)$ *Mutants generation*. We applied traditional mutation to common programming languages features such as arithmetic (e.g. $++$, $--$) and relational operators (e.g. $==$, $!=$). $iii)$ *Mutant execution*. We created a class for testing each mutant and evaluating whether or not it remained alive. .

Tables 3 and 4 depict the results for our test suite using the PIT framework and incremental testing of CONFPROD and DIAGPROD in the FaMa framework, respectively. Both tables present the total

Operation	Choco2Reasoner		
	Mutants	**Alive**	**Score**
ConfProd	178	29	83.7
DiagProd	170	24	85.8
Total	**348**	**53**	**84.8**

Table 3: PIT Mutants execution results

Operation	Choco2Reasoner		
	Mutants	**Alive**	**Score**
ConfProd	304	0	100
DiagProd	304	0	100
Total	**608**	**0**	**100**

Table 4: Own Solution Mutants execution results

number of tested mutants, the number of alive mutants and mutation score for applying ConfProd and DiagProd to the test cases. The number of test cases is lower in Table 3. Our testing solution considers the following increasing mutations: i) Changing relational operators (e.g. $==$, $!=$); ii) Changing logic connectors (e.g. $\&\&$, $||$); iii) Changing math operators ($+$ by $-$, $*$ by $/$ and vice-versa); iv) Returning the opposite value in the consistency check function . Hence, the number of test cases in Table 4 is only 304 for each tested operation. Tables 3 and 4 show 84.8% and 100% for the adequacy measurement of the test suite of both tested operations, respectively. We used the Choco2 reasoner in this evaluation.

5 RELATED WORK

The works of [11] and [13] describe processes for the maintenance of consistent knowledge bases to look for failures in those bases, and the main steps for implementing a knowledge-base recommender system to identify consistent products according to the customers' wishes and needs. Their focus is like the primary goal of our work's tested operations, i.e., minimal conflict and diagnosis detection of inconsistent configuration.

Concerning the FM test suite, Segura et al. [42] describe an implementation–independency set of test cases (FaMa Test Suite) for testing the functioning of different FM operations. Kitamura et al. [31] propose a test-case design method for the selection of features in the FM tree to perform black-box testing of FMs. The work of Lopez-Herrejon et al. [35] describe a suite of 19 "realistic" FMs and their products for a systematic Combinatorial Interaction Testing (CIT) of software system configurations. Galindo et al. [20] present TESALIA, a framework for selecting the most value SPL products for testing applications regarding their value and cost of the given set of constraints. Like the work of [42], our test suite represents an implementation-independent set of test cases.

Regarding testing algorithms, Lopez-Herrejon et al. [34] define a framework to assess and compare three pairwise testing algorithms of CIT in SPLs. [21] describe an approach based on constraint optimization and heuristic search to reach a max testing coverage. [23] introduce a model-based approach to reduce the number of regression tests using abstraction techniques and focusing on the changing properties being tested. Our test suite focuses on minimal preferred conflict and minimal preferred diagnosis of feature model configurations.

For mutation testing, the work of Henard et al. [24] proposes two mutation operators to derive erroneous FMs (mutants) from an original FM to check the validity of the original FM products toward the mutants. Ferreira et al. [17] describe a mutation approach to select products for the feature testing of SPLs. Jacob et al. [32] propose seven mutation operators for injecting variability faults in feature-oriented programming solutions.

Concerning the multi-objective nature of the testing process of SPLs, Lopez-Herrejon et al. [33] propose a zero-one mathematical linear program for solving the multi-objective problem of minimizing the number of test products and maximizing the pairwise coverage in FMs of SPLs. The work of Hierons et al. [25] devises the Grid-based Evolution Strategy (GrES) for the many-objective optimization problem in testing SPLs. We used mutation testing with PIT to get a high percentage of alived mutants. We also applied increasing and equivalent variations in the evaluated operations to get a 0% of alived mutants.

6 CONCLUSIONS

We designed FaMaProdConf TestSuite for the functional testing of minimal conflict detection and preferred diagnosis of SPL product configurations. We created a representative set of input-output combination by using popular techniques from the software testing community. In terms of the relevance of ConfProd and DiagProd operations in obtaining valid product configuration, this test suite aids in rapidly detecting faults to help create, test, and improve reliability and quality solutions for such tasks. These can be used either in isolation or in addition to other testing methods. Results obtained from mutant testing with ConfProd and DiagProd demonstrate the effectiveness of FaMaProdConf TestSuite. We applied mutation testing on ConfProd and DiagProd integrated into the FaMa framework using one reasoner. Since tested solutions do not depend on the reasoner used, our solution provides the necessary heterogeneity for evaluation. We initially obtained an average mutation score of 84.8% that we refined by developing a mutation testing solution to get an average mutation score of 100%. We are currently developing new solutions for those goals, and this test suite will be very useful for showing their functional validity. we will design additional mutant solutions and evaluate the adequacy measurement of the test suite with other solutions for the conflict detection and diagnosis such as such as MergeXPlain [44] and FlexDiag [16], respectively. ParallelQuickXPlain and ParallelFastDiag are part of our work-in-progress.

Acknowledgements

This work has been partially funded by the EU FEDER program, the MINECO project OPHELIA (RTI2018-101204-B-C22); the TASOVA network (MCIU-AEI TIN2017-90644-REDT); and the Junta de Andalucia METAMORFOSIS project.

REFERENCES

[1] FAMA tool suite. `https://www.isa.us.es/fama/`, 2020. Accessed: 2020-31-05.

[2] Real world mutation testing. `https://pitest.org/`, 2020. Accessed: 2020-28-05.

[3] S. Apel, D. Batory, C. Kästner, and G. Saake, *Feature-Oriented Software Product Lines: Concepts and Implementation*, Springer Publishing Company, Incorporated, 2013.

[4] B. Beizer, *Black-Box Testing: Techniques for Functional Testing of Software and Systems*, Verlag John Wiley Sons, Inc, 1 edn., 1995. MR: referenziert aus Binder1999 (u.a. im Hinblick auf data flow testing strategies).

[5] D. Benavides and J.A. Galindo, 'Automated Analysis of Feature Models: current State and Practices', in *Proceeedings of the 22nd International Systems and Software Product Line Conference - Volume 1, SPLC 2018, Gothenburg, Sweden, September 10-14, 2018*, p. 298, (2018).

[6] D. Benavides, S. Segura, and A. Ruiz-Cortés, 'Automated Analysis of Feature Models 20 Years Later: A Literature Review', *Information Systems*, **35**(6), 615–636, (September 2010).

[7] L. Copeland, *A Practitioner's Guide to Software Test Design*, Artech House, Inc., USA, 2003.

[8] K. Czarnecki, S. Helsen, and U. Eisenecker, 'Formalizing cardinality-based feature models and their specialization', *Software Process: Improvement and Practice*, **10**, 7–29, (01 2005).

[9] J. De Kleer, 'Mininimum cardinality candidate generation', *20th International Workshop on Principles of Diagnosis (DX'09)*, (2009).

[10] A. Felfernig, D. Benavides, J. Galindo, and F. Reinfrank, 'Towards Anomaly Explanation in Feature Models', in *Proceedings of the 15th International Configuration Workshop*, (August 2013).

[11] A. Felfernig, G. Friedrich, D. Jannach, and M. Stumptner, 'Consistency-Based Diagnosis of Configuration Knowledge Bases', *Artif. Intell.*, **152**(2), 213–234, (February 2004).

[12] A. Felfernig, L. Hotz, C. Bagley, and J. Tiihonen, *Knowledge-based Configuration: From Research to Business Cases*, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1 edn., 2014.

[13] A. Felfernig, K. Isak, and T. Kruggel, 'Testing Knowledge-based Recommender Systems', *OGAI Journal (Oesterreichische Gesellschaft fuer Artificial Intelligence)*, **4**, 12–18, (01 2005).

[14] A. Felfernig, S. Reiterer, M. Stettinger, and J. Tiihonen, 'Intelligent Techniques for Configuration Knowledge Evolution', 51–58, (2015).

[15] A. Felfernig, M. Schubert, and C. Zehentner, 'An efficient diagnosis algorithm for inconsistent constraint sets', *Artif. Intell. Eng. Des. Anal. Manuf.*, **26**(1), 53–62, (February 2012).

[16] A. Felfernig, R. Walter, J.A. Galindo, D. Benavides, S. Polat Erdeniz, M. Atas, and S. Reiterer, 'Anytime Diagnosis for Reconfiguration', *Journal of Intelligent Information Systems*, **51**(1), 161–182, (2018).

[17] J.M. Ferreira, S.R. Vergilio, and M.A. Quináia, 'Software Product Line Testing Based on Feature Model Mutation', *International Journal of Software Engineering and Knowledge Engineering*, **27**, 817–840, (2017).

[18] I. Forgacs and A. Kovacs, *Practical Test Design: Selection of Traditional and Automated Test Design Techniques*, G - Reference, Information and Interdisciplinary Subjects Series, BCS Learning & Development Limited, 2019.

[19] J. Galindo, D. Benavides, P. Trinidad, A. Gutiérrez-Fernández, and A. Ruiz-Cortés, 'Automated Analysis of Feature Models: Quo Vadis?', *Computing*, **101**(5), 387–433, (2019).

[20] J.A. Galindo, H. Turner, D. Benavides, and J. White, 'Testing Variability-intensive Systems Using Automated Analysis: An Application to Android', *Software Quality Journal*, **24**(2), 365–405, (June 2016).

[21] A. Gotlieb, M. Carlsson, D. Marijan, and A. Pétillon, 'A New Approach to Feature-based Test Suite Reduction in Software Product Line Testing', in *Proceedings of the 11th International Conference on Software Engineering and Applications*, pp. 48–58, (01 2016).

[22] D. Graham, E.V. Veenendaal, I. Evans, and R. Black, *Foundations of Software Testing: ISTQB Certification*, International Thomson Business Press, 2008.

[23] A. Gujgiczer, M. Elekes, O. Semeráth, and A. Vörös, 'Towards Model-Based Support for Regression Testing', in *Proceedings of the 24th PhD Mini-Symposium*, pp. 26–29, Budapest, Hungary, (January 2017). Budapest University of Technology and Economics, Department of Measurement and Information Systems.

[24] C. Henard, M. Papadakis, G. Perrouin, J. Klein, and Y. Le-Traon, 'Assessing Software Product Line Testing Via Model-Based Mutation: An Application to Similarity Testing', *Proceedings - IEEE 6th International Conference on Software Testing, Verification and Validation Workshops, ICSTW 2013*, 188–197, (03 2013).

[25] R.M. Hierons, M. Li, X. Liu, J.A. Parejo, S. Segura, and X. Yao, 'Many-Objective Test Suite Generation for Software Product Lines', *ACM Transactions on Software Engineering and Methodology*, **29**(1), (January 2020).

[26] S. Ibraheem and S. Ghoul, 'Software Evolution: A Features Variability Modeling Approach', *Journal of Software Engineering*, **11**, 12–21, (2017).

[27] IEEE, 'IEEE Standard for Software and System Test Documentation', *IEEE Std 829-2008*, 1–150, (2008).

[28] U. Junker, 'QuickXPlain: Preferred Explanations and Relaxations for Over-constrained Problems', in *19th National Conference on Artificial Intelligence*, pp. 167–172, San Jose, CA, (2004). AAAI Press.

[29] C. Kaner, J.L. Falk, and H.Q. Nguyen, *Testing Computer Software, Second Edition*, John Wiley Sons, Inc., USA, 2nd edn., 1999.

[30] K.C. Kang, S.G. Cohen, J.A. Hess, W.E. Novak, and A.S. Peterson, 'Feature-Oriented Domain Analysis (FODA) Feasibility Study', Technical report, Carnegie-Mellon University Software Engineering Institute, (November 1990).

[31] T. Kitamura, N.T.B. Do, H. Ohsaki, L. Fang, and S. Yatabe, 'Test-Case Design by Feature Trees', in *Leveraging Applications of Formal Methods, Verification and Validation. Technologies for Mastering Change*, eds., Tiziana Margaria and Bernhard Steffen, pp. 458–473, Berlin, Heidelberg, (2012). Springer Berlin Heidelberg.

[32] J. Krüger, M. Al-Hajjaji, T. Leich, and G. Saake, 'Mutation Operators for Feature-Oriented Software Product Lines', in *Proceedings of the 23rd International Systems and Software Product Line Conference - Volume A*, SPLC '19, p. 12, New York, NY, USA, (2019). Association for Computing Machinery.

[33] R. Lopez-Herrejon, F. Chicano, J. Ferrer, A. Egyed, and E. Alba, 'Multi-Objective Optimal Test Suite Computation for Software Product Line Pairwise Testing', *IEEE International Conference on Software Maintenance, ICSM*, 404–407, (09 2013).

[34] R. Lopez-Herrejon, J. Ferrer, F. Chicano, E.N. Haslinger, A. Egyed, and E. Alba, 'A Parallel Evolutionary Algorithm for Prioritized Pairwise Testing of Software Product Lines', 1255–1262, (2014).

[35] R. Lopez-Herrejon, J. Ferrer, F. Chicano, E.N. Haslinger, A. Egyed, and E. Alba. Towards a Benchmark and a Comparison Framework for Combinatorial Interaction Testing of Software Product Lines, 2014.

[36] A. P. Mathur, *Foundations of Software Testing: For VTU*, Pearson Education India, 2013.

[37] F. Meziane and S. Vadera, *Artificial Intelligence Applications for Improved Software Engineering Development: New Prospects*, Information Science Reference - Imprint of: IGI Publishing, Hershey, PA, 2009.

[38] H. Mohanty, J.R. Mohanty, and A. Balakrishnan, *Trends in Software Testing*, Springer, Singapore, 2017.

[39] I. Pill and T. Quaritsch, 'RC-Tree: A Variant Avoiding All the Redundancy in Reiter's Minimal Hitting Set Algorithm', in *Proceedings of the 2015 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW)*, ISSREW '15, p. 78–84, USA, (2015). IEEE Computer Society.

[40] R. Reiter, 'A Theory of Diagnosis from First Principles', volume 32, p. 57–95, GBR, (April 1987). Elsevier Science Publishers Ltd.

[41] P. Rodler, 'Understanding the QuickXPlain Algorithm: Simple Explanation and Formal Proof', *CoRR*, **abs/2001.01835**, (2020).

[42] S. Segura, D. Benavides, and A. Ruiz-Cortés, 'Functional Testing of Feature Model Analysis Tools: a Test Suite', *IET Software*, **5**(1), 70–82, (2011).

[43] S. Segura, A. Durán, A.B. Sánchez, D.L. Berre, E. Lonca, and A. Ruiz-Cortés, 'Automated Metamorphic Testing of Variability Analysis Tools', *Softw. Test. Verification Reliab.*, **25**(2), 138–163, (2015).

[44] K. Shchekotykhin, D. Jannach, and T. Schmitz, *Parallel Model-Based Diagnosis*, 547–580, Springer International Publishing, Cham, 2018.

[45] S. Tilley, *Software Testing in the Cloud: Perspectives on an Emerging Discipline: Perspectives on an Emerging Discipline*, Advances in Computer and Electrical Engineering (2327-039X), Information Science Reference, 2012.

[46] C. Vidal, 'Exploring efficient analysis alternatives on feature models', in *SPLC (B)*, pp. 150–155. ACM, (2017).

[47] C. Vidal, J.A. Galindo, J. Giráldez-Cru, and D. Benavides, 'Automated completion of partial configurations as a diagnosis task. Using FastDiag to improve performance', Graz, Austria, (2020).

[48] J. White, D. Benavides, D.C. Schmidt, P. Trinidad, B. Dougherty, and A. Ruiz-Cortés, 'Automated Diagnosis of Feature Model Configurations', *J. Syst. Softw.*, **83**(7), 1094–1107, (July 2010).

[49] Z. Yin, X. Ma, J. Zheng, Y. Zhou, L.N. Bairavasundaram, and S. Pasupathy, 'An Empirical Study on Configuration Errors in Commercial and Open Source Systems', in *Proceedings of the Twenty-Third ACM Symposium on Operating Systems Principles*, pp. 159–172, (2011).

Towards machine learning based configuration

Mathias Uta[1] and **Alexander Felfernig**[2]

Abstract. Nowadays, configuration technology is one of the most well-known and utilized applications of Artificial Intelligence (AI) which relies mainly on constraint-based approaches. Dependencies between features of the configured product are modeled as constraint satisfaction problems (CSP). This approach inherits some drawbacks considering the huge effort knowledge engineers have in maintaining knowledge bases, especially in complex configuration scenarios. In this paper, we propose an alternative configuration approach by utilizing machine learning (ML) algorithms and show that this technology might be a gamechanger for future configuration and recommendation approaches. To demonstrate the possibilities of ML in the configuration domain we implemented a prototype and showed its effectiveness in a short case study.

1 INTRODUCTION

Deep neural network (DNN) recommenders have been extensively researched for item recommendations such as movies, apps etc. The complexity of such recommendations is defined by the sparse information available to make suggestions to the user. In contrast, product configuration recommendation tasks can be performed on very specific user input, enriched during the process of configuration. The user is guided through the configuration process and provides relevant input to the system. Whereas, this fact simplifies the recommendation task, the fact that the available solutions are restricted by constraints complicate it.

In the domain of product configuration, traditional recommender systems are knowledge based, meaning they rely on explicitly expressed rules and constraints. These rules need to be formalized with big effort based on domain expert knowledge. Further downsides of the usage of constraints is the high maintenance effort which must be performed to sustain a conflict free knowledge base and the performance of constraint-based recommendations.

In this paper, we introduce a neural network that can recommend valid configuration options based on the given user input without utilizing constraints. Therefore, we apply a machine learning algorithm able to learn on basis of historic configurations. We apply our implementation to complex configuration tasks (components of a high voltage switchgear system configurator) to demonstrate its effectiveness. Finally, we show the increasing accuracy of our implementation in correlation to the percentage of parameters defined by user input during the configuration.

The remainder of this paper is structured as follows. In Section 2 we discuss related work and give an overview about state-of-the-art recommendation technologies. We continue in Section 3 with the presentation of our prototype implementation of a neural network to perform machine learning based recommendations for configuration tasks. To demonstrate the effectiveness of our model we apply the neural network in a case study to a complex configuration task in Section 4. Finally, we conclude our results and give an overview about further research activities in Section 5.

[1] Siemens Gas & Power GmbH & Co. KG, Erlangen, Germany, email: mathias.uta@siemens.com

[2] Graz University of Technology, Graz, Austria, email: alexander.felfernig@ist.tugraz.at

2 RELATED WORK

In commercial applications (e.g. product configurators) as well as on online platforms (e.g. online marketplaces and online video platforms) the amount of information, products and options to choose from are extensively overwhelming and overburden users [1]. In consequence, already starting in the early 2000s recommendation systems have been researched in the product configuration domain [2] and in the online platform domain [3].

Basic recommendation technologies which are applied in both domains are collaborative filtering, popularity-based, entropy-based and, utility-based feature recommendation [4]. Especially collaborative filtering, a technology, where the nearest neighbor items or features are proposed to the user, is extensively used for item recommendation such as movies, apps etc., but is also applied for product configurators. Unfortunately, features in product configuration scenarios are, unlikely to movies and apps, constraint in the sense of possible combinations the product allows.

Based on these restrictions, knowledge-based recommenders have been developed for CSPs. They mainly aim to utilize the product knowledge of the knowledge base to relax invalid configurations by providing explanations in such situations [5]. In contrast, in the online platform domain the collaborative filtering technology has been extended by applying ML algorithms by taking advantage of the capabilities deep neural networks supply to improve the quality of recommendations. The downside of this approach is the extreme memorization of former recommendations and the subsequent categorization of users. To counteract this behavior, Cheng et al. proposed a neural network which allows also a generalization to widen the user suggestions [6]. Because of the successful implementation of deep neural networks in item recommenders, recent research aims to evaluate the performance of those solutions for solving CSPs. Whereas, Wang & Tsang already showed in the early 1990s that neural networks can solve CSPs [7], Xu, König & Kumar showed recently that binary CSPs can be solved with a very high accuracy [8]. Furthermore, in the domain of product configuration the value of historic configurations has been recognized and methods to utilize this knowledge for recommendations have been researched by applying methods like matrix factorization [9].

In consequence, a convergence of both research domains can be observed and an application of ML and DNNs to configuration tasks is a logical conclusion of the scientific development.

In the following sections we show a novel approach in applying a neural network to a non-binary CSP which allows feature value recommendations for product configurators.

3 NEURAL NETWORK

We trained a DNN to find a valid solution for a configuration problem by making a single assignment to a variable that is globally consistent with the previously selected variable feature values.

We used one-hot encoding to represent the assignment of a variable, meaning that each possible feature value is represented as one bit in a vector. The assigned i-th value of the feature corresponds, consequently, with the assignment of 1 to the i-th bit of the vector, while all other bits remain 0. If no bit is raised the variable is unassigned. We have chosen fully connected layers rather than convolutional layers since they have proven to be more effective for combinatorial problems [10].

As depicted in **Figure 1** the network is designed with only three layers, which have shown the most accurate learning behavior and the least deviation from the target function. The input-layer consists of as many neurons as assigned input variables are given to the network, followed by a hidden-layer with the same number of neurons. Finally, the number of neurons in the output-layer corresponds to the number of possible values of the variable that should be predicted. For example, five variables of a configuration task are already selected by the user and the sixth variable with three possible values should be predicted by the ML algorithm. The resulting neural network will have the shape of 5-5-3 neurons.

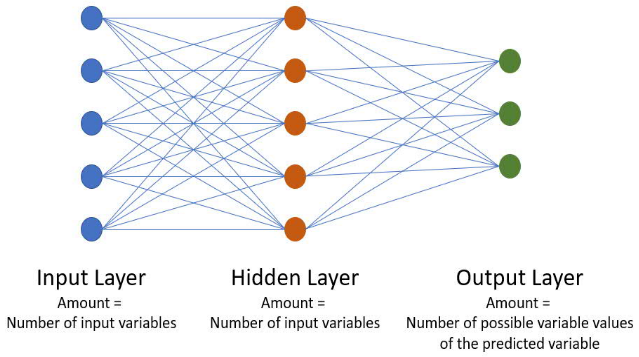


Figure 1. “Non-Deep” feedforward neural network

During the training phase existing configurations (where all six variables are already assigned), so called labelled examples, are processed through the network. Patterns within the training sets are determined and saved via a weight which is assigned to each neuron. Based on this resulting model a validation is performed by predicting unlabeled examples (where the sixth variable is unassigned) and depended on the model quality and its ability of generalization, the accuracy of the predicted values is high or low.

Extensive tests with our configuration datasets have surprisingly shown that the accuracy is lowered, and the loss is increased if more than these three layers are added to the network. Widen as well as deepening of the network reveals less performance in the sense of time required to train the model and, in predictions made. Therefore, we refer to this “non-deep” feedforward neural network, if a neural network is mentioned in the remainder of this paper.

We have trained our model based on randomly selecting a test and a validation set, where the training set included 80% of the dataset and the remaining 20% have been used to validate the model. The choice of the training hyper-parameters have been made based on several test executions. Optimizer “Adam” [11] has shown the best performance in compared to other gradient decent optimizers like “ADAGRAD”, “RMSProp” and “SGD” [12]. “Adam” uses adaptive estimation of first order and second order moments, which slows down the adjustment of neuron weights the more steps have been done. The selected parameters for the “Adam” optimizer were an initial learning rate of 0.001, β_1=0.9 and β_2=0.999. Each neuron is utilizing a “ReLu”-activation function. The learning was performed by utilizing randomly selected mini batches (10% of the training set) for as many steps till no improvement of the loss was observable.

To demonstrate the effectiveness of this model we will introduce complex configuration tasks in the following section and discuss the achieved results.

4 CASE STUDY

As a case study we have extracted configuration data from a product configurator developed and maintained by Siemens AG. This implies some restrictions according to the public availability of the data presented in this paper. The referenced configurator is used to guide users through the complex configuration of high-voltage switchgears. Based on the selection of variables, the result of the configuration is a bill-of-material. Available variable values are highly interdependent. The knowledge and its formalization is realized by constraints resulting in a typical CSP. The variables which are specified during the configuration process can be categorized into “basic” and “determination” variables. While “basic” variables describe the general conditions which are given, such as voltage level and ambient conditions, “determination” variables are defining specific components of the switchgear dependent on these basic variables (such as type of control cubicle or circuit breaker). The combination of the selected “determination” variables leads to the full bill-of-material necessary to build the specified product.

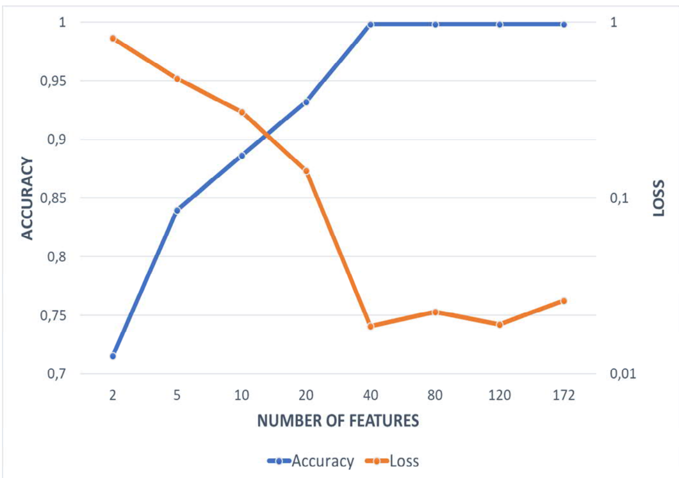


Figure 2. Accuracy and loss over number of input variables (control cubicle example)

For training purposes, 3200 valid high-voltage switchgear configurations were utilized to test the performance of the previously introduced neural network. Each configuration consists of 172 variables to describe the switchgear and a set of nine bill-of-material positions to describe a circuit breaker (one component of the resulting switchgear). All variables and bill-of-material positions had between 2 and 14 possible values.

The first example, which has been investigated more thoroughly, is the prediction of a "determination" variable, namely the "Control cubicle type" to choose for the high-voltage switchgear. The control cubicle can occur in eight different variants (Val 1 – Val 8) and is constrained by several other variable selections like "Control type", "Cable type", "Voltage level" etc. To validate the prediction performance and the resulting loss of the predicted function in correlation to the degree of specifications done already by the user, a test series has been executed.

To simulate the configuration process, the first prediction is performed only by applying the information about the user who is configuring ("End customer name" and "End customer country"). Then like for a normal configuration, "basic" variables (rated values) are subsequently added to the configuration (number of variables: 5, 10 and 20). They are followed by variables relevant for the "Control cubicle type" selection, which are subsequently added to the configuration (number of variables: 40 and 80). Finally, all other variables are added to the model (number of variables: 120 and 172). **Figure 2** depicts the achieved results in terms of prediction accuracy (**Equation 1**) and cross entropy loss (**Equation 2**), which are usually applied for classification problems.

$$Accuracy = \frac{TP}{TP + FP} \quad (1)$$

TP = true predicted variable values
FP = false predicted variable values

$$-\sum_{c=1}^{M} y_{o,c} \log (p_{o,c}) \quad (2)$$

M = number of possible variable values
y = binary indicator (0 or 1) if variable value c is the correct classification for prediction o
p = predicted probability observation o is variable value c

Even with the availability of only the user information, a prediction accuracy of 71.5% can be observed. The accuracy rapidly rises with the enrichment of the model with further variable selections till it converges to a maximum accuracy of 99.8%. The maximum value is reached already with 40 determined variables and stays stable till the maximum variables are added. As expected, an oppositional development can be found for the loss function which reaches a minimum where the first accuracy maxima is reached (number of variables: 40). But with the addition of further variables a slow increase of the loss is observed, which could be explained with the increasing complexity of the learned model.

To further investigate the recommendations made by the neural network **Table 1** shows the distribution of the predicted variable values compared to the expected distribution.

Based on the results depicted in **Table 1** it can be stated that a scarcity of features to learn from, leads to a prediction behavior which neglects rarely included values (Val 7 and Val 8). Instead, frequently appearing values like Val 4 and Val 5 are predicted even more often since they are the most likely choices if not enough information is included in the dataset, to make better predictions. Only with the enrichment of the configuration with further details a convergence to the correct value selection can be observed. Nevertheless, it is worth to emphasizes that the neural network can predict even variable values which occur very rarely in the data – a very promising behavior which implies that the model learns the constraints included in the data.

Table 1. Distribution of predicted variable values compared to expected distribution (control cubicle example)

Number of features	Val 1	Val 2	Val 3	Val 4	Val 5	Val 6	Val 7	Val 8
2	65	83	56	242	127	0	0	0
5	60	92	50	253	98	20	0	0
10	63	90	59	217	105	30	6	3
20	61	90	58	183	108	58	13	2
40	70	90	57	191	107	49	7	2
80	70	90	57	191	107	49	7	2
120	70	90	57	191	107	49	7	2
172	70	90	57	191	107	49	7	2
Expected	**70**	**91**	**57**	**191**	**106**	**49**	**7**	**2**

Conspicuous is the wrong prediction of one Val 2 configuration, which can be explained by an anomaly in the validation data. For all 90 correctly predicted selections of Val 2 is extraordinal one of the constraining input variables selected with only one specific value. This variable is for the wrongly predicted selection differently allocated. Consequently, extreme deviations from the normal distribution can lead to false learning results, which might be prohibited by closely examining the validation data.

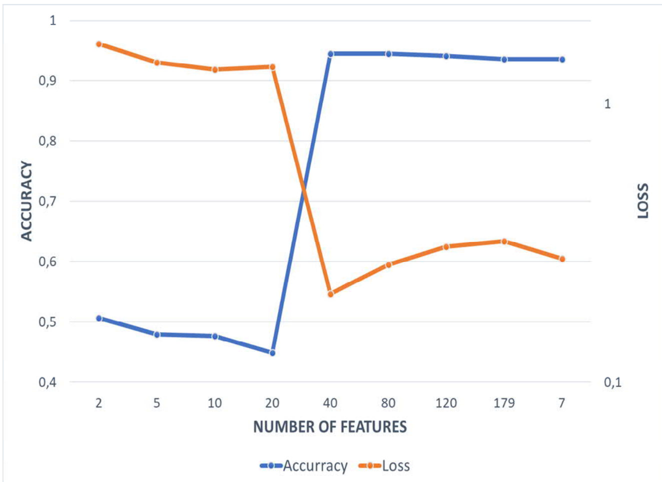


Figure 3. Accuracy and loss over number of input variables (bill-of-material circuit breaker example)

The second example, which was investigated based on our configuration dataset, was an even more challenging configuration task. We applied our model to predict the position of the bill-of-material of a high-voltage switchgear component, a circuit breaker. This position has 14 possibilities, which are in the constraint-based environment of the configurator dependent on the selection of seven "determination" variables. The results of the similar test series, as it was applied in our first scenario, are shown in **Figure 3**. The accuracy achieved by only applying the user information ("End customer name" and "End customer country") to the model is ca. 50%. The addition of further "basic" variables has interestingly no impact on the accuracy. Only the loss is slightly reduced. Even, if the accuracy is not as good as for the previously described simpler example, it is still noticeable that the model allows a 7-times better performance than guessing would.

With the introduction of the constraint relevant variables an extreme improvement is achieved. Again, the best result (94.4% accuracy) is observed with the least meta information included in this scenario (40 variables) but does not reach the level of the simpler prediction task. More variables lead, as already suspected, for complex predictions to a reduction of the accuracy (93.5%) and a rise in loss. The obvious assumption that the highest accuracy can be accomplished by only including the seven relevant variables into the model could not be confirmed (93.5% accuracy). Consequently, the neurons learn something from the data which is not explicitly formulated in the configuration knowledge base. However, overloading the model leads to a decreasing prediction performance and an optimum of variables involved exists.

5 CONCLUSION AND FUTURE WORK

In this paper, we have shown the that a ML algorithm is able to learn complex constraints within the context of product configuration. Even though the available dataset only included 3200 configuration examples – a very restricted amount of training data compared to other ML environments – the results presented, hypothesize an application of this technology in the configuration domain and motivate further research in that area.

Nevertheless, a full replacement of constraints seems improbable since prediction accuracy will most likely never reach 100%. But in the environment of product configuration, a probability of misinterpretation is unacceptable. Therefore, a user supporting role of ML as a recommendation application seems more promising. The capabilities shown in our examples have already achieved very good results (71.5% and 50.6%) with the rare input of user data and reaches values of over 95% accuracy with all relevant data included. To supplement these results more case studies and a deeper understanding of the applies neural network are necessary.

On the one hand future research will have to focus on analysis of the behavior of single neurons and on the other hand on validation if the here applied "non-deep" feedforward neural network remains the most promising network structure for CSP solution predictions. Closer observation of the resulting weights of each neuron in comparison with the variation of network hyperparameters as learning rate, utilized optimizer, activation functions etc. is of high interest. Furthermore, a closer investigation of the in the second example observed higher accuracy of models with included meta information compared to the model with only constraint relevant variables is essential. It is of high interest to determine the minimal number of variables to achieve the highest possible accuracy with the given dataset.

Additionally, neural networks with the capability to predict more than one variable at a time are desirable. In the product configuration domain, huge amounts of variables must be determined and a recommendation for the whole set of missing variables would be very helpful. Finally, further research on the integration of ML recommendation in the user interface is necessary to assure a reasonable usage of the new technology. For example, a visualization of possibilities for each selectable variable value might be a better representation of the prediction than the direct classification by the algorithm.

REFERENCES

[1] Fleischanderl, G., Friedrich, G., Haselboeck, A., Schreiner, H. & Stumptner, M. Configuring Large Systems Using Generative Constraint Satisfaction. IEEE Intelligent Systems, Special Issue on Configuration 13(4): 59–68, (1998).

[2] Coester, R.; Gustavsson, A., Olsson, R., and Rudstroem, A. Enhancing Web-Based Configuration with Recommendations and Cluster-Based Help. Paper presented at the Workshop on Recommendation and Personalization in ECommerce, Malaga, Spain, (2002).

[3] Herlocker, J, Konstan, J., Terveen, L., and Riedl, J. Evaluating Collaborative Filtering Recommender Systems. ACM Transactions on Information Systems 22, ACM Press, 5-53, (2004).

[4] Falkner, A., Felfernig, A., & Haag, A. (1). Recommendation Technologies for Configurable Products. AI Magazine, 32(3), 99-108. https://doi.org/10.1609/aimag.v32i3.2369, (2011).

[5] Felfernig, A.; Schubert, M.; Friedrich, G.; Mandl, M.; Mairitsch, M.; & Teppan, E. Plausible Repairs for Inconsistent Requirements. In Proceedings of the 21st International Joint Conference on Artificial, (2009).

[6] Cheng, H.-T. et al. Wide & Deep Learning for Recommender Systems. In Proceedings of the 1st Workshop on Deep Learning for Recommender Systems (DLRS 2016). Association for Computing Machinery, New York, NY, USA, 7–10, (2016).

[7] Wang, C.J. & Tsang, E. Solving constraint satisfaction problems using neural networks, IEE Conference Publication, 295 – 299, (1991).

[8] Xu, H., Koenig, S. and Kumar, T.K.S. Towards Effective Deep Learning for Constraint Satisfaction Problems. In: Hooker J. (eds) Principles and Practice of Constraint Programming. CP 2018. Lecture Notes in Computer Science, vol 11008. Springer, Cham, (2018).

[9] Erdeniz, S.P., Felfernig, A., Samer, R. & Atas, M. Matrix factorization based heuristics for constraint-based recommenders, In Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing (SAC '19), Association for Computing Machinery, New York, NY, USA, 1655–1662, (2019).

[10] Galassi, A., Lombardi, M., Mello, P. & Milano, M. Model Agnostic Solution of CSPs via Deep Learning: A Preliminary Study. 254-262. 10.1007/978-3-319-93031-2_18, (2018).

[11] Kingma, D. P. & Ba, J. Adam: A Method for Stochastic Optimization, Computer Science, Mathematics, ICLR, San Diego, (2015).

[12] Ruder, S. An overview of gradient descent optimization algorithms, arXiv preprint arXiv:1609.04747, (2016).

Configuration assisted through conversational agents (chatbots and voicebots)

Nicolás Afonso[1] and **José A. Galindo**[2] and **David Benavides**[3]

Abstract. Recent years have seen a rise in the use of technologies supporting natural language understanding as technological bases for implementing dialog systems (a.k.a. conversational agents). For example, conversational agents are present in many web front ends supporting sales and first customer contact. Separately, existing software variablity-intensive systems complexity has been increased dramatically due to the rise of cloud based offer with the appearing of multiple commercial providers. These facts require professionals to move frequently between different knowledge fields in short time to react in front a changing market. This results on an increasing demand of highly skilled employees. We think that such changes could be addressed if we serve this configuration through a conversational agent using the same conversational agents which are widely used in other fields.

This paper presents our ongoing work towards realizing a configuration virtual assistant for variablity-intensive systems. We think users can interact with this assistant through a conversational agent, and they shall be guided (driven by questions) in the configuration process following natural language. To realize this vision, we propose to draw on latest achievements in the field of natural language processing and understanding. We also briefly elaborate on the main challenges, and prove these concepts by developing a prototypical solution. The aim of this work is twofold, to semi-automate the configuration process and to explore enabling disabled people to overcome the barriers of traditional human computer interfaces. We understand despite, in origin, this was not the primary target of this work, it represents a beneficial to society which should not be skipped.

To demonstrate the feasibility, the team was producing a technology demonstrator which is intended to proof the correctness of exposed approach. This technology demonstrator relies on RESDEC, a Wordpress plugin recommending developed by our team which makes extensive use of Feature Model Analysis.

1 INTRODUCTION

A variability-intensive software line system is a set of software applications aimed at providing a set of similar functionalities. The crucial problem that a software producer have to cope with is to produce each software application belonging to this software line in an efficient way. For this reason Classen et al.[10] do include the ability to produce a large family of different systems efficiently in the definition of variability-intensive systems

Software product lines (SPL) are about developing a set of different software products that share some common functionality. In same sense software product line engineering is "about producing a set of related products that share more commonalities than variabilities" [3]. As an example, consider Helm charts[4]. In helm charts, we define a blueprint of a deployment of different containers and Kubernetes components to be deployed in a Kubernetes cluster. Helm charts can be customized by overwriting deployment variables defining the variability-intensive system, giving different software product lines depending on the configuration. Actually, Kubernetes could be considered itself as a variability-intensive system due to the large number of available components and add-ins and deployments performed in Kubernetes are themselves software product lines.

Feature models [21] have become the *de facto* standard to represent common and variable characteristics in variablity-intensive systems and Software Product Lines [32]. In previous example, helm charts are a way to define feature models in infrastructure as code approaches.

The foresee of public cloud revenues increase [31] allows predicting a progressive increment on the demand of IT professionals in coming years on the field of variablity-intensive systems. These professionals need to face a specific learning curve depending on the cloud provider in addition to the technology required to release their solutions. In example, a user creating an eCommerce site in Wordpress hosted by a cloud provider [18] will need to manage two separated domain areas systems; the Cloud vendor specific technology and the Wordpress stack. Both domain areas match in the definition of variablity-intensive systems. In these domains, there might exist multiple implementation components for each feature. This makes the selection of the most suited implementation a difficult and error prone process.

We refer as natural language processing (NLP) as "an area of research and application that explores how computers can be used to understand and manipulate natural language text or speech to do useful things" [9], this includes areas such as sentiment analysis or machine translation. A subset of NLP is natural language understanding (NLU) understood as "the ability of a system to detect the intentions of the user when this is expressed using human language as opposed to programming languages" [30].

[1] Strategic Technology and Incubation, Schneider-Electric Andover, Massachusetts, USA, email: nicolas.afonso@se.com

[2] Dpto. de Lenguajes y Sistemas Informáticos, University of Seville Spain, email: jagalindo@us.es

[3] Dpto. de Lenguajes y Sistemas Informáticos, University of Seville Spain, email: benavides@us.es

[4] https://helm.sh/ - Helm, package manager for kubernetes.

We'd like to remark that most of practitioners still relying on manual processes to configure feature models for these technologies as probes the work from guerriero et al about the adoption of infrastructure as code techniques[17]. In example, most of the deployments performed on the cloud make use of their commercial consoles (which are mostly intended for exploration) to deploy their solutions manually. These manual processes are error prone due to the previously described complexity and the low adoption of automated deployment on this systems [16]. Therefore, commonly, the process ends up with wrong or incomplete products. As an alternative to this manual solution, we consider a computer aided system to automatically configure feature models will reduce the technical debt of configured solutions.

Last, but not the least, job market is seriously restricted for disabled professionals such as visually impaired persons or those with neural degenerative diseases driving into disability bias for those professionals. In example, a skilled professional who suffered a traumatic loose of vision or hands, would need to re-orient his professional career while the experience would still be valid.

In this paper we evaluate the use of a conversational agent to assists multiple variablity-intensive systems. We'll go through the identified challenges and propose a potential solution by creating a generic conversational interface assisted by domain specific conversational agents. We support our study through a proof of concept where we can validate the feasibility of this approach. This goal is probing we use natural language processing and feature model analysis so in the future, we can use natural language technologies to generate infrastructure as code files to support deployment of solutions responding to the definition of software product line

2 CHALLENGES

In this section we present the main technological barriers identified so far when configuring a variablity-intensive system by relying on conversational mechanisms.

2.1 Challenge 1: Analysis of feature models to assist on configuration

This challenge consist on linking existing natural language understanding technologies with those technologies which could assist on performing the configuration. In other words, our team was challenged to being able to execute analysis of feature models using a conversational agent as we explain in this block. As we expose below, we reuse existing work to probe the problem can be overcome.

As exposed above (see 1), feature models are one of the most popular ways to represent common and variable characteristics in software variablity-intensive systems. These feature models support automatic analysis based on reasoning algorithms generically called auto-analysis of feature models (AAFM) [3]. Thus, we consider using automated analysis of feature models to support the configuration, detect potential incompatibilities or suggests features to optimize and complement a solution.

The first challenge is how a conversational agent can interact with an automated analysis of feature models platform when, traditionally , most of them are monolithic desktop solutions, that is, highly coupled solutions which are not intended to be integrated with third party solutions to reuse their functionality. The work done by Rodas et al. [27, 26, 28], proved that such automated analysis of feature models platforms can be split into two subsystems and isolate the analysis as a back end from an independent front end. By interacting with a recommendation system which exploits such automated analysis of feature models capabilities, RESDEC allowed guiding in the assistance of users on configuring Wordpress through a web application. RESDEC functionality is coped around three analysis, a basic cold start for new users, a rated base analysis based on users reviews and a third one based on feature analysis. We considered reusing this platform to integrate our conversational agent with the analysis provided by RESDEC solution.

Despite RESDEC is a Wordpress plugin recomendation engine, it makes extensive use of the technologies related to feature model analysis and suits our purposes.

2.2 Challenge 2: Choosing a natural language understanding framework

Natural language understanding (NLU) frameworks are based on the capability of a system to detect the intent of a user when expressing a question or command. [23]. In order to process this information, NLU platforms needs to handle a conversational context which prevents the virtual assistant to loose the information during a conversation. In example, RESDEC is built on top of three different algorithms chosen by end user at the beginning of the configuration. The virtual assistant should keep this information in the algorithm through the entire configuration process.

There exist multiple NLU commercial and open source solutions. Initial experiences performed in industry are driving us to consider there is not a best in class solution. Assuming this hypothesis which will be a matter of a future work, we must consider that different solutions can be addressed by different providers. In addition, most of the commercial solutions such Azure LUIS, AWS Lex or Google Dialogflow are cloud based. That can be challenging for systems developed in high security environments such as defense where there is a important component on configuring different software product lines disconnected from external networks.

In order to be able to support as many scenarios as possible, the chosen NLU framework should be able to work on premise solutions without depending on external network connections.

2.3 Challenge 3: Domain Orchestration

A problem which has not been satisfactory solved yet in NLU frameworks is related to manage conversations which could include one or more domains of knowledge. A conversational agent is relatively simple to be created to assist on a narrow topic, in example, we could have a conversational agent travel booking, and another to support a book store portal. However, there will be troubles if we merge both domains in a single conversational agent, because when the user requested "I want to book a hotel", the word embedding would show very similar distance between buying a book about hotels, or setting a stay in a city.

However, and as stated at the beginning of this work, a conversational agent should not be focus in a single configuration domain.

In example, a Wordpress configuration domain is potentially related with the cloud configuration where solution will be deployed. To illustrate this dependency, we can consider the dependency between Wordpress and the backend database. This backend database (usually MySQL) is deployed in a different way in Azure or AWS

The intuitive solution is creating a conversational agent for each domain. Some examples of different domains are the previously mentioned Wordpress, Linux and AWS. Under this approach, each specific domain would be handled by a specific conversational agent.

However, the huge number of different conversational agents have been detected to cause fatigue in users due to market saturation [6]. A potential solution has been risen through the concept of meta chatbot [1]. A meta chatbot concept is a unique conversational interface where different conversational agents are plugged. Meta chatbot acts as a conversation proxy, routing such conversation to the dedicated NLU implementation for a particular domain. Chaves et al. demonstrated that a meta chatbot solution is preferred by users to an ecosystem of conversational agents [8]. However, this approach drives into other challenges to make such approach feasible. In example, Amazon's Alexa uses an activation utterance to activate the skills and selecting the proper end conversational agents. Other vendor solution is producing dispatching agents, like the tool released by Microsoft [25], but it involves retraining the model for each change and with proper limitations of Microsoft LUIS [24].

Because the number of potential relationships, we can consider that the different configuration domains match in the definition of software product lines according to Apel et al. [2]. In advance we'll consider each domain as a separated software product line (SPL). Some examples of these SPL would be the one dedicated to configure Wordpress, Linux, or Public Cloud Providers like AWS.

In addition, there can exist relationships between different domains. In example, exposing a Wordpress site in an AWS Virtual Private Cloud requires a gateway to be open to external connections in a dedicated port. Because Wordpress is a mainly a website, the configuration assistant should be able to suggest gateway configuration.

3 PROPOSED APPROACH

Due to previously exposed analysis (see 2), we propose an idea to support configuration of multiple variability-intensive related systems by using a meta chatbot. We were unable to find any reference to conversational agents enabling configuration of variablity-intensive systems and supported automated analysis of feature models capabilities. Instead of limiting himself to the execution of orders, the agent can enrich the relationship with final users [5]. Some examples of this enrichment could be warning users on incompatibilities, suggesting components and providing access to frequent answered questions.

Each configuration domain matches in the definition of software product line (see 2.3). A software product line (SPL) can be defined with a variability model. This variability model will present a product configuration supported by the automated analysis of feature models (AAFM) operations.

In our proposal, these software product lines are accessed by specific NLP/NLU components (fig. 1). These NLP/NLU components are intended to support a domain specific conversation. In other hand, there will be a unique wrapping conversational interface to act as a meta chatbot. This conversational interface will appear a single conversational agent, but it is only a facade of an orchestration between different domain specific conversational agents [22]. We consider the previously mentioned domain specific conversational agents are open to use not only NLU capabilities, but also NLP such as language detection [12]. For this reason, we refer domain specific conversational agents as NLP/NLU components in our proposal.

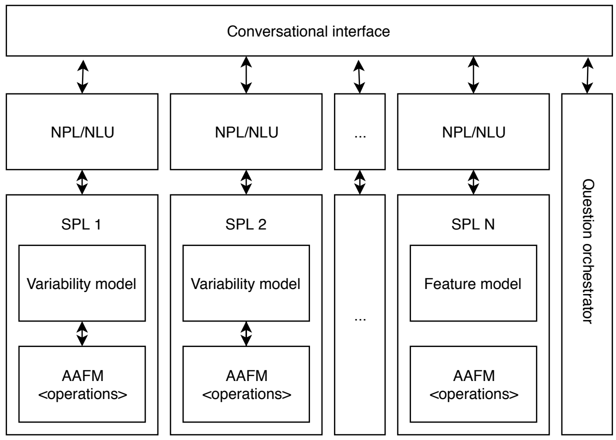


Figure 1. Components in our approach

The NLP/NLU components are exploited by a unique common conversational interface which uses an orchestrator to route the request to the proper NLP/NLU component. In that way, this conversational interface will appear to final user as a single conversational agent for system configuration while actually is a facade for the different conversational agents supporting the configuration

3.1 Automated analysis of feature models operations

For the implementation of the configuration primitives, we are relying on our previous in RESDEC [29] and FaMa [7]. Concretely, RESDEC presents the set of plugins available for configuring a Wordpress site [5] by relying on different recommendation systems. Also, FaMa enables the configuration of feature models by presenting the set of relationships within a model, their possible feature selection and propagating the selection of features.

RESDEC leverages the work done in the area of automated analysis of feature models by Benavides, Galindo et al. [4, 13, 14]. This work is intended to address off the difficulties[29] on supporting configuration systems, but has been used through traditional graphical user interfaces so far (fig. 2)

Despite RESDEC is a recommendation system, the use of feature models makes it a a good candidate to probe the feasibility of its use for assist in configuration based on feature models.

[5] www.wordpress.org

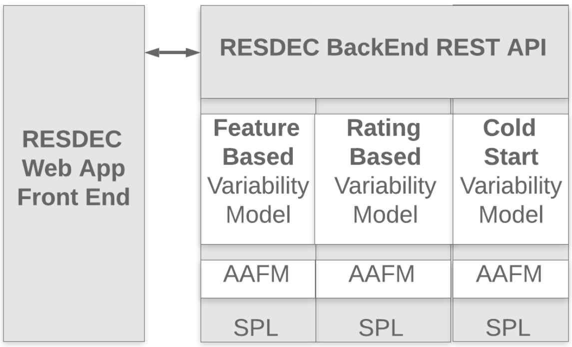


Figure 2. RESDEC - Recommendation System

3.2 NLU/NLP

We considered on-premise based solutions such as RASA ([6]) providing a full NLU/NLP Open Source framework. Most of the commercial solutions provide very short options to configure the model training of conversation dataset as they have proprietary algorithms. Some other frameworks, specially those which are based on open source solutions, provide a wider family of options for training the dataset model. Previously mentioned RASA provides the chance to choose among different algorithms, which in RASA are called NLU pipelines [20]. In example, for small training models, models can be trained using spaCy. For Frequent Questions, RASA can use BERT depending on the way it's configured model. Larger training models require supervised word vectors.

3.3 Orchestration

We propose a solution based on the concept of federation, so different NLU/NLP components can be plugged. Indeed, the unique conversational interface presents a single front end. This unique conversational interface simulates the existence of a single conversational agent, while each NLU/NLP component will have to handle their specific domains of knowledge. This federation is based on a subscription model in which each NLU/NLP component will be connected to a specific module in charge of performing Orchestration among all different subscribed subsystems. RASA provides orchestration features on RASA NLU/NLP Components, but orchestration on different technologies (in example, Google Dialogflow or AWS LEX Components) is not supported with this solution. A technology agnostic orchestrator will be a matter of a future work.

4 STATE OF ART ON OUR PROOF OF CONCEPT

To validate this idea, we're presenting our prototype that shows the feasibility of configuring variability-intensive systems through a meta chatbot. The solution is already available under request as docker images. Source code of is shareable under request too. We copy in this paper some screenshots from the actual solution to illustrate the concepts.

In our solution, we were reusing RESDEC backend features. As exposed above, RESDEC provides three recommendation engines, the cold start, the feature based and rating based, supported by different feature models. These feature models are exposed using a REST API built on Django. Such REST API had been exploited so far by an existing front-end web application using websockets. Each of these Feature Models are accessible via their own conversation flows, representing different conversation domains.

Following our proposed architecture, we used RASA Actions server (fig. 3) to handle the Orchestration. As previously explained (see 3.3), there is a limitation as NLU/NLP components must be based in RASA. A different NLU/NLP component was created to attend each one of the Software Product Lines supported by RESDEC. We define each conversation managed by a specific NLU/NLP component as conversation flow.

To support the orchestration, an additional talk flow was added. This talk flow is intended to spice conversation and drive the user conversation. It's referred generically Chit-Chat Flow and is based on RASA Smalltalk capabilities. These smalltalk capabilities are in charge of addressing human interactions for simulating human behavior, handling greetings and even jokes.

Finally, to offer a user interaction, Conversational Interface is exploited using a chat React widget customized for connecting through a web page. Is should be noticed the channel is an agnostic connection to the conversational agent and can be replaced by a voice channel by configuring the channel properly

In this way, we were able to probe we could workaround the challenges described on this paper. However, in order to simplify the reading of the paper, we were focusing in Cold start flow, which is the simplest case so the number of screenshots could be reduced

[19] to support speech capabilities.

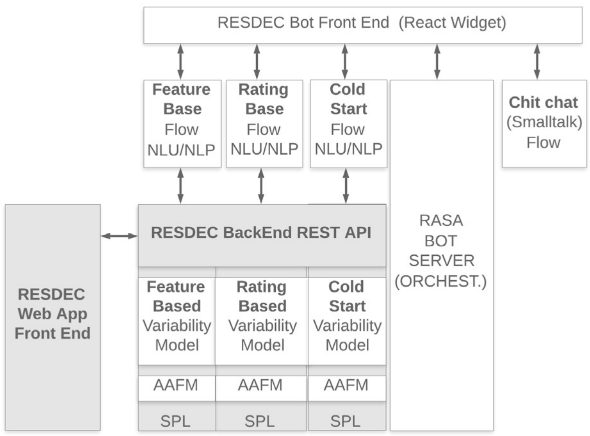


Figure 3. Rick - RESDEC Virtual Agent

[6] http://rasa.com

4.1 Conversation Demo

4.1.1 Chit chat Flow

The conversation begins when the user initiates the communication (fig. 4). Rick introduces the platform. In this example we wait for the user to take the initiative in the conversation and ask a question which can be resolved without recurring to the back end.

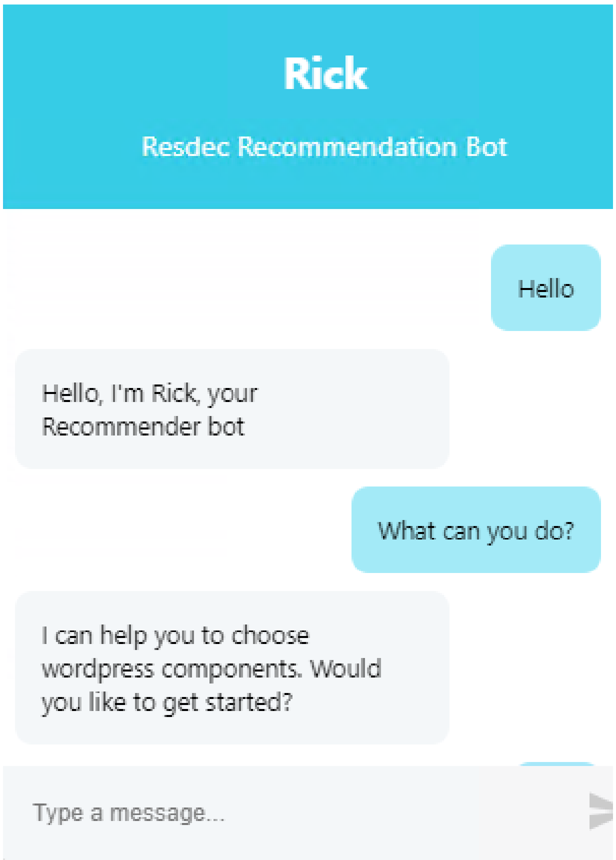


Figure 4. Rick - Chit Chat Flow in Action

4.1.2 Choosing Software Product Line and launching Cold Start Flow

The conversation follows up a natural flow directed by the chit chat flow. At some point (fig. 5), the orchestrator detects the intent as belonging to an Software Product Line (SPL) Domain, and pass the control to proper flow. To recommend a Wordpress plugin, Rick requests a key word for searching its specific variability model.

4.1.3 Finishing flow and returning control to Chit Chat Flow

Once all parameters have been gathered (in this example, there was only one single parameter), the flow executes the request against the back end (fig. 6), displays the result and returns the control to the chit chat flow.

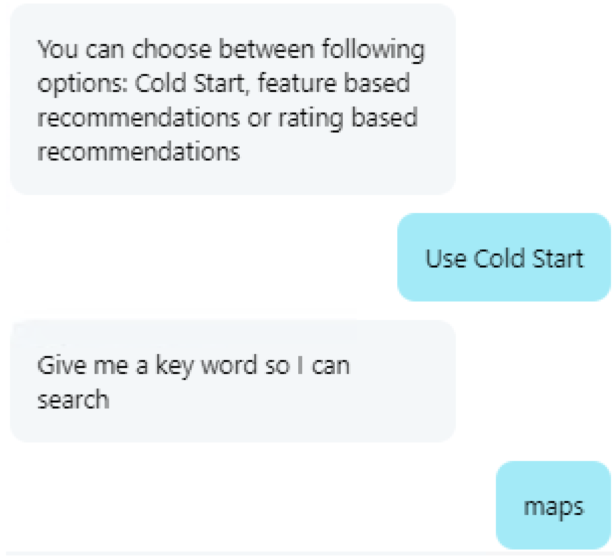


Figure 5. Rick - Cold Start Flow

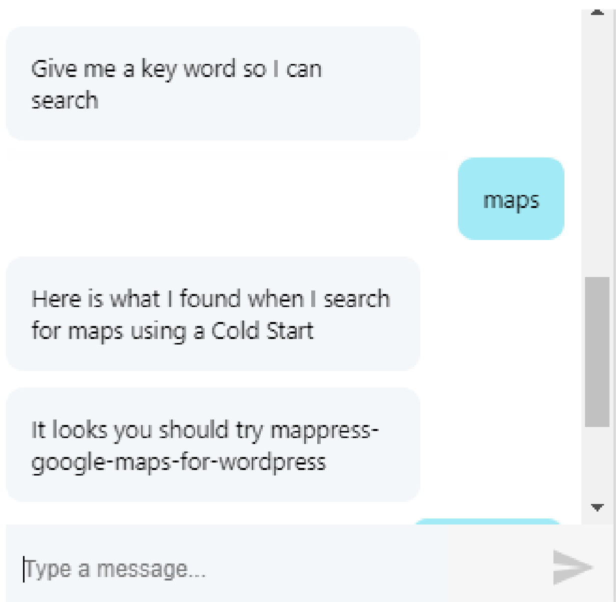


Figure 6. Rick - Flow Ending

5 RELATED WORK

In this section, we present some of the related works.

Multi-product lines. We reuse existing work around RESDEC Wordpress recommendation platform which have been extended with a conversational interface following our architectural vision.

Meta chatbot and bot federation Commercial meta chatbots are nowadays based on integrating specific conversational agents from multiple sources (i.e. Alexa Skills). However, those skills are selected or integrated during the build. That is, we cannot add a new bot that integrates with others in execution time [11]. Our approach enables the use of a bot federation that can add new bots and automatically extends the available actions that the meta chatbot can perform.

Galindo et al. [15] presented Invar, a tool that enables the configuration of multiple variability models. However, Invar does not cope with the orchestration of multiple VIS nor with the use of conversational interfaces.

6 CONCLUDING REMARKS AND FUTURE WORK

We present a solution for the problem of enabling configuration of variablity-intensive systems by relying on a simulated conversational agent composed of several domain specific NLP/NLU components serving specific variablity-intensive systems. We also explain the way in which we explore the feasibility of this work, which must be considered as work in progress.

In this sense, we have identified as future work the possibility of designing a platform agnostic orchestrator, therefore different flows could be handled by different technologies. Another pending work is interacting with a speech technology to provide access to same conversational capabilities through voice systems, such as Twilio.

Last, but not the least, we want to link the work from this paper with several studies performed in our team related to kubernetes, to generate infrastructure as Code files as outcome of the conversation.

Acknowledgements

This work has been elaborated in collaboration between Schneider-Electric Strategic Technology and Incubation Team in Andover, Massachusetts, USA and Dpto. de Lenguages is Sistemas Informáticos from University of Seville in Spain, in the frame of Schneider Digital Edison Program. It has also been partially funded by the EU FEDER program, the MINECO project OPHELIA (RTI2018-101204-B-C22); the TASOVA network (MCIU-AEI TIN2017-90644-REDT); and the Junta de Andalucia METAMORFOSIS project.

REFERENCES

[1] Pelumi Aboluwarin. Chatbots - igniting division of labour in ai, 2016.
[2] Sven Apel, Don Batory, Christian Kästner, and Gunter Saake, *Feature-oriented software product lines*, 2016.
[3] David Benavides, Sergio Segura, and Antonio Ruiz-Cortés, 'Automated analysis of feature models 20 years later: A literature review', *Information systems*, **35**(6), 615–636, (2010).
[4] David Benavides, Sergio Segura, and Antonio Ruiz-Cortés, 'Automated analysis of feature models 20 years later: a literature review', *Information Systems*, **35**(6), (2010).
[5] Robbert-Jan Beun, Eveliene de Vos, and Cilia Witteman, 'Embodied conversational agents: Effects on memory performance and anthropomorphisation', (2003).
[6] Eileen Brown. Seven out of ten americans have chatbot fatigue, 2019.
[7] *Systems and Software Variability Management - Concepts, Tools and Experiences*, eds., Rafael Capilla, Jan Bosch, and Kyo Chul Kang, Springer, 2013.
[8] Ana Paula Chaves and Marco Aurelio Gerosa, 'Single or multiple conversational agents?: An interactional coherence comparison', in *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, p. 191. ACM, (2018).
[9] Gobinda G Chowdhury, 'Natural language processing', *Annual review of information science and technology*, **37**(1), 51–89, (2003).
[10] Paul Clements and Linda Northrop, *Software product lines*, Addison-Wesley Boston, 2002.
[11] Hao Fang, Hao Cheng, Maarten Sap, Elizabeth Clark, Ari Holtzman, Yejin Choi, Noah A Smith, and Mari Ostendorf, 'Sounding board: A user-centric and content-driven social chatbot', *arXiv preprint arXiv:1804.10202*, (2018).
[12] J-M. Favre, D. Gasevic, R. Lämmel, and E. Pek, 'Empirical language analysis in software linguistics', in *Software Language Engineering*, volume 6563 of *LNCS*, 316–326, Springer, (2011).
[13] José A. Galindo, David Benavides, Pablo Trinidad, Antonio Manuel Gutiérrez-Fernández, and Antonio Ruiz-Cortés, 'Automated analysis of feature models: Quo vadis?', *Computing*, **101**(5), 387–433, (2019).
[14] José A Galindo, David Benavides, Pablo Trinidad, Antonio-Manuel Gutiérrez-Fernández, and Antonio Ruiz-Cortés, 'Automated analysis of feature models: Quo vadis?', *Computing*, **101**(5), 387–433, (2019).
[15] José A. Galindo, Deepak Dhungana, Rick Rabiser, David Benavides, Goetz Botterweck, and Paul Grünbacher, 'Supporting distributed product configuration by integrating heterogeneous variability modeling approaches', *Information & Software Technology*, **62**, 78–100, (2015).
[16] Somya Garg and Satvik Garg, 'Automated cloud infrastructure, continuous integration and continuous delivery using docker with robust container security', in *2019 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR)*, pp. 467–470. IEEE, (2019).
[17] Michele Guerriero, Martin Garriga, Damian A Tamburri, and Fabio Palomba, 'Adoption, support, and challenges of infrastructure-as-code: Insights from industry', in *2019 IEEE International Conference on Software Maintenance and Evolution (ICSME)*, pp. 580–589. IEEE, (2019).
[18] Dave Hilditch. how to: Setting up a wordpress cluster for huge sites, 2018.
[19] Rasa Inc. Rasa messaging and voice channels, 2019.
[20] Rasa Inc. Rasa pipeline configuration, 2019.
[21] Kyo C Kang, Sholom G Cohen, James A Hess, William E Novak, and A Spencer Peterson, 'Feature-oriented domain analysis (foda) feasibility study', Technical report, DTIC Document, (1990).
[22] A. Kleppe, *Software Language Engineering: Creating Domain-Specific Languages Using Metamodels*, Addison-Wesley Professional, 1 edn., 2008.
[23] Robert B Loatman, Stephen D Post, Chih-King Yang, and John C Hermansen. Natural language understanding system, April 3 1990. US Patent 4,914,590.
[24] Microsoft. Boundaries for your luis model and keys.
[25] Microsoft. Use multiple luis and qna models, 2019.
[26] Jorge L Rodas, David Méndez-Acuña, José A Galindo, David Benavides, and Jesennia Cárdenas, 'Towards testing variability intensive systems using user reviews', in *2015 10th Computing Colombian Conference (10CCC)*, pp. 39–46. IEEE, (2015).
[27] Jorge Rodas-Silva, José A Galindo, Jorge García-Gutiérrez, and David Benavides, 'Resdec: online management tool for implementation components selection in software product lines using recommender systems', in *Proceedings of the 23rd International Systems and Software Product Line Conference-Volume B*, p. 63. ACM, (2019).
[28] Jorge Rodas-Silva, José A Galindo, Jorge García-Gutiérrez, and David Benavides, 'Selection of software product line implementation components using recommender systems: An application to wordpress', *IEEE Access*, (2019).
[29] Jorge Rodas-Silva, José A. Galindo, Jorge García-Gutiérrez, and David Benavides, 'Selection of software product line implementation components using recommender systems: An application to wordpress', *IEEE Access*, **7**, 69226–69245, (2019).
[30] Bhargav Srinivasa-Desikan, *Natural Language Processing and Computational Linguistics: A practical guide to text analysis with Python, Gensim, spaCy, and Keras*, Packt Publishing Ltd, 2018.
[31] Conn Stamford. Gartner forecast worldwide public cloud revenue, 2019.
[32] Pablo Trinidad, Antonio Ruiz Cortés, Joaquin Pena, and David Benavides, 'Mapping feature models onto component models to build dynamic software product lines.', in *SPLC (2)*, pp. 51–56, (2007).

Supporting Feature Model-Based Configuration in Microsoft Excel

Alexander Felfernig[1] and **Viet-Man Le**[1] and **Thi Ngoc Trang Tran**[1]

Abstract. Feature model-based configuration is a process of selecting desired features from a collection of features (called a feature model) which satisfy pre-defined constraints. Configurator development can be performed by different stakeholders with distinct skills and interests, who could also be non-IT domain experts with limited technology understanding and programming experience. Therefore, a simple configuration framework is required to facilitate the participation of non-IT stakeholders in configurator development processes. In this paper, we propose an approach that uses Microsoft Excel to represent configuration knowledge. Our approach shows how to formulate an Excel worksheet from a feature model and also supports the formulation of corrective explanations that help stakeholders resolve configuration inconsistencies.

1 INTRODUCTION

Knowledge-based configuration encompasses all activities related to the configuration of products from predefined components while respecting a set of well-defined constraints that restrict infeasible products [15]. Configuration has been applied in various domains, such as *financial services* [9], *requirements engineering* [16], and *telecommunication* [10]. In configuration systems, knowledge bases often play a crucial role in reflecting the real-world product domain. Many communication iterations between domain experts and knowledge engineers are necessary to develop and maintain a configuration knowledge base. In this context, *feature models* [13] have been recognized as conventional means to facilitate the collaborative model development. Similar to UML-based configuration models [7], feature models provide a graphical representation that improves the understandability of knowledge bases as well as the efficiency of underlying development processes [5]. Moreover, these models help stakeholders decide on relevant features and learn about existing dependencies between features.

Feature model management is usually supported by tools that support feature model creation and analysis. In the current literature, there exist plenty of supporting tools for feature model such as FEATUREIDE [19], S.P.L.O.T. [14], VARISALES [20], and PURE::VARIANTS [18]. Although the advantages of these tools are evident, users might face some difficulties when using them, e.g., unintuitive interfaces [14], programming-skill requirements [19], propositional-logic knowledge [14, 19], and different concepts beyond feature models [20].

Microsoft Excel[2] has been recognized as one of the most widely used spreadsheet applications in modern society. This tool enables non-programmers to perform programming-like tasks in a visual tabular approach. The current literature has witnessed several studies (e.g., [3, 8]) that leverage Excel to tackle configuration problems. The popularity and usability of Excel gave us the idea of using this tool to support configurator development processes of non-IT stakeholders. The contribution of our paper is to propose a novel approach that utilizes Excel worksheets as a complementary means to model configuration knowledge on the basis of feature model concepts. Besides, our approach generates corrective explanations, which are helpful for stakeholders to resolve inconsistency issues.

The remainder of the paper is structured as follows. A brief revisit of feature model-based configuration is presented in *Section 2*. In *Section 3*, we show how to organize an Excel worksheet for a feature model and present a method to create corrective explanations that help stakeholders resolve inconsistencies. Discussions on the pros and cons of the presented approach and an outlook in terms of future work are presented in *Section 4*. Finally, the paper is concluded in *Section 5*.

2 FEATURE MODEL-BASED CONFIGURATION

2.1 Definitions

In feature modeling, feature models represent all possible configurations of a configuration task in terms of *features* and their *interrelationships* [2, 13]. Features are organized hierarchically as a tree structure, where *nodes* represent the features, and *links* represent relationships between nodes. *Features* and *relationships* are equivalent to the *variables* and *constraints* of a CSP[3]-based configuration task [12]. Each variable f_i has a specified domain $d_i = \{true, false\}$. An example feature model of a Smartwatch is depicted in Figure 1. The detailed description of this model is presented in *Subsection 2.2*.

For the following discussions, we introduce the definitions of a *feature model configuration task* and a *feature model configuration* (solution) [6, 12].

Definition 1 (**Feature model configuration task**) A feature configuration task is defined by a triple $(\mathbf{F}, \mathbf{D}, \mathbf{C})$, where $\mathbf{F} = \{f_1, f_2, ..., f_n\}$ is a set of features, $\mathbf{D} = \{dom(f_1), dom(f_2), ..., dom(f_n)\}$ is the set of feature domains, and $\mathbf{C} = CF \cup CR$ is a set of constraints restricting possible configurations, $CF = \{c_1, c_2, ..., c_k\}$ represents a set of feature model constraints, and $CR = \{c_{k+1}, c_{k+2}, ..., c_m\}$ represents a set of user requirements.

Definition 2 (**Feature model configuration**) A feature model configuration **S** for a given feature model configuration task (F, D, C) is an *assignment* of the features $f_i \in F, \forall i \in [1..n]$. S is **valid** if it is *complete* (i.e., each feature in F has a value) and *consistent* (i.e., S fulfills the constraints in C).

[1] TU Graz, Austria, emails: {alexander.felfernig, vietman.le, ttrang}@ist.tugraz.at

[2] www.office.com

[3] CSP - Constraint Satisfaction Problem

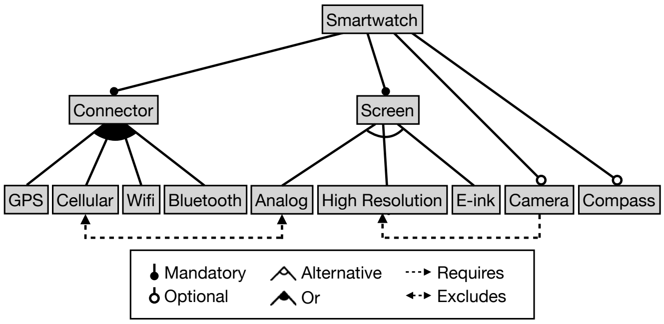


Figure 1: A simplified feature model of the Smartwatch Configuration

Based on the aforementioned definitions, in the next subsection, we introduce feature model concepts[4], which are commonly applied to specify configuration knowledge [12]. Besides, we exemplify a Smartwatch feature model (see Figure 1) to explain the concepts.

2.2 Feature Model Concepts

A feature model (configuration model) consists of two parts: *structural part* and *constraint part*. The former establishes a hierarchical relationship between features. The latter combines additional constraints that represent so-called *cross-tree constraints*.

Structurally, a feature model is a rooted tree, where nodes are features. Each feature is identified by a unique *name*, which exploited to describe possible *states* of a feature (i.e., *"included in"* or *"excluded from"* a specific configuration) [12]. The root of the tree is a so-called *root feature* f_r, which is involved in every configuration ($f_r = true$). Besides, each feature can have other features as its *subfeatures*. The relationship (constraints in CF) between a feature and its subfeatures can be typically classified as follows:

- *Mandatory relationship*: A *mandatory* relationship between two features f_1 and f_2 indicates that f_2 will be included in a configuration *if and only if* f_1 is included in the configuration. For instance, in Figure 1, `Connector` and `Screen` show mandatory relationships with `Smartwatch`. Since `Smartwatch` is the root feature, `Screen` and `Connector` must be included in all configurations.
- *Optional relationship*: An *optional* relationship between two features f_1 and f_2 indicates that if f_1 is included in a configuration, then f_2 *may* or *may not* be included in the configuration. In Figure 1, the relationship between `Smartwatch` and `Camera` is *optional*.
- *Alternative relationship*: An *alternative* relationship between a feature f_p and its subfeatures $C = \{f_1, f_2, ..., f_k\} (C \in F)$ indicates that if f_p is included in a configuration, then *exactly one* $f_c \in C$ must be included in the configuration. For instance, in Figure 1, the relationship between `Screen` and its subfeatures (`Analog`, `High Resolution`, and `E-ink`) is *alternative*.
- *Or relationship*: An *or* relationship between a feature f_p and its subfeatures $C = \{f_1, f_2, ..., f_k\} (C \in F)$ indicates that if f_p is included in a configuration, then *at least one* $f_c \in C$ must be included in the configuration. For instance, in Figure 1, the relationship between `Connector` and its subfeatures (`GPS`, `Cellular`, `Wifi`, and `Bluetooth`) reflects an *or* relationship.

In the *constraint part*, additional constraints are integrated graphically into the model to set cross-hierarchical restrictions for features.

[4] For further model concepts, we refer to [1, 2]

According to [12], the following constraint types are used for the specification of feature models:

- *Requires*: A *requires* constraint between two features (f_1 requires f_2) indicates that if feature f_1 is included in the configuration, then f_2 must also be included. For instance, in Figure 1, if a `Camera` is included in a configuration, then a `High Resolution` screen must be included as well.
- *Excludes*: An *excludes* constraint between two features (f_1 excludes f_2) indicates that both f_1 and f_2 must not be included in the same configuration. For instance, in Figure 1, `Analog` screen must not be combined with `Cellular` connectivity.

The mentioned relationships and constraints can be translated into a static CSP representation using the rules in Table 1.

Table 1: Semantics of feature model concepts in static CSPs (P, C, C_i, A, and B represent individual features).

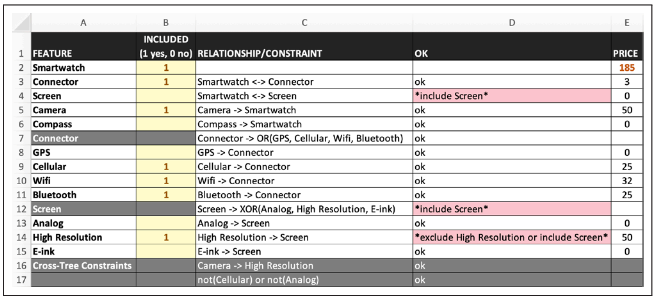
Relationship/Constraint	Semantic in static CSP
$mandatory(P, C)$	$P \leftrightarrow C$
$optional(P, C)$	$C \rightarrow P$
$or(P, C_1, C_2, \ldots, C_n)$	$P \leftrightarrow (C_1 \vee C_2 \vee \ldots \vee C_n)$
$alternative$ $(P, C_1, C_2, \ldots, C_n)$	$(C_1 \leftrightarrow (\neg C_2 \wedge \ldots \wedge \neg C_n \wedge P))$ $\wedge (C_2 \leftrightarrow (\neg C_1 \wedge \neg C_3 \wedge \ldots \wedge \neg C_n \wedge P))$ $\wedge \ldots \wedge (C_n \leftrightarrow (\neg C_1 \wedge \ldots \wedge \neg C_{n-1} \wedge P))$
$requires(A, B)$	$A \rightarrow B$
$excludes(A, B)$	$\neg A \vee \neg B$

3 MODELING A FEATURE MODEL IN AN EXCEL WORKSHEET

In this section, we present our approach to utilize an Excel worksheet to represent feature models, for both structural and constraint parts. An Excel worksheet represents three elements of a feature model: (1) *names*, (2) *states*, and (3) *relationships/constraints*. The *names* represent the structure of a feature model. The *states* store the current state of features in a specific configuration (e.g., "*included*"/"*excluded*"). The *relationships/constraints* are represented in two forms. First, *text-based rules* are exploited to enable stakeholders to understand the relationship between features. Second, *Excel formulae* are used to generate corrective explanations that help stakeholders to resolve configuration inconsistencies. These formulae are translated from the relationships/constraints using logical test functions.

The translation of a feature model to an Excel worksheet can be conducted in the following steps:

- **Step 1**: Put feature names in the **first column**. The features conform to one of the following orders:
  - *Breadth-first order*: The list of feature names is retrieved by traversing level-by-level in the feature model. The process starts with the root feature f_r, then comes to the subfeatures of the root feature before moving to other features at the next level. This process is repeated until the final level is reached.
  - *Depth-first order*: The list of feature names is retrieved by traversing the feature model in a depth-first fashion. The list starts with the root feature f_r, then follows the path of corresponding subfeatures as far as it can go (i.e., from the root feature to its leaf features). The process continues until the entire graph has been traversed.
- **Step 2**: Reserve cells in the **second column** to save the *states* of the features. The cells will be filled in the configuration phase where users manually change the value of the cells to find configurations. The value of the cells is a binary value (`1/0`) or a logical



	A	B	C	D	E
1	FEATURE	INCLUDED (1 yes, 0 no)	RELATIONSHIP/CONSTRAINT	OK	PRICE
2	Smartwatch	1			185
3	Connector	1	Smartwatch <-> Connector	ok	3
4	Screen		Smartwatch <-> Screen	*include Screen*	0
5	Camera	1	Camera -> Smartwatch	ok	50
6	Compass		Compass -> Smartwatch	ok	0
7	Connector		Connector -> OR(GPS, Cellular, Wifi, Bluetooth)	ok	
8	GPS		GPS -> Connector	ok	0
9	Cellular	1	Cellular -> Connector	ok	25
10	Wifi	1	Wifi -> Connector	ok	32
11	Bluetooth	1	Bluetooth -> Connector	ok	25
12	Screen		Screen -> XOR(Analog, High Resolution, E-ink)	*include Screen*	
13	Analog		Analog -> Screen	ok	0
14	High Resolution	1	High Resolution -> Screen	*exclude High Resolution or include Screen*	50
15	E-ink		E-ink -> Screen	ok	0
16	Cross-Tree Constraints		Camera -> High Resolution	ok	
17			not(Cellular) or not(Analog)	ok	

Figure 2: The template of the Excel worksheet for a Smartwatch feature model, where features are listed in breadth-first order.

value (`TRUE/FALSE`), which can be used to represent two states of a feature ("*included*"/"*excluded*").

- **Step 3**: Fill the **third column** with *text-based rules* that represent the relationships/constraints. An arbitrary logic form can be applied to represent relationships/constraints. The text-based rules can be represented according to relationship/constraint type as follows:
  - *Mandatory* and *Optional*: Each relationship is placed in the row of the feature that participates in the relationship (see cells $C3$–$C6$ in Figure 2). The feature is in the left part of the rule (except for a *mandatory* relationship, where the feature is in the right part of the rule).
  - *Alternative* and *Or*: Insert a new row above the subfeatures to store the relationship (see rows 7&12 in Figure 2). For instance, to represent an *alternative* relationship between `Screen` and its subfeatures (`Analog`, `High Resolution`, `E-ink`), we insert a new row above the subfeatures (see row 12 in Figure 2), and in cell $C12$, we add a corresponding rule of the *alternative* relationship. Besides, for each subfeature, add a *requires* constraint to check the consistency between the subfeature and its parentfeature (see cells $C8$–$C11$, $C13$–$C15$ in Figure 2).
  - *Constraints* are located at the end of the relationship list (see constraints in rows 16 & 17 in Figure 2).

- **Step 4**: Convert relationships/constraints into logical test formulae and save in the **fourth column**. The returns of these formulae are used as *textual explanations* that describe the consistency of feature assignments or suggest corrective solutions when inconsistencies occur. Tables 2–6 provide formula templates to generate such explanations according to *six* relationship/constraint types. The templates are derived from truth tables [17], where the last column shows how an explanation can be formulated (e.g., *"ok"* if consistent, *"include feature A"* if inconsistent).

  Besides textual explanations, *visual explanations* are exploited to graphically represent warnings concerning the inconsistency of the corresponding relationships/constraints. In Excel, the warnings can be created using conditional formatting. For instance, in our example, we set a conditional formatting to color a cell in column D with light red if the formula of this cell does not return the string *"ok"* (see cells $D4$, $D12$ & $D14$ in Figure 2).
- **Step 5**: Integrate services such as *pricing* and *capacity* of the product configuration domain into the **remaining columns**. For instance, in our example, the *fifth column* shows the price of each feature and the total price of a configuration (see Figure 2).

Table 2: The truth table and the derived Excel formula template for *Mandatory* relationships.

A	B	$A \leftrightarrow B$	Explanation
0	0	1	ok
0	1	0	include A
1	0	0	include B
1	1	1	ok

Derived Excel formula template:

```
=IF(A_ref=0,IF(B_ref=1,"*include A*","ok"),
IF(B_ref=0,"*include B*","ok"))
```

Table 3: The truth table and the derived Excel formula template for *Optional* relationships and *Requires* constraints.

A	B	$A \rightarrow B$	Explanation
0	0	1	ok
0	1	1	ok
1	0	0	exclude A or include B
1	1	1	ok

Derived Excel formula template:

```
=IF(A_ref=1,IF(B_ref=0,"*exclude A or include B*"
,"ok"),"ok")
```

Table 4: The truth table and the derived Excel formula template for *Or* relationships.

A	B	C	$A \leftrightarrow (B \vee C)$	Explanation
0	0	0	1	ok
0	0	1	0	include A or exclude A's subfeatures
0	1	0	0	include A or exclude A's subfeatures
0	1	1	0	include A or exclude A's subfeatures
1	0	0	0	include B or C
1	0	1	1	ok
1	1	0	1	ok
1	1	1	1	ok

Derived Excel formula template:

```
=IF(B_ref+C_ref=0,IF(A_ref=1,"*include B or C*","ok"),
IF(A_ref=0,"*include A or exclude A's subfeatures*",
"ok"))
```

Table 5: The truth table and the derived Excel formula template for *Alternative* relationships.

A	B	C	$(B \leftrightarrow (\neg C \wedge A)) \wedge$ $(C \leftrightarrow (\neg B \wedge A))$	Explanation
0	0	0	1	ok
0	0	1	0	include A
0	1	0	0	include A
0	1	1	0	include 1 out of B, C
1	0	0	0	include 1 out of B, C
1	0	1	1	ok
1	1	0	1	ok
1	1	1	0	include 1 out of B, C

Derived Excel formula template:

```
=IF(B_ref+C_ref=1,IF(A_ref=0,"*include A*","ok"),
IF(A_ref+B_ref+C_ref=0,"ok","*include 1 out of B, C*"))
```

Table 6: The truth table and the derived Excel formula template for *Excludes* constraints.

A	B	$\neg A \vee \neg B$	Explanation
0	0	1	ok
0	1	1	ok
1	0	1	ok
1	1	0	exclude A or B

Derived Excel formula template:

```
=IF(A_ref=1,IF(B_ref=1,"*exclude A or B*","ok"),"ok")
```

4 DISCUSSION

Our first experiment in the Smartwatch domain has shown that our approach is feasible for modeling configuration knowledge and exploring feature models. In particular, the usage of Excel-spreadsheet interface paradigm helps to maintain the most important benefits of feature models (the *feature hierarchy*) and provide stakeholders with an overview of product variants. Besides, the formulation of corrective explanations enables non-IT stakeholders to resolve configuration inconsistencies in the configuration phase. Thereby, these explanations help to preserve the restrictions for configurations.

Our approach allows stakeholders to represent configuration knowledge as an executable representation in Excel, which is also applicable in other spreadsheet programs such as *Numbers* and *OpenOffice Calc*. The representation is quite straightforward and appropriate for in-complex feature models, and therefore helpful for small/medium companies to overcome challenges concerning configurator implementation and utilization (e.g., high costs or considerable chances of failure [11]). Besides, the popularity and usability of Excel facilitate the participation of non-IT stakeholders in configuration development processes. Consequently, Excel-based configurators can be exploited to reduce efforts and risks related to configuration knowledge acquisition. Moreover, manually selecting or deselecting features in an Excel worksheet to find configurations might provide users with a simulation of how a configuration task is done. Thus, our approach can be applied in further scenarios, such as including customers in open innovation processes and enabling teachers to give students easy hands-on experiences with feature modeling.

Our proposal has two limitations that need to be improved within the scope of future work. First, the translation of relationships/constraints into Excel formulae is an error-prone activity. Thus, the development of an automated tool to translate a specific feature model into an Excel worksheet will be one of our future plans. The second limitation lies in the built-in reasoning engine of Excel (i.e., Excel solver). The Excel solver requires the settings of necessary parameters to find solutions, which could be a bit challenging to end-users. Besides, the Excel solver is able to find only one configuration at once instead of a set of configurations. Therefore, the utilization of constraint-based solving add-ins presented in [3, 4, 8] can be a potential solution to resolve this issue of our approach.

5 CONCLUSION

In this paper, we proposed an approach that leverages the usability of Microsoft Excel to model configuration knowledge. A Smartwatch feature model in an Excel worksheet was presented to illustrate the approach. We provided a guideline on the representation of an arbitrary feature model in an Excel worksheet and introduced a method to generate corrective explanations for consistency resolving. Our approach utilizes spreadsheet programs as simple tools to support the product configuration process of non-IT experts, while overcomes challenges concerning configurator implementation and utilization.

REFERENCES

[1] Don Batory, 'Feature Models, Grammars, and Propositional Formulas', in *International Conference on Software Product Lines*, eds., Henk Obbink and Klaus Pohl, pp. 7–20, Berlin, Heidelberg, (2005). Springer Berlin Heidelberg.

[2] David Benavides, Sergio Segura, and Antonio Ruiz-Cortés, 'Automated Analysis of Feature Models 20 Years Later: A Literature Review', *Information Systems*, **35**(6), 615 – 636, (2010).

[3] Lucas Bordeaux and Youssef Hamadi, 'Solving Configuration Problems in Excel', in *Proceedings of the 2007 AAAI Workshop*, Configuration, pp. 38–40, Menlo Park, California, (2007). The AAAI Press.

[4] Siddharth Chitnis, Madhu Yennamani, and Gopal Gupta, 'ExSched: Solving Constraint Satisfaction Problems with the Spreadsheet Paradigm', in *16th Workshop on Logic-based Methods in Programming Environments (WLPE2006)*, (2007).

[5] Alexander Felfernig, 'Standardized Configuration Knowledge Representations as Technological Foundation for Mass Customization', *IEEE Transactions on Engineering Management*, **54**(1), 41–56, (2007).

[6] Alexander Felfernig, David Benavides, José Galindo, and Florian Reinfrank, 'Towards Anomaly Explanation in Feature Models', in *ConfWS-2013: 15th International Configuration Workshop (2013)*, volume 1128, pp. 117–124, (08 2013).

[7] Alexander Felfernig, Gerhard Friedrich, and Dietmar Jannach, 'UML as Domain Specific Language for the Construction of Knowledge-based Configuration Systems', *International Journal of Software Engineering and Knowledge Engineering*, **10**(04), 449–469, (2000).

[8] Alexander Felfernig, Gerhard Friedrich, Dietmar Jannach, Christian Russ, and Markus Zanker, 'Developing Constraint-Based Applications with Spreadsheets', in *Developments in Applied Artificial Intelligence*, eds., Paul W. H. Chung, Chris Hinde, and Moonis Ali, volume 2718 of *IEA/AIE 2003*, pp. 197–207, Berlin, Heidelberg, (2003). Springer.

[9] Alexander Felfernig, Klaus Isak, Kalman Szabo, and Peter Zachar, 'The VITA Financial Services Sales Support Environment', in *Proceedings of the 19th National Conference on Innovative Applications of Artificial Intelligence - Volume 2*, IAAI'07, pp. 1692–1699. AAAI Press, (2007).

[10] Gerhard Fleishanderl, Gerhard E. Friedrich, Alois Haselbock, Herwig Schreiner, and Markus Stumptner, 'Configuring Large Systems Using Generative Constraint Satisfaction', *IEEE Intelligent Systems and their Applications*, **13**(4), 59–68, (July 1998).

[11] Cipriano Forza and Fabrizio Salvador, *Product Information Management for Mass Customization: Connecting Customer, Front-office and Back-office for Fast and Efficient Customization*, Palgrave Macmillan UK, 11 2006.

[12] Lothar Hotz, Alexander Felfernig, Markus Stumptner, Anna Ryabokon, Claire Bagley, and Katharina Wolter, 'Chapter 6 - Configuration Knowledge Representation and Reasoning', in *Knowledge-Based Configuration*, eds., Alexander Felfernig, Lothar Hotz, Claire Bagley, and Juha Tiihonen, 41 – 72, Morgan Kaufmann, Boston, (2014).

[13] Kyo Kang, Sholom Cohen, James Hess, William Novak, and A. Peterson, 'Feature-Oriented Domain Analysis (FODA) Feasibility Study', Technical Report CMU/SEI-90-TR-021, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, (1990).

[14] Marcilio Mendonca, Moises Branco, and Donald Cowan, 'S.P.L.O.T.: Software Product Lines Online Tools', in *Proceedings of the 24th ACM SIGPLAN Conference Companion on Object Oriented Programming Systems Languages and Applications*, OOPSLA '09, pp. 761–762, New York, NY, USA, (2009). ACM.

[15] Sanjay Mittal and Felix Frayman, 'Towards a Generic Model of Configuraton Tasks', in *Proceedings of the 11th International Joint Conference on Artificial Intelligence - Volume 2*, IJCAI'89, p. 1395–1401, San Francisco, CA, USA, (1989). Morgan Kaufmann Publishers Inc.

[16] Gerald Ninaus, Alexander Felfernig, Martin Stettinger, Stefan Reiterer, Gerhard Leitner, Leopold Weninger, and Walter Schanil, 'INTELLIREQ: Intelligent Techniques for Software Requirements Engineering', in *Proceedings of the Twenty-First European Conference on Artificial Intelligence*, ECAI'14, p. 1161–1166, NLD, (2014). IOS Press.

[17] Kenneth H. Rosen, *Discrete Mathematics and Its Applications*, McGraw-Hill Higher Education, 5th edn., 2002.

[18] Olaf Spinczyk and Danilo Beuche, 'Modeling and building software product lines with eclipse', in *Companion to the 19th Annual ACM SIGPLAN Conference on Object-Oriented Programming Systems, Languages, and Applications*, OOPSLA '04, p. 18–19, (2004).

[19] Thomas Thüm, Christian Kästner, Fabian Benduhn, Jens Meinicke, Gunter Saake, and Thomas Leich, 'FeatureIDE: An Extensible Framework for Feature-Oriented Software Development', *Science of Computer Programming*, **79**, 70–85, (January 2014).

[20] Juha Tiihonen and Andreas Anderson, 'Chapter 26 - VariSales', in *Knowledge-Based Configuration*, eds., Alexander Felfernig, Lothar Hotz, Claire Bagley, and Juha Tiihonen, 309 – 318, Morgan Kaufmann, Boston, (2014).

Production Reconfiguration with ASP

Richard Taupe and **Andreas Falkner**[1]

Abstract. With increasing demand for individualized products, the need for flexible production processes is also increasing. Formalized descriptions of factories (cyber-physical production systems, CPPS) and specifications of products enable configurators to plan the production process on demand, by assigning production operations for concrete products to available machines. We propose an architecture that facilitates dynamic reconfiguration of the production process when deviations from the original plan are necessary. Reconfiguration is triggered by events from the data stream generated by the CPPS. Configurations are computed using a declarative problem encoding in Answer Set Programming (ASP). We report on a proof-of-concept implementation of this architecture in a simulated environment and on first evaluation results.

1 INTRODUCTION

In Cyber-Physical Production Systems (CPPS), highly individualized products are manufactured by adaptive, self-organized technical systems (factories) and components (machines). Most of these systems and their components are reconfigurable, i.e., production plans (schedules) are not known before production is triggered and machine configurations may change over time. Operations that produce parts of the product are dynamically assigned to machines providing the necessary skills to execute these operations properly. Optionally, this assignment shall be optimal w.r.t. a certain key performance indicator (KPI) such as maximal throughput or minimal carbon footprint [1, 15]. The specific problem that is solved by our work is how to change this assignment of operations to machines to maintain optimality or to be able to produce the product at all if conditions change, for example if a machine breaks down, is overloaded, or needs more resources to execute operations than expected.

This paper reports on results from the project DynaCon[2] (Dynamic knowledge-based (re)configuration of cyber-physical systems) [6] where we integrate the scientific results into simulations based on an existing CPPS environment at Siemens. We regard a CPPS as a plant, in which methods of discrete manufacturing are employed to manufacture products. A product is defined by the precise list of its concrete parts (Bill of Materials, BOM) and the set of operations necessary to build it (Bill of Processes, BOP). The CPPS itself is defined as a set of machines (Cyber-Physical Production Units, CPPUs), each of which offers a set of skills to execute operations of a BOP.

To produce all products for a customer order on a CPPS, all operations in the BOP need to be assigned to machines in the CPPS in a way so that production, as a whole, is feasible and efficient. This configuration of the production plan is a reconfiguration problem if the current state of the production and partial execution of an existing production plan are taken into account. In this paper, we describe how answer set programming (ASP) can be used for this task and evaluate its performance based on a somewhat simplified example.

Introductions to ASP for usage in object-oriented and large-scale (re)configuration problems can be found in [10,17,18]. ASP has been employed to solve similar planning tasks, such as (offline) solving of intra-logistics problems [8, 11] or production scheduling [9]. Offline optimization of the production process concurrently with product configuration has already been under investigation by the configuration community [16]. The reconfiguration of production processes is related to issues of assemble-to-order (ATO), make-to-order (MTO), and engineer-to-order (ETO) industrial situations [19] as well as the (semi-)automated generation of manufacturing processes [12]. ANSI/ISA-95 is a relevant standard regarding the exchange of information between a manufacturing system and other business processes of a company, such as enterprise planning [20].

The remainder of this document is organized as follows: In Section 2 we present a running example which includes a product, a factory, and a production process, all of which are configurable. After formally describing the problem, architecture, and interfaces in Section 3, we show the main aspects of implementation with ASP in Section 4. Section 5 reports on experimental results and concludes this paper.

2 EXAMPLES

The production of simple, configurable smoke detectors serves as a motivating example which is based on a real production process. We define the product and its properties, a CPPS where the product family can be produced, and an example production lot based on a concrete customer order, i.e. fully configured, concrete products (BOM and BOP). Furthermore, we describe some scenarios where machines fail and the production plan must be reconfigured, resulting in changed assignments of operations to machines.

2.1 Example Product

A smoke detector comprises a base plate where all functional components are mounted, and a cover. Functional components are the battery, an optical sensor consisting of one or two LEDs and an optical receiver, and optional alarm equipment – visual with a lamp and/or acoustic with a siren. For convenience, a type code uniquely defines each individual configuration by a string of length 8.

Figure 1 specifies all parts of the smoke detector, including their configurable properties (attributes) and constraints among them. Each box represents a part or an assembly of parts and contains in the lower section the sequence of production operations necessary to produce the part or assembly, respectively.

[1] Siemens AG Österreich, Corporate Technology, Vienna, Austria,
email: richard.taupe@siemens.com, andreas.a.falkner@siemens.com

[2] `ainf.aau.at/dynacon`

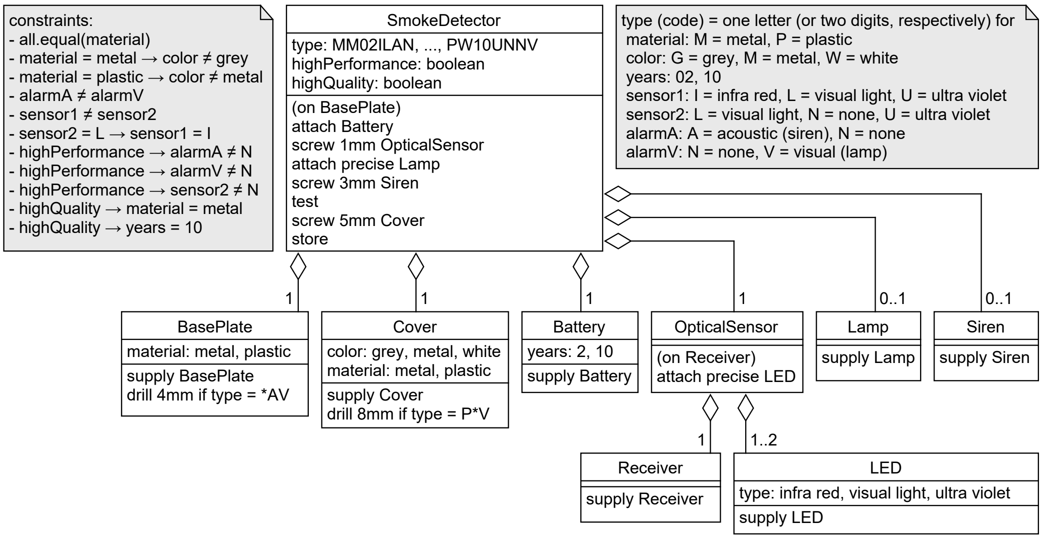


Figure 1. Example product: Smoke detector with properties (attributes) and necessary production operations

2.2 Example Factory

Our plants (CPPS) consist of several hexagons which have a universal robot in the middle and up to six different work places (CPPUs implemented by machines or human workers) around the outside. Each CPPU offers specified skills such as drilling, screwing, etc. with specified parameters (e.g. diameter of the drill hole). To simplify the example, we assume that each CPPU also has a Kanban store where necessary material is supplied regularly by a separate process. The hexagons are connected among themselves and to the environment outside of the CPPS with conveyor belts and/or automated guided vehicles (AGVs).

The right-hand side of Fig. 2 sketches our factory for producing smoke detectors. It is configured to use two hexagons (m1x, m2x) for assembly, one conveyor belt (m30) and an AGV (m40) to transport product parts between the hexagons, and two conveyor belts (m11, m21) to store the completed products. Each hexagon comprises the following CPPUs: A grab robot (mx0) which transports assembled parts between the CPPUs of the hexagon and is also able to execute simple assembly tasks such as attach. A drill robot (mx2) which can drill holes of a diameter of 1 to 8 mm in metal and plastic and execute screwing tasks. An assembly robot (mx3) to undertake precise attachment operations and screwing of small diameters. Two testbeds (mx4, mx5) for ensuring the functionality and quality of the product. CPPUs mx2 and mx3 have Kanban stores which supply the necessary parts.

2.3 Example Production

For production, we assume customer orders for a certain number of configured smoke detectors, e.g. of type MW10INAV where the customer preference is for a product with white color, both alarms, and high quality which necessitates a metal construction with a battery life of 10 years. However the customer does not need a high performance product and thus selected only one sensor.

The left-hand side of Figure 2 shows the BOM and the BOP of such a configured product (p1). All parts (p1.x) and their sub-parts (p1.4.x) are listed in a tree. Together with the concrete parameter settings (attribute values), they form the BOM. In addition, we introduce intermediate assemblies (p1a – p1e) to specify the concrete production steps. The necessary operations (the BOP, comprising all production and transportation steps) are shown in the middle – together with an assignment to machines. We do no show the pre/post relationship of the operations directly, but it can be easily derived from the parts tree. While BOM and BOP, except for some transportation operations, are the same for all products of the same type, the assigned machines can, and will, be different for different concrete products – for example, attaching the battery (p1.3) to the base plate (p1b) could be done by the grab robot and in the other hexagon (m20) as well. The right-hand side shows the configured factory setting (CPPS) with all machines (CPPUs).

2.4 Example Reconfiguration

The assignment of all operations is done by a reconfigurator. For an idle factory where no products are currently being produced, this corresponds to the configuration of the production plan. In general, even that task can be seen as a reconfiguration (of an empty plan). However, the factory will seldom be idle, so that we will typically be always in a real reconfiguration scenario, i.e. adapting an existing, non-empty, partially executed plan.

To qualitatively test the performance of the reconfigurator, we define the following scenario that makes reconfiguration necessary after machine failures such as a machine break-down, skill failure, or shortage of supply:

- Start with four products, each configured and assigned as in Fig. 2
- Event: drill skill of machine m12 fails
- This requires all remaining metal drills to be executed by the only remaining drill robot (m22) and to use the AGV (m40) for transportation to m13 – it may even lead to substantial reconfiguration of other operations, such as attaching batteries of the remaining products also in hexagon m2x to reduce transportation effort
- Event: conveyor belt m30 fails

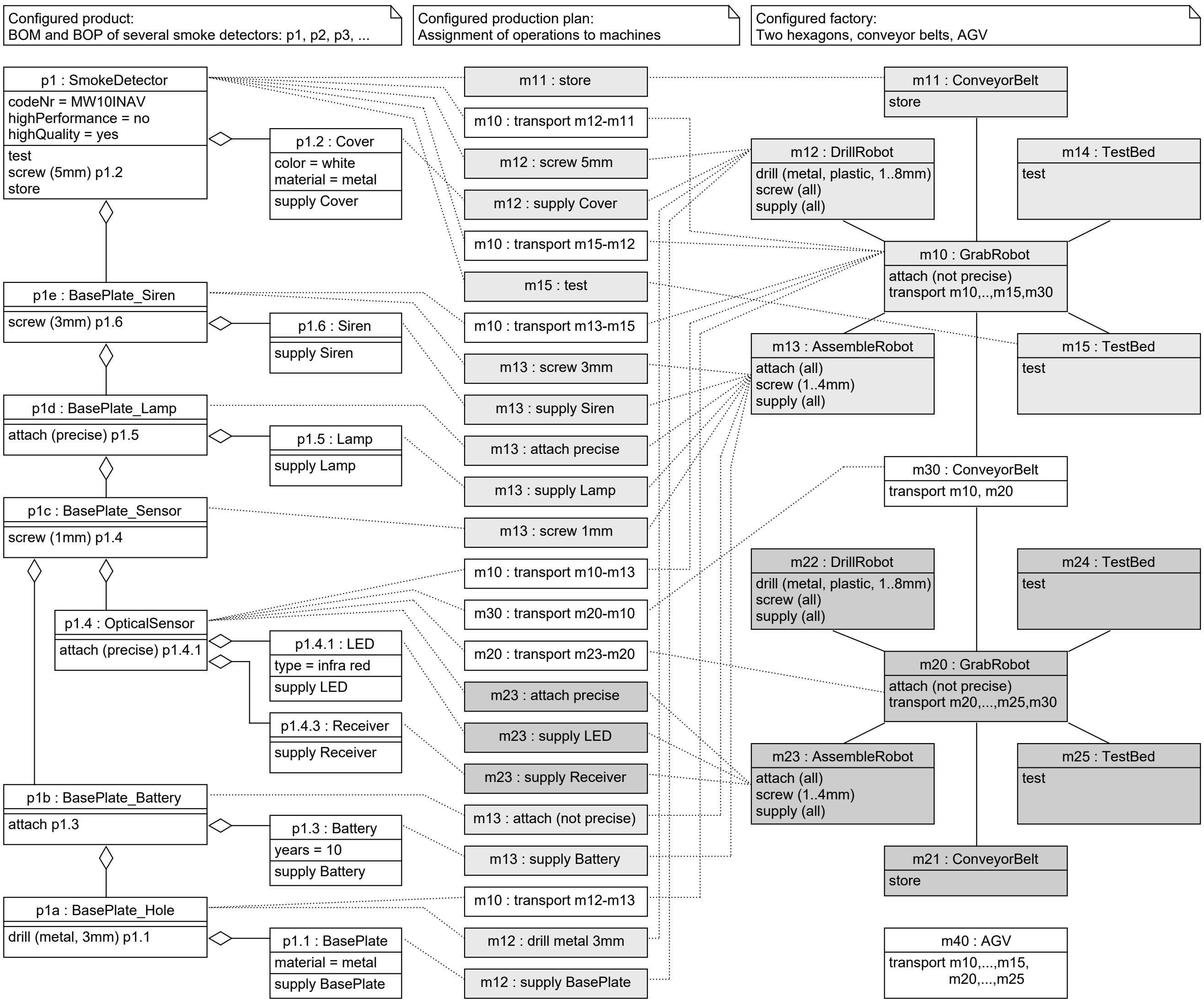


Figure 2. Example for production of a configured smoke detector (high quality and two alarms): BOM and BOP, assignment of operations to machines, and CPPS (two hexagons, conveyor belts, AGV)

- This requires to use the AGV (m40) or to execute all operations on the second hexagon, depending on the summarized efficiency
- Event: second drill robot (m22) fails
- The remaining products cannot be completed – however, the system could recommend to the customers that they relax the requirement for high quality products and switch to plastic for the remaining products (type = PW10INAV) which needs an additional drill for p1.2

3 DYNAMIC RECONFIGURATION

In this section, we present our general approach to dynamic reconfiguration of production processes. We will focus on architectural and theoretical aspects, while implementation-specific aspects are described in Section 4.

3.1 Architecture

Figure 3 shows our proposed architecture. The diagram is divided into three parts: The CPPS is located on the right hand side. Profacto, located in the left, serves as a user interface and facilitates customization of factories, configuration of products, and creation of customer orders. The central part, connecting CPPS and Profacto, is named "DynaCon" after the project in which it is being developed [6].

The two main components of DynaCon are a stream reasoner and the reconfigurator. The task of the stream reasoner is to aggregate a continuous stream of machine status and process information to events such as "Machine M started execution", "Assembly of Part P finished", etc. For more details and an implementation, see [7]. The reconfigurator assigns production operations for products to machines such that a desired KPI is optimized, e.g., throughput of products is maximized or average makespan is minimized. Reconfigura-

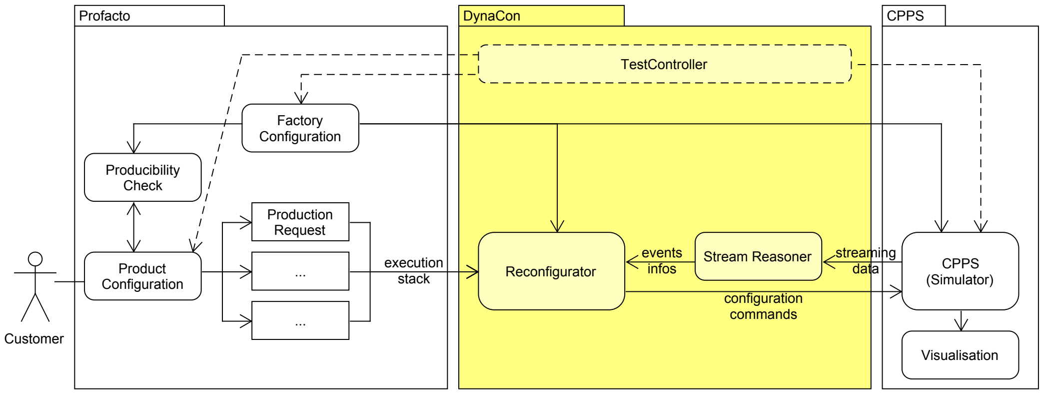


Figure 3. Architecture relating DynaCon to Profacto and the (simulated) CPPS

tion of production is executed for newly ordered products and whenever machine failures require changes to the existing production plan.

The configuration of products and factories is done with Profacto [2, 4, 5] – a prototypical framework by Siemens. It contains a marketplace in an ecosystem for product and factory line engineering. In this marketplace, various stakeholders can model and sell factory equipment, configure factories, and model and configure products. In the current approach, Profacto is responsible for providing:

- a product configurator, which allows the customer to pass orders consisting of configured products to DynaCon,
- the producibility check, a tool that checks whether a specific product can be produced by a given factory,
- and data about a configured factory.

Customer orders are put on the execution stack and passed to DynaCon, which acts here as a production planning system within a Manufacturing Execution System (MES), i.e., it assigns production operations to machines.

An optional Test Controller is responsible for the setup of test data and the simulated CPPS. Processes occurring in the (simulated) CPPS can be visualised by a visualisation component to support comprehension of its inner workings by human users.

3.2 Interface between CPPS and Reconfigurator

The CPPS under consideration can be real or simulated. To be more flexible and to be able to test larger plants, we implemented a simple simulator that provides an interface for communicating with the reconfiguration component (DynaCon). DynaCon can also communicate with a real plant if that plant implements the core functionalities of this interface.

The necessary object-oriented data structures are defined by the following classes:

- A *Plant* consists of a set of *Machine*s.
- A *Machine* is associated to a set of *OfferedSkill*s.
- An *OfferedSkill*, a skill offered by a machine, has one or more KPI (key performance indicator) parameters[3]. While an instance of *OfferedSkill* uniquely represents a skill of a concrete machine, its *name* can refer to a type of skills and be unique only in combination with the *ID* of the machine offering the skill.
- *OfferedTransportSkill* is a subclass, which additionally specifies *source* and *target* of the transport, both of type *Machine*.
- Another subclass is *OfferedSupplySkill*, which introduces preassembled parts into the plant. The *assemblyName* is used to specify the part type and to assign supply operations only to those skills that are able to supply the corresponding assemblies.
- An *Operation* denotes the execution of a skill. It refers to an *OfferedSkill* as well as a set of *predecessors*, which are *Operation*s whose execution must be finished before starting this operation. An operation must have one of two types:
  - A *ProductionOperation* modifies some assemblies. It is associated to a list *inAssemblies* (the assemblies processed by it) and to an *outAssembly* (the *Assembly* produced by it).
  - A *TransportOperation* transports some assembly without modifying it. It simply refers to precisely one *transportedAssembly*.

  While an instance of *Operation* represents an individual operation for a concrete product, its *name* can refer to a type of operations and is recommended to match the *name* of the operation's *requiredSkill*.[4]
- An *Assembly* has a unique *id* and a human-readable *name*, and it is associated to its *producingOperation* (the opposite of a *ProductionOperation*'s *outAssembly*) and its *consumingOperation* (the opposite of a *ProductionOperation*'s *inAssemblies*). Assemblies with the same name are considered to be of the same type.

These data structures are, of course, just an abstraction of the real world. We only deal with data necessary to simulate basic operational behaviour of a CPPS, most importantly the passing of time.

Regarding the nomenclature of assemblies and parts: What we call *Assembly* can either be a supplied or created part – shown as a leaf node in the BOM – or an intermediate assembly or a finished product. Assemblies are produced by *ProductionOperation*s. Small parts or materials like screws, glue, paint, etc. are ignored in this use case.

Table 1 lists the core interface methods to be implemented by a plant. Columns labeled *Parameters* contain names of classes, in-

[3] Currently, we have a predefined set of KPIs (duration, energy consumption, carbon footprint) and implementation can easily be extended with an optimization function which combines several KPIs. A pareto front is not yet supported.

[4] This is not a strict requirement, though. For example, taxonomies of skills are also supported in principle.

Table 1. CPPS Core Interface Methods

Method	Parameters	Description
Submit Operation	*Operation*	submit an operation, i.e., an assignment of a production operation to a machine skill, for execution
Revoke Operation	*Operation*	revoke an operation from a machine where it has previously been submitted for execution

Table 2. Event Listener Interface Methods

Method	Parameters	Description
Execution Finished	*Operation*	the execution of an operation has ended successfully
Machine Failed	*Machine*	a machine has failed
Machine Recovered	*Machine*	a machine has recovered from failure
Skill Failed	*OfferedSkill*	a skill offered by a machine has failed
Skill Recovered	*OfferedSkill*	a skill has recovered from failure

stances of which are passed as arguments. Note that classes are structured further, e.g., *Operation* contains a reference to a machine skill.

The methods listed in Table 2 are implemented by an event listener such that the reconfigurator can receive information from the CPPS (simulator).

3.3 BOM and BOP

While neither Bill of Materials (BOM) nor Bill of Processes (BOP) are explicitly modeled in the data structures, both can easily be derived from them. The BOM of a product can be seen as the set of *Assemblies* that are used in *Operation*s to build that product (i.e. the transitive closure of sub-parts). The BOP is the set of all involved *Operation*s. For each concrete product, BOM and BOP are combined as an *Order*, which contains an association to all *Operation*s of the BOP and all *Assemblies* of the BOM.

A more structured, tree-like view of BOM and BOP can be computed as follows:

Let the name of each class C at the same time denote the set of all instances of this class. Let $c.a$ denote the set of objects associated to instance c of class C via association a.

Any *ProductionOperation* without *outAssembly* removes an end product from the plant, which means that the single element of *inAssemblies* of this operation is an end product. More formally, the set of end products $P \subseteq Assembly$ is defined as

$$P = \bigcup \{o.inAssemblies \mid o \in ProductionOperation, o.outAssembly = \emptyset\}.$$

Conversely, any *ProductionOperation* without *inAssemblies* introduces a sub-assembly to the plant, e.g., by a supplying operation or by additive manufacturing.

Let $A \subseteq Assembly$ be a set of assemblies. Then the set of all assemblies contributing to A is defined inductively as follows:

$$\begin{aligned} A_0(A) &= A \\ A_n(A) &= A_{n-1}(A) \cup \\ &\quad \bigcup \{a.producingOperation.inAssemblies \mid a \in A_{n-1}(A)\} \\ A(A) &= \lim_{n\to\infty} A_n(A) \end{aligned}$$

For an end product $p \in P$, the set of contributing assemblies is $A(p) = A(\{p\})$.

Definition 1. *The* BOM *for an end product $p \in P$ is a graph (V, E), where $V = A(p)$ and $E = \{(super, sub) \mid super, sub \in V$ and $\exists o \in ProductionOperation$ s.t. $sub \in o.inAssemblies, super \in o.outAssembly\}$.*

Definition 2. *The* BOP *for an end product $p \in P$ is a graph (V, E), where $V = \{a.producingOperation \mid a \in A(p)\} \cup \{p.consumingOperation\}$ and $(pre, post) \in E \iff pre \in V$, $post \in V, (pre \in post.predecessors$ or $(pre.outAssembly \neq \emptyset$ and $pre.outAssembly \subseteq post.inAssemblies))$.*

Definition 3. *An* Order *is a triple (p, o, a), where $p \in P$ is an end product, $o = \{a.producingOperation \mid a \in A(p)\} \cup \{p.consumingOperation\}$, and $a = A(p)$.*

To be consistent with the associations to *Order*, for each $p \in P$ it must hold that $p.order.product = p$, that $p.order.assemblies = A(p)$ and that $p.order.operations \supseteq \{a.producingOperation \mid a \in A(p)\}$ and for all $o \in p.order.operations \setminus \{a.producingOperation \mid a \in A(p)\}$ it holds that either $o \in TransportOperation$ and $o.transportedAssembly \in A(p)$ or $o \in ProductionOperation$ and $o.inAssemblies \subseteq A(p)$.

In the example of Fig. 2, the BOM of p1 is a graph with vertices p1, p1e, p1.2, etc., and edges (p1, p1e), (p1, p1.2), etc. The vertices of the BOP graph correspond to the operations listed in the bottom part of each box representing an assembly in Fig. 2, and the edges of the BOP graph correspond to the part-subpart and precedence relationships between those operations. For example, there is one edge going from the operation “supply Cover” in p1.2 to the operation “screw (5mm) p1.2” in p1.

3.4 Configuration and Reconfiguration

Reconfiguration of a production plan does not need to consider the existing assignments of non-completed operations to machines because there is no penalty for changing the assignment. In contrast, it is important to optimize only for the new situation, ignoring already completed operations. Therefore reconfiguration can be implemented as configuration with the following inputs: a set of orders (comprising BOM and BOP of a configured product, together with the completion state of the operations) and the factory state (machines, skills, availability). Output is a feasible production plan for the non-completed operations which is optimal with respect to some predefined KPI.

Definition 4. *A* production plan *is a set of pairs* (o, m)*, where* $o \in$ *Operation and* $m \in$ *Machine. It comprises all non-completed operations of all orders.*

Definition 5. Reconfiguration *is the task to assign a machine* $m \in$ *Machine to each non-completed operation* $o \in \bigcup\{p.order.operations \mid p \in P\}$ *so that a given KPI is optimal. The assignment must use only the available skills of the available machines.*

The reconfigurator is triggered by placing a new order (Submit Operation of Table 1) and by events from the CPPS (Machine/Skill Failed/Recovered of Table 2). Whenever on operation is completed (Execution Finished), it is removed from the production plan.

4 IMPLEMENTATION

Our reconfiguration component is based on an ASP encoding. Answer Set Programming (ASP) is a logic-based approach to declarative programming [10, 14, 17, 18]. We assume familiarity with the language of ASP and describe parts of our encoding in this section. A preliminary encoding for our reconfiguration problem is available on our website.[5]

4.1 Predicates

The major portion of data needed to be supplied to the reconfigurator consists of an instantiation of the data structures described in Section 3.2. Classes and associations are represented by facts over the predicates listed in Table 3 (for static information) and Table 4 (for dynamic information).

The input for the reconfigurator contains facts over these predicates describing the current plant layout and existing production operations. Note that the association **operation_requiredSkill** only exists for operations that have already been planned. Operations that are not yet assigned to a skill are part of a new order and the task of the reconfigurator is to assign them to offered skills (i.e., to machines). To determine to which offered skill an operation can be assigned, the reconfigurator must either get the results of skill matching as an input or do skill matching itself [3]. Here, we use a simplified version of skill matching that only checks for *OfferedSkill*s whose *name* equals that of the *Operation*.[6]

Additional dynamic information is represented by prediactes in Table 5: The reconfigurator has to know about the availability of machines, skills, and supplied assemblies. Machines and skills are unavailable when they have failed (until they are restored), and offered supply skills can additionally be unavailable when there is insufficient supply (until supply is replenished). In the input for the reconfigurator, only unavailability is represented by facts over the unary predicates **unavailable_machine** and **unavailable_skill**, where each fact contains the corresponding ID as a term.

To reason about the workload of machines and skills, the reconfigurator also needs to know which operations, that are already assigned to skills, are currently being executed or waiting to be executed. Therefore, the additional input predicate **operation_status/2** associates an operation id with one of three states: waiting, executing, finished.[7]

Changes in the CPPS may cause existing operations to end up being assigned to unavailable skills or machines. In this case, the reconfigurator must be executed. Its output is represented by the predicate **assign_operation_offeredSkill/2** which instruct the CPPS to assign an operation for a product part to an offeredSkill of a machine. The reconfigurator creates an atom for each non-completed operation, identifying operations and offeredSkills by their unique IDs.

4.2 Encoding

Our encoding (Listing 1) is based on the well-known generate-define-test approach [13]. The "generate" part (Lines 1 to 4) of the encoding contains a choice rule that unfolds the search space consisting of possible operation-to-skill assignments (Line 2) and two rules defining which operations are assignable: namely, those that are not yet assigned and those currently assigned to an unavailable skill.

The "test" part (Lines 9 to 22), which uses a predicate defined in the "define" part (Lines 6 to 7), contains the constraints restricting the solution space (only the most important constraints are shown in Listing 1): Each operation must be assigned to exactly one skill which can execute this operation, and there are additional restrictions for the assignment of transport and supply operations.

Line 25 contains a simple optimization statement that minimizes the overall carbon footprint by summing up the known carbon footprints from all used skills. Makespan optimization could also be implemented, but this would involve planning at which time point each operation is executed.

5 PRELIMINARY RESULTS AND CONCLUSIONS

We have proposed an architectural solution that facilitates dynamic reconfiguration of production processes. In this architecture, a central reconfigurator is triggered when new production orders are placed or when there are severe changes to parameters of the plant, e.g. failures of machine skills or whole machines.

Our proof-of-concept implementation of the reconfigurator is based on Answer Set Programming (ASP) and is embedded into a simulated CPPS environment. Architecture and implementation have been complemented by a systematic definition of an interface between CPPS and reconfigurator based on object-oriented data structures, by a formalisation of BOM and BOP, and by a definition of the reconfiguration task.

Our centralized planning component allows to compute a production plan that is globally optimal w.r.t. a given KPI (e.g., carbon footprint), assuming that KPI values for operations executed by machines are known or can be estimated with sufficient certainty, which is usually the case. Changes to the plan are computed and executed as soon as necessary.

We conducted qualitative and quantitative experiments based on the example in Section 2 and found that the approach is adequate for modelling use cases like ours. The second column of Table 6 shows that the models stay small, comprising 1,299 facts for the plant and 141 facts for each ordered product, configured as in Figure 2. This supports the common assumption that knowledge-based approaches such as ASP allow for cost-efficient specifications of configuration problems that can easily be changed or extended.

[5] `https://git-ainf.aau.at/DynaCon/website/-/wikis/Dissemination-activities#supplementary-material`

[6] Another possibility would be to do skill matching outside of the reconfiguration component. Then the reconfigurator would receive all possible matchings (the sets of offered skills that can execute each production step) as input and not have to do skill matching at all.

[7] An alternative to this would be to provide the reconfigurator with an aggregation of this data, e.g., the number of operations waiting per machine.

Table 3. ASP predicates for representing the static part of instantiations of classes and associations

Predicate name	Arity	Terms
plant	2	id, name
machine	2	id, name
offeredSkill	3	id, name, durationInSeconds
offeredTransportSkill	1	id
offeredSupplySkill	2	id, assemblyName
offeredProductionSkill	1	id
machine_plant	2	id of machine, id of plant
offeredSkill_machine	2	id of offeredSkill, id of machine
offeredTransportSkill_source	2	id of offeredTransportSkill, id of source machine
offeredTransportSkill_target	2	id of offeredTransportSkill, id of target machine

Table 4. ASP predicates for representing the dynamic part of instantiations of classes and associations

Predicate name	Arity	Terms
operation	2	id, name
transportOperation	1	id
productionOperation	1	id
assembly	2	id, name
order	1	id
operation_requiredSkill	2	id of operation, id of offeredSkill
operation_predecessor	2	id of operation, id of predecessor operation
transportOperation_transportedAssembly	2	id of transportOperation, id of transported assembly
productionOperation_inAssembly	2	id of productionOperation, id of in assembly
productionOperation_outAssembly	2	id of productionOperation, id of out assembly
assembly_order	2	id of assembly, id of order
operation_order	2	id of operation, id of order

Table 5. Additional ASP predicates for representing dynamic information

Predicate name	Arity	Terms
unavailable_machine	1	id of machine
unavailable_skill	1	id of skill
operation_status	2	id of operation, status (waiting, executing, or finished)

```
% GENERATE:
{ assign_operation_offeredSkill(O,S) : offeredSkill_id(S), not unavailable_skill(S) } :- assignable_operation(O).
assignable_operation(O) :- operation_id(O), not operation_preassigned(O).
assignable_operation(O) :- operation_status(O,waiting), operation_requiredSkill(O,S), unavailable_skill(S).

% DEFINE:
operation_assigned(O) :- assign_operation_offeredSkill(O,S).

% TEST:

% each operation that is not yet assigned to a skill must be assigned to exactly one skill:
:- operation_id(O), not operation_preassigned(O), not operation_assigned(O).
:- assign_operation_offeredSkill(O,S1), assign_operation_offeredSkill(O,S2), S1 < S2.

% each operation must be assigned to a skill whose name matches the operation's skill name (skill matching):
:- assign_operation_offeredSkill(O,S), operation(O,OName), offeredSkill_name(S,SName), OName != SName.

% each transport operation must be assigned to a skill that matches the two machines between which the assembly has
    to be transported:
:- transportOperation_transportedAssembly(TO,Assembly), productionOperation_outAssembly(SourcePO,Assembly),
    productionOperation_inAssembly(TargetPO,Assembly), operation_assigned_to_machine(SourcePO,SourceMachine),
    operation_assigned_to_machine(TargetPO,TargetMachine), not
    transportOperation_machines(TO,SourceMachine,TargetMachine).

% a supply operation must be assigned to a supply skill with matching assemblyName:
:- assign_operation_offeredSkill(O,S), offeredSupplySkill(S,SAName), productionOperation_outAssembly(O,A),
    assembly(A,AName), SAName != AName.

% OPTIMIZATION:
#minimize { C,O : offeredSkill_carbon(S,C), assign_operation_offeredSkill(O,S) }.
```

Listing 1. Most relevant parts of the reconfigurator ASP encoding

Table 6. Preliminary evaluation results

Concrete products (#)	Facts (#)	Constraints (#)	Execution (sec)	Solve (sec)	Optimize (sec)	Models (#)
1	1,440	192,659	0.44	0.02	0.20	11
2	1,581	385,318	1.41	0.03	3.48	21
4	1,863	770,636	2.70	0.06	>3,600.00	>40
8	2,427	1,541,272	5.47	0.12	-	-
16	3,555	3,082,544	15.86	0.25	-	-
32	5,881	6,165,088	34.51	0.48	-	-

For qualitative evaluation, we simulated the scenarios of Section 2.4 with a customer order of four products of type MW10INAV. The returned production plans were feasible for the first configuration as well as for the subsequent reconfigurations after machine failures. Response time (<3 sec) was suitable for interactive use, even when considering unsatisfiability in the last scenario. Optimization delivered near optimal results after cut-off at 15 sec.

For quantitative evaluation, we use customer orders with doubling sizes. Problem sizes increase proportionally, and total execution time, which includes grounding, grows only slightly faster (see fourth column of Table 6). The times were measured with clingo 5.3.0 running on a 64-bit CPU with 2.40 GHz and 16 GB RAM. Solving alone takes less than a second for orders comprising up to forty configured products. Optimization, however, is not performing well with our straight-forward approach: already for four smoke detectors it runs into a time-out after one hour while the 40th model is checked for optimality The reason is the enormous number of alternatives: several billions even for only one product.

In future, we will try to generalize the results by conducting more experiments with other realistic examples and improve the encoding to achieve better runtime performance, especially for optimization.

ACKNOWLEDGEMENTS

This work has been conducted in the scope of the research project DynaCon (FFG-PNr.: 861263), which is funded by the Austrian Federal Ministry of Transport, Innovation and Technology (BMVIT) under the program “ICT of the Future” between 2017 and 2020. See https://iktderzukunft.at/en/ for more information.

REFERENCES

[1] Birte Caesar, ‘Engineering support for variability modeling for contextsensitivereconfiguration of collaborative manufacturing systems’, in *VaMoS*, pp. 14:1–14:2. ACM, (2020).

[2] Deepak Dhungana, Andreas A. Falkner, Alois Haselb¨ock, and Herwig Schreiner, ‘Smart factory product lines: a configuration perspective on smart production ecosystems’, in *SPLC*, pp. 201–210. ACM, (2015).

[3] Deepak Dhungana, Andreas A. Falkner, Alois Haselb¨ock, and Richard Taupe, ‘Enabling integrated product and factory configuration in smart production ecosystems’, in *SEAA*, pp. 266–273. IEEE, (2017).

[4] Deepak Dhungana, Alois Haselb¨ock, and Richard Taupe, ‘A marketplace for smart production ecosystems’*, in Customization 4.0*, pp. 103–123. Springer, (2018).

[5] Deepak Dhungana, Alois Haselb¨ock, and Stefan Wallner, ‘Generation of multi-factory production plans: Enabling collaborative lot-size-one production’, in *Euromicro SEAA*, (2020). To appear.

[6] Thomas Eiter, Gerhard Friedrich, Richard Taupe, and Antonius Weinzierl, ‘Lazy grounding for dynamic configuration - efficient largescale (re)configuration of cyber-physical systems with ASP’, *KI*, 32(2-3), 197–198, (2018).

[7] Thomas Eiter, Paul Ogris, and Konstantin Schekotihin, ‘A distributed approach to LARS stream reasoning (system paper)’, *Theory Pract. Log. Program.*, 19(5-6), 974–989, (2019).

[8] Felicitas Fabricius, Marco De Bortoli, Maximilian Selmair, Michael Reip, Gerald Steinbauer, and Martin Gebser, ‘Towards ASP-based scheduling for industrial transport vehicles’, in *Joint Austrian Computer Vision and Robotics Workshop,* (2020). To appear.

[9] Gerhard Friedrich, Melanie Fr¨uhst¨uck, Vera Mersheeva, Anna Ryabokon, Maria Sander, Andreas Starzacher, and Erich Teppan, ‘Representing production scheduling with constraint answer set programming’, in OR, pp. 159–165. *Springer*, (2014).

[10] Gerhard Friedrich, Anna Ryabokon, Andreas A. Falkner, Alois Haselb¨ock, Gottfried Schenner, and Herwig Schreiner, ‘(re)configuration using answer set programming’, in Configuration Workshop, volume 755 of *CEUR Workshop Proceedings*. CEUR-WS.org, (2011).

[11] Martin Gebser, Philipp Obermeier, Thomas Otto, Torsten Schaub, Orkunt Sabuncu, Van Nguyen, and Tran Cao Son, ‘Experimenting with robotic intra-logistics domains’, *Theory Pract. Log. Program.*, 18(3-4), 502–519, (2018).

[12] Giray Havur, Alois Haselb¨ock, and Cristina Cabanillas, ‘Automated multi-perspective process generation in the manufacturing domain’, in *Business Process Management Workshops, volume 362 of Lecture Notes in Business Information Processing*, pp. 81–92. Springer, (2019).

[13] Vladimir Lifschitz, ‘Answer set programming and plan generation’, *Artif. Intell.*, 138(1-2), 39–54, (2002).

[14] Vladimir Lifschitz, *Answer Set Programming*, Springer, 2019.

[15] Hermann Meissner and Jan C. Aurich, ‘Implications of cyber-physical production systems on integrated process planning and scheduling’, *Procedia Manufacturing*, 28, 167 – 173, (2019). CARV2018.

[16] Paul Pitiot, Michel Aldanondo, ´ Elise Vareilles, and Paul Gaborit, ‘Concurrent configuration and planning problems: Some optimization experimental results’, in *Configuration Workshop*, volume 958 of CEUR Workshop Proceedings, pp. 41–46. CEUR-WS.org, (2012).

[17] Gottfried Schenner, Andreas A. Falkner, Anna Ryabokon, and Gerhard Friedrich, ‘Solving object-oriented configuration scenarios with ASP’, in *Configuration Workshop*, volume 1128 of CEUR Workshop Proceedings, pp. 55–62. CEUR-WS.org, (2013).

[18] Gottfried Schenner and Richard Taupe, ‘Techniques for solving largescale product configuration problems with ASP’, in *19th International Configuration Workshop*, (2017).

[19] Abdourahim Sylla, E Vareilles, Michel Aldanondo, Thierry Coudert, Laurent Geneste, and Paul Pitiot, ‘Concurrent configuration of product and process: moving towards eto and dealing with uncertainties’, in *18th International Configuration Workshop*, (2016).

[20] Hakki Ozgur Unver, ‘An ISA-95-based manufacturing intelligence systemin support of lean initiatives’ *The International Journal of Advanced Manufacturing Technology,* 65(5-8), 853–866, (2013).

A development approach towards user-centered front-ends for knowledge-based engineering configurators: a study within planning of robot-based automation solutions

Eike Schäffer[1] and **Sara Shafiee** and **Tobias Frühwald** and **Jörg Franke**

Abstract. Configurators are well-established strong researched expert systems due to the high popularity and gained benefits. Nevertheless, the aspects of user-centered design are rarely researched and applied during the development of configurator front-ends. However, when addressing a mass consumer market or non-experts end-users, the front-end represents the required crucial knowledge that bridges the end-user's current level of knowledge to the experts' knowledge. In order to push the application of knowledge-based engineering configurators for a mass market in SMEs, a user-centered configurator frond-end for the concept planning of robot-based automation systems is being developed within the platform project ROBOTOP. In this paper, the design of an architectural user-centered layer and a ten step development approach of user-centered front-ends are presented.

1 INTRODUCTION

The automation of production processes as well as digitalization in engineering within the framework of Industry 4.0 promises automated workflows, higher speeds and lower costs in both production and engineering. Nevertheless, system integration causes major costs of automation solutions, which makes many solutions uneconomical for small and medium-sized companies (SME). Continuous engineering with focus on simulation is mentioned in this context to reduce costs and time as well as to improve quality. [1]

1.1 Motivation and aim

Despite intensive efforts, pure simulation approaches provide limited benefits without the inclusion of expert knowledge as well structured approaches to prepare knowledge. Nevertheless, people and project specific solutions are common practice and often prevent scalability to other projects and sustainable cost reduction [2] [3]. Especially for effective and scalable knowledge reuse, knowledge-based configurators and associated methods have established themselves as a well-proven approach [4] [5].

However, the use of knowledge-based configurators is primarily established in expert and business-to-business (B2B) communities and scaling on the mass customized consumer market is still limited [6]. Reasons are routed in the development of front-ends of configurators, which are mostly not user-centered and are given low priority and development capacity. As a result, such systems are often too complex for targeted non-expert users and in some cases, users have to answer questions during configuration processes that include partly the configuration solution, rather than user-requirements. According to Steve Jobs, design is how something works [7]. A user-oriented design can therefore not be imposed on a finished solution, but should rather be an integral part of the development process [7].

The aim of the paper is therefore to show how user-centered approaches for the B2B can be used to make complex configurators accessible to a wider audience who are not business experts or non-expert customers. This is addressed by a development approach towards a user-centered engineering configurator using as an example the concept planning of robot-based automation solutions.

1.2 Structure of the paper

The paper is divided into five further sections: Firstly, the theoretical background of the scientific idea is explained, showing both the status and the challenges in the development of expert tools focusing on knowledge-based engineering configurators (KBEC), as well as benefits and solutions from the field of user-centered development. Secondly, the research method is briefly presented. Thirdly, the development approach towards user-centered front-ends for KBEC is introduced, which is divided into an architectural user-centered layer integration and the development approach. Fourthly, the use case is introduced, the implementation of the concept and method is demonstrated and validated by means of user-testing and expert discussions. Fifthly, a summary of the main findings and an outlook on further development opportunities is given.

2 THEORETICAL BACKGROUND

The following sub-sections summarize the relevant theoretical background of expert systems respectively KBEC and user-centered development, which have been mostly considered as independent fields of research. Due to the high complexity and variance of engineering tasks and the non-expert target group, a need-based interface [8] [9] is targeted instead of a parameter-based interface. The highly relevant knowledge about the voice of the customer [9] (SME) and the system integrator is addressed within the front-end.

[1] Institute for Factory Automation and Production Systems (FAPS), Friedrich-Alexander-University Erlangen-Nuremberg (FAU), Germany, contact: linkedin.com/in/eike-sch%C3%A4ffer/

2.1 Situation of developing expert tools

The research of expert tools such as knowledge-based configurators (KC) is mainly focused on the modeling of expert knowledge and its implementation and integration into existing IT landscapes [5] [6] [10] [11]. The development of KC is primarily project-based, less focusing on sustainable cost-reduction for development and on cross-project knowledge elements reuse [12].

One reason for the expensive developments is that most configurators are single project or product configurators [12] and are developed based on a given fixed product structure and less for dynamic, general planning tasks based on generalized knowledge. As a result, the user domain is often too small to justify developments that are more extensive.

One solution to reduce the development costs for configurators is a cross-project reuse of partial results of configuration projects. Schäffer et al. [13] present a collaborative and work-sharing development process for knowledge-based engineering configurators (KBEC) considering a cross-project reuse using the example of the collaborative configuration model development for engineering configurators based on eight-step model.

2.2 Complication of KBEC

Another limitation to the potential mass utilization of expert tools focusing on KBEC is the requirement of professional knowledge which makes them difficult to understand and use by non-experts [9] [14] [15]. Consequently, the potential target audience is limited which in quintessence also limits the available budget for development. Since no knowledge elements can be imported from previous projects for technical or contractual reasons [16], they have to develop the configurator from scratch each time. Therefore, each part of the configurator can only be developed to a limited extent within the economic limits, and learning effects cannot be accumulated over time. Consequently, only one front-end view beneath the available possibilities is developed for a specific configurator project which also limits the potential target audience [9].

Apart from budgetary limitations of KBEC, there is often no awareness of customer needs [9] with respect to user-centered design (UCD) approaches and their benefits within configuration projects; especially as front-end development is frequently presented as a secondary sideline activity. Therefore, UCD is briefly introduced. The goal is to make existing and new configurators accessible to a wider audience.

2.3 User-centered design

User-centered design (UCD) offers numerous advantages and frequently mentioned benefits are: Increased customer satisfaction (33 %), improved usability (20 %), increased revenue (19 %) and reduced customer support costs for the software (18 %) [17] .

Usability is the level to which a solution can be used by a specific user group in a concrete user context to achieve a defined goal. The degree of usability fulfillment can be divided into three levels: (1) effectiveness, (2) efficiency and (3) user satisfaction, see Figure 1. The first one focuses on precision and the degree of task fulfillment. The second one highlights the users' necessary competence, the required time for usage and therefore the cost-efficiency e.g. [18] for task fulfillment. The third stage focuses on the system level of user acceptance and user friendly interface, and therefore on the satisfaction of the users [6]. [19] [20]

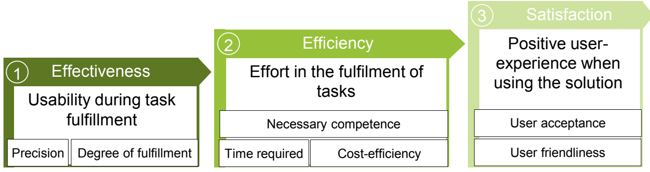


Figure 1. The three stages of the usability; from effectiveness to user satisfaction [19] [20]

One of the central lessons from the UCD according to Nielsen [21] is, that the developer is not the typical user and therefore it is essential to involve targeted user groups in the development process. In addition, the Standish Group estimates that 50 percent of all software functions remain unused or are rarely used [22]. Complexity is not only a "creator"-problem, but also a "user and consumer"-problem. By involving users in the development process, there is enormous potential for complexity reduction, both for the development team and the users. The UCD approach also fits well within modern microservice [23] and micro-frontend architectures [24] [25].

It is therefore important to observe end-users and to involve them in the development process within frequent and regular feedback cycles, e.g. in the form of a mock-up, i.e. a scale model for presenting overall impression, or a click-prototype, i.e. a partially interactive demonstrator of a user interface that simulates certain interactions. Concept decisions should be made based on statistical data. Therefore, concepts and prototypes are used as tools for the validation of hypotheses of the development team [20] [21] [26].

3 RESEARCH METHODOLOGY

The research methodology is a front-end study approach which can be summarized as an architecture and development approach empowered by proof of concept test-design (user-centered front-end of KBEC). The study is based on a single use case which focuses on the concept planning of robot-based automation solutions (ROBOTOP) [27]. The study addresses a typical problem of system integrators distributing automation solutions to SMEs, analysing the automation potential, informing SMEs, finding a first conceptual solution for the contract and keeping the sales costs low at the same time [3]. In particular, since SMEs which acquire an initial automation solution also have a high demand for knowledge. To validate this approach user-testing as well as expert interviews are performed. The final prototype was presented to and discussed with experts from the field of sales of robot-based automation solutions and configuration development.

4 APPROACH TOWARDS USER-CENTERED FRONT-ENDS FOR KBEC

The fourth chapter is divided into two parts, the general architectural classification of user-centered configurators and the development concept of user-centered front-ends for configurators.

4.1 Architectural user-centered layer

Configurators established in the consumer and B2B market focus on the advisory salesperson, which manually translates the logical structure to the end-user. Therefore, a top layer should be introduced as a surface of a pure one-to-one parameter input configurator, which translates the vague requirements of the customer into concrete parameters and solutions similar to advice from an expert salesperson. This layer serves metaphorically as a digital bridge between the non-expert-user-groups and the expert knowledge integrated inside the logical configuration model, see Figure 2.

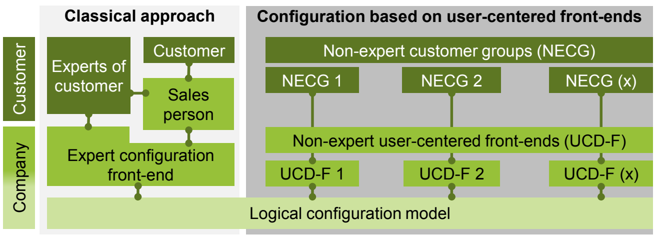


Figure 2. Comparison between classic configuration architecture with user-centered configuration architecture

In order to integrate the specific understanding, a user-centered front-end development approach for knowledge-based configurators will be introduced and presented for one specific user group within concept planning of robot-based automation solutions for SMEs.

4.2 Development approach of user-centered front-ends

The development concept is divided into three main phases consisting of ten sub-steps. The range of functions for first user-centered KBEC was deliberately limited because technical complexity would otherwise be too extensive. Therefore, the concept of a minimal viable prototype was applied. In addition, an important concept for the user-centered KBEC is that the users are only asked questions that can be answered within their knowledge domain concerning product and production process (user-requirements). The general classification into product, process and resource is known as PPR-model, introduced by Steinwasser [28]. Questions concerning the technical design of the resource (configuration solution) especially the automation solution will not be raised. Therefore, the "Best Practice"-based-configuration-approach is developed whereby a "Best Practice" is a successful realized automation solution from past engineering projects needed for initial starting points [9] or default configuration parameters [29] [30]. Parallelly, the configuration model was developed and described by Schäffer et. al. [13]. For the sake of simplicity, the concept is based on the assumption, that a logical configuration model is simultaneously developed and the Best Practices are already available. The approach is based on the general user-centered approach and has been extended and concretized within the scope of the configuration project [31]:

I) Preparation:

1. Defining the strategic objective of the configuration project based on major economic or social challenges.
2. Selecting the business case and target group, whereby the size of the target group as well as economic market demands are taken into account.
3. Analyzing the current situation and processes of the selected business case and target group based on literature review and expert interviews.

II) Initialization:

4. Concept workshop to present of the current situation and discuss with various stakeholders from management, sales, engineering, marketing and customers. Using flipcharts, whiteboards and a moderator's toolbox, first concepts can be initialized.
5. First concept design sequence is created using a graphical program or presentation tools e.g. PowerPoint.
6. Qualitative feedback, based on the individuals of the first concept workshop to improve the concept design.
7. Click-prototype based on the improved concept design via rapid prototyping tool e.g. Axure (www.axure.com) or balsamiq (www.balsamiq.com) in combination with a CAD environment e.g. Inventor (www.autodesk.de/products/inventor) to create the 3D-scenes of the Best Practices.

III) User-centered optimization:

8. User-testing and optimization based on the click-prototype, with iterative cyclic optimizations.
9. The first milestone achievement when the click-prototype contains all essential functions and contradictory user feedback is given. Then the first prototype can be released for the platform or live system to be integrated.
10. Multi-user-testing and optimization within online platform or live system and also further development of click-prototypes for the purpose of easy testing of new ideas.

5 REALIZATION AND VALIDATION

As a result of applying the approach, a front-end for the rough concept planning of robot-based automation solutions was created and the process was validated. For a better understanding, the developed use case and goal is firstly described (preparation). The strategic objective as well as the business case and target group selection (see chapter 5.1) was based on the PAiCE project call of the German Federal Ministry of Economic Affairs and Energy (BMWi) [32]. Secondly, the results are presented, as well as the main findings resulting from the user-centered development process. Finally, the development process and the results are evaluated based on user-tests and expert discussions.

5.1 Use case: concept planning of robotbased automation solutions

The target group of the user-centered configurator example are small and medium-sized manufacturing companies (SME) that have expertise in a specific product and its manufacturing process, but not within automation solutions. Typically, manual production with increasing production quantities are the starting point for further automation. In order to realize a turnkey automation solution, SMEs usually contract a system integrator. Nevertheless, many SMEs are non-lucrative customers, having a low automation-knowledge-level which leads to high consultative effort for sales as well as having smaller budgets compared to companies e.g. automotive OEMs or large tier-1 suppliers. Therefore, many automation solutions are not realized due to economical reasons or insufficient knowledge about their benefits. Additionally, there is the lack of planning tools, for the early basic concept planning. Configurators in general are suitable, but a complete user input of required parameters would be far too complex and time-consuming for non-experts, especially since in this phase there is a high degree of planning uncertainty.

Therefore the goal of the user-centered KBEC ROBOTOP is to support the idea generation phase for SMEs, whereby the result is a first concept that can be handed over to a system integrator for further engineering, see Figure 3. Based on the recommendation of Industry 4.0, the XML-based, open format AutomationML [33] was selected to transfer the final configuration results to system integrators. Therefore, we described the work-sharing building of data models as well as first implementations. [34]

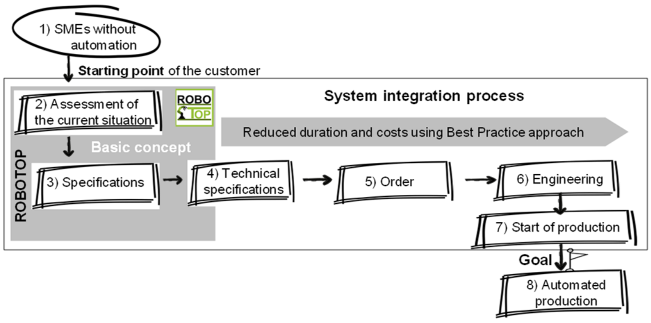


Figure 3. Scope of the user-centered KBEC for basic planning of robot based automation-solutions along the system integration process [35]

5.2 Exemplary implementation

To explain the method, essential results along the implemented validated click-prototype (in the next sections called prototype) are presented. The implementation presented in the following can be viewed as a video sequence in German at the following source [35]. As the prototype was originally developed for the German market, it has been translated within this paper into English for the sake of better communication. The result is based on more than 50 feedback loops with SMEs, system integrators and engineering experts. The prototype was incremental optimized in terms of understandability (less user comprehension questions), simplicity (as little user input as possible) and minimal completeness (only relevant information for concept planning).

1) Initial requirement entering for Best Practice filtering: Therefore, in the first step general user-requirements of SMEs such as production process description and product specifications are utilized for further Best Practice filtering in step two, see Figure 4.

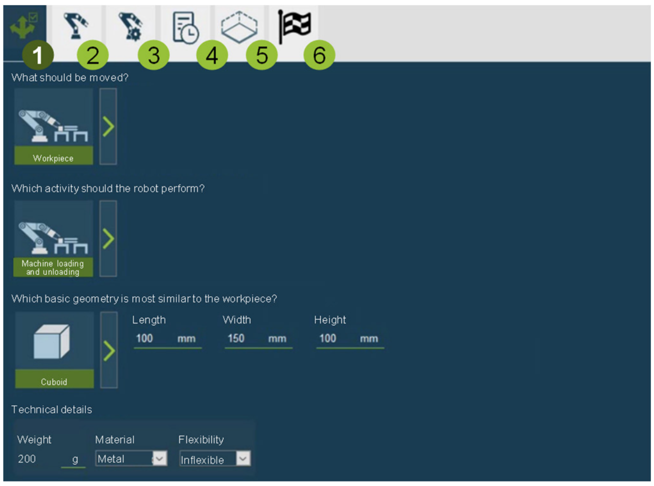


Figure 4. General condition-parameter entering (1) for Best Practice selection in the next step (2) [35]

2) Best Practice selection: The selected Best Practice serves as a starting-solution and parameter-set for the further customized configuration in step three to five. The Best Practices were conceived and condensed on the basis of a literature search, previous projects and expert discussions. The acquisition of Best Practice takes place within the scope of the ROBOTOP platform and is available to all SMEs in a consolidated database.

Based on the previous entered filtering criteria, suitable Best Practices are displayed, from which the user can select based on various qualitative evaluations such as primary advantage, cycle time, price and customer rating. Parameters for which concrete values were not available, such as price, were abstracted using rough price ranges see Figure 5, e.g. from (€€€ - €€€€) thus medium (€€€) to higher prices (€€€€).

Figure 5. Best Practice selection as initial starting point [35]

3 - 5) Customization configuration: Based on the selected and pre-loaded Best Practice parameter setup, the SME customizes the initial solution to fit most of the individual needs. The advantage of the Best Practice configuration approach is that a complete solution can be displayed as 3D-visualization right from the start, see Figure 6. This approach is therefore very user-friendly, as it provides direct and easy solution to understand the feedback. Special care was taken to ensure that the parameters with the highest technical dependencies are asked in the first phases of the user interaction with regard to changes to the overall concept.

Figure 6. Adaptation configuration (left) with 3D-visualisation (right) [35]

More complex and technically irrelevant questions are initially filled with the Best Practice-parameters. These are partially individualized in the course of steps three to five. The process parameters, such as supply and delivery condition as well as needed cycle time are individualized. Subsequently, additional basic conditions are included, such as space restrictions, desired autonomy time, protection class and data connection. By entering the shift information of the previous, manual production, the required cycle times and from these the economic potential of the automation solution can be calculated.

6) Summary and contact establishment: Lastly, the main results of the configuration are summarized. Based on request, the configuration, contact data and other requirements will be forwarded to a system integrator respectively contractor who can support the further engineering process.

5.3 Validation through user-testing

The prototype was cyclically tested and further developed along with the user-centered development process using an HTML-prototype developed within the rapid prototyping environment Axure. The configurator was evaluated for comprehensibility testing the potential SME customers and with the system integrators to ensure the technical completeness and inclusion of relevant information. This fact that the configurator requires only a few simple queries, leads us to a predefined process. The generated displays of an initial 3D-concept based on Best Practices from step three was rated positively by the users. In general, the user-centered approach was evaluated positively, even though only conceptual and technical information could be provided, since e.g. commercial information such as prices are very customer-specific. Therefore, commercial information is not provided online by system integrators and components manufacturers.

5.4 Validation through expert discussion

Furthermore, the final prototype was presented to and discussed with several experts from the field of sales of robot-based automation solutions and configuration development. The general requirement for a tool for the preliminary concept planning could be confirmed. However, sales staff in particular were sometimes sceptical about the concept, maybe because they consider their position to be endangered by such tools. The approach to develop a configuration front-end by click-prototypes via Axure was new for all experts but was considered very promising. In some cases, the cost of creating individual front-ends for different target groups was considered too high. Moreover, cost-reducing methods and therefore research projects were classified as very reasonable.

6 CONCLUSION AND OUTLOOK

In this paper, we introduced an approach for user-centered front-ends for knowledge-based engineering configurators (KBEC), considering the PPR-model (product, process and resource) from engineering. Hence we introduced the idea of an architectural user-centered layer and a ten step development approach of user-centered front-ends. Therefore, we designed and implemented an exemplary user-centered front-end within concept planning of robot-based automation solutions. The user-centered frond end is developed in parallel to the configuration logic. This serves both to clarify the requirements and to narrow down the required logic. For the implementation of a user-centered front-end, possible example software tools were identified. The steps 1-6 of the user-centered front-end could also be transferred as a general pattern to further configuration of engineering tasks. A first fusion of configuration development and user-centered approaches with focus on the front-end were shown, requiring consideration in the future projects.

As a next step, the prototypes can be analytically evaluated and optimized based on a larger number of test users. Tools to record and analytically evaluate the click-path and termination-rates of users from websites or HTML prototypes can be used, e.g. Open Web Analytics or Google Analytics. Various links to different variants of the prototype can also be distributed to specific target groups. This enables a successive further development of the configuration front-ends. Also a connection of user-centered front-end development with executable process models based on BPMN and the Process-driven approach [36], could be a promising combination for dynamic process integration within configurators.

ACKNOWLEDGEMENTS

The research and development project ROBOTOP (01MA17009E) is funded by the Federal Ministry of Economic Affairs and Energy (BMWi). It is part of the technology program “PAICE Digitale Technologien für die Wirtschaft” and is guided by the DLR Project Management Agency “Information Technologies/ Electromobility”, Cologne. We would like to thank our project partners Infosim, ICARUS Consulting, Robert Bosch, SCHUNK, IPS, wbk and LPS for their valuable contributions.

REFERENCES

[1] Dagmar Dirzus; VDI/VDE-Gesellschaft mess- und automatisierungstechnik: Durchgängiges Engineering in Industrie 4.0-Wertschöpfungsketten, 2016

[2] Brossog, Matthias; Schäffer, Eike: Prozesssimulation. In: Müller, Rainer; Franke, Jörg; Henrich, Dominik; Kuhlenkötter, Bernd; Raatz, Annika; Verl, Alexander (Hrsg.): *Handbuch Mensch-Roboter-Kollaboration*. München: Hanser, 2019, S. 284–297

[3] Schäffer, Eike; Schulz, Jan-Peter ; Franke, Jörg:Robotiklösungen im Baukastenprinzip für den Mittelstand. In: BigData Insider (2019). URL-https://www.bigdatainsider.de/robotikloesungen-im-baukastenprinzip-fuer-denmittelstand-a-882401/

[4] *Knowledge-Based Configuration*: Elsevier, 2014

[5] Shafiee, Sara ; Kristjansdottir, Katrin ; Hvam, Lars ; Forza, Cipriano: How to scope configuration projects and manage the knowledge they require. In: *Journal of Knowledge Management* 22 (2018), Nr. 5, S. 982–1014

[6] Wang, Yu ; Wu, Jiacong ; Lin, Li ; Shafiee, Sara: An online community-based dynamic customisation model: the trade-off between customer satisfaction and enterprise profit. In: *International Journal of Production Research* (2019), S. 1–29

[7] Walker, Rob: The Guts of a New Machine. In: *The New York Times*, NOV. 30 (2003)

[8] Anwar, Sadaf ; Motla, Yasir Hafeez ; Siddiq, Yasir ; Asghar, Sohail ; Hassan, M. Shabbir ; Khan, Zahid Iqbal: User-Centered Design Practices in Serum Development Process: A Distinctive Advantage? In: RASHEED, Haroon (Hrsg.): *2014 IEEE 17th International Multi-Topic Conference (INMIC) : 8 - 10 Dec. 2014, Bahria University Karachi Campus, Pakistan*. Piscataway, NJ : IEEE, 2014, S. 161–166

[9] Randall, Taylor ; Terwiesch, Christian ; Ulrich, Karl T.: Principles for User Design of Customized Products. In*: California Management Review* 47 (2005), Nr. 4, S. 68–85

[10] Shafiee, Sara ; Hvam, Lars ; Bonev, Martin: Scoping a Product Configuration Project For Engineering -to-Order Companies. In: *International Journal of Industrial Engineering and Management*, 2014 (5 (4)).

[11] Shafiee, Sara: Conceptual Modelling for Product Configuration Systems. Kgs. Lyngby, Technical University of Denmark, Department of Management Engineering. Dissertation. 2017

[12] Felfernig, Alexander ; Hotz, Lothar ; Bagley, Claire ; Tiihonen, Juha: Knowledge-based configuration: From research to business cases. Waltham, MA : Morgan Kaufmann, 2014

[13] Schäffer, Eike; Fröhlig, Simon; Mayr, Andreas; Franke; Jörg: A method for collaborative knowledge acquisition and modeling enabling the development of a knowledge-based configurator of robot-based automation solutions, Bd. 86. In: DIETRICH, Franz (Hrsg.): *7th CIRP Global Web Conference –Towards shifted production value stream patterns through inference of data, models, and technology (CIRPe 2019*) : Elsevier, 2019 (Procedia CIRP, 86), S. 92–97

[14] Zhang, Linda L.: Product configuration: a review of the state-of-theart and future research. In: *International Journal of Production Research* 52 (2014), Nr. 21, S. 6381–6398

[15] Puppe, Frank: Einführung in Expertensysteme. Zweite Auflage. Berlin, Heidelberg: Springer, 1991 (Studienreihe Informatik)

[16] Zhang, Linda L. ; Lee, Carman K.M. ; Akhtar, Pervaiz: Towards customization: Evaluation of integrated sales, product, and production configuration. In: *International Journal of Production Economics* (2020), S. 107775

[17] Mao, Ji-Ye ; Vredenburg, Karel ; Smith, Paul W. ; Carey, Tom: The state of user-centered design practice. In: *Communications of the ACM* (2005), Vol. 48, No. 3, S. 105–109

[18] Marcus, Aaron: User Interface Design's Return on Investment: Examples and Statistics. In: *BIAS*, Randolph G.; MAYHEW, Deborah J. (Hrsg.): Cost-justifying usability. Boston: Academic Press, 1994

[19] ISO 9241-11. 1998. ISO 9241-11: *Ergonomic requirements for office work with visual display terminals (*VDTs)

[20] DIN EN ISO 9241-110. 2006. ISO 9241-110: Ergonomie der Mensch-System-Interaktion – Teil 110: Grundsätze der Dialoggestaltung

[21] Nielsen, Jakob: Usability Engineering. Burlington : Elsevier Science, 1993 (Interactive Technologies)

[22] Aghina, Wouter ; Handscomb, Christopher ; Ludolph, Jesper ;West, Dave ; YIP, Abby: *McKinsey Global Institute: How to select and develop individuals for successful agile teams : A practical guide*

[23] Schäffer, Eike; Leibinger, Hannes; Stamm, Axel; Brossog, Matthias; Franke, Jörg: Configuration based process and knowledge management by structuring the software landscape of global operating industrial enterprises with Microservices, Bd. 24. In: Denkena, Berend; Thoben, Klaus-Dieter; Trächtler, Ansgar (Hrsg.): *4th International Conference on System-Integrated Intelligence: Intelligent, Flexible and Connected Systems in Products and Production*: Elsevier, 2018 (Procedia Manufacturing, 24), S. 86–93

[24] Schäffer, Eike ; Mayr, Andreas; Fuchs, Jonathan; Sjarov, Martin; VORNDRAN, Johannes; FRANKE, Jörg: Microservice-based architecture for engineering tools enabling a collaborative multi-user configuration of robot-based automation solutions, Bd. 86. In: Dietrich, Franz (Hrsg.): *7th CIRP Global Web Conference – Towards shifted production value stream patterns through inference of data, models, and technology (CIRPe 2019)* : Elsevier, 2019 (Procedia CIRP, 86), S. 86–91

[25] Schäffer, Eike; Pownuk, Tobias; Walberer, Joonas; Fischer, Andreas; Schulz, Jan-Peter ; Kleinschnitz, Marco ; Bartelt, Matthias ; Kuhlenkötter, Bernd ; Franke, Jörg: System architecture and conception of a standardized robot configurator based on microservices. In: Franke, Jörg (Hrsg.): *Tagungsband 3. MHI Kongresses* : Springer, 2018, S. 159–166

[26] Anwar, Sadaf; Motla, Yasir Hafeez; Siddiq, Yasir; Asghar, Sohail; Hassan, M. Shabbir; KHAN, Zahid Iqbal: User-Centered Design Practices in Serum Development Process: A Distinctive Advantage? In: RASHEED, Haroon (Hrsg.): *2014 IEEE 17th International Multi-Topic Conference (INMIC): 8 - 10 Dec. 2014, Bahria University Karachi Campus, Pakistan. Piscataway, NJ :* IEEE, 2014, S. 161–166

[27] Yin, Robert K.: Case study research: Design and methods. 4th ed. Thousand Oaks : Sage, 2009 (Applied Social Research Methods 5)

[28] SteinwasseR, Peter: Modulares Informationsmanagement in der integrierten Produkt- und Prozeßplanung. Zugl.: Erlangen, Nürnberg, Univ., Diss., 1996. Bamberg : Meisenbach, 1996 (Fertigungstechnik - Erlangen 63)

[29] Schäffer, Eike; Bartelt, Matthias; Pownuk, Tobias ; Schulz, Jan-Peter; Kuhlenkötter, Bernd ; Franke, Jörg: Configurators as the basis for the transfer of knowledge and standardized communication in the context of robotics, Bd. 72. In: WANG, Lihui (Hrsg.): *51st CIRP Conference on Manufacturing Systems*: Elsevier, 2018 (Procedia CIRP, 72), S. 310–315

[30] Schäffer, Eike ; Mayr, Andreas ; Huber, Tobias ; Höflinger, Tobias ; Einecke, Matthias ; Franke, Jörg: Gradual tool-based optimization of engineering processes aiming at a knowledge-based configuration of robot-based automation solutions, Bd. 81. In: Peter Butala, Edvard Govekar, Rok Vrabič (Hrsg.): *52nd CIRP Conference on Manufacturing Systems (CMS)* : Elsevier, 2019 (Procedia CIRP, 81), S. 736–741

[31] Garrett, Jesse James: *The elements of user experience: Usercentereddesign for the Web and Beyond. 2*. ed. Berkeley, Calif. :New Riders, 2011

[32] Bundesministerium für wirtschaft und energie ; forschungsprojekte ; VDI/VDE Innovation + technik GMBH: Digitale Technologien für die Wirtschaft : (PAiCE). Berlin, 10.01.2018. URL https://www.bmwi.de/Redaktion/DE/Publikationen/Digitale-Welt/paice-digitale-technologien-fuer-die-wirtschaft.html– Überprüfungsdatum 2018-12-23

[33] Automationml consortium: Whitepaper AutomationML: Part 1 - Architecture and General Requirements. URL https://www.automationml.org/o.red.c/publications.html – Überprüfungsdatum 2020-04-14

[34] Schäffer, Eike ; Penczek, Lars ; Mayr, Andreas ; Bakakeu, Jupiter ; Franke, Jörg ; Kuhlenkötter, Bernd: Digitalisierung im Engineering : Ein Ansatz für ein Vorgehensmodell zur durchgehenden, arbeitsteiligen Modellierung am Beispiel von AutomationML. In: *Industrie 4.0 Management 2019* (2019), Nr. 1, S. 61–66. URL https://www.industrie-management.de/node/256

[35] Schäffer, Eike ; THI, Kim ; FRANKE, Jörg: Bedarf und Konzeptvorstellung des Forschungsprojekts ROBOTOP für einen webbasierten Konfigurator für den Massenmarkt (2020). URL http://dx.doi.org/10.13140/RG.2.2.15152.30720

[36] Stiehl, Volker ; Danei, Marcus ; Elliott, Juliet ; Heiler, Matthias: Effectively and Efficiently Implementing Complex Business Processes: A Case Study. In: LÜBKE, Daniel; Pautasso, Cesare (Hrsg.): *Empirical Studies on the Development of Executable Business Processes.* [S.l.] : SPRINGER NATURE, 2019, S. 33–57

Towards a Modular Distributed Configuration Model for Autonomous Machines

Lothar Hotz[1] and **Stephanie von Riegen**[2] and **Rainer Herzog**[3] and **Raoul Pascal Pein**[4]

Abstract. Today, the ability to quickly and flexibly adapt to changes in product lines is becoming increasingly important, so that production systems must be adapted to changes. Those changes can come from customers, from percepting the production environment and the machine, or from newly developed components of component suppliers. This development towards a more flexible, heterogeneous and dynamic production is known as one of the key aspects of *Industry 4.0*. A production system, consisting of many different individual parts, that was once configured based on different configuration models for the specific needs of a certain production line, must then be reconfigured to meet the new requirements. In this paper, we will focus on the demands for individual phases of the reconfiguration and will share first insights into distributed configuration models that will be able to provide solutions based on single components from multiple suppliers.

1 INTRODUCTION

Production plants are composed of various individual components, which are dimensioned according to initial requirements. If the properties of the individual components are recorded in a formal ontology, a configuration model, which also takes into account dependencies between the individual parts, can be used to determine a suitable production plant configuration for the initial requirements. Changes in production requirements or the provision of new, optimized individual components may result in a whole series of new solutions to be considered and simulated before adaptation is used.

Based on an architecture for adaptation, in this paper, we identify and explain, using an example of a conveyor system that delivers metal plates to a CNC machine, which tools are needed to implement an adaptation: Percepting runtime parameters and interpret them for triggering adaptations, the computation of possible reconfiguration proposals is started based on relaxation. If one or more reconfiguration proposals are available for the adaptation, a decision on these must first be made in an evaluation phase. This decision is to be supported by a simulation environment in which the effects of the adaptation proposals are modelled. In addition, a planning phase is necessary in which the necessary prerequisites for implementing the change are modelled. Finally, the adaptation is carried out in the implementation phase. These steps are to be observed by a monitoring process that can provide information about the current adaptation phase at any time. Our framework is intended to consider the RAMI 4.0 specifications [16] and taking profit from the standardisations covered by the Asset Administration Shell (AAS). We do this by introducing distributed configuration modules which each represent a certain subsystem through a subconfiguration model and interfaces to each other.

In Section 2, we present an overview of related work. In the following Section 3, we provide the use case by which we will define our architecture. Section 4 introduces the use of distributed configuration modules for our scenario and Section 5 the concept of adaptation that we consider. Section 6 combines the aspects in the overall architecture and Section 7 summarizes the paper.

[1] HITeC, University of Hamburg, Germany, email: hotz@informatik.uni-hamburg.de
[2] HITeC, University of Hamburg, Germany, email: svriegen@informatik.uni-hamburg.de
[3] HITeC, University of Hamburg, Germany, email: herzog@informatik.uni-hamburg.de
[4] University of Hamburg, Germany, email: pein@informatik.uni-hamburg.de

2 RELATED WORK

The basic prerequisites for our approach are uniform data exchange formats and coordinated interfaces for software modules that can also be used and reused quickly by small and medium-sized companies as has also been stated by the PAiCE consortium [8]. Standardisation of formats and interfaces is an essential aspect to enable a broad application of leading technologies. Therefore, we will make use of the framework of the Asset Administration Shell (AAS), in which all physical objects such as machine components, tools, factories but also products are combinedly represented. This combination of each physical object with its AAS forms an *Industry 4.0 Component*. The AAS provides a minimal but sufficient description of an asset for exact identification and designation in its header part. The body part of the AAS consists of a number of independently maintained submodels. These represent different aspects of the relevant asset, i.e. properties and functions that can be used for different domains, such as a description regarding safety or efficiency, and various process capabilities are outlined in addition. If the asset comprises Industry 4.0 (I4.0) compliant communication infrastructure, it can be deployed directly to the asset, otherwise it is located in an affiliated IT system [14, 19]. This basic AAS structure also enables a structured recording of product variants, which can be used by a configuration model.

The development towards I4.0 has been accompanied since decades by research [17], which has already developed partial solutions that address various aspects. Hoellthaler et al. designed a decision support system for factory operators and production planners. This system is intended to support them in responding appropriately on changing production requests by adding, changing or removing production resources. Based on optimization and material flow simulation, a result is computed that shows the best solution in terms of the highest number of parts produced and the lowest manufacturing costs per part, as well as alternative solutions [7]. Zhang et al. propose a five-dimensional model-driven reconfigurable Digital Twin (DT) to manage reconfiguration tasks and a virtual simulation to verify the applicability of system changes [18]. Contreras et al. demonstrate on the basis of a mixing station, which steps are necessary to design a RAMI 4.0 compatible manufacturing system [2]. Patzer et al. investigate the implementation of the AAS based on a specific use case with a clear focus on security analysis. Together with the description of their practical experiences, they provide recommendations for the implementation of the AAS on similar use cases [13]. In contrast to those approaches, we consider the use of knowledge-based configuration technologies [5], especially the configuration model describing the variants of a machine as well as the use of constraint programming for dealing with dependencies and relations, as a basic source for handling adaptations (see Section 6).

3 USE CASE

In our use case, a system for cutting sheet metal consists of a supply station, a loading unit, a CNC laser cutting machine and an unloading unit. The supply station is the location of the raw material sheets. The loading unit is necessary to transport a single metal sheet from the supply station to the CNC laser cutting machine. The loading unit is given an order for this purpose, which includes the following information: the source coordinate of the raw material sheets (supply station), the target coordinate (CNC laser cutting machine), the sheet thickness and how the sheets can be lifted. When lifting the sheet, a sheet thickness measurement is performed to ensure that only one sheet is lifted, as oiled sheets may stick together. The target value of the sheet thickness measurement is given by the order data, the actual value is determined by a sensor. Both values must be the same within a tolerance, otherwise the process of lifting the sheet metal is repeated up to three times. The sheets are lifted by a suction matrix.

If the number of failed separation attempts exceeds a tolerable value or if the sheet metal separation leads to a delay of the entire production process, there are several possibilities to adapt the system. Increasing the speed in a later step of the production can compensate the cycle rate. A spreading magnet could also be installed which supports the sheet metal separation.

4 TOWARDS A MODULAR, DISTRIBUTED CONFIGURATION MODEL

A plant consists of several components, which in turn are composed of several subcomponents. If component manufacturers provide configuration modules that are covering their components in such a way that dependencies to other components can either be fulfilled within the same module or only occur in standardized form, then these modules can be used for flexible adaptation to changes in product lines.

For the definition of a configuration module, we start from the definitions found in the configuration terminology [9]: A *configuration* is an instance of a *configuration model*, which in turn is specified in machine-readable, semantically interpretable form. It covers all the variants of parameterized system components, which can be aggregated to configurations (or *machine descriptions*). As an additional modeling facility, we introduce *adaptation points* in the configuration model that lead to *concrete adaptation points*, i.e., components or parameters of a configuration which might change in the future during machine processing. Hence, not every component might be subject of adaptation. A configuration process results in an *initial configuration* of a machine, which is delivered with the machine to a plant operator. During the configuration process *partial configurations* are subsequently concretized through user requirements and their impacts on component parameters. Such impacts emerge from dependencies between components which are represented through constraints. Constraints can describe the compatibility between several components, or the limits of a single component, e.g. the maximum speed, torque, or the outer dimensions of a motor. Other types of constraints are plant-related (e.g. maximum floor height) or production-related. All these constraints will be taken into account by a constraint solver[5] whenever a configuration or reconfiguration process is executed.

In order to build a modular, distributed configuration model, standardisation of formats and interfaces is fundamental. Hence, it is intended in this consideration that within the framework of the *Platform Industrie 4.0* [15] a standard is to be created based on the AAS in which arbitrary component providers can integrate their products by following the given format. A standardised description of the properties and requirements of all components of a supplier can then be used as a module for a configuration model.

A *configuration module* consists of following parts:

1. a configuration model for a self-contained subsystem of a system or plant (*subconfiguration model*),
2. *subconfiguration model interfaces*, i.e., parameters that can be used in other configuration modules,
3. and an *addressable identification* (e.g. URL) that allows the use of this subconfiguration module.

Industry associations could provide key servers that would maintain a centralized library of addresses and short descriptions of modules of this type. Although the effort involved in agreeing such standardisation should not be underestimated, and though we know that many manufacturers consider their own accomplishments as part of their intellectual property, and therefore refuse to standardize their products, we focus here on the advantages, as the economic benefits can be considerable[6]. From a business point of view, the possibility of making component descriptions available as modules would, on the one hand, offer advantages for a plant manufacturer, since he can reduce previous dependencies on individual component suppliers and the optimum component for the respective application can be found from a large selection in a simple, automated way. On the other hand, even a small component manufacturer can make its products available on this basis without a great deal of marketing effort, and, thus, win new customers to whom it was previously unknown. Niche products in particular should benefit most from this. At this point, we would like to limit ourselves essentially to introducing the idea and advantages of configuration modules. The basic structure of a module is given with the specification of the AAS, while the exact definition of the interfaces should be determined in inter-trade organisations or even smaller committees, specifically for the respective application. The question of how exactly we will form an overall configuration model from the subconfiguration models is still future work, but solutions for similar tasks have already been found in the past [4, 6, 12].

5 CONCEPT OF THE ADAPTATION PROCESS

We expect a machine to be accompanied by a complete description of the currently installed (parameterized) components, here called configuration. This can be a special submodel of the associated AAS. Each configuration holds a number of constraints, that can be represented as part of a AAS submodel. An overview of the general

[5] see, e.g., the Choco solver `https://choco-solver.org/`
[6] Probably the best example for the beneficial impact of an open standard [3] is the success story [1] of the IBM PC [11] in the early 1980s.

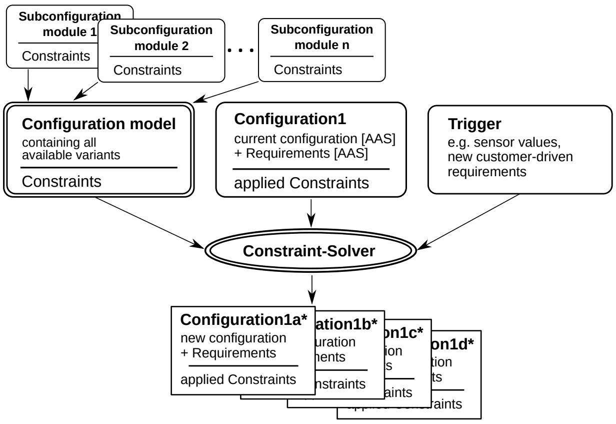


Figure 1. Input for the Adaptation Process

concept for Autonomous Adapting Machines is given in [10]. The standardisation and provision of components by several manufacturers in the configuration model as described in Section 4 alone can accelerate flexible changes, if the configuration model (as the union of the subconfiguration models) is used as described below.

The requirements (see Figure 1) describe the operators' demands that the plant must meet, such as minimal throughput (e.g. 500 units per day). The following example should clarify the difference between constraints and requirements. Assuming available supply stations can achieve a throughput of either up to 400 (SS4) or 600 (SS6), but not 500. Therefore the SS6 station is selected. However, this supply station requires the choice of a loading unit that is also able to deliver the throughput of up to 600 (LU6). There is a choice of another loading unit (LU5) that would be able to provide a throughput of up to 500, but which is not compatible with the selected supply station SS6. Thus, LU5 is fulfilling the requirements of the plant operator, but the component-related constraint which is linked to SS6 implies the choice of LU6 since LU5 does not fit.

Constraints are stored not only in the configuration model but also as "applied constraints" in the actual configuration (see Figure 1). These applied constraints enable the computation of machine-specific impacts on triggers (i.e., *affected parameters*, see below).

A reconfiguration process can be activated by a trigger, which could be determined either continuously or event-driven. A trigger can be an update of a components (e.g. new components of the component manufacturer, such as a provision of an optimized drive-system), a new requirement of the plant operator to the asset (e.g. a customer could demand that his products be manufactured only with machine parts that meet certain criteria), which will lead to changes of the requirements that are part of the adaptation model, or a sensor value (e.g. temperature, log entries).

If a trigger for adaptation occurs, the constraint solver determines whether the current configuration is sufficient to make the desired changes. If this is not the case, the configuration model will be included in the process to reconfigure the asset (see Sec. 6 for details). The result is the suggestion of one or more configurations. As the current configuration is also considered, possible solutions might be prioritized according to the fewest number of required changes. The plant operator can check and evaluate the results by applying them, if available, to the DT and might either immediately acknowledge the change (e.g. in case of the installation of a new firmware) or the proposed solution might require additional development activities to carry out the change. If no DT is accessible the evaluation has to be done by manually creating appropriate simulations.

6 ARCHITECTURE

The architecture of our Autonomous Adapting Machine (ADAM) is shown in Figure 2.

The current configuration of the asset is referenced here as "machine description" and is defined according to the standard of the AAS. In addition to the machine description, the structural model also contains the adaptation model. The adaptation model is a part of the configuration model and contains possible variants of the initial configuration, hence, the variants which are known when the machine is initially configured. The adaptation model is initially delivered besides the machine description.

The "trigger", depicted in Figure 1, as well as the runtime parameters and target values of the order to be processed on the plant, are fed into the system via the "process data". Trigger information can be provided by component or plant manufacturers. In a specific format, it is recorded for which component and with which setting an adaptation of the system could be considered.

The process data is the input to the "optional data evaluation" component that might filter out irrelevant or anonymise too sensitive data, which might contain intellectual property of the asset operator. This component may also include an anomaly detection functionality (e.g. by a machine learning module) that identifies non-essential errors or undesired, non-obvious patterns. This anomaly detection might also in turn create triggers.

In the component "determination of adaptation" the provided triggers are evaluated. This component processes three substeps: computation of affected parameters, reconfiguration of the machine, and the identification of concrete adaptation points. First, starting from the triggers, the affected parameters of the machine are computed. This means, it is checked, whether the plant is affected by such an adaptation trigger. In our example, if the number of failed attempts during sheet metal separation of the loading unit is too high, this would be the trigger, the affected parameter in the machine is the cycle rate or the component selection of the component which supports the sheet metal separation. The affected parameters are relaxed, i.e., their potential values are set to those known from the configuration model. The resulting partial configuration is the starting point of the second step, i.e., the reconfiguration. With commonly known configuration tools this reconfiguration is processed and computes a new configuration. Thus, in total, the trigger result leads to a configuration request. Here, the constraint solver will come into action. If no connection to cloud services is available, the solver will try to find a suitable solution based solely on the variants of the adaptation model. However, the more advanced scenario is made possible by access to the "ADAM Cloud" service designed in the project, see Figure 2. By transmitting the structural model and the treated process data to the ADAM Cloud, which will have access to product services of asset or component manufacturers ("solution clouds"), a more advanced configuration model is build up, making use of probably daily updated configuration modules (introduced in Section 4, here depicted as "solution clouds") and optimized solution candidates will be returned by the "ADAM Cloud".

The third step compares this new configuration with the initial configuration for computing potential adaptations points. The result of the adaptation determination is a selection of solution candidates, which are summarized in a "computed adaptations" object. In a next step a candidate can be evaluated on a DT. The suggestion of a possible adaptation, accompanied by a human expert, will then be put into the planning stage ("adaptation planning"). The final step is the adaptation of the plant. After that the new machine description and

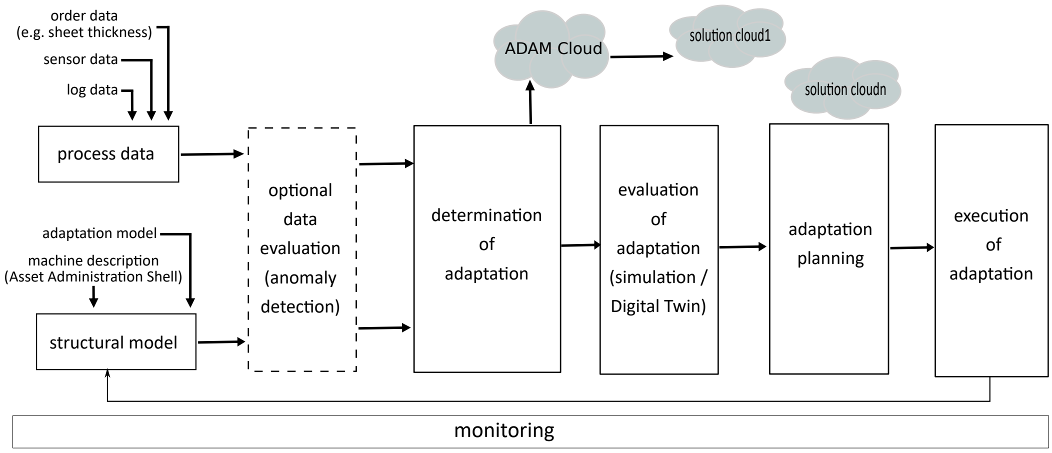


Figure 2. ADAM Architecture

an updated adaptation model is written back to the structural model. The whole process is accompanied by the aforementioned "monitoring component".

7 SUMMARY

In this paper, we described first steps on the basis of reconfiguration techniques for a modular distributed configuration model for autonomous machines. We introduced the notion of a configuration module to cope with distinct configuration models for different parts of the machine.

To support the needs reflected by Industry 4.0, we propose the use of configuration technologies not only in the beginning of a product life cycle, but also during runtime of machines in production, also called reconfiguration. Knowledge about variants and dependencies, as well as reasoning methods known from the area of knowledge-based configuration can support the adaptation of machines. However, additional technologies, such as sensor evaluation, as well as adaptation planning, monitoring and simulation on the basis of DT have to be considered. During our research, we identified concrete application scenarios, one use case has been presented in this paper for guiding the research in the direction of autonomous adaptive machines. As next steps, we consider the adaptation of the knowledge base to the specifications of AAS and RAMI 4.0 as well as the implementation of the architecture.

ACKNOWLEDGEMENTS

This work has been developed in the project ADAM. ADAM (reference number: 01IS18077A) is partly funded by the German ministry of education and research (BMBF) within the research program ICT 2020.

REFERENCES

[1] *Influence of the IBM PC on the personal computer market*, ed., Adam Cornelius Bert, Chromo Publishing, August 2011.
[2] Juan David Contreras, Jose Isidro Garcia, and Juan David Pastrana, 'Developing of industry 4.0 applications', *International Journal of Online and Biomedical Engineering (iJOE)*, **13**(10), 30 – 47, (2017).
[3] Clifton Ericson, *Concise Encyclopedia of System Safety: Definition of Terms and Concepts*, John Wiley and Sons, 2011.
[4] A. Felfernig, Gerhard Friedrich, Dietmar Jannach, and Markus Zanker, 'Distributed generative CSP framework for multi-site product configuration.', in *Workshop on Configuration, ECAI2002, Lyon, France*, pp. 77–81, (07 2002).
[5] A. Felfernig, L. Hotz, C. Bagley, and J. Tiihonen, *Knowledge-Based Configuration: From Research to Business Cases*, Morgan Kaufmann Publishers, Massachusetts, US, 2014.
[6] Alexander Felfernig, 'Standardized configuration knowledge representations as technological foundation for mass customization', *IEEE Transactions on Engineering Management*, **54**(1), 41–56, (2007).
[7] Georg Hoellthaler, Martin Schreiber, Kilian Vernickel, Johannes be Isa, Jan Fischer, Nils Weinert, Roland Rosen, and Stefan Braunreuther, 'Reconfiguration of production systems using optimization and material flow simulation', *Procedia CIRP*, **81**, 133 – 138, (2019). 52nd CIRP Conference on Manufacturing Systems (CMS), Ljubljana, Slovenia, June 12-14, 2019.
[8] Johannes Hoos, Björn Kahl, Ronny Kreuch, and Arndt Lüder, *Interoperable Informationsmodelle in Industrie 4.0 Erfahrungen und Handlungsempfehlungen aus Praxissicht*, iit – Institut für Innovation und Technik in derVDI / VDE Innovation + Technik GmbH, February 2020.
[9] L. Hotz, A. Felfernig, M. Stumptner, A. Ryabokon, C. Bagley, and K. Wolter, 'Configuration Knowledge Representation & Reasoning', in *Knowledge-based Configuration – From Research to Business Cases*, eds., A. Felfernig, L. Hotz, C. Bagley, and J. Tiihonen, chapter 6, 59–96, Morgan Kaufmann Publishers, (2013).
[10] L. Hotz, St. von Riegen, R. Herzog, M. Riebisch, and M. Kiele-Dunsche, 'Adaptive autonomous machines – requirements and challenges', in *Configuration Workshop, 2019*, eds., L. Hotz, T. Krebs, and M. Aldanondo, pp. 61–64, (September 2019).
[11] IBM Archives. The birth of the IBM PC. `http://www-03.ibm.com/ibm/history/exhibits/pc25/pc25_birth.html`, 2020. [Online; accessed 30-May-2020].
[12] Dietmar Jannach and Markus Zanker, 'Modeling and solving distributed configuration problems: A csp-based approach', *Knowledge and Data Engineering, IEEE Transactions on*, **25**, 603–618, (03 2013).
[13] Florian Patzer, Friedrich Volz, Thomas Usländer, Immanuel Blöcher, and Jürgen Beyerer, 'The industrie 4.0 asset administration shell as information source for security analysis', in *Emerging Technologies and Factory Automation, IEEE International Conference on*, pp. 420 – 427, (2019).
[14] Plattform Industrie 4.0, *Details of the Asset Administration Shell*.
[15] Structure and organization of the Plattform Industrie 4.0. `https://www.plattform-i40.de/PI40/Navigation/EN/ThePlatform/Structure-Organization/structure-organization.html`. [Online; accessed 30-May-2020].
[16] *DIN SPEC 91345: Reference Architecture Model Industrie 4.0 (RAMI4.0)*, April 2016. Beuth Verlag GmbH, Berlin.
[17] B. Scholz-Reiter and M. Freitag, 'Autonomous processes in assembly systems', *CIRP Annals*, **56**(2), 712 – 729, (2007).
[18] Chenyuan Zhang, Wenjun Xu, Jiayi Liu, Zhihao Liu, Zude Zhou, and Duc Truong Pham, 'A reconfigurable modeling approach for digital twin-based manufacturing system', *Procedia CIRP*, **83**, 118 – 125, (2019). 11th CIRP Conference on Industrial Product-Service Systems.
[19] ZVEI e.V., *Struktur der Verwaltungsschale - Version 2, Fortentwicklung des Referenzmodells für die Industrie 4.0 - Komponente*, 2015.

Configuration Models Used for Design Review of Façade Systems in Building Renovation

Irene Campo Gay[1] and **Lars Hvam**[2]

Abstract. This article discusses how a configuration model may serve as a tool for conducting a design review of a product, and, thus, improve its design. There is a globally increasing demand for operational solutions for the renovation of multistory buildings, particularly in terms of sustainable methods. In this context, the renovation of façades to enhance buildings' energy performance is fundamental and entails a mass customization problem. The construction industry is familiar with mass customization methods, and literature suggests a few examples of configurators' use for customizable façade systems. This article presents a framework and a case study where the configurator model is used to evaluate, improve, and clarify details missing in the product design, i.e., a façade system. The framework includes a method to structure and relate the models of the existing building façade and the new façade system to be mounted based on modular elements. The façade system was portrayed utilizing a so-called product variant master and the structuring principles from the domain theory on how to model mechanical products. Besides, the study attempts to address multiple stakeholders' requirements across the configurator development process, i.e., the architecture, supplier, and construction companies. The case study demonstrated that configuration models are a valuable tool to assess the façade system design and can reveal several design details that still require to be specified.

1 INTRODUCTION: MASS CUSTOMIZATION IN CONSTRUCTION

The construction industry has experienced two clear general tendencies during the past 70 years, and it is now moving towards a new trend drawn in principles of mass customization [27].

In the early 1920s, the concept of mass-produced buildings started to generate interest promoted by visionaries such as Le Corbusier. He exposed the need for reviewing building construction practices evoking the automotive production example implemented by Henry Ford [11].

Years later, after the end of World War II, the shortage of dwelling instigated significant international developments in construction. In 1947 the Danish government established the "Housing Ministry," which, together with the social housing companies, urged the construction industry to the mass-production paradigm that lasted until the late 1970s [20]. The first phase of this period entails the decade of the 1950s with the construction of the first elementary buildings [12]. The second phase embraces the decades of the 1960s and 1970s in which the main two subjects were the modular coordination and the dimensional standardization [14, 21]. In these terms, the Danish Building Act from 1961 stated that all the new projects for rented apartments had to be planned following the predefined modular principles. Moreover, in line with the international modular convention, different Danish Standard Specifications (DS) and Recommendations (DS/R) were conceived to constitute the dimensional basis for the production of prefabricated components for buildings [13].

Though, from 1980, mass production in the construction industry yielded a new individual customization paradigm in which the principal aim was customizing unique buildings for each customer [27].

At this juncture, the construction industry is currently embarking on a new path that points at achieving the benefits of mass production without renouncing to the construction of unique buildings. Hence, by adopting a mass customization strategy, the construction industry could achieve the same benefits that diverse manufacturing industries have already reaped during the past decades [10]. It has been proved that this new scenario of prefabricated construction elements and on-site assembly can enhance the quality while lowering the prices and can pave the way for reaching sustainable construction [29].

The structure of the paper is as follows. Section 2 presents the addressed problem. Section 3 shows how the research in question differs from previous works. In section 4, the relevant theories that support the research study are evaluated. Section 5 describes the research method. Section 6 proposes a novel framework for a configuration model development to check and review a façade system design. A case study to validate the proposed approach is presented in section 7. Section 8 discusses the relevance of the results and analyses possible improvements. Finally, the paper concludes with section 9.

2 PROBLEM STATEMENT

There is a global demand for industrialized solutions to face the renovation of the large number of buildings erected during the decades of 1960 and 1970. This study is part of a project launched by the Danish construction industry to improve the buildings' energy performance and develop industrialized solutions for the refurbishment of façades.

Moreover, the view of buildings as one-of-a-kind has led to a fragmented value chain [26]. Hence, there is a need for developing operational solutions in terms of cooperation methods to liaise with the different stakeholders, i.e., the customer, architecture firm, constructors, and suppliers [3].

Within this framework, a configurator project could serve to identify missing definitions on the product design and to clarify the decisions that still have to be made.

[1] Department of Management Engineering, Technical University of Denmark, Denmark, email: ircag@dtu.dk
[2] Department of Management Engineering, Technical University of Denmark, Denmark, email: lahv@dtu.dk

This paper (1) proposes a framework to model a façade system for a product configurator underpinned by a product variant master (PVM) model and (2) presents a method to review the design of a façade system by developing a configuration model.

3 RELATED WORK

The literature presents a few initiatives that have approached the challenge of developing configurators for customizable façades.

Barco et al. developed a configurator for mass customization façade renovation, mainly focusing on the combinatorial challenge of dimensioning and positioning building envelopes [2, 1]. It presents extensive work on structuring such knowledge and coping with the constraints problem when programming the configuration system. It should be noted that the current study does not present a combinatorial challenge since it targets specific buildings whose envelopes are predefined as per the Danish Standards from 1960 and 1970s.

On the other hand, Farr et al. considered the advantages of utilizing BIM as a generic configurator to promote customization [6]. Also, Piroozfar et al. investigated how modularization and configuration principles could be applied in a BIM platform to support the Architecture, Engineering, and Construction industry [23].

In this regard, previous work has mainly focused on designing an information model to be computerized. Conversely, this paper centers the attention on a previous stage of the configuration model development—the translation of knowledge from the real world into a sub-set of information that can be structured into a phenomenon model.

Furthermore, few studies have addressed the formalization of phenomenon models in the construction industry. Ramaji et al. proposed the Product Architecture Model (PAM) to depict the information of modular buildings [24] while Hvam and Thuesen suggested the use of the PVM model to structure construction information models [16]. However, presumably, none has depicted a façade system, and, particularly the relationship between an existing façade and a new façade system.

Hence, there is room for improvement in methodologies and tools that can elucidate missing design specifications when developing configuration systems.

4 THEORETICAL BACKGROUND

4.1 Structure and modular coordination

The development of the proposed model relies on the systems theory. Provided that the new configurable façade system is to be mounted on top of the existing façade system, it is essential to ensure consistency within the system levels. By understanding the new configurable façade system as the core level, two super-system levels are identified; the existing façade system and the working site system (see Figure 1).

This kind of systems' correlation and properties inheritance is, for instance, common in building renovation projects such as façade renovation [6] and on new balconies installation.

In these terms, the new façade system's modularity relies on the existing façade system and, to a lesser extent, to the working-site. The modular coordination of the existing façade was defined as per the standards established at the time [19] and delineates the scope of the new façade system. Hence, the backbone of the façade system aggregation hierarchy is the same for both existing and new building envelopes, see Figure 2. A façade can be understood as the repetition

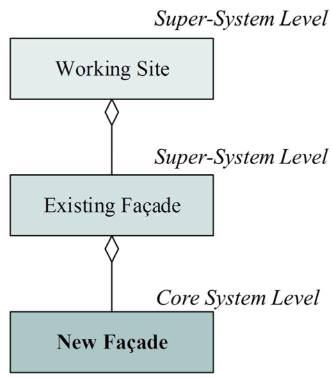


Figure 1. System levels depicted through aggregation structure.

of the same modular units. In turn, these modules are composed of panels that may have a certain number of openings.

Consequently, for example, the height of the new modular system panels is determined by the standard story height defined on the Danish Standards *DS1000*.

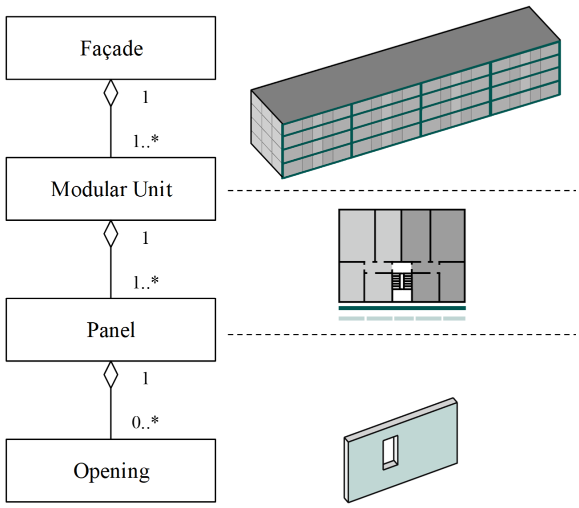


Figure 2. Façade system hierarchy in UML notation based on the Danish Standards conceived during the 1960s and 1970s decades.

4.2 Modeling method

Research has borne out the need to extract and represent expert knowledge in formal product models to develop configuration systems [9]. A configurator is based on a product model range, which includes a subset of knowledge from the real world provided by the domain experts. Figure 3 illustrates the major phases of the knowledge translation from the real world to an IT system, in this case, the configurator.

The first step is to scope and structure the knowledge that is to be incorporated into the configurator. In this regard, a phenomenon model is built containing relevant and standardized information regarding both the product and its lifecycle properties.

Few studies have analyzed the generation of formal phenomenon models in the construction industry. On the one hand, Ramaji et al. propose the Product Architecture Model to particularly represent the

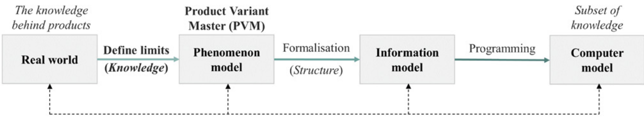


Figure 3. From the real-world to an IT system. Adapted from [4].

information of modular buildings [24]. Hvam and Thuesen suggest using the PVM model to capture the products' knowledge, including the specification of the parts and their functions and properties [16, 15]. This research applied the PVM representation model widely used in the manufacturing industry to the building industry. The creation of the model required a novel approach focusing on the interaction between the systemic levels.

The PVM method is structured based on the rationale of domain theory. It provides a holistic perspective of the product range from three different dimensions: the customer, the engineering, and the part viewpoint. The systems theory is also a cornerstone for identifying the system boundaries and interactions with the environment. Lastly, the formalization of the method relies on object-oriented modeling techniques and, thus, integrates three kinds of relations between elements. First, the *part-of connection* or *aggregation* mapped on the generic structure side of the model. Second, the *kind-of connection* or *inheritance* depicted on the variant side of the PVM. And, finally, the *instance connection* which shows when an object needs another object to fulfill its responsibilities defined on the leftmost side [7]. The *generic structure* of the model contains the classes described with a series of attributes and constraints. Besides, it is possible to specify the number of units with the cardinality property. Differently, the variant structure presents the alternatives for an element [18] (See Figure 4).

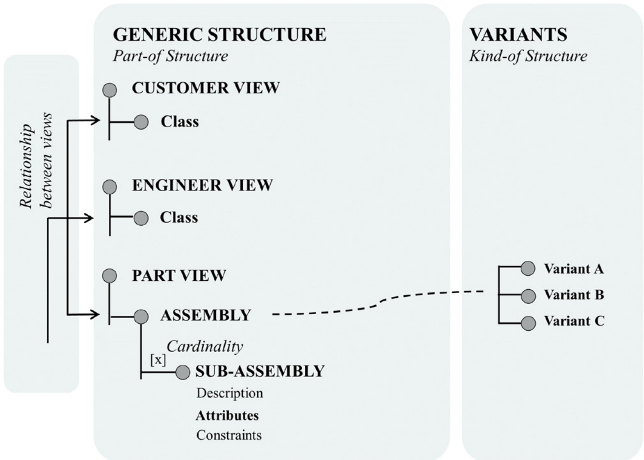


Figure 4. Basic notation of the product variant master (PVM) model.

5 RESEARCH METHOD

In accordance with the problem definition, the project relies on a partnership company case. The exploratory case study research method was adopted to gain a deep understanding of configuration model projects used for the design development and review of a product range. Besides, the single case study aimed to extensively analyze the benefits and applicability of a framework to model a novel façade system [28].

First, two main topics were explored during the literature review. On the one hand, the historical buildings to be renovated and, on the other hand, the previous work on construction configurators and, more specifically, façade configurators.

The knowledge acquisition and data processing were elaborated through qualitative research. Initially, weekly semi-structured interviews were performed with the architectural firm to formalize and define the new façade system knowledge. These semi-structured interviews were also performed once per week with the construction supplier company, which led the project. The meetings included interviews with different experts to address subjects such as energy-saving, product and production costs, and lifecycle management. Furthermore, three of these meetings were also supported by university researchers to broaden the knowledge on the different mentioned topics. Finally, monthly semi-structured interviews were also conducted with the construction company to discuss the technicalities and cost drivers.

During the early phases of configuration projects, decision-making is considerably important and is a task that can be complex and time-consuming. Thus, the purpose was to efficiently gain an in-depth understanding of the knowledge to be included in the configurator. Later, the meetings were used during the configurator iterations for solving conflicts and validating the model.

The interviews were underpinned by a set of small workshops designed to facilitate the interaction between relevant actors. Besides, a major workshop assisted by all the parts to be affected by the configurator was performed. It included the different potential configurator users—the social housing building owners—, all the different stakeholders and other secondary companies involved. The session facilitated the clarification of expectations and the evaluation of the configurator requirements within the participants' purview.

6 PROPOSED FRAMEWORK

As acknowledged by researchers, decisions made during the conceptual modeling phases of configuration projects have a crucial influence on the product's turnover and cost [17]. In addition, distilling and mapping the information from domain experts' knowledge is one of the most tedious activities [9]. Diverse methodologies have been presented to assist in establishing the configurator's scope, identifying the project goals, and acquiring the required knowledge [25]. However, these methods are intended to facilitate the cooperation within departments of a company and do not focus on the objectives' misalignment between differing businesses.

Moreover, a particularity of the project was the simultaneous development of the product range, along with the configuration system. Consequently, an iterative design method was proposed to enable the requirements clarification for the product range's formalization. The suggested framework was intended to check the façade system design and elucidate the model specifications that needed to be addressed to develop the configurator. Besides, it also enabled an ongoing dialogue between the pertinent stakeholders—an architectural firm, a construction company, and a supplier enterprise—and the configuration team.

The emphasis was placed on the ability to differentiate between functional requirements, i.e., crucial for the development of the project, and nonfunctional requirements, i.e., preferable such as performance and reliability [5, 10].

The sequential process is illustrated in Figure 5. The first step is to define the aesthetic preferences from an architectonic perspective. Based on that, the second step concerns the description of the technical requirements from two angles: the compelling technicalities and the preferred technicalities. The final step involves delimiting the financial implications subject to the previous choices.

The framework is designed to contemplate knowledge creation through conversion between two dimensions, *tacit knowledge* and *explicit knowledge*. In this respect, step 1 is presumed mainly tacit, challenging to formalize and communicate due to its subjective character. Conversely, steps 2 and 3 embrace knowledge of explicit nature, meaning that it is rather formal and systemic [22]. Such features also determined the framework' sequence.

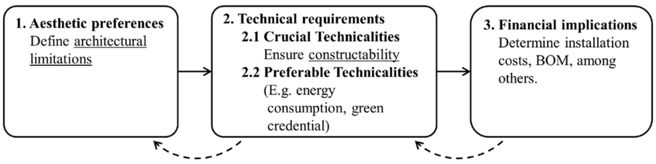


Figure 5. The proposed framework for requirements definition in construction configuration projects.

Dealing with different requirement magnitudes lay bare the necessity for an iterative framework. Particular emphasis should be placed on matching the aesthetic preference with the rest of the requirements since some aspects might remain subjective. Given that the construction industry is in a pivotal moment, some parts might be reluctant to embrace the new mass customization trend.

Equally important is to distinguish between the crucial technicalities that signify the buildability of the project and the preferable technicalities that upgrade its quality. The last ones may include the reduction of energy consumption or improving green credentials.

Cost drivers' aspects such as the installation costs, products, and production costs, inter alia, should be deferred until a later stage when the aesthetics and technical aspects have been addressed.

7 CASE STUDY

The use of configuration projects for the product range conception from the ground up is not unprecedented [2, 1]. It is known that the configuration system industry has worked in projects where the development of the configurator was parallel to the product range design. However, to the best of the authors' knowledge, the phenomenon of utilizing configuration systems as a review design tool per se has yet been little analyzed.

The described framework's application on the case study brought to light several design specifications that required to issue a decision over the product range. Table 1 gathers an overview of the design details revealed by the configuration model that still needed to be completed.

The process involved first outlining the primary aesthetic preferences. It was crucial for this step to precede the decisions on technical and financial limitations. It could not be initiated under significant constraints since it also constitutes a creative process. The outputs could then be contrasted with feasible construction solutions and evaluated through cost assessment.

For instance, a fundamental requirement from the architectural point of view was the increment of the openings' area. This specification revealed new technical conditions. The potential options included cutting and covering the panel or removing the existing panel

Table 1. Examples of missing design specifications recognized through the configuration model development.

	Design detail	Factor	Target
1	U-value of new parts	Reduction in heat losses (%)	Thermal improvement
2	Reference U-value of the existing façade system		
3	R-value of new parts	Weighted Sound Reduction (%)	Acoustic improvement
4	Reference R-value of the existing façade system		
5	Parts prices	Cost Drivers	Quotation (± error margin)
6	Production prices		
7	Installation prices		
8	Parts CAD (parts library)	2D/3D representation	Façade Visualization
9	Panels CAD (feasible combinations definition)		
10	Area of new openings (dimensions' specification)	Area improvement (%)	Visual improvement
11	Reference area of existing façade openings		
12	CO_2 emissions associated with all processes	CO_2 reduction (%)	Sustainably saved CO_2
13	Reference CO_2 emissions of standard façade renovation processes		

and replacing it with a new one. Finally, the second option was discarded due to the high costs and the need for reallocating the tenants during the renovation course (see Figure 6).

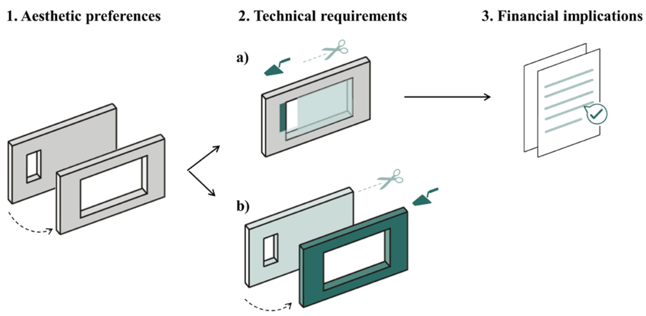


Figure 6. Example of the framework application.

In contrast, previous to the configurator project inclusion on the façade's renovation project, the aforementioned design decisions could not be easily held. The optimal design solution had been recognized as the one maintaining the existing openings' area. The reasons were the lower technical difficulties implying shorter construction duration and the overall cost reduction due to the decrease in installation operations and material expenses. Nevertheless, this compromise had not contemplated the added value of offering bigger openings to the customers. Thus, the different stakeholders had been unable to reach an agreement that could satisfy all the parties.

For these reasons, it can be stated that the use of the proposed framework improved the system design through the configuration model development. It constituted a considerable improvement compared to prior to applying this method in terms of information quality, efficiency, and error reduction, which could be achieved by identifying the different requirements interdependency, shortening the process by facilitating a smoother communication and analyzing the missing design features.

The acquired knowledge was scoped and contained in a PVM model. The product range model was constructed through a top-down approach meaning that the system was first segmented into a few main modules. The demands of different stakeholders compromise a façade system, which can lead to an infinite solution space. In order to delimit the number of elements and possible combinations, the project was conducted over a conceptual procedure where the focus was on larger modules. The main objective was to depict a clear relationship between the two systems, the existing and the new façade. It implies that a priori, the product range does not cover all the individual components but can be steadily extended.

Rules, attributes, and complex notation written in the programming language can be confusing for people without IT-related knowledge. The PVM technique served not just as an ontological model to formalize the information but also as a communication tool to bridge the gap between domain experts and configuration engineers. Hence, the different stakeholders could benefit from seeing their requirements aligned in a common model and the relationship between them through the *instance connection* depiction.

In this particular case study, using the framework for the phenomenal model's development was well-received by both the contractor and the constructor company, although the last one was not used to standard procedures and formalized models. However, the architectural firm was more reluctant to its adoption since it was perceived as a limitation for the creative process. Withal, it was demonstrated that the configurator played a significant part in clarifying the architects' choices and promoting decision making.

The model was gradually built up under an iterative process while the project was scoped through decisions on the product range standardization. Simultaneously, when the model was mature enough, the configurator was programmed and upgraded accordingly to the PVM evolution.

The time investment in the PVM model construction pays off during the programming phase of the configuration. Moreover, it also serves as documentation for the project. It has been demonstrated that undocumented projects can be detrimental to the configurator's optimal performance and for subsequent upgrades of the IT system [8].

Figure 7 presents an example of the PVM model. The attributes, variables, and constraints were consistently mapped, and its logic was validated with the domain experts. It should be stressed that both the user of the configurator and the customer were identified as the same person, the building owner. On this basis, the existing façade system was specified on the PVM's *Customer View*, considering that it concerns to the customer. Likewise, the *Part View* concerning the new façade system was structured and generated according to the *Customer View*. For instance, the panel frame's width was established as a result of the horizontal 300 standard modules adopted for the existing façade system.

8 DISCUSSION

This project's main aim was to develop a modular system to renew buildings' envelopes from the 1960s and 1970s. For this reason, a proposed framework was used to determine the different stakeholders' requirements and enhance the specification process. The purpose was to cope with the needs that emerged from diverse construction industry businesses—an architectural firm, a supplier company, and a construction enterprise. Under this procedure, the aesthetic preferences, technical requirements—including crucial and preferable features—and financial limitations were iteratively and sequentially defined. This method aided to streamline communications and optimize

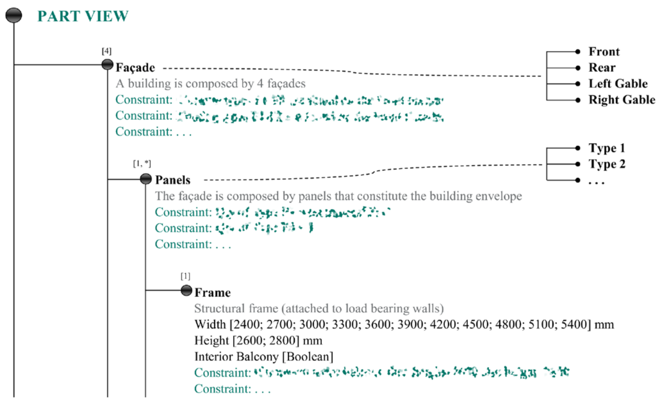


Figure 7. Extract from the product variant master model.

the formalization of the knowledge.

Moreover, the potential benefits of employing a configuration model for the design review were demonstrated since it enabled the identification of several pendant design details that had yet to be addressed. Utilizing a configurator project to evaluate and improve a product design is not an unusual practice on configuration projects; however, the topic has not been intrinsically explored. The consciousness of using configuration models for the design review could launch new research lines regarding the advantages and applicability of configuration system projects.

Besides, the project was formalized and documented through a PVM tool utilized as an ontological model to contain and communicate the system's knowledge. The information was organized according to the domain systems theory through a top-down approach. This architecture facilitated the data correlation between different systems, such as the interaction of the façade system with the working site.

Attention should also be drawn to the fact that the observed advantages are based on a single case analysis. An analysis of a new company case could shed light on further interests of the use of configuration models for the product range's enhancement and could contribute to a reassessed framework for requirements specifications.

9 CONCLUSION

This paper confirmed the applicability of configuration system models as design review methods. With a formal framework, it was possible to recognize the different design specifications that still needed to be tackled. The process was facilitated using a PVM architecture, which has demonstrated to be an adequate model for the organization and structuration of buildings information. In this respect, the main benefits of the method mentioned above are listed as: the quality improvement of the formalized knowledge model, reduced time, and increased efficiency in terms of stakeholders' requirements alignment and reduction of information errors, including missing design specifications. All in all, it entailed a complete and improved product design. An apparent limitation of the project could be the evaluation of the methodologies on a single study case. In future research, it is aimed to refine the framework methodology through new construction case studies. Moreover, the use of configuration models for revisiting the design architecture has been seldom researched in the academia. Hence, there is a potential for investigating the advantages of using configuration models to evaluate the product's design. Future work should extend this study to the benefit analysis of using configuration projects for the design review, also beyond the construction industry.

REFERENCES

[1] A. F. Barco, E. Vareilles, P. Gaborit, and M. Aldanondo, 'Industrialized building renovation: Manufacturing through a constraint-based on-line support system', *Ieee International Conference on Industrial Engineering and Engineering Management*, **2016-**, 7385789, 947–951, (2015).

[2] Andrés F. Barco, Élise Vareilles, Paul Gaborit, and Michel Aldanondo, 'Building renovation adopts mass customization: Configuring insulating envelopes', *Journal of Intelligent Information Systems*, **49**(1), 119–146, (2017).

[3] Martin Bonev, Michael Wörösch, and Lars Hvam, 'Utilizing platforms in industrialized construction: A case study of a precast manufacturer', *Construction Innovation*, **15**(1), 84–106, (2015).

[4] AHB Duffy and MM Andreasen, 'Enhancing the evolution of design science', in *Proceedings of ICED*, volume 95, pp. 29–35, (1995).

[5] C. Ebert, 'Dealing with nonfunctional requirements in large software systems', *Annals of Software Engineering*, **3**, 367–95, 367–395, (1997).

[6] Eric R.P. Farr, Poorang A.E. Piroozfar, and Dexter Robinson, 'Bim as a generic configurator for facilitation of customisation in the aec industry', *Automation in Construction*, **45**, 119–125, (2014).

[7] Ulf Harlou. Developing product families based on architectures: Contribution to a theory of product families, 2006.

[8] Anders Haug and Lars Hvam, 'The modelling techniques of a documentation system that supports the development and maintenance of product configuration systems', *International Journal of Mass Customisation*, **2**(1/2), 1–18, 1–18, (2007).

[9] Lars Hvam, Martin Bonev, Anders Haug, and Henrik Mortensen, Niels, 'The use of modelling methods for product configuration in industrial applications', *Proceedings of the 7th World Conference on Mass Customization, Personalization, and Co-creation (mcpc 2014), Aalborg, Denmark, February 4th - 7th, 2014*, 529–539, (2014).

[10] Lars Hvam, Niels Henrik Mortensen, and Jesper Riis, *Product Customization*, Springer Publishing Company, 2008.

[11] Peter Kerr and Joe Wollen, *Autopia: cars and culture*, Reaktion Books, 2002.

[12] M. Kjeldsen, *Halvtredsernes byggeri*, Teknisk forlag,, 1961.

[13] M. Kjeldsen, *INDUSTRIALISED BUILDING IN DENMARK*, 1965.

[14] M. Kjeldsen, *Industrialized housing in Denmark*, Byggecentrum,, 1988.

[15] Anders Kudsk, Lars Hvam, Christian Thuesen, Martin O'Brien Grønvold, and Magnus Holo Olsen, 'Modularization in the construction industry using a top-down approach', *Open Construction and Building Technology Journal*, **7**, 88–98, (2013).

[16] Anders Kudsk, Christian Langhoff Thuesen, and Lars Hvam, 'Modules in historic building construction according to the copenhagen building code', *Proceedings of the World Conference on Mass Customization, Personalization, and Co-creation*, (2011).

[17] Niels Henrik Mortensen, Ulf Harlou, and Anders Haug, 'Improving decision making in the early phases of configuration projects', *International Journal of Industrial Engineering (online)*, **15**(2), 185–194, (2008).

[18] Niels Henrik Mortensen, Lars Hvam, and Anders Haug, 'Modelling product families for product configuration systems with product variant master', *Ecai 2010*, (2010).

[19] J.F. Munch-Petersen, *Facadeelementer*, volume 4, 1971.

[20] J.F. Munch-Petersen, J.F. Munch Petersen, and DTH, *Politiske teknologiske initiativer*, volume 149, 1981.

[21] H. Nissen, *Industrialized building and modular design*, Cement and concrete association,, 1972.

[22] Ikujiro Nonaka, 'A dynamic theory of organizational knowledge creation', *Organization Science*, **5**(1), 14–37, (1994).

[23] Poorang Piroozfar, Eric R.P. Farr, Lars Hvam, Dexter Robinson, and Sara Shafiee, 'Configuration platform for customisation of design, manufacturing and assembly processes of building façade systems: A building information modelling perspective', *Automation in Construction*, **106**, 102914, (2019).

[24] Issa J. Ramaji, Ali M. Memari, and John I. Messner, 'Product-oriented information delivery framework for multistory modular building projects', *Journal of Computing in Civil Engineering*, **31**(4), 04017001, (2017).

[25] Sara Shafiee, Katrin Kristjansdottir, Lars Hvam, and Cipriano Forz, 'How to scope configuration projects and manage the knowledge they require', *Journal of Knowledge Management*, **22**(5), 982–1014, (2018).

[26] Christian Thuesen and Lars Hvam, 'Rethinking the business model in construction by the use of off-site system deliverance: Case of the shaft project', *Journal of Architectural Engineering*, **19**(4), 279–287, (2013).

[27] Christian Thuesen, Jens Stissing Jensen, and Stefan Christoffer Gottlieb, 'Making the long tail work - reflections on the development of the construction industry the past 25 years', *Association of Researchers in Construction Management, Arcom 2009 - Proceedings of the 25th Annual Conference*, 1111–1120, (2009).

[28] Chris Voss, 'Case research in operations management', *Researching Operations Management*, 162–195, (2008).

[29] Hossein Zabihi, Farah Habib, and Leila Mirsaeedie, 'Definitions, concepts and new directions in industrialized building systems (ibs)', *Ksce Journal of Civil Engineering*, **17**(6), 1199–1205, (2013).

Configuration and Mass Customization of Domotics to support SMEs and their Customers

Gerhard Leitner [1] and **Martin Stettinger** [2] and **Anton Josef Fercher** [3]

Abstract. In the recent years, smart home systems and components, also known as domotics, have been literally mushrooming. Today hundreds of different systems are available on the end-consumer market. However, because of the difficulty for the end-consumer to overlook the variety of products, the percentage of real smart homes (fully equipped and not only characterized by the presence of single smart components) is far behind the estimations. A related problem is that off-the-shelf products cannot be easily installed and require the support of professionals.

The latter, often working in small and medium-sized enterprises (SMEs), have to deal with the situation of having to install products of their own portfolio in heterogeneous technical infrastructures, and, increasingly, combine them with products from doubtful origin. Due to a lack of compatibility/interoperability of smart home systems, this can constitute a big challenge. Even minimal standards, such as those present in other fields of ICT (e.g. USB connectors), are not existing in the domotic area. These and other problems finally led to a still weak penetration of smart home technology. However, there are ways to overcome the problems involved, one of them is Mass Customization in Combination with Configuration presented in this paper. The approach is applied within the Interreg funded project Mass Customization 4.0 (MC 4.0) and aims at designing (mass) customized solutions for domotic systems based on configuration technology.

1 INTRODUCTION

Today hundreds of different smart home systems or components are available, in the industrial and public building sector, but increasingly also on the end-consumer market. The latter is characterized by a variety of mainly off-the-shelf products. Today, every store for electronic appliances, hardware store or even supermarkets, offers smart home products from different brands that promise a better life to their buyers in terms of comfort, safety, energy-saving, and so forth. A literature review by [1] illustrates the diversity of smart home systems on the basis of the technical foundations (Zigbee, WLAN, Bluetooth), interoperability issues, or related costs.

However, it is difficult to give a precise number of different systems available, because they differ from country to country, depending on, for example, governmental regulations for electrical devices. To get an idea of the magnitude, we refer to the smart home software platform OpenHAB (Open Home Automation Bus) [8], which aims at integrating smart home components and systems of different manufacturers. OpenHAB supports the integration of around 400 different systems, in different categories that are, in a broader sense, domotics. The range goes from infrastructural technologies such as Zigbee or Bluetooth, over Webservices (Weather, Calendaring, etc.) to domotic systems and subsystems from different manufacturers (Ikea, Philips) to control heating, lighting and cooling. The latter constitute the majority of devices supported by OpenHAB.

The sheer number of systems highlights the difficulty for the end consumers to keep an overview of the available products and to identify whether a specific product could support their needs. Supporting information and advertising material cannot answer all questions, because off-the-shelf systems typically cannot just be placed into the living environments such as smartphones or other stand-alone smart devices. Domotics have to be integrated and their installation and maintenance at least require the support of technically experienced people, oftentimes certified professionals, such as electricians.

The latter - speaking of the typical situation in Austria - are employed in small and medium-sized enterprises (SMEs) with an - in general - small and manufacturer-specific product portfolio. Due to the diversity and manifold availability of smart devices emphasized above, they are increasingly confronted with the problem of having to compete with companies that address customers via alternative channels (online or consumer markets) and, as a result, increasingly have to deal with the integration and combination of their own products with products and devices from such other sources of supply.

An example of a situation a professional might have to deal with occurred in the past in the household of one of the authors. The house was equipped with basic smart home functionality, such as controls for lights, heating and blinds. When the children in the house were small, their sleep was monitored with a baby-phone. During this period, the blinds in the household occasionally did not work properly. The smart home's central gateway operated on LAN/WLAN to enable remote control (via Web interface) and on 433 MHz radio frequency for the communication with the attached components (such as the blinds). Due to low robustness (the system was one of the first generation wireless smart systems, around the year 2005) malfunctions occurring from time to time were rather the rule than the exception. The standard strategy in such cases was to check all components of the system for appropriate connection, reset and restart the software-based components, but in the concrete case, no improvement of the situation resulted. It finally turned out that the baby-phone also operated on 433 MHz radio frequency and interfered with the smart home system. Many other examples from past projects could be given, illustrating the challenges for consumers and local professionals that are related to the proliferation of smart home systems.

[1] Department for Informatics-Systems, University of Klagenfurt, Austria, email: gerhard.leitner@aau.at
[2] Department for Informatics-Systems, University of Klagenfurt, Austria, email: martin.stettinger@aau.at
[3] Selectionarts GesmbH. Graz/Klagenfurt, Austria, email: a.fercher@selectionarts.com

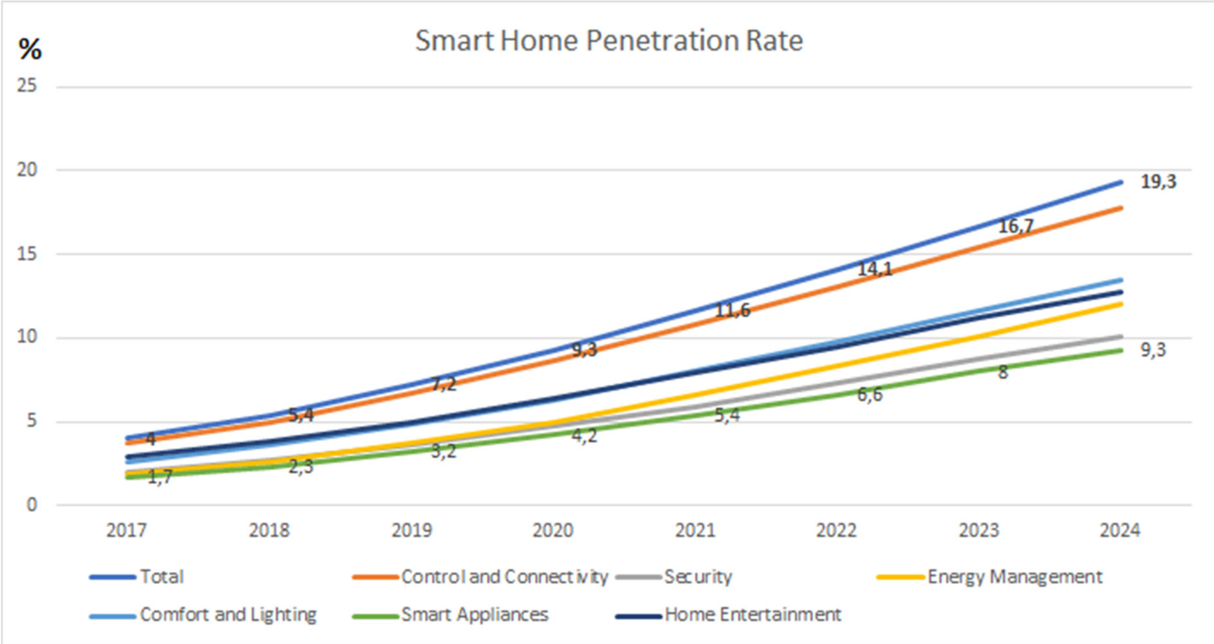


Figure 1. Penetration Rates and Estimations for the development of Smart Homes, adapted from [5]

The project MC 4.0 addresses the related challenges, by the involvement of companies from the field of domotics representing the perspective of the industry. The second important column of the project is an appropriate scientific approach providing the industrial partners and their end consumers with state-of-the-art know-how. Research challenges in different domains have to be faced in this context. To be able to better support end-consumers as well as SMEs, which both cannot be assumed to be ICT experts, a special focus is put on the user experience/usability of the digital tools developed in the course of the project [6]. The other research gap to be filled is to find appropriate tools to apply and adapt to the area of domotics systems. In the case of configurators, related work on, for example, system interoperability, system configuration [10] or distributed configuration systems [16], [17] have to be taken into account.

The remainder of the paper is structured as follows. In the background section, the project MC4.0 and the research-led approach is presented, followed by a detailed description of tools and methods applied to overcome the emphasised problems. The paper finishes with a conclusion and an outlook on future work.

2 BACKGROUND

The activities described in this paper take place in the context of MC 4.0, which constitutes a cross-border cooperation between Italian and Austrian partners from industry and academia.

2.1 Overview of the MC 4.0 project

One of the main goals of the project is to establish a communication platform in the area of domotics, consisting of two main parts. The first part is represented by real world support centers at different locations in the project area, the second is an online portal providing the possibility for information exchange and bringing together experts and interested companies working in the field of smart living. The sustainable goal of the project is to support small and medium-sized enterprises to develop mass-customized smart home solutions for their customers, give them instruments to stand out from the competition (specifically from online supply channels) and in this way improve their market position. This is realized by several means, one of the most important ones is to deal with the aforementioned large number of available smart products and the resulting number of combinatorial solutions. The major problem to overcome in this regard is the still insufficient compatibility/interoperability of available smart home systems and their components. Minimal standards, which exist in other areas of ICT (e.g. USB connectors) are hardly existing in the domotic area. These and other problems led to the current situation in the smart home sector in which real smart homes (fully equipped and not only characterized by the presence of single smart components) are far behind the estimations. Although the growth rates are high, the spread of smart homes is not significant when compared to other technologies. The concept and term *smart home* has been announced in 1984, at about the same time when the personal computer was introduced. The latter (and its derivates) is present in almost 100% of homes in developed countries, smart home systems are [5] in 2020 still in the single-digit percent range, see Figure 1.

3 ADOPTED APPROACH

The methods applied in MC 4.0 to deal with the related obstacles and challenges are Mass Customization and Configuration. Before going into detail on the application of the methods, the central concepts are defined.

At first it is important to illustrate our understanding of the concept of Smart Home. According to [15], *..a smart home is a home that incorporates advanced automation systems to provide the inhabitants with sophisticated monitoring and control over the building's functions. For example, a smart home may control lighting, temperature, multi-media, security, window and door operations, as well as many* *other functions."* A state-of the art smart home would also enable remote access via internet/smartphone and individual programming. However, most of the off-the-shelf systems do not provide such a level of "real smartness", they hardly integrate different subsystems in a building (e.g. lighting, shadowing and multi-media), provide adequate programming facilities or advanced AI-functions such as learning from user behaviour, adaptive functions, etc.

The next important concept is Mass Customization. Several definitions of Mass Customization exist. We refer to the one of [13], because it describes best what is planned in the course of MC 4.0. According to [13] Mass Customization is the *"...Production of personalized or custom-tailored goods or services to meet consumers' diverse and changing needs at near mass production prices. Enabled by technologies such as computerization, internet, product modularization, and lean production, it portends the ultimate stage in market segmentation where every customer can have exactly what he or she wants."*

The third central concept of MC 4.0 is Configuration, as an enabling technology for Mass Customization. According to [14] *"Configuration is a basic form of design activity where the target product is composed from a set of predefined parts in a way which is consistent with a given set of constraints."* Configurators are applied in a process where users specify, change and tune their requirements and the configuration system provides feedback. The challenge for configuration is that it should, in principle, allow for a combination of all instances represented in a system, more precisely in the system's knowledge base.

Potentials, but also challenges for Mass Customization and Configuration in the smart home domain are manifold. MC 4.0 is focused on two main areas. The first is energy-related aspects in the context of homes, such as the thermal insulation of buildings, the energetic conceptualization of heating and cooling, and other climate/building envelope related aspects. The potentials of configuration in this domain are discussed, for example, in [2].

The other area, which is the focus of this paper, is domotic appliances aka. smart homes. The first important step in the application of configuration technology in the context of smart homes is to utilize the appropriate technological basis, preferably based on state-of-the-art AI. When taking a closer look at the end consumer market of smart home solutions, there are already observable attempts to support consumers in the configuration of individual smart homes by utilizing AI. As example, Busch & Jäger provide tools labelled as smart home configurators on their websites (see [9]). They are based on designing a floor-plan (or uploading an existing one) on the basis of which smart components are proposed that can make the customer's living environment smarter. Merten [11] call their configurator the *Wiser* configurator. The configuration process starts by asking the users a few questions about their needs and goals (renovation or new building, local or remote control, infrastructure to be integrated such as heating, lighting, etc.). The dialog finally results in a list of smart products which the provider considers as useful, related to the selections done by the user in the previous steps of the dialogue. The two examples are representative for the state-of-the-art of similar platforms (mainly websites) labelled *Configurators*. However, to our understanding, the majority of those systems are - in the sense of [14] - recommender systems rather than configurators. They offer pre-defined standard solutions instead of individualized ones combining "all possibly allowed instances" [14], or, in other words, solutions that optimally fit to a consumer's circumstances rather than representing standard product combinations. With a few exceptions (e.g. the possibility to use own floor plans) these systems do not appropriately take into account the current living situation of a user, the infrastructural conditions and the needs the customer wants to have fulfilled by a smart system.

Our approach to configuration is a different one. Our goal is to focus on the end-customers and their living conditions and guide them in the process in order to find appropriate solutions. We refer to an approach which Mayer [4] calls a goal-driven approach where users tell what they want. A critical feature - and relevant research issue - in this regard is the user interface of the configurator [6] which has to provide appropriate customizability (e.g. is available on different platforms), offer appropriate starting points and the possibility of incremental refinement (the latter ideally in a way to support "playing around" with different variants, in a sense of gamification/game-based configuration [3]). Although the end-customer and their needs in the context of configuration is the focus of this paper, the access to the system for the SMEs is of equal importance. Both target groups and their requirements and needs are emphasised in the following.

3.1 End Customer

The end customers have to be "picked up" from where they are; most probably in a situation when they are considering making their living environment smart. For this purpose, our approach is to initially ask a few questions about the customer's goals (which is also done in other concepts, such as the aforementioned [9, 11]). Goals can be, for example, enhancing safety and security, increasing comfort, saving energy, etc. Another step would be to get information about the infrastructural conditions of the living environment that should be made smart. Aspects such as the number of floors, the age of existing infrastructure (wiring, fusebox) are of essential importance for the generation of possible solutions, the configurator would perform. A segment of a possible decision tree and the features addressed is given in Figure 2.

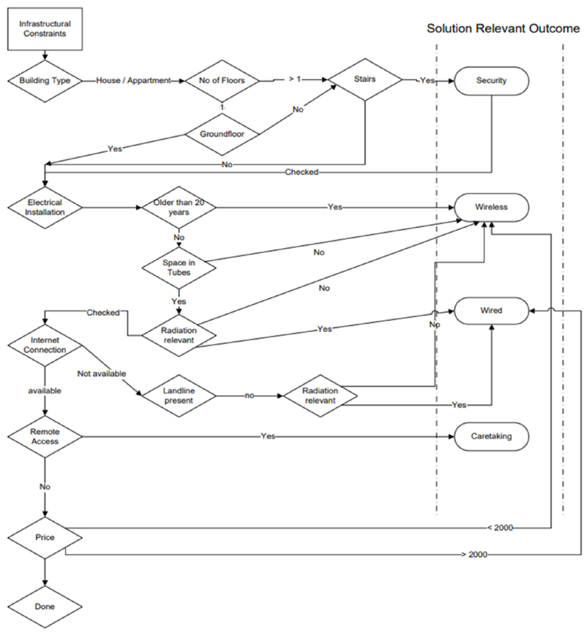


Figure 2. Dialogue decision tree example

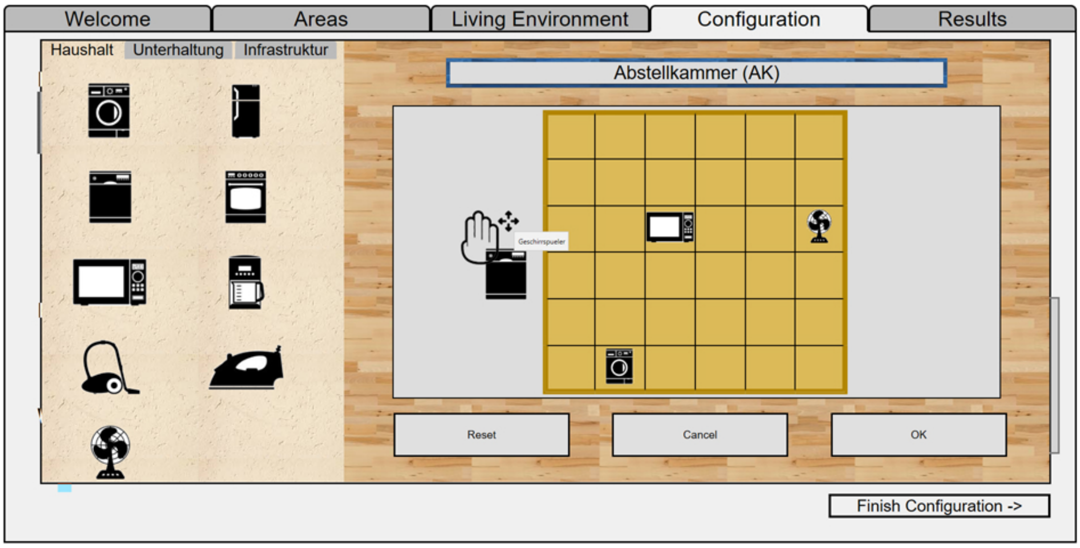


Figure 3. Snapshot of the Configurator Interface Prototype - showing how to place existing components

Another central feature that differentiates our approach from others on the market is that the proposed solutions take into account appliances and components which are already present in the household of question. Information on these aspects can be conveyed to the system by the possibility to drag and drop example components (such as TV sets, light bulbs, radiators, furniture) on the appropriate position of a floor-plan. This step is represented in Figure 3, showing one screen of a conceptual prototype of the configurator to be developed. Competitor systems ask users about needs, partly take into account conditions such as floor plans, but quickly propose smart components whose relation to the user's living conditions and needs are questionable. By asking users more detailed questions about the current situation, the quality of information (what [4] calls semantic information) is considerably higher and the solutions that can be proposed are more appropriate. Additionally, the collected information is more useful for the professional who is utilizing the system as a supplier offering a customized and individual solution to the customer.

Based on the information provided by the customer, the configurator ideally can already pre-calculate appropriate solutions in the back end, or at least propose suitable components to the professional who evaluates the request of a customer. The pre-calculation is based on attributes of appliances that are stored in the system. As an example, when customers state that they would like to increase safety and put a kitchen stove on the floor plan, the stove shall be integrated with an auto switch-off function (either when a smoke detector is triggered or when all inhabitants have left the house). In Austria, kitchen stoves are typically operated by 3-phase 380V current, which would require specific components for the integration in a smart system [7]. However, this knowledge is commonly available and would just have to be entered (e.g in the form of attributes) into the configurator system. Another important category of knowledge to be considered in this context is individual knowledge of a certain professional (electrician, installer). The possibility of entering and sharing (at first in the consortium of MC 4.0, later in a larger community) such knowledge should also be supported by the platform. An example in this regard could be the documentation of the interference problem between the baby-phone and the smart blinds mentioned above in a kind of forum.

3.2 Professionals, SMEs

Most of the configurators available online (such as the two of [9, 11] mentioned above) provide an option to contact a professional who would be able to consult customers in their wish to utilize smart technology. However, this is often based on a switch of tools (Configurator -¿ E-Mail) and efforts that have been invested by the users (e.g. designing a floor plan, answering questions in a dialogue) are not appropriately exploited (as has been emphasized in the previous section). With our approach, users would be enabled to provide information in a significantly higher detail and quality in the system that is also used by the professional. In this way no information is lost and it makes it easier for the professional to calculate more serious price estimations based on more precise information about the technical possibilities of enhancing the living environment with smart technology. Information on the existing infrastructure (as shown in Figure 3), for example, enables the professional to identify misunderstandings or false expectations on the side of the customer and allows for clarification/correction and the proposal of alternative solutions or enhancing features, which offer themselves in the given constellation.

Another difference to other approaches and an important goal of MC 4.0 is to increase the exchange of information and know-how on a regional, cross-border (regions in northern Italy and southern Austria) and local level. Locality is important in general, but specifically turned out to be important during the Covid-19 crisis, when the drawbacks of globalization and the limits of remote service providers were drastically recognizable. It is a problem, when systems do not work appropriately and suppliers have their support facilities online or at remote locations, maybe even overseas. Therefore, support, information exchange and improvement of cooperation in MC 4.0 is not only based on an online platform, but also by offline facilities, the so-called DEA (DEvelopment and Application) Centers, which will be a contact point for end consumers as well as companies working or being interested in the field of smart living.

3.3 Technological considerations

The already mentioned challenge of choosing appropriate technology has been partly addressed in the previous sections, e.g. in connection with interface design issues. In the case of configurators, it is important to take into consideration related work on system interoperability, system configuration (cf. e.g.[10]) and distributed configuration systems (cf. e.g. [16, 17]). A critical part is the backend-system offering a variety and flexibility that is needed for the described purposes. The planned solution is based on the system VariPDM [18], which constitutes a base system that contains the core data model and basic functionality for product life cycle management. Most importantly, in terms of requirements and constraints of the project, the backend-system can easily be integrated/connected to existing technical infrastructures of the involved professionals and SMEs. In this way, efforts for adopting a new ICT system are reduced and the provision of customized products, combined with reduced costs and increased efficiency is realistic. Some of the features of VariPDM are a flexible data model that allows for adaptation during and after the implementation. Items in the system can be either fixed or configurable products, allow for the creation of relationships to establish product structures, and the application of rules (e.g. revision rules). Mechatronic-oriented product management enables the coverage of product data originating from different design disciplines, such as mechanical engineering, electronics and software development, respective attributes can be mixed in one and the same product. This is specifically important in a heterogeneous domain such as smart living. The same flexibility provides for the management of related documents, items and product data, as well as user management.

4 DISCUSSION AND CONCLUSION

In this paper, we have presented the approach, related challenges and problems of applying mass customization and configuration technology in the context of the project MC 4.0 which is located in the domain of Smart Living. The goal of the MC 4.0 project is to enable local SMEs to provide custom Smart Living solutions to their customers, and to support them in the realization of those solutions. This requires considerations regarding the basic technology, not only in the phase of the configuration, which is in the focus of this paper.

A related challenge to be addressed in the course of the project is how to deal with the different requirements of SMEs involved in the project in terms of interfaces to existing ICT-infrastructures, product portfolios and business segments. The platform should, for example, support electricians as well as plumbers, companies offering shadowing as well as those being experts in multi-media solutions (e.g. multi-room audio, projection). Supporting the individual work method is as important as putting a focus on the integration of these different viewpoints to provide a "one-stop-shop" for domotic questions and solutions to the customer.

The future challenge is to also provide support in the phases of implementation and maintenance. A platform that offers itself for this purpose has already be mentioned, Openhab [8], which already supports the integration and combination of around 400 systems and components in the smart home sector. It is the task of the scientific partners, the scientists and researchers involved in MC 4.0 to establish the configurator platform described and the features required on the side of the end consumers as well as the professionals on the one hand, and also support the target group in the installation and maintenance of the resulting solutions on the other. This includes further challenges and questions, for example, how to deal with solutions the configurator would propose, which are based on the combination of components that are optimally covering the needs of a customer but are from different manufacturers. Due to contractual restrictions, an SME might not be able to provide this solution. A possibility to solve the problem could be that the MC 4.0 offers contracting and cooperation alternatives which make possible the implementation of a smart home system consisting of components from more than one manufacturer. Dealing with such problem will be subject of future work in MC 4.0.

ACKNOWLEDGEMENTS

The project MC 4.0 (ITAT1057) is funded by Interreg, Italia-Austria, 2014-2020. We thank the reviewers for their valuable comments and advices on the enhancement of this paper.

REFERENCES

[1] Danbatta, S. J., & Varol, A. (2019, June). Comparison of Zigbee, Z-Wave, Wi-Fi, and Bluetooth Wireless Technologies Used in Home Automation. In ISDFS, 2019, 1-5.

[2] Fuerstner, I., & Anisic, Z. (2009). Adaptive Product Configuration for thermal insulation of buildings. Annals of DAAAM & Proceedings.

[3] Fogli, D., Lanzilotti, R., Piccinno, A., & Tosi, P. (2016, June). AmI@ Home: A game-based collaborative system for smart home configuration. In: Proc. of the int. working conf. on advanced visual interfaces, 308-309.

[4] Mayer, S., Verborgh, R., Kovatsch, M., & Mattern, F. (2016). Smart configuration of smart environments. IEEE Transactions on Automation Science and Engineering, 13(3), 1247-1255.

[5] https://www.statista.com/outlook/283/100/smart-home/worldwide#market-revenue Accessed: 10.06.2020

[6] Leitner, G., Felfernig, A., Blazek, P., Reinfrank, F., & Ninaus, G. (2014). User interfaces for configuration environments. Chap, 8. In Knowledge-based Configuration: From Research to Business Cases, 89-106.

[7] Leitner, G., Felfernig, A., Fercher, A. J., & Hitz, M. (2014). Disseminating ambient assisted living in rural areas. Sensors, 14(8), 13496-13531.

[8] Open Home Automation Bus - https://www.openhab.org/addons/ Last Access: 10.06.2020

[9] Busch & Jäger Configurators 1) https://smart-home-configurator.my.busch-jaeger.de/#/system/welcome, 2) https://www.yoursmarterhome.at/konfigurator/

[10] Kristianto, Y., Helo, P., & Jiao, R. J. (2015). A system level product configurator for engineer-to-order supply chains. Computers in Industry, 72, 82-91.

[11] Merten "Wise" Configurator https://www.merten.de/wiser-konfigurator/

[12] https://www.feelsmart.de/conf/de/basic/das-intelligente-haus/de/KNX/rooms

[13] http://www.businessdictionary.com/definition/mass-customization.html

[14] Falkner, A., Felfernig, A., & Haag, A. (2011). Recommendation technologies for configurable products. Ai Magazine, 32(3), 99-108.

[15] M. Schiefer, "Smart Home Definition and Security Threats," 2015 Ninth International Conference on IT Security Incident Management & IT Forensics, Magdeburg, 114-118.

[16] Kristjansdottir, K., Shafiee, S., Bonev, M., Hvam, L., Bennick, M. H., & Andersen, C. S. (2016, September). Improved Performance and Quality of Configurators by Receiving Real-Time Information from Suppliers. In 18 th International Configuration Workshop (p. 39).

[17] Ardissono, L., Felfernig, A., Friedrich, G., Goy, A., Jannach, D., Petrone, G., ... & Zanker, M. (2003). A framework for the development of personalized, distributed web-based configuration systems. Ai Magazine, 24(3), 93-93.

[18] Variantum https://www.variantum.com/#solutionsmarker

Including Maintenance Services in the Solution Space — Considering Life Cycle Costs in Product Configuration

Nicola Ganter [1] and **Daniel Kloock-Schreiber** [1] and **Paul Christoph Gembarski** [1] and **Roland Lachmayer** [1]

Abstract. For manufacturers of capital goods who aim to bring competitive and profitable products to the market, it is beneficial to optimize the products with regard to the costs that arise over their entire life cycle. Especially in the early development phase, decisions that are primarily made to reduce initial costs must be compared with the resulting follow-up costs of later phases of the life cycle. A significant share of the follow-up costs can be incurred in connection with maintenance and repair activities. In order to influence these follow-up costs in the product design, a suitable maintenance and repair strategy must be selected according to the specific application conditions. According to the chosen strategy, requirements for the product design must be derived. This requires a solution space that represents product and process/service components and their dependencies and parameters. In this paper such a solution space is set up as well as a configurator for its exploration, which enables the automated adaptation of the design to specific application conditions. The configurator contains a knowledge base with information about the geometry, lifetime and price of each component. A configuration regarding maintenance intervals and initial costs, executed with a slider in a CAD software, is carried out using a single-stage gearbox as an application example. Such a configuration is particularly profitable for the design of technical products in product-service systems, especially for use- or result-oriented provider models.

1 INTRODUCTION

An important factor for companies is the increasing demand for customized products and systems, which is accompanied by ever more demanding requirements in terms of performance, costs and delivery times. Therefore it is necessary for companies to design a customized system of products and services in a short time and at the same time optimize time and resources of the design process [4,25,39]. To meet the demand for customized solutions, companies set up mass customization techniques, such as the integration of configuration software (product configuration systems) [3, 24].
At the same time, it can be seen that companies are no longer differentiating themselves only through the physical characteristics of their products [23], which has led to an expansion and strengthening of (after-sales) service activities. With these activities they can already generate at least three times the turnover of the original purchase during the lifetime of the product [6, 31, 49].

[1] Leibniz University of Hannover, Institute of Product Development, email: Ganter@ipeg.uni-hannover.de

1.1 Motivation and Aim

Offering products and services enables manufacturing companies to transform their business model from a pure product supplier to a provider of product-service systems (PSS) [21,36,42]. With PSS the product manufacturer aims to keep not only the production costs of the physical product low, but also the costs resulting from the service part [36]. These costs, which usually occur after product manufacture, are largely determined by decisions in the product development phase [16,43]. For example, maintenance costs in the operating phase can only be influenced to a limited extent, since the main decisions regarding spare parts, required maintenance intervals or dismantling efforts were made in product development [46].
A method to estimate costs incurred during the life cycle of a product is life cycle costing (LCC). The LCC method aims to identify and explain relationships in the generation of costs, which can therefore be systematically influenced in order to optimize the life cycle costs of a system [46]. Therefore, a better integration of information from the different phases of the product life cycle is necessary [40]. Within the scope of LCC assessment, technologies, components or materials are compared to find the most cost-effective solution regarding life cycle costs for the specific operating conditions among the options. An optimum or compromise must often be identified between the mostly opposing investment costs and costs in later phases of the product life cycle, such as operating costs or maintenance costs [46]. After the development of a base product or PSS, the customized product/PSS is specified with regard to the requirements of the customer and the use case [20]. For a cost-efficient specification in the sense of LCC, it is necessary to jointly consider and modify the physical product, services and associated business processes with regard to the specific requirements [41]. However, it is challenging to incorporate information from different phases of the product life cycle and from different business units for the task of specifying a tailored product [20].
The aim of this paper is to present a modeling approach for a configurator that incorporates service-related characteristics and considers the dependencies between characteristics of services and the technical product. This modeling approach for a configurator is profitable on the one hand to specify a product cost-effectively in the sense of LCC and on the other hand to match the product and the service part in a PSS. The modeling approach is based on a data model comprising technical-mechanical and service-related parameters and its representation in a constraint-based model [34, 35]. As a basis of a consistent model for the joint design of product and process, CAD models are suitable, as they are an important tool in product development and contain information of the physical product and the assembly. Furthermore, they provide a great potential for (automated) variant creation through parametric design [13,33] as well as the pos-

sibility to simulate service-related or technical-mechanical properties of the product, e.g. in maintenance ergonomic analysis [40]. The proposed approach is to extend a parametric CAD model from product development with service-related parameters and their relationships to the technical parameters of the physical product, so that both the physical product and the associated services can be configured in terms of technical-mechanical and service-related constraints and requirements, e.g. the desired maintenance interval, maintenance time, operating costs or service life [33]. The output of the resulting costs of the product-service configuration enables the user of the configurator to evaluate the input parameters in a LCC analysis.

1.2 Structure of the Paper

After the introduction and motivation, the further sections of the paper are structured as follows: Section 2 describes the theoretical background to geometry-based solution space modeling in modern CAD systems and the product and process configuration. Furthermore, an overview of the basics of life cycle analysis and maintenance activities, which are relevant for the presented product service configuration, is given. Based on this, Section 3 describes the model of the extended CAD solution spaces in its structure and implementation. This is also illustrated in Section 4, which shows an exemplary implementation of a solution space, as well as a configurator on a gearbox, where an optimization with regard to different maintenance intervals is possible. The paper concludes with a discussion and an outlook on further research potential in Section 5.

2 THEORETICAL BACKGROUND

2.1 CAD-based Solution Space Modeling

A design solution space is the set of all machine systems that fulfill a given set of functions [19], in general a solution space is the totality of all (theoretically) feasible solutions for a task [27]. For business models of mass customization, a stable solution space is necessary from which the customer-specific variant is derived [32]. In the development, a solution space is described because the designer defines not only the product form, but also the variant design and the associated control and configuration concept for the components [17, 37]. To do this, knowledge must be explicitly implemented into digital prototypes. This is made possible in particular by the fact that mathematical and logical constraints can also be defined between the parameters of the models. Compared to rigid (conventional) geometry modeling, a parametric CAD model is able to represent such a solution space [37]. Most CAD systems enable the extension of classic CAD models with parameters, formulas and rules, and it is also possible to link the model to a spreadsheet [44].

In addition to conventional and parametric CAD systems, VDI Guideline 2209 [47] contains two further types of CAD systems that offer additional functionality for creating models with variable geometry and implementing design knowledge (see figure 1). Feature-based systems are an extension of parametric CAD systems. A feature consists of several geometric elements with parametrics and rules of behaviour, so it can be understood as a semantic information object [17]. To a limited extent, features can adapt to their environment. Furthermore, there are Knowledge-based CAD systems that can draw conclusions from current design situations (through geometric and also background information). In general, knowledge-based engineering (KBE) aims at the automation of routine design tasks. To realize this, two different categories of knowledge must be considered, domain and control knowledge. These are shown in figure 2.

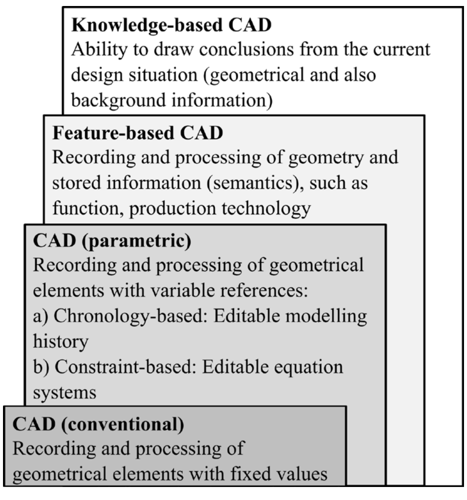


Figure 1. Overview of the principles of 3D modeling [47]

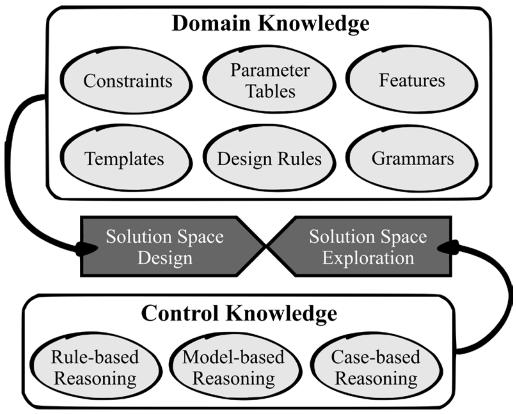


Figure 2. Knowledge Modeling in KBE [35]

Domain knowledge describes a solution space. The solution space is built-up with constraints (e.g. dimensioning formulas that restrict the parameters of the CAD model), templates (as reusable building blocks), parameter tables (for parametric part families such as standard parts), features (a semantic information object, formed by connected geometry elements with parameters and rules of behaviour [45]), design rules (correspond in their structure to an IF-THEN-ELSE construct and are used for the formulation of both domain and control knowledge [18]) or grammars (map related synthesis operations through a vocabulary of elements in combination with a set of rules [5]) [7, 30].

Control knowledge determines the way a solution space is explored. Existing literature refers to inferences and reasoning techniques to adapt the configuration to new or changed requirements. The following three different techniques can be used for this [20, 30]:

- Rule-based reasoning: Knowledge representation is based on IF-THEN-ELSE statements. Rules are executed procedurally, they can execute subordinate rules or delete them from memory to perform more complex tasks.
- Model-based reasoning: The possible solution space is described by a constraint-based physical and/or logical model or the repre-

sentation as allocation and resource consumption.

- Case-based reasoning: Knowledge is not explicitly modeled as a constraint- or rule-based model. The necessary knowledge is stored here in examples (previously approved solutions). A simple case-based reasoning system can retrieve individual cases that already exist, or compile a set of cases that represent the most appropriate case. Sophisticated systems can modify existing cases or mix several cases to adapt them to new situations.

2.2 Configuration of Product and Process

In configuration design, a system is assembled from fully pre-defined elements and linked via interfaces [26]. A product can be regarded as a set of components, and the product configuration can be seen as an attempt to find at least one set of components that meets all constraints and customer requirements (although the solution set can also be empty). This can be supported by a configurator, which consists of a knowledge base (where the generic model of the product is stored) and control knowledge that help in finding solutions or selecting components. According to Aldanondo and Vareilles [1] is the basic requirement in terms of support to ensure that the configured product is consistent with the generic model and requirements, and that all constraints are satisfied.
Configuration problems are typical tasks for constraint-based reasoning [2, 10, 11, 38]. The aim of constraint-based reasoning is to find solutions where all constraints are fulfilled (constraint satisfaction) [29]. In order to solve so-called constraint satisfaction problems, constraint networks are set up and a constraint solver has the task to find solutions considering the given or defined constraints (boundary conditions). If configuration is seen as the solution of a constraint satisfaction problem, it can be extended to a global configuration approach that covers not only the configuration of products but also the cycle of requirement and process configuration. In this way a joint configuration of product and production process can be achieved [1, 11].
For the joint configuration of product and processes, Aldanondo and Vareilles [1] use a domain concept to describe the solution space. The approach describes selectable product features and their characteristics (as the domain of product properties), in the domain of design they have a connection to product components (which realize the features). In the third domain, manufacturing process chains (to which resources such as manufacturing equipment or processing times are assigned) are described, with which the components are produced and connected. In the model, the characteristics, components and process chains are formulated as a constraint network. The approach proposes a two-level modeling method that links the product to an operation level through a routing level. However, this method is also applicable when a multilevel process decomposition is required, so further decomposition steps can be performed and additional domains can be introduced for intermediate process levels. Here, too, the manufacturing operations are modeled as a simple set of resources [1].
In most common CAD systems parameters can be related to each other through logical and mathematical dependencies [12]. To provide a combined configuration of products and processes, a holistic data model is also necessary [34]. While CAD documents normally represent order-neutral data [44], an approach presented by Kloock-Schreiber et al. [22] shows the possibility of extending CAD models based on the stored data of individual product and process models to support the documentation (by storing feedback e.g. on current component conditions), or the execution of service processes (by providing additional information). Service planning and configuration can benefit from such a model by using the information available in CAD about neighborhood relationships (to simulate installation and removal processes and their sequence), number of components, e.g. screws (to estimate and plan the duration of processes) and additional information such as tightening torques (for required tools and ergonomic optimization) or part conditions (for planning the next replacement intervals).
Up to now, research on product and process configuration has mainly concentrated on processes in the early phases of the product life cycle, such as fabrication, assembly or distribution, which are only slightly offset in time from product or PSS development [20]. Moreover, in the literature, products and services/processes are configured simultaneously but often separately or only by one-sided influence of the product configuration on the service/process configuration. A common configuration through a constraint-based approach of the product and services/processes in terms of their requirements, constraints and relations is not yet documented.

2.3 Maintenance Activities

In life cycle management the scope of maintenance activities is not limited to the use phase, but there are close relationships to other phases of product life cycle, such as design, production, and end of life phases. Essential maintenance activities within the design phase are the design for maintainability and the selection of a suitable maintenance strategy, i.e. according to which principles and to what extent maintenance should be performed [40, 46]. Maintenance strategies can basically be categorized according to the factors in figure 3.

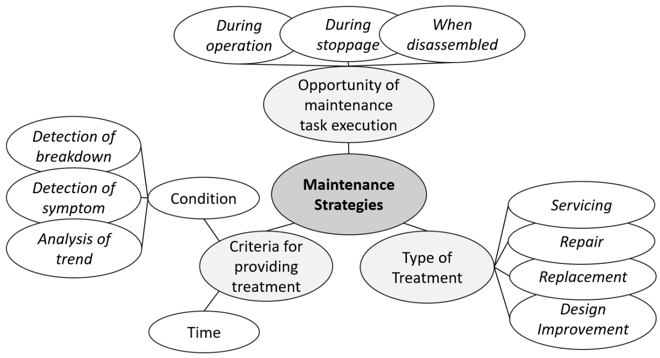


Figure 3. Factors to categorize Maintenance Strategies; according to [40]

The selection of a suitable maintenance strategy from the options in figure 3 is based on both a technological and a managerial evaluation [40]. A decisive factor for the evaluation is the agreed technical availability. According to DIN 31051 [8], availability is defined as the ability of an object at given conditions and when it is required to be in a state in which it fulfills the required functions, presupposing that the necessary external resources are provided. If a product fails to fulfill a specified function a deficiency exists [8,9]. This deficiency, however, only corresponds to a nonconformity if it occurs more frequently than the required availability. Therefore, a required availability is achievable despite the performance of maintenance measures, provided that the maintenance and repair work can be carried out during breaks in operation of a product [48]. Besides availability, the following factors, among others, are decisive for the selection of a suitable maintenance strategy [40, 46]:

- Technical specifications and performance data of the product

- Operating conditions
- Scheduled service life
- Characteristic wear and resulting functional failures (wear and failure analysis)
- Execution of maintenance activities by own employees or externally purchased services
- Applicability of maintenance technologies
- Severity of damage to the system environment
- Probability of total system loss.

Based on the chosen maintenance strategy, the maintenance plan is created which contains the maintenance tasks, such as inspection, monitoring, diagnosis and treatment as well as their scheduling. Afterwards, a prediction of the expected maintenance costs can be given [46].
Therefore, an essential task of product development in the context of life cycle management is providing the design data for the maintenance strategy planning [40]. Furthermore, the maintainability of the design has to be ensured in the product development phase by considering the corresponding design guidelines. Moreover, the required maintenance intervals can be controlled by decisions in product development, e.g. the choice of low-wear materials or technologies allows an extension of the intervals [28]. In addition, the selection or design of critical components can be optimized with regard to the specific application scenario, which includes the maintenance intervals, planned service life and required availability.
In summary, in terms of life cycle management during product development, a suitable maintenance strategy must be selected according to the technical-mechanical and managerial, service-related requirements and restrictions of the specific application. In addition, the physical product must be customized with regard to the selected maintenance strategy.

3 EXTENDED CAD SOLUTION SPACE

In order to facilitate decisions in product or PSS development by means of a life cycle cost assessment, the dependencies between design decisions and service parts from different phases of the product life cycle must be taken into account. To achieve this, the solution proposed in this paper is to extend the solution space by managerial, service-related parameters and the inter-dependencies between these and the technical-mechanical parameters. For the joint configuration of a physical product and associated services, the innovative aspect compared to the existing literature on product process configuration is that the mutual dependencies of technical-mechanical and business, service-related parameters are modelled. In the proposed approach, the dependencies of all parameters are formulated using constraints. The specific application scenario of a product or PSS determines the values or value ranges of certain parameters in the constraint network, for instance the parameters availability, number and duration of operating breaks and maximum installation space in the network in figure 4. By the constraints as well as control knowledge, the remaining parameters in the model are determined. Once all parameters have been assigned, the constraint satisfaction problem is consistently solved. In addition to the configured parameters of both domains, the resulting required resources and life cycle costs are calculated and output. This requires the assignment of resources such as costs, time, or qualification of the executing personnel to the service process chains. If resources impact the life cycle costs, they have to be converted into costs in order to be considered in the LCC assignment.

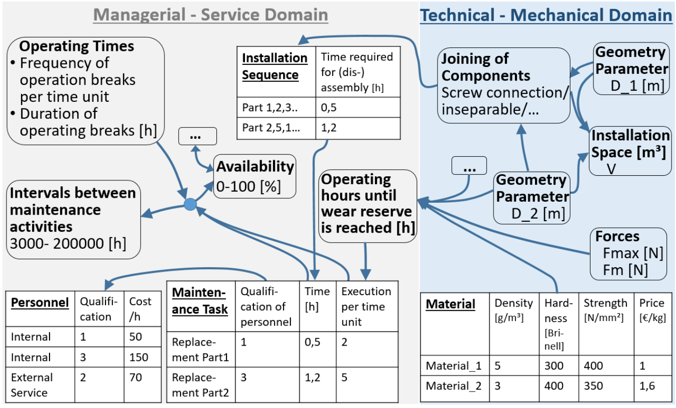


Figure 4. Excerpt of a Constraint Network for a Product-Service System

A knowledge-based CAD assembly model enables the described configuration of a product and its associated services. In the remainder of this paper, the extension of the CAD model of a technical product to a knowledge-based model containing the inter-dependencies of a technical product and service/process relevant characteristics is discussed and exemplified. The extension of the solution space is illustrated using the example of the integration of services in connection with maintenance, repair and overhaul.

3.1 Model Structure

The domain knowledge of a product-service configurator is extended by parameters that formulate service characteristics and their variants, such as the required maintenance interval, the time required for disassembly and assembly, or the service life of a component. These service characteristics are created as user-defined parameters in the CAD assembly model.
Likewise, the domain knowledge is extended by connections between the parameters within the domains and across domains. The inter-domain relationships are formulated by mathematical or logical references in the form of formulas, rules and parameter tables. In the formulas and rules, parameters of both domains can be included to determine the value of a certain parameter. For a common product-service configuration the technical-mechanical parameter tables are expanded by service-related parameters and resources, such as the price of a material or the qualification of the task executing personnel. In this way (product) variants described by the table rows can be selected with regard to service-related parameters and resources. In addition, parameter tables are used as templates for service processes, e.g. for the process steps and resources required to provide spare parts through replacement cycles.
Besides formulas, rules and parameter tables, the domain knowledge contains parametric and feature-based parameter connections within the technical-mechanical domain. The entire domain knowledge is stored in the CAD model or in embedded files of a spreadsheet software. The latter should be chosen for parameter tables with a large amount of data as well as for complicated calculations, such as a lifetime calculation.
In principle, the product and its associated services can be configured with regard to each parameter that is defined in the established constraint network. According to the requirements and restrictions that arise for the product and its associated services in the specific application scenario, technical-mechanical and service-related parameters in the constraint network are assigned concrete values/characteristics or their value ranges are limited. Possible restrictions imposed by the

user may be that the maintenance tasks cannot be performed by the operator's personnel, which technologies are available for repair and the limitations of the installation space.

The parameters set according to the technical-mechanical and service-related requirements and restrictions represent the leading parameters for the rule- and model-based exploration of the solution space. By means of rules, conclusions can be drawn from a leading parameter value. This is especially beneficial for designing the physical product according to the service-related parameters, e.g. for activating features, templates or parts according to the value of a service-related parameter. Furthermore, rules enable parameter values to be determined based on the value of a leading parameter, without the relationship between these parameters being describable by a formula. In addition, for parameters whose relationship is mathematically described, a suitable formula based on the leading parameter can be selected through rule-based reasoning, e.g. according to the required service life, it is selected whether a calculation of load capacity is carried out for service strength or fatigue strength. Through the combination of individual rules, decision trees with nested iterative conclusions are modelled. For example, if the required availability is 100%, components are designed to be very robust and wear-resistant and maintenance tasks are scheduled for operation breaks. If the number or duration of the planned operation breaks is not sufficient, maintenance tasks have to be performed during operation.

As the rules and the constraints are followed, parameters are set up incrementally, thus eliminating degrees of freedom. During this process, the restrictions are considered that have been entered by the configurator user, as well as restrictions resulting from generally valid regulations regarding technical-mechanical and managerial service-related aspects, e.g. from legislation concerning break times of personnel as well as from the mandatory safety factors of components calculated by load-bearing capacity analyses. In addition, restrictions derived from already defined parameters are considered, e.g. the defined handling processes and available transport containers limit the maximum weight of a component [15].

By means of the described design and control knowledge, the product and its associated services are defined jointly with regard to the requirements and restrictions set by the operator. The result for the physical product is output as the configured CAD model. The service-related parameters and resources that are particularly relevant for the operator, such as life cycle costs, are displayed in a message box in the CAD system. The parameters of the associated services are recorded in a spreadsheet software file. There, process flows and work schedules can be automatically generated.

3.2 CAD Environment

To dynamically create and modify an assembly in terms of technical-mechanical and service-related parameters, the software Autodesk Inventor Professional 2020 is suitable [14]. The software enables the implementation of service-related parameters as user-defined parameters and their processing. For a product-service configuration it is beneficial that the input values for parameters can be user-defined text or true / false in addition to numerical values. The software allows to define the relationships of parameters by formulas, equations or by means of features and templates. For standard components and elements, features and templates are provided in the software within a content center. In addition, parameter tables can be used to create part families, so-called iPart factories, containing part variants of different sizes, materials, or mounting configurations.

Rules are formulated in the software using the ilogic programming language in order to draw conclusions based on parameter values. Ilogic further allows values to be read and inserted in Microsoft Excel files that are embedded in the CAD model. The access to implemented MS Excel files enables the input of parameter values from large tables or complicated calculations, whose input values can be parameters of the CAD model. For entering parameter values, the software also provides customizable user interfaces connecting the input to the parameters, properties and rules of an Inventor part, assembly or drawing document.

4 APPLICATION EXAMPLE

This section describes the structure and possibilities of the product or PSS configurator, which takes into account the mutual dependencies of product and service by means of a constraint-based approach. The proposed modelling approach is illustrated using a simple functional unit and the associated maintenance and repair services. In the selected application example, the physical product is the gear unit shown in figure 5, which consists of the components shafts, keys, gearwheels, bearings, casing components as well as retaining rings, rotary shaft seals and screws. The basic function of the unit is to convert an offered torque of a driving machine to the torque and rotational speed required for a driven machine. The service part in the application example covers the planned maintenance tasks.

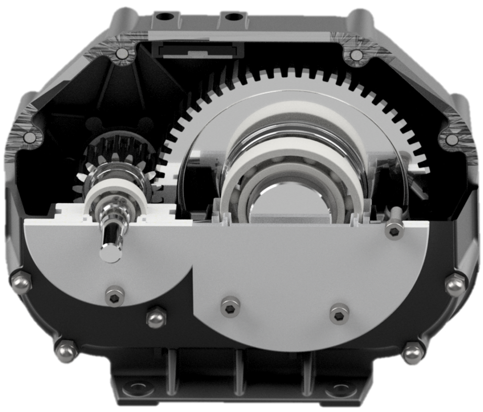


Figure 5. Cutaway view of a gearbox CAD assembly model

The product-service configurator allows the gearbox to be designed cost-effectively in terms of LCC for the boundary conditions in the specific application scenario. These boundary conditions include not only the technical-mechanical requirements and restrictions such as required performance data, environmental conditions, installation space and weight but also service-related aspects, such as desired maintenance interval. The output of the manufacturing costs of a product configuration supports the user in the selection of a suitable maintenance interval in the sense of a LCC evaluation, since the configurator outputs the hard-to-estimate manufacturing costs resulting from the service-related decision of the desired maintenance interval. By varying the input values, the user can compare the maintenance intervals and manufacturing costs of different configurations and select the most suitable configuration for the specific application. In this way, the conflicting goals of high maintenance intervals and low manufacturing costs are resolved.

4.1 Configurator

The solution space of the considered product-service system is implemented in a CAD model of the software Autodesk Inventor Professional 2020 and an embedded Microsoft Excel 2013 file. The solution space contains the geometric parameters that were created during the design of the gear unit. During the modelling the parameters were related to other geometrical parameters by formulas or logical relations. In addition, templates and parameter tables are used to describe the parameters and parameter relationships of parts. Standard parts, such as bearings have been added to the CAD model as templates from the inventor content center, whereby all bearing variants available in the solution space are inserted individually from the content center. Parameter tables stored in the Excel file are used to relate geometric parameters, e. g., to set the width and diameter of the groove of the retaining ring depending on the shaft diameter. This is performed by writing the shaft diameter into a cell in the first sheet of the Excel file using ilogic and searching for the parameter in the table of retaining ring parameters using the VLOOKUP command. The associated parameters groove width and groove diameter are output to the first sheet of the Excel file and read out using ilogic. In addition to individual parameters, parameter tables define the variants of retaining rings and rotary shaft seals, which are stored in the CAD model as iParts.

The solution space further contains user-defined technical-mechanical and service relevant parameters and resources. The parameter tables of the standard components are enhanced by the resources price and weight. For the other components, e. g. the shafts, the volume and mass are automatically calculated in Inventor by defining its material. The part costs is calculated using the mass and the material price per kg, which is stored in a parameter table of the material properties in Excel. According to the following technical-mechanical and service-related user-defined parameters the configuration is carried out: Torque and speed at input and output, the mode of operation of the driving and driven machine and the desired or required maintenance interval of the gear unit. In addition, the user decides whether to optimize the installation space and weight of the gear unit. Due to the defined constraint network, the configuration can also be based on other parameters than those mentioned above. However, the use of a form in Inventor determines which entries the user can make on the interface. The defined user interface used in the configurator is depicted in figure 6.

The solution space is explored rule- and model-based, which is illustrated by the example of the bearings in figure 7. Model-based parameters are defined and templates are activated or deactivated according to the entered parameters through the constraint network. Moreover, the appropriate variants of the iParts are activated model-based. For the iParts retaining ring and radial shaft seal, the variants whose inner diameter corresponds to the shaft diameter are activated. The relationship between the user-defined and the geometric parameters in the CAD model is described by formulas implemented in Excel. The required input parameters for the calculations are written by ilogic into cells on the first sheet of an Excel file. In the further sheets of the Excel file, the formulas for the calculation are stored as well as other input parameters that are required for the calculation, e. g. the physical quantities of a material. The result of a calculation is written into a cell on the first sheet of the Excel file, from where it is read out by ilogic and saved to a parameter in the CAD model. In this way the calculations of safety factors of a keyway according to DIN 743-1 are implemented. Using ilogic, the input parameters for the calculation (shaft diameter, acting loads (bending and torsional

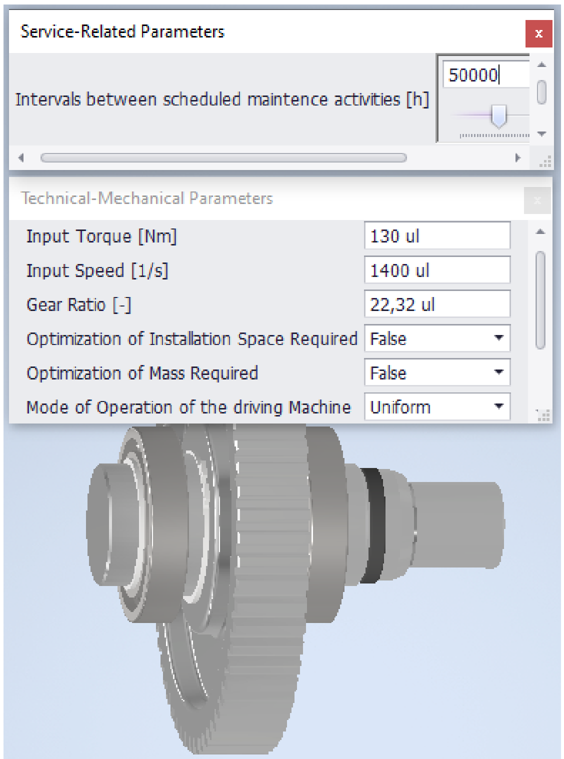


Figure 6. Configuration of Gearbox Components with regard to the desired Maintenance Interval and technical-mechanical Parameters

moment) and the material) are written into the Excel file and the calculated safety is read out. A rule is used to examine whether the calculated safety exceeds a given safety. If this is not the case, the shaft diameter is increased and the calculations are repeated. The input parameters of the calculations implemented in Excel can also be service-related parameters. For example, the desired/required service life is included in the calculations for selecting a suitable bearing. Furthermore, rules are used to set a mathematical relation between two parameters in dependence on a parameter value, e.g. that the shaft diameter D1 on which the gear is located is calculated for $D2 <= 50mm$ according to $D2 + 9mm = D1$ (with D2=shaft diameter on which one of the bearings is located) and for $50mm > D2 <= 80mm$ according to $D2 + 11mm = D1$.

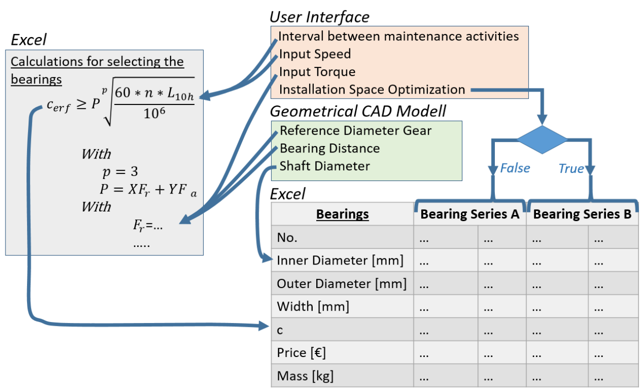


Figure 7. Excerpt from the Constraint Network of the Application Example

Rule-based, the user inputs “Installation Space Optimization Required” or “Weight Optimization Required” are considered in the configuration. The installation space optimization is achieved by selecting a special bearing series whose width and outside diameter are smaller than those of the standard bearing series. Due to the user input “Weight optimization required”, a material with a lower specific density than the standard material is selected for the shafts and gears. In addition, a gearwheel is selected which has a lightweight construction due to cut-out cheeks. The choice of installation space or weight optimization leads to higher manufacturing costs, which are considered through the material price per kg and varying prices for the gear and bearing variants.
After successful modification of the CAD model according to user input, the weight, installation space and manufacturing costs of the configuration are displayed in a message box.

4.2 Usage of the Configurator

The described configurator allows a gearbox to be configured to a desired maintenance interval in addition to the technical-mechanical requirements of the application. Thereby, information from later phases of the product life cycle is integrated into the product development. For example, the configurator enables a gearbox to be tailored for use in a continuous process in which it must meet high demands on availability and in which failure of the gearbox leads to high downtime costs. Furthermore, customization can take into account that the gearbox is difficult to access for maintenance, e.g. because of long distances that maintenance personnel and the required spare parts have to travel. Under these conditions, the most cost-effective configuration of the gear unit in terms of LCC will probably be a configuration with regard to long maintenance intervals using low-wear and long-life components. The decision which maintenance interval is most suitable for the application case is made within a LCC analysis, in which an optimum has to be found with regard to the conflicting goals of high maintenance intervals and low manufacturing costs. The output of the production costs in the configurator creates a database that supports the user in decision-making based on a LCC analysis.

5 CONCLUSION AND OUTLOOK

In this paper an approach is proposed to extend an existing parametric CAD model from the development of a product to a constraint-based model, which contains besides the technical product characteristics also service relevant characteristics and their mutual dependencies. The extended model enables the specification of a product or PSS to meet the requirements of the specific application scenario and customer needs with a high degree of product customization. The proposed approach is illustrated for the CAD model of a gear box and services related to maintenance. With the built tool the gearbox can be configured in terms of technical-mechanical and service-related parameters, such as desired maintenance interval, allowing information from different phases of the product life cycle to be considered in product and process design. The configurator was implemented based on a CAD model in Autodesk Inventor Professional 2020 containing an embedded file from the spreadsheet software MS Excel. In principle, an interface between the CAD system and a spreadsheet software is required to implement the proposed approach, so that parameters from calculations and parameter tables can be read from the spreadsheet software.
In contrast to previous studies on a common product and process configuration, a constraint-based approach creates not only a simultaneous but a joint configuration of products and services/processes. In a constraint-based approach, the configuration of product and service is not only based on a common database of customer-specific constraints and requirements as in [22], but product and service are matched. Therefore, the proposed modelling approach enables a high degree of product customization due to the extended solution space.
On the one hand, this approach is helpful for decision-making through LCC analysis in product customization, since the modelling approach relates services/processes of different phases of the product life cycle to technical characteristics of the product. On the other hand, the proposed modelling approach allows to align the product and service part in a PSS, enabling a high degree of customization and a cost-efficient design of the physical product and the service part of a PSS.
Further work is required to adapt the product in the application example stronger to its associated services and establish a larger database for LCC analysis. Therefore, additional service-related parameters and resources have to be integrated in the model, such as parameters that allow a configuration with respect to the time available for maintenance work. This includes the time for disassembly/assembly, which is determined by the assembly sequence and the time for (dis)assembly of individual parts. The extension of the constraint network by these parameters would, for example, support a LCC analysis regarding the question whether the design of six screws (with standard hexagon and a screw-in time of 60 seconds) or three more expensive screws (with special head and a screw-in time of 15 seconds) on the bearing pot is the more cost-effective solution for the present application. Furthermore, an implementation of further parameters regarding the manufacturing process will allow to increase the prediction accuracy for the manufacturing costs of the gearbox, since besides the pure material costs for the gears and shafts, the machining costs can also be taken into account. In addition, the implemented constraint network could be extended by service-related parameters as well as managerial decisions in order to suggest a suitable maintenance strategy for the physical product according to the customer-specific use. If service processes are considered more detailed in the configurator, the output of the configured processes should be implemented. The service processes can be output in an embedded file of a spreadsheet software, where the generation and output of work plans can be implemented as well.
For the proposed extension of a CAD model by service-related parameters and the constraints to technical-mechanical parameters, empirical knowledge is necessary to formulate these inter-dependencies explicitly. If the constraints cannot be formulated based on empirical knowledge, it is partly possible to determine them by simulations with the CAD model, e.g. a deterioration analysis provides information about the “operating hours until the wear reserve is reached”. Another way to provide the parameter constraints is to build up a case base of service (sub)processes or existing design solutions that are suitable for specific service processes. In this way, appropriate cases can be found even for fuzzy requirements.
By jointly configuring product and process parameters with regard to defined requirements and restrictions, information from different life cycle phases is integrated into the product and process design to support a configuration in terms of life cycle costs. Further work is required to configure systematically the most cost-efficient variants in the sense of LCC with regard to the customer-specific requirements, in other words to achieve the global optimum instead of a local one. For this purpose, optimization algorithms must be implemented in the configurator.

ACKNOWLEDGEMENTS

This research was conducted in the scope of the research project "RePARE- Regeneration of product and production systems through additive repair and refurbishment" (Funding reference number 033R229) which is funded by Federal Ministry of Education and Research (BMBF) within the funding measure "Resource-efficient Circular Economy – Innovative Product Cycles" (ReziProK).

REFERENCES

[1] M. Aldanondo and E. Vareilles, 'Configuration for mass customization: how to extend product configuration towards requirements and process configuration', *Journal of Intelligent Manufacturing*, **19**(5), 521–535, (2008).

[2] R. Barták, M. A. Salido, and F. Rossi, 'Constraint satisfaction techniques in planning and scheduling', *Journal of Intelligent Manufacturing*, **21**(1), 5–15, (2010).

[3] P. Blazek, M. Partl, and C. Streichsbier, *Configurator Database Report 2014*, Lulu. com, 2014.

[4] J. P. Cannon and C. Homburg, 'Buyer–Supplier Relationships and Customer Firm Costs', *Journal of Marketing*, **65**(1), 29–43, (2001).

[5] A. Chakrabarti, K. Shea, R. Stone, J. Cagan, M. Campbell, N. V. Hernandez, and K. L. Wood, 'Computer–Based Design Synthesis Research: An Overview', *Journal of Computing and Information Science in Engineering*, **11**(2), (2011).

[6] M. A. Cohen, N. Agrawal, and V. Agrawal, 'Winning in the aftermarket', *Harvard business review*, **84**(5), 129, (2006).

[7] J. J. Cox, 'Product Templates - A Parametric Approach to Mass Customization', *CAD Tools and Algorithms for Product Design*, 3–15, (2000).

[8] DIN. 31051: Fundamentals of maintenance, 2019.

[9] DIN EN. 13306: Maintenance - Maintenance terminology, 2018.

[10] A. Felfernig, L. Hotz, C. Bagley, and J. Tiihonen, *Knowledge-Based Configuration: From Research to Business Cases*, Newnes, 2014.

[11] P. C. Gembarski, *Komplexitätsmanagement mittels wissensbasiertem CAD: Ein Ansatz zum unternehmenstypologischen Management konstruktiver Lösungsräume*, Ph.D. dissertation, Leibniz University Hannover, 2018.

[12] P. C. Gembarski, 'Three Ways of Integrating Computer-Aided-Design and Knowledge-Based-Engineering', in *Proceedings of the Design Society: DESIGN Conference*, (2020).

[13] P. C. Gembarski and R. Lachmayer, 'Product-Service-Systems - What and why Developers can learn from Mass Customization', *Enterprise Modelling and Information Systems Architectures*, **13**, 1–16, (2018).

[14] P. C. Gembarski, H. Li, and R. Lachmayer, 'KBE-Modeling Techniques in Standard CAD-Systems: Case Study-Autodesk Inventor Professional', in *Managing Complexity*, 215–233, Springer, (2017).

[15] P. C. Gembarski and R.Lachmayer, 'Solution Space Development: Conceptual Reflections and Development of the Parameter Space Matrix as Planning Tool for Geometry-based Solution Spaces', *International Journal of Industrial Engineering and Management (IJIEM)*, (2018).

[16] T. Günther, 'Life Cycle Costing', *WISU Das Wirtschaftsstudium*, **10**, 900–912, (1997).

[17] M. Hirz, W. Dietrich, A. Gfrerrer, and J. Lang, *Integrated Computer-Aided Design in Automotive Development*, Springer, 2012.

[18] A. A. Hopgood, *Intelligent systems for engineers and scientists*, CRC press, 2012.

[19] V. Hubka, *Theorie Technischer Systeme: Grundlagen einer wissenschaftlichen Konstruktionslehre*, Springer-Verlag, 2013.

[20] L. Hvam, N. H. Mortensen, and J. Riis, *Product Customization*, Springer Science + Business Media, 2008.

[21] E. Keller, 'Customer Support: Delivering Enhanced Services through Intelligent Device Management', *Service Business Magazine*, (2003).

[22] D. Kloock-Schreiber, L. Domarkas, P. C. Gembarski, and R. Lachmayer, 'Enrichment of Geometric CAD Models for Service Configuration', in *21th International Configuration Workshop*, p. 22, (2019).

[23] Y. Koren, *The Global Manufacturing Revolution: Product-Process-Business Integration and Reconfigurable Systems*, Wiley series in systems engineering and managements, Wiley, Hoboken, 2010".

[24] K. Kristjansdottir, S. Shafiee, and L. Hvam, 'How to identify possible applications of product configuration systems in engineer-to-order companies', *International Journal of Industrial Engineering and Management*, **8**(3), 157–165, (2017).

[25] M. Krömker, K. D. Thoben, and A. Wickner, 'An infrastructure to support concurrent engineering in bid preparation', *Computers in Industry*, **33**(2-3), 201–208, (1997).

[26] N. R. Milton, *Knowledge Technologies*, volume 3, Polimetrica sas, 2008.

[27] J. Ponn, 'Systematisierung des Lösungsraums', *Handbuch Produktentwicklung*, 715–742, (2016).

[28] J. Reichel, G. Müller, and J. Haeffs, *Betriebliche Instandhaltung*, Springer, 2018.

[29] S. Rude, *Wissensbasiertes Konstruieren*, Shaker, 1998.

[30] D. Sabin and R. Weigel, 'Product configuration frameworks - a survey', *IEEE intelligent systems*, 42–49, (1998).

[31] N. Saccani, P. Johansson, and M. Perona, 'Configuring the after-sales service supply chain: A multiple case study', *International Journal of production economics*, **110**(1-2), 52–69, (2007).

[32] F. Salvador, P. M. De Holan, and F. Piller, 'Cracking the Code of Mass Customization', *MIT Sloan management review*, **50**(3), 71–78, (2009).

[33] D. Schreiber, P. C. Gembarski, and R. Lachmayer, 'Datamodels for PSS Development and Configuration: Existing Approaches and Future Research', *World Conference on Mass Customization, Personalization and Co-Creation (MCPC 2017)*, **9**, (2017).

[34] D. Schreiber, P. C. Gembarski, and R. Lachmayer, 'Modeling and configuration for product-service systems: State of the art and future research', *International Configuration Workshop (CWS 2017)*, **19**, (2017).

[35] D. Schreiber, P. C. Gembarski, and R. Lachmayer, 'Developing a constraint-based solution space for product-service systems', *International Conference on Mass Customization and Personalization - Community of Europe (MCP-CE 2018)*, **8**, (2018).

[36] G. Seliger, A. Buchholz, and W. Grudzein, 'Multiple usage phases by component adaptation', in *Proceedings of the 9th CIRP International Seminar on Life Cycle Engineering, Erlangen*, pp. 47–54, (2002).

[37] J. J. Shah, 'Designing with Parametric CAD: Classification and comparison of construction techniques', *Geometric Modelling Proceedings of the Sixth International Workshop on Geometric Modelling*, **6**, 53–68, (2001).

[38] M. Stumptner, G. E. Friedrich, and A. Haselböck, 'Generative constraint-based configuration of large technical systems', *AI EDAM*, **12**(4), 307–320, (1998).

[39] A. Sylla, E. Vareilles, M. Aldanondo, T. Coudert, L. Geneste, and K. Kirytopoulos, 'Customer/Supplier Relationship: reducing Uncertainties in Commercial Offers thanks to Readiness, Risk and Confidence Considerations', in *Advances on Mechanics, Design Engineering and Manufacturing*, 1115–1122, Springer, (2017).

[40] S. Takata, F. Kimura, F. J. A. M. Van Houten, E. Westkämper, M. Shpitalni, D. Ceglarek, and J. Lee, 'Maintenance: Changing Role in Life Cycle Management', *Annals of the CIRP*, **53**(2), 643–656, (2004).

[41] O. Thomas, P. Walter, and P. Loos, *Konstruktion und Anwendung einer Entwicklungsmethodik für Product-Service Systems*, Hybride Wertschöpfung, Springer, Berlin, Heidelberg, 2010.

[42] A. Tukker, 'Eight types of product–service system: eight ways to sustainability? experiences from SusProNet', *Business Strategy and the Environment*, **13**(4), 246–260, (2004-07).

[43] D. G. Ullman, *The mechanical design process*, McGraw Hill Higher Education, 2009.

[44] S. Vajna, C. Weber, H. Bley, and K. Zeman, *CAx für Ingenieure: eine praxisbezogene Einführung*, Springer-Verlag, 2009.

[45] VDI. 2218: Information technology in product development - Feature Technology, 2003.

[46] VDI. 2884: Purchase, operating and maintenance of production equipment using Life Cycle Costing (LCC), 2005.

[47] VDI. 2209: 3D product modelling, 2009.

[48] VDI. 3423: Technical availability of machines and production lines - Terms, definitions, determination of time periods and calculation, 2011.

[49] R. Wise and P. Baumgartner, 'Go downstream', *Harvard business review*, **77**(5), 133–141, (1999).

Toward Data-Driven Modeling of Configurable Products

Albert Haag[1]

Abstract. Modeling, including knowledge-based modeling, is a major cost factor when realizing product configuration in a business. A product model often represents a considerable investment, best not to be lost even as configuration technology advances. This is particularly true where models encode a lot of business or engineering data.The specific representation used in modeling is determined by the configurator tools a business deploys. In practice, this makes a model proprietary to the configuration environment, which has the major disadvantage of tying the investment in modeling to the chosen configurator. Given the increasing diversity of business environments, configuration environments need to evolve rapidly. Being able to *"model once, configure anywhere"* would be a highly desirable objective. *Data-driven modeling* is a step in this direction. In this paper data-driven modeling is taken to mean making relevant content of a model externally accessible and modifiable in tabular form. Tables can be considered a non-proprietary, easily understood form of representation. They can be managed and queried using established tools, such as spreadsheets or conventional databases. The intention of this short paper is to explore in more detail what a move towards data-driven modeling might entail, and to generally provoke *"out-of-the-box"* thinking about modeling configurable products. The paper is based on personal experiences of the author; a more thorough and complete compilation of ideas and references is ongoing and must follow in a later work.

1 INTRODUCTION

The basic motivation for this short paper are current developments in the areas of *CPQ* (*Configure Price Quote*) and *Mass Customization*. The views expressed in this paper are based on my (the author's) personal experiences. For grammatical simplicity, I use the first person singular to make this clear[2]. The term *configuration* is used in the same way as in [8].

The representation used for a model[3] is determined by the chosen configuration technology, specifically by the configurator tools a business deploys. In practice, this makes a model proprietary to the configuration environment used, which can bring with it the following issues:

- The need for skilled specialized modeling experts, which is both a cost factor and a bottleneck.
- Re-use of common content between distinct but related models is not well supported.
- Keeping different models of the same product aligned can be tedious and costly; particularly, if different configurators are used (e.g. a sales model versus a manufacturing model of a product).
- Switching configurator technology entails remodeling.
- Queries to a model are possible only via the configurator. Integration with other business processes such as shipping and invoicing requires special consideration.
- Increasingly, communication between the configurator and external graphical tools (e.g. CAD systems) is a central requirement. Synchronization between a configuration model and a graphical representation of a product is a challenge of its own, which is facilitated if both keep the relevant content in tabular form.
- Models are versioned as a whole in many current configuration environments. Being able to separately version content that changes frequently is highly desirable. Model content in tabular form can be modified and versioned easily, independent of any model which makes use of it.

We may observe ongoing improvements to individual commercial modeling environments in response to requirements by their users[4]. In this paper, I intend to address the issues of modularity and proprietariness from a general data-driven perspective:

- Exposing content that is subject to frequent, independent change in tables is one way to improve the modularization of a model.
- Where there is a standard way of exchanging tabular data between different configurators, such data can be seen as non-proprietary.

If one distinguishes "sustainable innovation" from "disruptive innovation", then the ongoing efforts to continuously improve modeling tools fall under "sustainable innovation", whereas I see efforts to make configurators more open as "disruptive innovation". My proposals in this paper for data-driven modeling fall into the latter category.

The content of this paper is organized as follows: Section 2 serves to establish a common understanding of the setting and its relevance. Section 3 describes the potential of tables for configuration in themselves. As one of the problems is that relational tables don't scale with an increasing number of product features, possibilities of extending the tabular paradigm to non-relational tables is discussed. The proposal on how to better enable data-driven modeling is the subject of Section 4. Several open challenges are posed in Section 5, and Section 6 contains an outlook on other and future work.

2 EXPECTATIONS AND THEIR LIMITS

2.1 Declarative and Procedural Paradigms

A model that in effect implements an executable program is termed to be *procedural*. A set of IF, THEN, ELSE rules to be processed in a

[1] PMH - Product Management Haag GmbH, Germany, email: albert@product-management-haag.de

[2] This also avoids a mistaken interpretation of using the first person plural as *"pluralis majestatis"*. The first person plural is used to either include co-authors where previous work is referenced or where there is intent to include the reader.

[3] The term *model* is taken to be synonymous with *knowledge base* where modeling is knowledge-based.

[4] An assessment of commercially available CPQ systems is published by Gartner [9].

particular order is procedural by this definition. In contrast, a model that contains only a description of objects and their relationships, such as might be expressed in predicate logic, without any demand on the order in which this information is later processed, is termed to be *declarative*. Declarative models are easier to verify than procedural ones. One currently dominant paradigm in product configuration is to apply constraint programming techniques [20], as evidenced in [8]. This is a declarative paradigm, as the problem definition of a CSP (*Constraint Satisfaction Problem*)[5] says nothing about how to best solve it.

Notwithstanding this general bias toward declarative modeling, we must keep in mind that various different configuration settings may each pose their own problem solving and representational challenges. For example, in previous work [17, 11] we have argued that the real problem underlying configuration in many cases is not solving a CSP, but providing decision support in the context of a dynamic multi-criteria optimization problem[6]. We may observe that it is not hard to find a valid configuration when shopping for a car, but it may be hard to find one that has all of our desired features. While declarative modeling is an ideal, many models will be hybrid in practice, i.e. contain some procedural elements as well. The placement of components in racks, for example, is sometimes most easily handled by a (perhaps pre-existing) program that encodes the way an engineer would decide. Another motivation for using procedural rules is to provide strong user guidance for the configuration process.

2.2 Use of Standards in Product Modeling

Various proposals for standardizing the representation of product models have been made, primarily from an academic perspective. One approach is to extend the scope of CSP problems to be able to more completely represent realistic problems and to define a machine readable format for this. The XCSP 3.0 format [4] exemplifies this approach. Versions of XCSP have been used for disseminating CSP problems for benchmarking and research collaboration in the academic CP (*Constraint Programming*) community, as evidenced by [21]. Another proposal is to make use of a standard UML (*Unified Modeling Language*) representation in product modeling [7].

In the business community, relational database tables are a widely understood form of representing business data in general. Although there are some differences between different *Database Management Systems* (DBMS), the *Structured Query Language* (SQL) [22] forms a kind of standard in practice. In this sense, relational databases can be considered as a non-proprietary form of representing data. Moreover, relational tables relate naturally to constraints in a product model. In a classic CSP with a finite number of variables with finite domains, any constraint relation could be represented in its extensional form as a table, subject to size limitations. Such a table could also be browsed and maintained without a DBMS using a spreadsheet. Conversely, a table listing valid combinations of product features can be interpreted as a constraint. The ubiquitous benchmark model of a Renault Megane (*RM model*) [21] is an example of a realistic model that can be represented entirely in 113 relational tables (see [16] for a more thorough discussion). Where configurator systems support importing or referencing tables as constraints, and where it is possible to represent a constraint extensionally in a database table, such tables can be seen as standardized elements of the product model. The modeling effort reduces to declaring the constraints and pointing each to a table defining its content.

Outside academia, modeling remains largely proprietary in commercially deployed configurators. This may be due to the fact that these configurators evolve in response to the diverse needs of their clients. These customer requirements transcend the scope of any one academic community and more directly address problems that arise from practical business considerations.

2.3 Tried and True Modeling Aids

2.3.1 *Spreadsheets*

Let us first note that the use of spreadsheets in modeling is wide-spread[7] and must be included as a *tried and true* modeling aid. Spreadsheets are well suited to manage relational tabular data and can be treated as a database. As opposed to a DBMS, however, they also lend themselves to more generally maintaining data in tabular arrangements. The data in a spreadsheet cell can be interpreted in custom ways. One established and widely used feature is to store a list of values in a spreadsheet cell, interpreted as a subdomain of a CSP variable.

Spreadsheets typically allow programming add-on functionality, which can be useful to implement custom approaches to modeling, perhaps unique to a particular enterprise. This includes custom interfaces for exchange with the modeling environment of the configurator in use. The organization of data in a spreadsheet document into *sheets* and *workbooks* naturally supports versioning and modularization of the data, independent of that offered by the configurator.

2.3.2 *Databases*

Databases offer much of the same functionality as spreadsheets. They are better able to deal with large tables and offer better data safety. In many businesses, bulk data that is central to a configuration model, such as the specifications of components that appear as parts in a product model, may already be available in database tables. On the other hand, when compared to spreadsheets, a DBMS has a larger footprint, and the use of SQL, while wide-spread, requires a steeper learning curve. Also, the strict relational paradigm is less easily adapted to custom approaches to modeling.

2.3.3 *Variant Tables*

The *SAP Variant Configurator* (SAP VC) family of configurators [10] has provided a transparent use of tables in product models from its inception in the early 1990s. *Variant tables* are used to list valid combinations of product features. They are modeling elements that can be referenced in rules and constraints in a product model[8]. Their content can be maintained independently from other elements of the

[5] See [20] for a definition of a CSP and on constraint programming in general. One chapter of the book deals specifically with configuration [18].

[6] I take *dynamic optimization criteria* to mean that the utility functions used to guide optimization cannot always be defined in advance as part of the model. For example, the willingness to take risks when configuring a mortgage may vary greatly even during a single configuration setting.

[7] There is a slew of links on the internet on spreadsheet-based modeling; search for "excel-based product configuration", for example.

[8] It is possible to import the 113 RM tabular constraints directly as SAP variant tables for use by the SAP Variant Configurator (*SAP VC*). The import of a table from a spreadsheet via the standard transactions CU60E and PMEVC presupposes that variant table declarations are already in place (*"Some modeling required"*). Members of the SAP centric CWG (*Configuration Workgroup*) [5], can view details on running the RM model in the SAP VC in [14, 15].

model. The motivation for companies to make use of this in their configuration projects is two-fold[9]:

- It allows product data engineers to maintain critical parts of a product model without assistance from modeling specialists.
- It is possible to update the model dynamically with rapidly changing information, such as pricing or availability data by just updating the content of tables.

Use of variant tables in conjunction with constraints is seen as a "modeling best practice" [6, 2]. However, the use of variant tables is not limited to constraints. They can also be referenced in *procedures* (a form of procedural rules). Already since the mid 1990s, the SAP VC supports the use of variant tables with multi-valued cells and cells containing numeric, real-valued intervals.

In this paper, I will use of the term *variant table* to generically refer to lists of valid (or invalid[10]) combinations of product features, either explicitly as a relational table, or in a suitably compressed form, e.g. allowing multiple values and/or intervals in a cell.

3 THE TABULAR PARADIGM

The conceptually easiest case in product configuration is when all variants of a configurable product can be enumerated as a catalog in one relational table. In that case, the table represents the one and only constraint in the model. Constraint solving reduces to querying this table. Queries can be formulated in SQL (*Structured Query Language* [22]) by any part of the business needing access to the model.

There are several reasons why this is not generally feasible in practice, foremost size and expressiveness. A simple product like a T-shirt with a modest amount of choices for *color*, *size*, *style*, etc. can already lead to a catalog of hundreds of millions of variants, which cannot be feasibly listed in a relational table. Also, an exhaustive list of all combinations is not possible in relational form if some domains are infinite. While it is not necessarily a goal to construct an overall catalog of all variants, the same problems can arise in conjunction with tables representing individual constraints.

The use of multiple values and intervals in cells in variant tables is a practical response to these problems. The challenge in such extensions is to keep the modeling advantages of tables intact. In previous work [12, 16], we have argued that an SQL-like interaction with tables can be preserved under various forms of table compression. Moreover, we may expect such compression to scale with increasing choices for variant tables, due to expected regularities in the feature combinations they express. The size of a compressed product catalog can actually be seen as a measure of the product's complexity [12].

3.1 C-Tuples

One approach at table compression is to partition the rows of a table into blocks of unconstrained subsets termed *c-tuples*. The following example is taken from [16]. Table 1 has two c-tuples that represent the eleven combinations (rows) of Table 2. The first c-tuple represents the first three combinations. The second c-tuple the remaining eight.

Table 1: Classic T-shirts in C-Tuples

Imprint	Size	Color
MIB	S;M;L	Black
STW	M;L	Black;White;Red;Blue

Table 2: Simple T-shirt table of variants

Imprint	Size	Color
MIB	S	Black
MIB	M	Black
MIB	L	Black
STW	M	Black
STW	M	Blue
STW	M	Red
STW	M	White
STW	L	Black
STW	L	Blue
STW	L	Red
STW	L	White

C-tuples harmonize naturally with constraints. Let S_{ij} be the set of values recorded in the i-th row of the j-th column of Table 3 (e.g. $S_{12} = \{S; M; L\}$ and $S_{23} = \{Black; Red; White; Blue\}$ in Table 1). Table 3 is a table of n abstract c-tuples, which can be directly interpreted as the logical expression (1):

Table 3: C-Tuples table

x_1	x_2	$\dots$	x_k
S_{11}	S_{12}	$\dots$	S_{1k}
S_{21}	S_{12}	$\dots$	S_{1k}
$\dots$	$\dots$	$\dots$	$\dots$
S_{n1}	S_{n2}	$\dots$	S_{nk}

$$\begin{aligned}
&(x_1 \in S_{11} \wedge x_2 \in S_{12} \dots \wedge x_k \in S_{1k}) && \vee \\
&(x_1 \in S_{21} \wedge x_2 \in S_{22} \dots \wedge x_k \in S_{2k}) && \vee \\
&\dots && \vee \\
&(x_1 \in S_{n1} \wedge x_2 \in S_{n2} \dots \wedge x_k \in S_{nk}) && (1)
\end{aligned}$$

3.2 C-Tuples in the Wild

A c-tuple can be represented transparently in a spreadsheet, by placing more than one value in a cell. A row in the spreadsheet is then taken to represent the Cartesian product of all combinations that can be formed by picking one value from each cell, c.f. Tables 1 and 2.

Support of c-tuples in SAP variant tables was due to early demand by clients. C-tuples were not, however, considered a *modeling best practice* [3]. The reason for this is probably that they are hard to maintain without additional tool support. They are being used, however, as noted in [16], and they are important as increasing variability in products leads to an exponential increase in valid combinations, no longer representable in relational form. I have reason to believe that the use of c-tuples in custom approaches to modeling with spreadsheets is common but further investigation would be needed to confirm this.

The proximity of a table comprised of c-tuples such as Table 3 to a constraint as shown in expression (1) suggests that it should be easy for any constraint-oriented configurator to import constraints from tables compressed to c-tuple format.

[9] I first saw this technique of deliberately farming out those parts of the model subject to frequent change to variant tables in a customer solution using the SAP OS/2 configurator [10] (the precursor to the SAP VC) already 25 years ago.

[10] A variant table listing invalid combinations is termed a *negative* variant table. Being able to formulate exclusions is an important customer requirement. However, since negative tables do not directly map to constraints, I do not include them in the discussion in this paper.

There is no standardized way of representing c-tuples in databases[11].

3.3 Added Expressiveness

When the same set of values occurs in multiple c-tuples in a variant table then it makes sense to refer to it by a symbolic name as shorthand. We made use of this feature in the T-shirt example in [16]. Symbolic names were there defined for *standard colors*, *adult sizes*, and *vintage imprints*. Particularly frequent in practice is the case that a cell in a c-tuple encompasses all values possible for that column, and to use a wildcard symbol such as ("$*$") as an abbreviation for this. It also makes sense to refer to the complement of a set using a negation operator such as ("$\neg$") where this results in a shorter expression. For example, in Table 1 the set of sizes $\{M; L\}$ might be referred to by $\neg S$ (i.e. "not small"). Using both symbolic names for sets and the negation operator, Table 1 can be reformulated as Table 4, which is close to a logical formulation of two constraints stating that the imprint MIB comes only in color Black and the imprint STW does not come in small sizes while retaining its interpretation as a table.

Table 4: T-shirt: C-tuples with wildcard symbols and negation

Imprint	Size	Color
MIB	$*$	Black
STW	$\neg S$	$*$

In my experience, the desire to use wildcard symbols is very common in practice and falls almost naturally within the tabular paradigm. There is no agreement on a standard wildcard symbol, but ("$*$") is a prevalent contender. It is not infrequent to encounter SAP variant tables which are in relational form but contain empty cells, signifying a wildcard. Similarly, using real-valued intervals in table cells is a natural extension of the tabular paradigm possible in both spreadsheets and SAP variant tables. In my observation, the use of intervals is not nearly as prevalent as the use of wildcards.

These approaches can be generalized and formalized. A table will always be comprised of a finite number of symbols, but some of these symbols might refer to sets. We discuss dealing with symbols that refer to infinite sets (*quasi-finite symbols*) in [13]. However, beyond wildcards and the simple use of intervals, some additional meta-data is needed to interpret the symbols. This is a non-trivial extension of the tabular paradigm, as the tables are no longer self-contained.

3.4 Advanced Compression

We have discussed in previous work [12, 16] that stronger, more advanced compression based on *decision diagrams* is possible. We did not focus on this in [16], because standards for storing and exchanging such formats still need to be established. Apart from this lack of general availability and standardization, all remarks made about the utility of c-tuples would apply in even stronger form to this advanced compression as well. Indeed, some features, such as sorting a table may be easier using this advanced compression.

[11] Some databases such as Microsoft Access may actually support multi-valued fields (cells) in a similar way to spreadsheets, but this is outside of the relational paradigm and not a standard.

4 DATA-DRIVEN MODELING

The essence of data-driven modeling is to model in a way that relevant parts of a model, particularly those subject to frequent change, are expressed as tables where possible. Changes and queries to such tables can be made by product engineers without special training in modeling. This is already a proven strategy to some extent[12]. Data-driven modeling does not entail switching modeling paradigms, but simply places more stress on a conscious use of tables[13].

What I propose here, is an attempt to formulate common conditions and expectations that will enable the use and re-use of tables in multiple different product models and between different configurator environments. It can be understood as a proposal for a minimal standard for variant tables, seen generically.

Agreement on such a standard would not only facilitate the sharing and exchanging of relevant parts of a model, but would also enable enterprises to standardize their use of modeling aids such as spreadsheets to handle tables in a common way. It is also desirable that data already present in tables can be incorporated into a product model with minimal effort. Furthermore, a standard SQL-like interface to tabular content in a model would facilitate the integration with other applications like the business processes and graphics applications. It should not matter, if the table is used to embody a constraint or is to be used in some other way.

The proposal is based on the following observations:

- A constraint based configurator should to be able to map variant tables to constraints and to provide an import/export functionality between such constraints and tables.
- Compression to c-tuples greatly empowers the use of tables, because larger tables can be maintained and these can be processed more efficiently. Models should allow variant tables with c-tuples.
- In the absence of an agreed database representation for c-tuples, a file format intelligible to a spreadsheet may serve as an exchange format for tables with c-tuples (see below).
- Support for wildcard symbols is a requirement in conjunction with c-tuples. On the other hand, intervals and other *quasi-finite symbols* [13], would be considered "nice-to-have". As would named sub-domains.
- *Filtering queries* expressing the selection or exclusion of product features are the central type of query we may expect a variant table to support, whether compressed to c-tuples or not. These queries can easily be reformulated in SQL.
- Table content is comprised of symbols, which relate to different datatypes. Minimally, we may expect agreement to distinguish between the datatypes *string*, *integer*, and *float*.

The inclusion of compressed tables with c-tuples in a data-driven approach is justified by the fact that they can closely mimic the behavior of an underlying uncompressed relational table [16, 12], but scale better with increasing domain sizes. Not only can tables with c-tuples support filtering queries (i.e. the selection and exclusion of values), but iteration and direct access to uncompressed data rows can be provided for use by applications outside configuration. There

[12] This strategy was used internally at SAP implementing a business configurator [17]. It is also corroborated by the observation in Footnote 9, as well as the fact that voicing this at the several recent CWG conferences met with general assent, without any dissent, c.f [14].

[13] Specifically, the vision that 80% of a SAP VC model could be in the form of SAP variant tables has been voiced in my presence at a CWG meeting by a consultant already years ago. The RM model is an example of a model entirely comprised of relational variant tables. It has been used as a benchmark in many different configurators.

are various ways c-tuples might be represented, depending also on whether naming of frequently used sub-domains is to be supported. A common universally understood format is a file where each c-tuple is represented as a line, the cells (fields) and the values within a cell separated by a column separator/value separator character, respectively[14]. Table 1 can be stored in such a file by defining the column separator symbol as a blank or tabulator character (“ ”) and the value separator symbol as a semicolon (“;”).

5 CHALLENGES

As already stated, we can divide improvements to modeling into *sustainable innovation* and *disruptive innovation*. Continuous improvement falls under *sustainable innovation*. Any efforts that need agreement between the different commercial configurator vendors and their clients fall under *disruptive innovation*.

Continuous development of individual modeling environments will likely improve the handling of tables. However, the vision I propose in this paper addresses a more radical shift in thinking: Businesses with *CPQ*, *Mass Customization*, or other forms of product configuration that follow the strategy of farming out relevant parts of their models to tables should come to expect that this will also pay off once a common standard of dealing with tables is established, by it becoming significantly easier and less costly to share and exchange critical data between different models and different configurators.

So, the main challenge is to establish such a consensus that allows for a smooth integration with external tables both in relational form and in c-tuple form. However, as use of c-tuples in conjunction with the SAP VC has shown, browsing and maintaining tables with c-tuples is not a trivial user experience. Therefore, a secondary challenge is to come up with new ways of user interaction with large variant tables in general. This is one of the topics of development that I am currently focussed on. Participation by the configuration community in this endeavor is welcome.

6 OTHER WORK AND OUTLOOK

This section covers both a look at other work, notably *XCSP 3.0*, and discusses the problem of data versioning and identity.

Looking at the *XCSP 3.0* specification [4] it becomes apparent that both extensional tabular constraints and advanced compression are dealt with (sections *Constraint extension* and *Constraint mdd* in [4]). The best fit for c-tuples seems to be *Constraint smart* [19]. *Smart tables* [19] is another approach to empower a tabular representation in a constraint setting. This must be looked at in further work, but it seems that the focus is not on the likeness of smart tables to database tables; rather on their expressiveness from the point of view of constraint solving.

These approaches in the CP community seem to strive to extend expressiveness and require dedicated training to use. In contrast, a data-driven approach strives to deviate as little as possible from the tried and true uses of tables in databases and spreadsheets. The goal there is to be as universally intelligible as possible, but it is not necessarily a goal to model exclusively with tables.

Finally, if the vision of a standard in data-driven modeling comes true, this means that relevant data is independent of a particular product model and outside a natural stewardship of any one system. The problem of data ownership and versioning must then be rethought. This is an area for further innovative ideas.

[14] This describes a variant of what is known as a *comma separated values* (.csv) file. It may be necessary to also specify a *text delimiter* symbol. All this is supported in most common spreadsheet tools.

REFERENCES

[1] C. Bessiere, ‘Constraint propagation’, In Rossi et al. [20], chapter 3.

[2] U. Blum¨ohr, M. M¨unch, and M. Ukalovic, *Variant Configuration with SAP, second edition*, SAP Press, Galileo Press, 2012.

[3] U. Blum¨ohr, M. M¨unch, and M. Ukalovic, ‘Variant tables in detail’, in *Variant Configuration with SAP, second edition,* SAP Press, chapter 2.6.3, 152–158, Galileo Press, (2012).

[4] Fr´ed´eric Boussemart, Christophe Lecoutre, and C´edric Piette, ‘XCSP3: an integrated format for benchmarking combinatorial constrained problems’, CoRR, **abs/1611.03398,** (2016).

[5] CWG. SAP Configuration Workgroup. https://www.configuration-workgroup.com, 2020. Accessed: 2020-06-

[6] eSpline. eSpline Recommended Variant Configuration Practices. https://www.espline.com/services/modeling-best-practices, 2020. Accessed: 2020-06-13.

[7] Alexander Felfernig, Gerhard Friedrich, Dietmar Jannach, and Markus Zanker, ‘Configuration knowledge representation using UML/OCL’, in *UML 2002 - The Unified Modeling Language, 5th International Conference, Dresden, Germany, September 30 - October 4, 2002, Proceedings,* eds., Jean-Marc J´ez´equel, Heinrich Hußmann, and Stephen Cook, volume 2460 of Lecture Notes in Computer Science, pp. 49–62. Springer, (2002).

[8] Knowledge-Based Configuration, eds., Alexander Felfernig, Lothar Hotz, Claire Bagley, and Juha Tiihonen, Morgan Kaufmann, Boston, 2014.

[9] Gartner. Configure, Price, Quote (cpq) Application Suites. https://www.gartner.com/en/information-technology/glossary/configure-price-quote-cpq-application-suites, 2020. Accessed: 2020-06-10.

[10] [A. Haag, ‘Chapter 27 - Product Configuration in SAP: A Retrospective’, in *Knowledge-Based Configuration*, eds., Alexander Felfernig, Lothar Hotz, Claire Bagley, and Juha Tiihonen, 319 – 337, Morgan Kaufmann, Boston, (2014).

[11] Albert Haag, The ATMS - *an assumption based problem solving architecture utilizing specialization relations*, Ph.D. dissertation, Kaiserslautern University of Technology, Germany, 1995.

[12] Albert Haag, ‘Managing variants of a personalized product’, *Journal of Intelligent Information Systems*, 1–28, (2016).

[13] Albert Haag, ‘Quasi-finite domains: Dealing with the infinite in mass customization’, in *Proceedings of the 20th International Configuration Workshop, Graz, Austria, September 27-28, 2018.*, p. (Submitted), (2018).

[14] Albert Haag. Making more of variant tables in standard vc, 2020.

[15] Albert Haag and Christophe Faure. Polishing a raw diamond: Variant table with multiple values per cell, 2018.

[16] Albert Haag and Laura Haag, ‘Further empowering variant tables for mass customization’, *International Journal of Industrial Engineering and Management (IJIEM),* **10**(2), (2019).

[17] Albert Haag and Steffen Riemann, ‘Product configuration as decision support: The declarative paradigm in practice’, *AI EDAM,* **25**(2), 131–142, (2011).

[18] Ulrich Junker, ‘Configuration’, in Handbook of Constraint Programming,eds., Francesca Rossi, Peter van Beek, and TobyWalsh, volume 2 of *Foundations of Artificial Intelligence*, 837–873, Elsevier, (2006).

[19] Jean-Baptiste Mairy, Yves Deville, and Christophe Lecoutre, ‘The smart table constraint’, in Integration of AI and OR Techniques in Constraint Programming - *12th International Conference, CPAIOR 2015, Barcelona, Spain, May 18-22, 2015, Proceedings*, ed., Laurent Michel, volume 9075 of Lecture Notes in Computer Science, pp. 271–287. Springer, (2015).

[20] *Handbook of Constraint Programming, eds.,* F. Rossi, P. van Beek, and T. Walsh, Elsevier, 2006.

[21] S. Subbarayan. Clib: Configuration benchmarks library. https://itu.dk/research/cla/externals/clib/, 2004. Accessed: 2019-06-03.

[22] WIKIPEDIA. Sql. https://en.wikipedia.org/wiki/SQL. Accessed: 2018-06-17.

Managerial Challenges in Designing an IT Service Configuration System

Franziska Schorr[1] and **Amartya Ghosh**[2] and **Lars Hvam**[3]

Abstract. Although the configuration field has seen significant theoretical contributions to product configuration since the late 1990s, few studies have used the Configuration Management Database (CMBD) — an information system that comprises a configuration system (CS) and a database — to study service configuration. Yet the information technology (IT) departments in every size of organisation often use the CMDB to configure and store the IT services they offer their business units. Using an exploratory case study at a large, multi-national IT department, we investigate the key managerial challenges that IT managers need to consider when designing IT service configuration. This paper finds that designing the conceptual IT service model, choosing a project management setup and maintaining service data quality are the key challenges that IT managers need to consider when designing IT service configuration.

1 INTRODUCTION

The past twenty years have seen significant advances in and theoretical contributions to configuration [1], which enables companies to design customised products and services while applying mass production principles [2], [3]. Sabin and Weigel (1998) define knowledge-based configuration as 'a special case of design activity where the artefact being configured is assembled from instances of a fixed set of well-defined component types which can be composed conforming to a set of constraints' [4]. A configuration system (CS) is an information technology (IT) system that supports configuration by generating product- and service-related data (e.g. design specifications based on customer requirements) [2].

A special case of configuration is the configuration of IT services. IT services are defined as the day-to-day provision to customers of IT infrastructure and applications, and support for their use [5]. One CS that IT departments commonly use to support this specific configuration problem is the Configuration Management Database (CMDB) [6]. IT departments use the CMDB to (i) automate IT service configuration management (CM) processes and (ii) digitise service data and knowledge [7].

Several studies have outlined the benefits that firms can realise from the successful implementation of product and service CSs [8], [9]. However, firms implementing CSs often fail to fully realise the benefits or end up terminating the CS projects before their implementation [10]. Scholars have identified the challenges (e.g. data quality, product modelling, organisational, IT-related) that firms face when designing product CSs [10], [11] and have proposed guidelines for addressing these challenges [10]. In order to avoid expensive CMDB design failures, both managerial and IT department personnel have to know the managerial challenges that are likely to arise during the CMDB design process [12]. However, the specific challenges that arise during the design of IT service CSs remain unknown. Consequently, it remains unknown if, and if so, how managerial personnel can refer to existing CS theories for solving these challenges.

Using an exploratory case study at a large IT department, we investigate the key managerial challenges that IT managers need to consider when designing IT service configuration. More specifically, we gathered empirical data on the CMDB design process of a multi-national firm collected over two years. We analyse the IT service CS design process, with a particular focus on which managerial decisions, dilemmas, and issues IT managers face during the CS design, and how they respond. We compare the managerial challenges that managers face during the IT service CS design with the challenges occurring during the product CS design.

The structure of the paper is as follows. Section 2 presents a review of the theoretical literature on IT service CM. Section 3 gives the research method. Section 4 presents the details of the case study, Section 5 gives the results, and Section 6 concludes and discusses implications for future research.

2 THEORETICAL BACKGROUND

This section summarises the theoretical background on IT service management and configuration, and the CMDB, which is an IT service CS. The section outlines the known managerial challenges that are likely to arise during the design of product and service CSs.

2.1 IT service management

IT service management (ITSM) is concerned with delivering IT services to service customers. An IT service uses processes, technology assets, activities, functions, key roles, and vendors to contribute to the value creation in firms [13]. A firm's internal IT department provides IT services to the firm's departments (HR, finance, business units) [14] and is solely responsible for managing the IT services. Business units use the provided IT services to

1 Department of Management, Technical University of Denmark, Denmark, email: frsc@dtu.dk

2 Department of Mechanical Engineering, Technical University of Denmark, Denmark, email: amgho@dtu.dk

3 Department of Management, Technical University of Denmark, Denmark, email: lahv@dtu.dk

support their business processes [6]. In this business-IT alignment, and with firms' growing dependency on IT comes their need for high IT service quality that IT departments achieve by using business-focused ITSM practices [13].

ITSM gained popularity when the Information Technology Infrastructure Library (ITIL) published a well-known ITSM framework [15]–[17]. ITIL describes and standardises the best-practice for ITSM, which involves understanding the customer needs, managing the service processes, ensuring the service performance and targeting continual service improvement [14]. The ITIL ITSM framework divides the ITSM area into five main processes: service strategy, service design, service transition, service operations, and continuous improvement [6].

The ITIL framework describes that IT service CSs are essential for IT departments' IT service and asset management processes [6]. To manage large and complex IT services and infrastructures, IT departments need to be in control over all their individual assets (e.g. applications, servers, databases, licenses, network, data centres, mobile devices) and their service assets (the IT service's asset configuration) [18]. IT departments achieve control over IT service specifications by using IT CM, which is responsible for controlling the IT service specifications over the service lifecycle by identifying, controlling, recording, reporting, auditing and verifying service assets and configuration items, including versions, baselines, constituent components, their attributes, and relationships [7]. However, the ITIL ITSM framework does not explain how firms can implement the configuration concept or design IT service CSs successfully [19].

2.2 IT service configuration

In the following sections, we establish that IT service configuration applies knowledge-based configuration. According to Sabin and Weigel (1998), the knowledge-based configuration requires three criteria: (i) knowledge-based configuration is a design activity, (ii) the artefact being configured is assembled from instances of a fixed set of well-defined components, and (iii) the components can be composed conforming to a set of constraints [20].

IT service configuration is a particular case of design activity. IT service configuration supports designers with the formalisation, verification, and control of the IT service design specifications [6]. The IT service design specifications frequently change over the IT service's long lifecycle (> 20 years). The changes result from changing customer requirements, maintenance changes, or innovation to the IT service design. Because of these frequent changes, controlling IT service specifications over time is a more critical data management challenge than defining its initial design specifications. Therefore, the IT service configuration is a core concept in service transition management [6], which assists in transitioning new and changed IT services into operations, ensuring that the service design requirements are effectively realised in service operations [7].

IT services can be assembled from instances of a fixed set of well-defined components. Most of the IT services' components are physical, tangible components (e.g. computing server and storage components, and network components) [18]. Therefore, the well-researched product configuration theories could be re-applied to configuring the physical components of an IT service. Additionally, IT services consist of less well-defined, and intangible components (e.g. software components, processes, key roles, and vendors) [6], [13]. Therefore, theories on service configuration [9], [21] and process configuration [22], [23], [19] could be applied to configure the intangible components of an IT service.

The IT service components must be composed conforming to a set of constraints for the IT service to be a configurable service. IT services consist of very diverse component categories, namely people, processes and technology [24]. Additionally, most IT service components and their interactions are intangible [25]. Therefore, the mentioned IT service characteristics could pose challenges for defining a manageable set of IT service configuration constraints.

2.3 Configuration management database

To perform service CM, the internal IT departments at firms use a CS, the CMDB [6], [7]. Figure 1 shows the typical architecture of a CMDB. The CMDB automatically imports IT asset data from the many sources that monitor IT assets. Using knowledge-based configuration principles, the CMDB configures IT services from assets, builds relationships between related IT assets and services, and validates the IT service specifications. The CMDB holds all asset and configured IT service data. Therefore, the CMDB serves as the single source of truth and supports ITSM processes (e.g. change, financial and disaster recovery management).

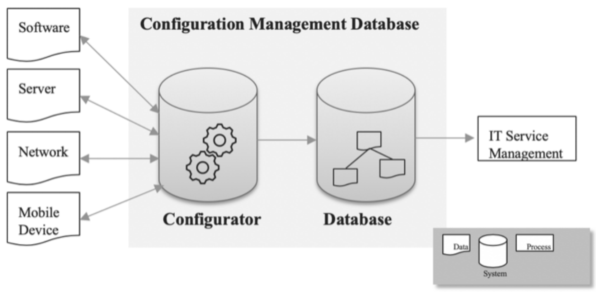


Figure 1: Typical architecture of a CMDB

Using a CMDB, IT departments improve their ITSM capabilities. Without a CMDB, IT service data is scattered across databases, or missing. For example, IT personnel manage servers independently of applications. With a CMDB, IT service data (e.g. attribute and relationship) is combined in one service knowledge management system. Using the now accessible and available service data allows IT management to leverage two significant managerial benefits. First, it enables IT departments to improve IT operational stability [18]. Service relationship data let IT personnel assess the business impact that results from IT service downtime, which can occur due to service maintenance, re-design, failure, or retirement. Second, it enables IT departments to provide IT cost transparency [18]. Using data on service component interdependencies lets IT personnel precisely allocate and charge IT costs. Moreover, the use of a CMDB reduces the need for manual data management and provides holistic IT service knowledge. Before firms can leverage the advantages, IT management must successfully implement a CS by overcoming the managerial challenges during its design.

2.4 Managerial challenges in CS projects

This section provides an overview of the findings from existing research on the managerial challenges faced during the design of product and service CSs.

Kristjansdottir et al. (2018) identifies five broad categories of managerial challenges that manufacturing companies face while developing, implementing and maintaining product CSs from a review of existing literature of product CSs [11]. These categories are organisational challenges, knowledge acquisition challenges, IT-related challenges, product-related challenges, resource constraints and product modelling challenges [11]. Haug et al. (2018) reviewed 8 cases of CS project termination at ETO firms and identified ten causes, related to issues of the project scope, CS design and poorly defined CS maintenance procedures, of project termination [10]. The paper highlighted the importance of holistic decision-making during the entire CS project cycle and prescribed a set of guidelines that firms must follow during the different project phases to avoid CS project failure [10].

Several studies have addressed the use of CSs to configure services in several domains, such as telecommunication services [26], ICT services [27] equipment maintenance services [23], [28], and insurance services [29]. A limited number of studies have highlighted the conceptual and managerial challenges that firms face while designing the service CSs. Comparing the characteristics of services against the characteristics of physical products, Heiskala et al. (2005) identified updating the service CS knowledge base, acquiring service configuration knowledge from multiple stakeholder groups, employing both configuration experts and domain experts in the CS design process, testing service configuration knowledge, handling errors in service specifications arising out of misunderstandings between the customer and the service provider, and ensuring the ease of use of service configurators for customers as potential challenges facing service providers [9]. Based on their experience of developing three service CSs, Tiihonen et al. (2014) faced the challenges of modelling stakeholders as resources and conceptualising services using product CS concepts during the design process [23]. While modelling a distributed web-based sales CS prototype for Internet Protocol Virtual Private Networks (IP-VPN) services, Ardissono et al. (2003) highlighted the need for a common approach towards conceptually modelling services across the supply chain [26]. The key business requirement for online and offline availability of the service CS for ICT services at Fujitsu highlighted the need for careful data governance through the sharing and synchronisation of data across global, regional and local databases [27].

Several studies have addressed the use of the CMDB for IT service configuration [12], [19], [30]. Keller et al. (2009) present a best practice approach for deploying CMDB software [30]. Brenner et al. (2006) found that defining the CMDB scope and structure, as well as filling and maintaining its content, are difficult and time-consuming tasks [12]. Jelliti et al. (2010) hint at the integration of the CMDB with external databases as well as the CMDBs architectural data model being managerial challenges [19].

The previous sections have shown that IT service configuration is a particular and very relevant CM topic. Our literature review revealed a lack of empirical evidence on the managerial challenges associated with IT service CS design. Knowledge about the managerial challenges provide researchers with directions to guide the development of new methods, tools, and frameworks for addressing them. Firms can utilise these learnings to foresee the problems that they will face while developing IT service CSs and make plans to overcome them. This study aims to add to the knowledge on IT service configuration by providing insights into the process of designing a CMDB, and by identifying the managerial challenges that arise during the CMDB design process. To guide our research, we formulate the following research question: *What are the managerial challenges that IT departments face during the design of IT service CSs?*

3 RESEARCH METHOD

We seek to answer the research question through a case study because it is a suitable method for researching our research problem. Three reasons motivated the choice of the case study methodology for this research. First, the case study answers research questions such as "how" and "why", and the case study method, therefore, fits perfectly to the research question of this research [31]. Second, case studies allow for in-depth studies of the unit of analysis [31]. Third, using a case study allows us to analyse the unit of analysis over time, in real-time, and in its natural setting. Gathering real-time data reduces hindsight bias that often occurs during case studies [32].

The paper is part of a two-year in-depth case study at the IT department of a transportation and logistics service provider (TLS), which is a global firm with headquarters in Denmark. The central unit of analysis in this paper is the CMDB design process at the TLS – because it is in this process that the managerial challenges arise. During the design process, management regularly must take decisions, justify their decisions, and choose between alternatives. Management continuously evaluates and decides if the investment in a CS is still worthwhile. We chose the design process and not the final CS as our unit of analysis. Studying a process in real-time allows us to collect suitable empirical data. If we were to analyse the process of successfully implemented systems retrospectively, we could only access data collected by the firm. Furthermore, we would not be able to gather data about managerial challenges in configuration design projects that had failed.

We selected the firm because the firm's IT department has recently started a new CMDB implementation project. Following the implementation of an initial CMDB prototype, the firm opted for a radical change in the design process and restarted the development of the CMDB. The firm's use of two distinct design approaches allows us to investigate the rationale behind the changes and consequently identify the managerial challenges in developing the CMDB. The firm's IT service portfolio is representative of portfolios of modern global firms that are supplying IT services to different global subsidiaries. The firm manages and designs its services themselves and relies little on outsourcing. In general, IT service design is quite homogeneous across companies and industries because IT services are assembled from the same technology components (e.g. software, server, or network components).

Conducting an in-depth case study would not be possible without the researchers' access to the firm [33]. During the study, one member of our research team worked at the firm. We did not have an active role in the CMDB design project but would from time to time consult on the CMDB design project due to our experience in the field of product configuration. Our entire team had access to and analysed the empirical case data.

The case study method lets us enrich the empirical discussion by using both primary and secondary data sources [31]. The data collected during this study is mainly qualitative. We conducted five interviews with the CMDB project stakeholders, which we present in Table 2. Table 1 shows an overview of the context area and topics covered in the interviews, and the supplementary data sources used for extending the empirical content and triangulating the interview data.

Table 1. Required data and data sources

Context area and topics discussed	IT CS design: • Design procedure • Design requirements • Managerial challenges • Solutions to managerial challenges
Data source	• CS • CM policy • Configuration software • Conceptual data models • Project plan and reports • Interviews with CMDB stakeholders

4 CASE DESCRIPTION

We conducted a case study at a TLS. The TLS manages, controls, and delivers logistics activities that are undertaken by shipping companies and their clients [34]. IT systems are critical in supporting the firm's business processes and business capabilities. At the TLS, the global IT department is responsible for providing IT services associated with the enterprise resource planning (ERP) systems, customer relation management (CRM) systems and industry-specific transport management systems.

When the TLS requires IT capabilities, the global IT department is responsible for managing the IT capability. First, the IT department would design an IT service for the TLS either by developing a new information system or by implementing off-the-shelf software. The IT department would then implement the IT service and provide its computing infrastructure (e.g. servers, storage, network) required for operating the IT service. The IT department would support the IT service over its lifecycle by providing processes for service support (e.g. incident management, security, service help desk), maintenance (e.g. updates, asset management), and continuous improvement (e.g. change management, service level management). The global IT department provides both standard IT services to the entire firm as well as customised IT services to their 83 local subsidiaries. To improve the management of their 1000+ services, the global IT executive management at the TLS decided to implement a CMDB. Annually, the TLS's IT department implements approximately 10,000 IT service changes, which requires IT personnel to document, update, control, maintain and change the IT service data.

In 2017, the TLS started an IT project to design the CMDB, which the IT executive management considered as strategically important for the TLS. Table 2 shows the key CMDB project stakeholders, their organisation and the decisions they make. The deliverables of the project are the design of the conceptual IT service model, the CM process and policy and the implementation of the CM software.

Table 2. CMDB project stakeholder overview

CMDB project stakeholder	Decisions and responsibilities
IT executive management	Approval of project progress, resources, deliveries and financing
Project manager	Project management
Project team	Project deliveries
Reference group	User requirements and quality
Change manager	Anchoring the CMDB in the firm

At the start of the CMDB project, the project stakeholders prioritised a fast CMDB design implementation over a long conceptual design planning phase. The IT executive management argued that implementing the CMDB allows them to assess and evaluate the tangible project progress and business value. The project team prioritised leveraging their existing resources such as data and data structures, CM software and personnel. Within six months, the TLS had implemented their first CMDB prototype that did not satisfy the stakeholders' data quality requirements. CMDB project stakeholders responded by making changes to their design strategy, which now set a higher focus on the conceptual CMDB planning by revamping the conceptual IT service model and data quality management.

5 RESULTS

In the following sections, we present and discuss our findings.

5.1 Managerial challenges in designing IT service CSs

The in-depth case study revealed that, during the IT service CS design process, managers are likely to face three significant managerial challenges.

5.1.1 Designing the conceptual IT service model

We identified the design of the conceptual IT service model as the first managerial challenge during the CMDB design process. The TLS's IT executive management decided that the CMDB project team should design conceptual model prototypes, then implement the model in the CM software, and continuously test, improve, and maintain the model. When designing the conceptual IT service model, the TLS followed these steps: (i) develop a service definition, (ii) define the service boundaries, (iii) define the service relationships, (iv) define the component types, (v) define the component attributes, (vi) define the component relationships, (vii) align the conceptual model with industry-standard data models and (viii) align the conceptual service models with the needs of the IT management processes. While we found that this overall CMDB design procedure was a good fit for the CMDB implementation at the TLS, designing a conceptual IT service model comes with practical challenges. We identified three issues that the TLS faced while conceptually modelling the IT services for configuration.

The first issue concerns the development of a service definition.

At the TLS, multiple service definitions exist, and the IT executive management faces the challenge of providing a unified definition. Some of the CMDB project reference group members argue that an IT service is a performance (e.g. a user experience or feature that the customer can order). In contrast, others argue that an IT service is a process (e.g. a procedure in which internal IT departments provide value to firms). The CMDB project team members argue that a service definition includes a combination of processes, performances, and tangible technology components. Their service definition aligns with the generic service definition (*'we define services as the application of specialised competences (knowledge and skills) through deeds, processes, and performances for the benefit of another entity or the entity itself'* [35]) and the IT service definition.

The lack of theoretical knowledge has adversely impacted the conceptual service modelling process at the TLS. Although the IT staff at the TLS had the academic knowledge about service definitions, they were unable to reconcile their definitions. The service definition is a difficult concept for both practitioners and scholars to grasp. The academic service definitions do not sufficiently explain the relationships between operand (technology components) and operant service resources (skills, knowledge, processes) in IT services. Theoretical knowledge of the managerial decisions about defining the service boundaries, component types, component attributes and interrelationships are lacking.

The second issue concerns the scoping of the conceptual IT service model. The model's scope covers the IT service components, their attributes and relationships, that the model includes.

Initially, the TLS used a conceptual service model with little detail and without following a modelling methodology. The conceptual model included various components, a high number of component attributes, and multiple but non-classified component relationships. Following the change to their design strategy, the TLS used a conceptual service model based on object-oriented modelling. The model included fewer components compared to their first model. Every component was modelled in detail, including its attributes and relationships. The model included different relationship types. We observed that the CMDB project management considers the complexity of the conceptual IT service model an issue in conceptually modelling IT services for configuration. One general finding of this study is that having a rigid conceptual data model on hand from the start, and sticking to the data model, was an enabler of CMDB implementation.

Modelling IT services for generating IT service knowledge that is useful for management is the critical scoping issue. This managerial issue translated into the CMDB project team having to choose between leveraging existing IT service data structures and designing new data structures based on future use-cases of the CMDB. Designing new data structures is a costly task; however, it will enable the firm to design the CS for the best usability.

Modelling IT services for configuration is another scoping challenge. This managerial issue translated into the CMDB project team having to design the conceptual model for configurability. IT services consist of many standard components. This characteristic of IT service composition is advantageous for managing the scope of the conceptual model. Because IT services are highly interconnected [25], they require extensive relationship modelling. This requirement drives the need for an extensive set of configuration constraints and consequently increases the IT service model scope.

The third issue concerns the alignment of the need for continuous conceptual IT service model changes with the model's implementation in the CM software. During the TLS's CMDB design, a lack of alignment led to the need to revamp the CMDB. During its initial design approach, the CMDB project team tested the conceptual service model for its fit with the entire service portfolio and populated the CMDB service-by-service. When the conceptual service model is not yet fully defined, incomplete service configurations are released and stored in the CMDB. Managerial issues arose when the CMDB project team had to incorporate changes to the service configuration as they resulted in inconsistent service configurations. The TLS responded by changing their approach, populating the CMDB service component-by-component. As a result, the CMDB stores incomplete service configurations up until the CMDB is fully populated. During its long implementation process, the CMDB is not entirely usable and contributes little to business value. This approach allows the CMDB project team to reuse configuration rules and use the economy of scale concept, and to reduce service data inconsistencies.

5.1.2 Choosing a project management setup

We identified the management of the IT service CS project as the second managerial challenge during the CMDB design process.

During the CMDB project, the IT executive managers consider the TLS's general ITSM maturity as low to medium, with the CMDB, which is the core IT service knowledge management system used at the TLS, being the cornerstone of the TLS's overall servitisation strategy. To servitise their IT operations, the TLS must implement the CMDB, engineer new ITSM processes (e.g. CM, service level management) and re-engineer existing processes (e.g. cost management). Because of the interdependent nature of the CMDB's relationship with the ITSM processes, the TLS faces a paradox when improving the ITSM maturity level. The ITSM processes depend on the knowledge deliveries from the CMDB [6], and the CMDB scope and functionality depend on well-defined user requirement deliveries from the ITSM processes.

We identified two limitations of the interdependencies among multiple explorative ITSM implementation projects (e.g. ITSM processes, CMDB).

The first limitation is the lack of established organisational ITSM knowledge. The capability of the CMDB users that have limited experience with using the CMDB to provide well-defined user requirements and, thereby, feedback and control on the CMDB design scope and quality is questionable. Scholars have mentioned the (i) weak requirement determination and (ii) short-changed quality assurance as typical project management mistakes that lead to IT system design failures [36]. To avoid design failures, scholars recommend implementing project management best-practices such as joint application development [36]. Scholars propose to manage related projects in multi-project programs together to obtain control and benefits not available through single project management [37].

The case results show that the IT executive management continuously urged the CMDB project management to align the CMDB with other departmental ITSM projects, processes, and systems. However, the CMDB project stakeholders ran into alignment challenges (e.g. on data model, timing of deliverables, decision authority, servitisation strategy) with the other initiatives.

The second limitation is the lack of clear project management governance. As the project management had neither full control nor autonomy over the project deliverables, the CMDB project was soon caught in a paradox that limited CMDB project progress. Scholars present, specifically for IT projects, a tension resolution strategy theory that defines how managers can respond to the TLS's alignment problems: management should find blended solutions when the program requirements are combinable, and management should urge project teams to find compromises through dedicated co-ordination mechanisms and roles when requirements are not combinable [38].

5.1.3 Maintaining service data quality

We identified the maintenance of the IT service data quality in IT service CSs as the third managerial challenge. Table 3 provides an overview of the most common data quality issues [39] and an explanation of how those issues limit the usefulness of CMDBs.

Table 3. CMDB data quality issues

Data quality issue	Managerial challenges
Availability	Missing data in the CMDB results in IT service knowledge limitations, which inhibit firms in leveraging their data assets
Accuracy	Incorrect data in the CMDB results in incorrect service knowledge, which misinforms decision-makers
Consistency	The lack of a single source of data truth leads to non-credible service knowledge, which is unreliable for decision-makers to use
Usability	The CMDB collects data and generates service knowledge that the firm cannot effectively use
Timeliness	The CMDB collects data and generates service knowledge that the firm cannot yet use

At the TLS, the CMDB project stakeholders initially decided that the data sources would govern the data quality because they believed in leveraging their existing data management processes. When the evaluation of the CMDB prototype revealed that the TLS's data quality was insufficient for the newfound IT service knowledge requirements, IT executive management handed over data governance to the CMDB project team. This handover improved the TLS's overall service data quality. The TLS used source systems for data import and validation. To support data governance, the TLS adopted the RACI method to assign each stakeholder a definite role in the data governance process. The TLS used knowledge-based configuration for generating component relationship data. The TLS invested in a dependency-mapping system for real-time data on component relationships.

To manage data quality, CMDB project team can take managerial design decisions regarding the technical CMDB design, more specifically, on the alignment between data sources and the CMDB. While most source systems provide component data, the CMDB is responsible for the knowledge-based configuration of the service components. Management can choose between multiple CMDB design scenarios to decide which system will govern the data.

In one scenario, the source systems are responsible for data quality governance. The CMDB imports the data and has no direct control over the data. IT personnel build the component interrelationships manually, requiring additional labour resources, and do not use knowledge-based configuration. In that case, the CMDB is rather a database system than a knowledge-based CS.

In another scenario, dedicated IT configuration managers are responsible for data quality governance by defining rules, standards, and policies for all data sources linked to the CMDB. The CMDB imports most data from the source systems and has direct control over the data. To manage missing data, CMDB can import component data from existing IT component management systems (e.g. software or IT infrastructure management systems). To generate missing data on component interrelationships, the CMDB can import data from dependency mapping systems that identify component relationships automatically in real-time.

5.2 Managerial challenges from CMDB design project compared with product CS projects

This section contrasts the managerial challenges that the TLS has experienced during the CMDB design process against those experienced by firms during the design of product CSs. Our analysis will help managers in identifying situations where they can refer to product configuration research to inform their decision-making.

5.2.1 Designing the conceptual model

Developing conceptual models is one area where all firms developing CSs face challenges. For the case study at the TLS, we identified three issues that are related to the conceptual service modelling activity. First, the CMDB project stakeholders had conflicting views of the IT service definition, which translated into disagreements about the conceptual IT service model. Scholars and practitioners generally agree on the definition of a configurable product. However, we find that domain experts often encounter disagreements over product knowledge while developing conceptual product models. Achieving consensus about product knowledge among domain experts is an essential criterion for the successful design of product CSs [40]. Firms have to ensure consensus on the product knowledge that is to be included in a product model. In the case of the CMDB, we do not yet have an answer as to how managers best can tackle the issue of having competing service definitions. Project management theories suggest either balancing the competing differences out or accepting them.

Second, when configuring complex products and services, firms face challenges in defining the scope of their conceptual models. In case of product CSs, firms have to make trade-offs between the need to cover the entire product portfolio in the CSs and the need to ensure that the CS model does not become too complex [11]. On the one hand, if CSs do not cover the entire product portfolio, its user acceptance may be adversely affected [11]. On the other hand, if the firm intends to cover the entire product portfolio within its CSs, then the underlying models may become too complex and adversely impact the ease-of-use and the user-friendliness of the system [11]. Many firms are unable to successfully implement CSs due to the complex nature of their products [11], [41].

IT service CSs must cover the entire IT service portfolio because else firms cannot use them for effective ITSM. Hence, IT service managers cannot exclude individual services or service components from the conceptual IT service model to limit its complexity. The homogeneity of IT services, which entails that an IT service is assembled from the same well-defined technology components from across firms and industries, makes it theoretically easier to cover the entire IT service portfolio within the scope of the CMDB. However, managers encounter and must manage the challenge of the given complexity of conceptual IT service models. The source of complexity in IT services lies in the variety of tangible and intangible component types (e.g. IT assets, people, processes) and inter-relationships between these components. From our analysis of the CMDB design project, we find the need to prioritise further research into methods for managing the complexity of IT services over mechanisms for IT service complexity reduction.

Third, we observed challenges arising out of the stakeholders' initial decision to fast-track the service model conceptualisation process for the CMDB. Since the CMDB project team decided to release the CMDB service configurations without completing the conceptual service model, they had to dedicate significant efforts to update the conceptual service model to reintroduce service features that had gone undiscovered during the analysis of previous services. In case of conceptual product modelling, eliciting product knowledge from domain experts and documenting it using tools, such as the product variant master (PVM) [2], Unified Modelling

Language (UML) class diagrams [42] and class-responsibility-collaboration (CRC) cards [2], are time-consuming activities [43]. These activities require specialist knowledge to be performed correctly. Once the product model is completed, the configuration engineer is responsible for implementing the product model within the CS software. Firms often overlook the documentation activity and instead opt to directly implement the acquired product knowledge into the CS knowledge base. This lack of documentation leads to problems in communication among the domain experts, knowledge engineers, and configuration engineers, and testing and validation of the product knowledge modelled within the CS. While a number of studies have presented conceptual models and ontologies for services [21], [44]–[47], there is a need for further rigorous validation and evaluation of these models in other service domains. In contrast to the product configuration domain, practitioners in the IT service domain lack well-known, widely applied conceptual service models and methodologies for the implementation of such models. Additionally, we observe a similarity in the behaviour of the firms regarding the insufficient allocation of resources and time towards documenting of the conceptual models before implementing them in the CS software.

5.2.2 *Maintaining service data quality*

Another challenge that both product and service CS projects face is managing data quality. At the TLS, poor data quality in the CMDB leads to an adverse impact on the quality of the service specifications and, consequently, the TLS's capability to manage its rapidly changing IT services and assets. In case of product CSs, poor master data quality in product CSs negatively impact the quality of product specifications, lead time, cost and price estimations, and, consequently, the economic performance of the firm [48], [49]. Firms often face challenges of having inconsistent data. Data pertaining to specific product configuration instances must be consistent across multiple IT systems (e.g. ERP systems, CRM systems, CAD system) for each lifecycle phase of the product [50].

The product configuration managers can solve these issues by designing technical integrations among these IT systems and establishing product configuration data quality management practices and tools [50]. However, there is limited empirical evidence of the successful implementation of these concepts (e.g. configuration lifecycle management) across all the lifecycle phases of a product. While IT service CSs face similar data quality challenges as product CSs, IT service CSs also have to accommodate the need for continuous changes or updates to the IT service design specifications. Similar to product CS projects, the CMDB project solves data quality issues by designing technical integrations between the CMDB and its data sources, and by establishing clear data governance mechanisms. We identify the need for research into data quality management for both product and service CSs.

6 CONCLUSION AND FUTURE RESEARCH

We set out to identify the managerial issues in designing IT service CSs. Through an in-depth case study at a TLS based in Denmark, we uncovered at least three significant problems that the firm faced during the design process of a CMDB. We synthesise these problems into three main categories:

Designing the conceptual IT service model: In our analysis, we found that though the TLS had a framework for scoping and formalising IT service knowledge within a conceptual IT service model, the TLS still faced challenges in the successful execution of the framework. Gaps in the literature on IT service composition and the management of complex IT service design contributed to the challenges that the TLS faced.

Choosing a project management setup: Our main finding was that, for firms developing CMDBs alongside their ITSM processes, managing the CMDB design project using project management best-practices and as a part of an ITSM program is beneficial.

Maintaining service data quality: Our main finding was that service data quality management was the TLS's most crucial area in CMDB design. We found that CMDB quality management can be supported by clear data governance and by aligning the CMDB technical design with data quality requirements. Our analysis also highlights future research into the development and implementation of data quality management mechanisms to be of benefit to both product and service configuration areas.

We contribute to the service configuration theory by identifying the managerial issues that firms face in designing service CSs. Moreover, we highlight the gaps in IT service configuration research, the fulfilment of which can help IT managers in addressing these challenges. We contrast our findings from the CMDB design project against product configuration literature. Our case description and findings can assist IT service managers in anticipating and addressing potential managerial issues early in the design process, and in identifying situations where they can refer to product configuration research to inform their decision-making.

From a methodological perspective, the limitations of the paper arise out of the single case study approach. We acknowledge that firms may face even more challenges while designing IT service CSs. As our research topic is largely unexplored in research, the single case study approach allowed us to present a detailed description of the case and its findings.

Our findings suggest that future research ought to focus on more cases on IT service CS design projects. With the growing importance of service configuration, we feel that this research area would benefit from further exploration, as firms require meaningful IT service configuration solutions.

REFERENCES

[1] L. L. Zhang, "Product configuration: a review of the state-of-the-art and future research," *International Journal of Production Research*, vol. 52, no. 21, pp. 6381–6398, 2014.

[2] L. Hvam, N. H. Mortensen, and J. Riis, *Product customization*. Springer Science & Business Media, 2008.

[3] C. Forza and F. Salvador, *Product information management for mass customization: connecting customer, front-office and back-office for fast and efficient customization*. Springer, 2006.

[4] S. Mittal and F. Frayman, "Towards a Generic Model of Configuration Tasks, IJCAI89, vol. 2." San Mateo, CA: Morgan Kaufmann, 1989.

[5] ISACA, *COBIT 5: A Business Framework for the Governance and Management of Enterprise IT.* ISACA, 2012.

[6] A. Cartlidge, A. Hanna, C. Rudd, M. Ivor, and R. Stuart, *An introductory overview of ITIL V3*. 2007.

[7] S. Lacy and I. Macfarlane, *Service transition*. TSO, 2007.

[8] K. Kristjansdottir, S. Shafiee, L. Hvam, M. Bonev, and A. Myrodia, "Return on investment from the use of product configuration systems--A case study," *Computers in Industry*, vol. 100, pp. 57–69, 2018.

[9] M. Heiskala, K.-S. Paloheimo, and J. Tiihonen, "Mass customization of services: benefits and challenges of configurable services," *Frontiers of e-Business Research (FeBR 2005),*

Tampere, Finland, pp. 206–221, 2005.
[10] A. Haug, S. Shafiee, and L. Hvam, "The causes of product configuration project failure," *Computers in Industry*, vol. 108, pp. 121–131, 2019.
[11] K. Kristjansdottir, S. Shafiee, L. Hvam, C. Forza, and N. H. Mortensen, "The main challenges for manufacturing companies in implementing and utilizing configurators," *Computers in Industry*, vol. 100, pp. 196–211, 2018.
[12] M. Brenner, M. Garschhammer, M. Sailer, and T. Schaaf, "CMDB - Yet another MIB? On reusing management model concepts in ITIL configuration management," *Lecture Notes in Computer Science*, vol. 4269, pp. 269–280, 2006.
[13] N. Shahsavarani and S. Ji, "Research in Information Technology Service Management (ITSM): Theoretical Foundation and Research Topic Perspectives," in *International Conference on Information Resources Management*, 2011.
[14] R. Johnston and G. Clark, *Service operations management: improving service delivery*. Pearson Education, 2008.
[15] T. Proehl, K. Erek, F. Limbach, and R. Zarnekow, "Topics and applied theories in IT Service Management," in *Proceedings of the Annual Hawaii International Conference on System Sciences*, 2013, pp. 1367–1375.
[16] A. Teubner and C. Remfert, "IT Service Management Revisited – Insights from Seven Years of Action Research," in *CONF-IRM 2012 Proceedings*, 2012.
[17] J. Iden and T. R. Eikebrokk, "Implementing IT Service Management: A systematic literature review," *International Journal of Information Management*, vol. 33, no. 3, pp. 512–523, 2013.
[18] L. Hunnebeck, C. Rudd, and S. Lacy, "ITIL Service Design," TSO, London, 2011.
[19] M. Jelliti, M. Sibilla, Y. Jamoussi, and H. Ben Ghezala, "A Model Based Framework Supporting ITIL Service IT Management," in *Enterprise, Business-Process and Information Systems Modeling*, 2010, vol. 50 LNBIP, pp. 208–219.
[20] D. Sabin and R. Weigel, "Product configuration frameworks-a survey," *IEEE Intelligent Systems and their applications*, vol. 13, no. 4, pp. 42–49, 1998.
[21] H. Akkermans, Z. Baida, J. Gordijn, N. Peiia, A. Altuna, and I. Laresgoiti, "Value webs: Using ontologies to bundle real-world services," *IEEE Intelligent Systems*, vol. 19, no. 4, pp. 57–66, 2004.
[22] M. Aldanondo and E. Vareilles, "Configuration for mass customization: how to extend product configuration towards requirements and process configuration," *Journal of Intelligent Manufacturing*, vol. 19, no. 5, pp. 521–535, 2008.
[23] J. Tiihonen, W. Mayer, M. Stumptner, and M. Heiskala, "Configuring services and processes," in *Knowledge-based Configuration--From Research to Business Cases*, Morgan Kaufmann Publishers, 2014, pp. 251–260.
[24] M. Matzner *et al.*, "Digital Transformation in Service Management," *Journal of Service Management Research*, vol. 2, no. 2, pp. 3–21, 2018.
[25] J. Salminen, "Digital Services: How Are They Different?," in *Proceedings of International Conference on Business, Information, and Cultural Creative Industry (ICBIC14), Taipei, Taiwan*, 2014, pp. 6–8.
[26] L. Ardissono *et al.*, "A Framework for the Development of Personalized, Distributed Web-Based Configuration Systems," *AI Magazine*, vol. 24, no. 3, p. 93, 2003.
[27] C. U. Tauscher, K. Haasis, and J. Menne, "Global Approach to Knowledge Sharing and Standardization for Promoting Growth in Service Business," *FUJITSU SCIENTIFIC & TECHNICAL JOURNAL*, vol. 54, no. 2, pp. 22–27, 2018.
[28] M. Dausch and C. Hsu, "Engineering service products: The case of mass-customising service agreements for heavy equipment industry," *International Journal of Services, Technology and Management*, vol. 7, no. 1, pp. 32–51, 2006.
[29] R. Winter, "Mass customization and beyond - Evolution of customer centricity in financial services," *Moving Into Mass Customization: Information Systems and Management Principles*, pp. 197–213, 2002.
[30] A. Keller and S. Subramanian, "Best practices for deploying a CMDB in large-scale environments," in *2009 IFIP/IEEE International Symposium on Integrated Network Management*, 2009, pp. 732–745.
[31] R. K. Yin, *Case study research and applications: Design and methods*. Sage publications, 2017.
[32] K. M. Eisenhardt and M. E. Graebner, "Theory Building From Cases: Opportunities and Challenges," *Academy of Management Journal*, vol. 50, no. 1, pp. 25–32, 2007.
[33] C. Karlsson, *Research Methods for Operations Management*, Second Edi. Routledge, 2016.
[34] S. Hertz and M. Alfredsson, "Strategic development of third party logistics providers," *Industrial Marketing Management*, vol. 32, no. 2, pp. 139–149, 2003.
[35] S. L. Vargo and R. F. Lusch, "Evolving to a New Dominant Logic for Marketing," *Journal of Marketing*, vol. 68, no. 1, pp. 1–17, 2004.
[36] R. R. Nelson, "IT project management: Infamous failures, classic mistakes, and best practices," *MIS Quarterly executive*, vol. 6, no. 2, 2007.
[37] Project Management Institute, *The standard for program management*. Project Management Institute, 2013.
[38] R. W. Gregory, M. Keil, J. Muntermann, and M. Mähring, "Paradoxes and the nature of ambidexterity in IT transformation programs," *Information Systems Research*, vol. 26, no. 1, pp. 57–80, 2015.
[39] R. Y. Wang, M. P. Reddy, and H. B. Kon, "Toward quality data: An attribute-based approach," *Decision support systems*, vol. 13, no. 3–4, pp. 349–372, 1995.
[40] K. Edwards and K. L. Jensen, "Framework for Assessing Configuration Readiness," in *Proceedings from the 3rd Interdisciplinary World Congress on Mass Customization and Personalization*, 2005.
[41] L. L. Zhang, P. Helo, A. Kumar, and X. You, "An empirical study on product configurators' application: Implications, challenges, and opportunities," in *Configuration Workshop*, 2015, pp. 5–10.
[42] A. Felfernig, G. E. Friedrich, and D. Jannach, "UML as domain specific language for the construction of knowledge-based configuration systems," *International Journal of Software Engineering and Knowledge Engineering*, vol. 10, no. 04, pp. 449–469, 2000.
[43] A. Haug, A. Degn, B. Poulsen, and L. Hvam, "Creating a Documentation System to Support the Development and Maintenance of Product Configuration Systems," in *Proceedings of the 2007 WSEAS International Conference on Computer Engineering and Applications, Gold Coast, Australia, January*, 2007.
[44] M. Heiskala, J. Tiihonen, A. Anderson, and T. Soininen, "Four-Worlds Model for Configurable Services," in *Joint Conference IMCM*, 2006, pp. 199–216.
[45] M. Sievänen, M. Heiskala, and J. Tiihonen, "Cost and pricing principles for service configurators," in *10th Conference on Manufacturing Accounting Research*, 2010, pp. 20–23.
[46] T. Böhmann, M. Junginger, and H. Krcmar, "Modular Service Architectures : A Concept and Method for Engineering IT services," *36th Hawaii International Conference on System Sciences*, pp. 1–10, 2003.
[47] F. Schorr and L. Hvam, "The Concepts of Modularization in ICT Service Modeling," *2019 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*, 2019.
[48] J. B. Rasmussen, A. Myrodia, L. Hvam, and N. H. Mortensen, "Cost of not maintaining a product configuration system," *International Journal of Industrial Engineering and Management*, vol. 9, no. 4, pp. 205–214, 2018.
[49] A. Haug, F. Zachariassen, and D. Van Liempd, "The costs of poor data quality," *Journal of Industrial Engineering and Management (JIEM)*, vol. 4, no. 2, pp. 168–193, 2011.
[50] A. Myrodia, T. Randrup, and L. Hvam, "Configuration lifecycle management maturity model," *Computers in Industry*, vol. 106, pp. 30–47, 2019.

Exploring features for digital social interaction between configurator users and their friends

Chiara Grosso[1] and **Cipriano Forza**[2]

Abstract. The increasing popularity of online social network platforms allows people to produce and maintain social relations. By 2011, Facebook was the largest social networking site, producing 35 billion online friendships per year. Previous studies on social recommendation have posited that the potential of social relations can be exploited to complement recommender systems implemented in e-commerce environments by collecting social interactive feedback while users are shopping online (e.g., ratings, clicks, and favorites). Such arguments are based on the likelihood that individuals will seek suggestions from friends before making a purchase decision and that their friends consistently provide good recommendations. In the domain of e-commerce for self-customized products, online sales configurators (OSCs) have become connected to social software (SSW). Complementing a configurator with SSW converts it into a highly social interactive medium able to support users with digital interactions. Prior research in social product customization has not determined the extent to which active social interaction features in a product self-design environment respond to user needs for digital social feedback. By using analytical reasoning to compare results and conclusions from previous research, the present study shall investigate how connected OSC-SSW modalities support users in their digital interactions with friends while they are shopping via an OSC. Results from the present paper will describe current social interaction features in an online sales configurator environment and propose end-user interfaces that connect users with friends (trustworthy recommendation sources) during ongoing product configuration. The paper seeks to further the debate on social product customization systems by providing insight on implementing interactive features that respond properly to configurator user demand for digital social interaction.

1 INTRODUCTION

Social networks employ a variety of media-rich, social, and decision-making components that include recommendation agents designed to assist users with digital communication as well as the sharing of ideas and opinions with connected users. This has prompted consumers to bring more of their online experiences onto social networks rather than engaging directly on company websites. Facebook, Instagram, Pinterest, and Twitter are the most popular platforms where customers share their buying habits [1].

Recently, socially enriched mass customization (MC) systems, social product customization systems in particular, have been found to systematically influence consumer choices [2–4]. Since 2003, MC research has recognized the importance of social influence processes in the context of self-designable products [5–7], highlighting the need for research on socially enriched variants of such systems.

Social presence has been identified as a key consideration in website design to overcome a lack of warmth, social cues, and face-to-face interaction. However, previous studies that have investigated interface features that may increase social presence in product configuration, helping users feel connected with their relevant others, such as social network friends, are mostly focused on group recommendation systems [8–9]. Online sales configurators (OSCs) are connected to social network platforms with social software (SSW) programs [10–11]. Each connection enables digital social interactions by providing different features for OSC users during their configuration process. Some of these features support users in sharing configured products within their Facebook accounts, sharing self-designed products with other users, and obtaining feedback via email or social profile accounts [10].

Prior research in this area has not explored the extent of social interaction features incorporated into product self-design that provide users with the interaction necessary for triggering social recommendations from sources users trust. This study is intended to extend social product customization system studies by using analytical reasoning to compare results and conclusions from previous research. This analytical reasoning will enable the present study to propose a configuration environment featuring social recommendation capabilities that connect users in one-to-one social interactions with their personal contacts while shopping for self-designed products.

2 THEORETICAL BACKGROUND

2.1 Social network and social influence in an online environment

Personal social networks are implicit in the offline shopping experience, while the inclusion of social interaction features in e-commerce is a recent phenomenon known as “social commerce” [12]. Social commerce allows personal social networks to gather information and enables customers to deploy their online purchasing behaviors with a co-shopping experience as they do in retail [13–14]. For example, individuals seeking to purchase something they are unfamiliar with will consult friends and family for advice. The rationale behind these behaviors is supported by theories such as consumer socialization [15], consumer behaviors [16], social learning [17], and social influence [18]. Upon purchasing a popular new product, individuals also tend to have an urge to tell everyone they know about it. The rationale behind this behavior is supported by the theory of hedonic shopping motivations [19]. Social influence plays a key role not only during the shopping process, when individuals may look for advice, but also in the post-purchase stage, when a person may look to share their experience or influence his/her friend’s purchase decision.

Theories on social influence exercised by relevant others on individuals have highlighted that customer preferences are similar to, or influenced by, socially connected friends [18]. Relevant others

[17] refers to people socially proximal to an individual, acting as socialization agents by maintaining constant interactions and informal relationships. Individual socially proximal referents are also people with major emotional significance and those an individual considers as major decision-process influencers, such as family, friends, and colleagues [20].

2.2 Social trust in online environment

Social networks can provide online sellers with new revenue opportunities while also providing consumers with product information and social rewards for sharing [21]. However, e-retailers have difficulty satisfying higher-level customer needs, such as the need for personal interaction while shopping online [22], and addressing trust issues in particular. For online users, gathering information from trustworthy sources and interacting with trustworthy people is crucial. Trust has often been reported as central to e-shopping intention [23].

Due to the vast amount of user interaction and user-generated content on the web, finding trustworthy sources within the vast amount of product/service reviews, ratings, and comments posted by web users presents a challenge [24]. Online reviews can vary widely in quality, and anonymous reviews tend to trigger source credibility issues. Recent research has pointed out that various levels of source anonymity are problematic, since ambiguous authorship often makes source motivations unclear [25–26]. Therefore, finding credible information online involves determining if a source is believable. Official, expert, generic, and unidentified online sources may or may not be in a superior position to provide the most accurate information depending on circumstance.

Possible negative sales impacts due to information quality and source anonymity concerns have driven e-commerce companies to invest in technologies that reduce digital consumer search costs for quality information [27]. Research on online influencers has found that the most trusted information sources for online users are their friends and family, sources that users trust more than any other source of information about products and services [28]. Complementing the e-shopping environment with features that connect customers to their personal network can support the consumer in being advised by friends, thus creating followers. These also engender social recommendation processes within the co-shopping environment [29–30].

2.3 Social recommender system

Just as in an offline environment, individuals are likely to seek suggestions from friends before making an online purchase decision, and friends consistently provide good recommendations [29–30]. In an e-commerce environment, recommender systems significantly support online customers in locating relevant information quickly by suggesting potentially interesting information [31–33].

Beyond recommender systems, automatic systems can recommend items to users, and social relations also play a key role in supporting customer decision making. Social relations provide an independent recommendation source, and various approaches have been proposed for social recommender systems, such as trust propagation and social regularization [34].

As stated by Huang et al. [30], social recommendation occurs when one individual recommends an item to another. The process has gained popularity and success in web applications, such as online sharing and shopping services. Social information from friends has proven useful to the improvement of social recommendation accuracy and trust propagation between socially connected individuals [35].

3 RELATED WORKS

3.1 Social customization systems

Consumers increasingly use social media to exchange product information, and many companies have tried harnessing social media technologies to foster consumer interactions during the use of a mass customization (MC) system. Studies on social product customization have noted that system design can affect consumer evaluations of customized products [2, 3].

However, existing social customization systems differ in terms of how consumers interact and provide input to each other by (i) interacting in user communities, (ii) encouraging consumers to post their own designs, (iii) encouraging consumers to comment on others' designs within a company-led community, (iv) developing community-based MC systems that are embedded within Facebook (e.g., Porsche and Audi community-based MC).

A recent study also described how online sales configurators are connected to social software [10]. Table 1 provides a brief description of these connections and whether they are active within the configuration environment. Specifically, integrated connections refers to connections that enable social interaction features on the same web page where the configuration process takes place. Co-located connections refers to connections that enable social interaction features on a different web page from the one where the configuration process takes place, for example, on the company home page [10].

By drawing a parallel[1] between customers' decision process while shopping [36] and the steps in the configuration process [6], each connection between OSCs and SSW differently supports users during various steps of the configuration/shopping and decision-making processes. Colocation-based modalities are connections between the SSW and OSCs that enable social interactive features to engage interactions with real persons only outside the configuration environment, while integration-based connection modalities are enabled within that environment. Following the parallelism between the configuration/shopping and decision-making process steps, and then combining the processes with colocation- and integration-based connection modality characteristics, makes it possible to determine which modalities support OSC users during each step.

Groups of colocation-based modalities, namely: M1, M5.1, M5.2, M6, and M7.2 (for shortness the M, stands for modality) and integration-based modalities, namely: M2.1, M3, M4, and M8, support users during the information-seeking step (when customers are looking for information about a product for configuration). This step in the customer decision-making process corresponds with the initial idea-development step, which is equivalent to the information-seeking step.

Previous studies on MC systems enriched with social interaction features, social media, and social software have yet to explore how to support users in choosing their own trusted referents to advise them while shopping via OSCs. The present study moves a step forward by proposing a social interaction feature for co-shopping

[1] In Engel et al. [36], customers' decision-making process is structured in the following steps: (a) need recognition (b) alternative evaluation (c) purchase (d) post purchase. Following Franke et al. [6], the configuration process is divided into the following steps: (a) initial idea generation; (b) intermediate evaluation, (c) final configuration evaluation.

between the OSC user and his/her friends following a co-shopping design model described in Kumar et al. [37].

Table 1. OSC-SSW connection modalities and variants

Name	Brief description	Positioning
M1	Icons on the company´s website connect configurator users to the company´s social media (SM) profile(s)	Co-located
M2.1	SM icons enable the user to automatically publish the configurator link on his/her social profiles	Integrated
M2.2	SM icons enable the user to automatically share a complete configuration in the user's social profile(s)	Integrated
M2.3	SM icons enable the user to automatically share a partial configuration on the user's social profile(s) while the configuration is in process	Integrated
M3	Direct browse/upload into the configurator's files shared in the user's social media profile(s)	Integrated
M4	Simplified configurator embedded into the company's social media profile	Integrated
M5.1	A company blog diary provides the user with content published by company representatives (e.g., information about brands, events, sponsorships)	Co-located
M5.2	A company blog post provides the user with additional information not available in the configurator environment, provided by the company itself and/or by other blog users	Co-located
M6	A company discussion forum connects the configurator user to his/her online contacts	Co-located
M7.1	Email sends the complete configuration to the user's online contacts	Integrated
M7.2	The company´s email as a company–customer-service channel	Co-located
M8	Instant message services connect the configurator user to company representatives	Integrated

Source: adapted from Grosso et al. [10]

To study users' experience when they are directly engaged in the design of their products and thus in the product value creation is especially relevant because of the specificity of the product self-design process. Specificity in terms of the customer experience refers to customers' decision-making process, and a number of choice tasks is required before an optimal solution is produced. Thus, to deploy this specific process and the different choice tasks, the end-user may need support for their decision-making process through contact with real persons in addition to the support provided by the product configurator systems and/or recommender systems [31–34].

3.2 Users' need to interact digitally with others

A recent study [11] that explored configurator users' need for digital interaction with others reported interesting data that are useful for our current reasoning. In particular, it reported that the majority of OSC users experienced the need for social interaction in their configuration experiences (88%) (see Fig. 1). Only 4% of OSC users did not experience a desire to interact with real people in any form during configuration experiences, while 8% did not provide a definitive answer regarding whether they perceived this need to be relevant or not.

The intent to interact with user contacts, reported in 75% of cases, underlines OSC-user demand for consulting referents from their circles and confirms the role of relevant others as trustworthy sources of social information and as referents aware of user tastes and habits [18–20], enabling preferential recommendations [30].

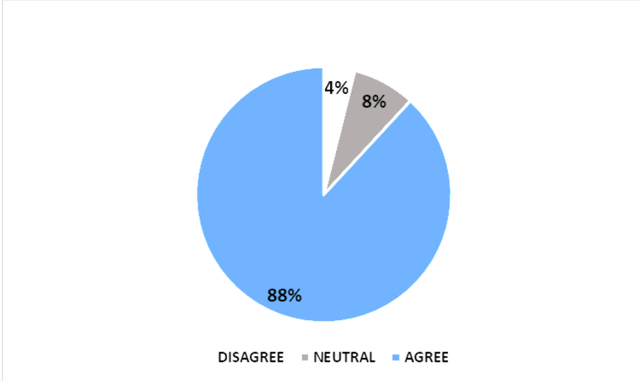


Figure 1. Experienced need for social interaction
Source: Grosso and Forza [11]

Results on users seeking interaction or advice instead of simply facing the configuration experience are shown in Fig. 2. The image indicates that the need to engage in human-assisted interactions varies depending on which type of referent is involved in the interaction (the "with whom" factor). This is not surprising given that different referents provide different kinds of information and support, but it raises the question on what determines configurator users' need for social interaction. Moreover, there is a surprising lack of OSC-SSW connections that enable interactions between users and their contacts. To fill this gap, the present study provides reasoning for a possible OSC-SSW connection and its social interactive features.

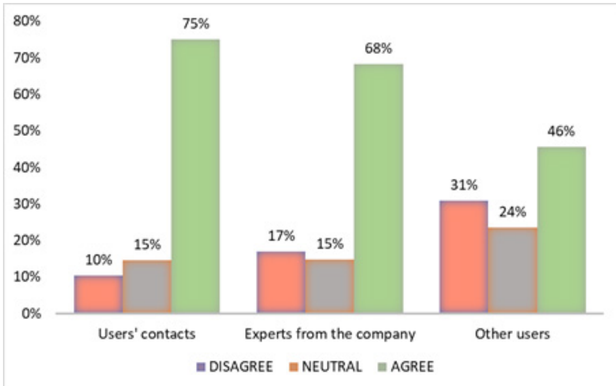


Figure 2. From whom users seek social interaction
Source: Grosso and Forza [11]

4 DISCUSSION

Previous studies have shown that users look for interaction with contacts from their personal network [11]. This result is consistent with the role of social influence in supporting customers' decision making [18, 30] and the need for social interaction that guides customers during their shopping processes [20]. The surprising data is the lack of OSC-SSW connection modalities that enable digital social interaction features between users and their contacts within the configuration environment. The connection modalities that support user–contact interactions (M2.1, M2.2, M2.3, and M7.2) enable users to share items to their Facebook account and via email (M7.2); thus, any interactions with others happen only outside of the configurator environment (i.e., on Facebook and/or via email). It follows that users must suspend their configuration process in order to be redirected to an external environment where they can be advised by people they trust. A parallel in retail shopping would a shopkeeper inducing customers to leave a shop to seek and obtain required support from a preferred referent.

Previous studies on social customization systems have investigated features enabling peer input from other consumers during the product configuration process [2–5]. The present study shows, on the one hand, the role that social recommendations between OSC users and friends can play in replacing social interaction while shopping. On the other hand, the lack of a feature to connect OSC users with contacts from their personal social network in real-time conversation has not been addressed.

The conceptual speculation of the present study proposes a possible end-user interface enabling one-to-one or one-to-many social interactions between users and their friends during product configuration. The feature, referred to as the "K-interactive feature," defines K as the key role connecting OSC users with relevant others. The feature consists of sending an email, link, or Facebook message invitation from within the configuration environment to someone outside of it; once the invitation is accepted, users and the invited person(s) can start a real-time conversation.

Once a user decides to invite a person from his/her network of friends, the environment provides options to offer an invitation (see Figs. 5–6 for implementation examples). As described in a previous study [37], real-time online co-shopping can be implemented using PUSH AJAX technology. As explained by Kumar et al. [37]: PUSH AJAX uses a technique called Long Poll to update server data to clients. Once a client establishes the connection, the server holds back the connection until the data is available, at that time the server sends the data through the held connection. Once, the co-browsing ends, the server closes the connection. "This way, real-time communication is achieved from server to client" (p. 474). Following Kumar et al. [37] instruction of PUSH AJAX enhancement and implementation for co-shopping environment, a speculation on its implementation into an OSC environment is provided. As for the implementation of real-time co-shopping design proposed by Kumar et al. [37], its implementation into a configuration environment requires:

- A client-side collaborative window with HTML iFrames
- An AJAX program
- A server side PUSH AJAX implementation
- A chat service

The client-side, or leader side, collaborative window consists of three HTML frames: (1) the first iFrame loads the pushed URL from the server and has provision to share the configurator URL with friends. (e.g., via email, link, or Facebook Messenger); (2) the second frame has provision to invite users' friends; and (3) the third frame has a chat window.

In Kumar et al.'s [37] co-shopping environment design, all the communications from client to server happen via AJAX routines. The server side PUSH AJAX is implemented using AJAX PUSH APIs, where AJAX PUSH empowers the server to push data from server to client and update any part of the client's page at any time (Figure 3).

An exemplification of how K-interaction feature graphics could appear to configurator users is provided in Figures 4–9. Figure 4 depicts how such an invitation button could be visually represented in the configuration environment. The configurator user initiates an invitation to his/her friends by clicking on the "Invite a friend" button (Fig. 4 and Fig. 5). An AJAX request goes to the server. The AJAX request contains the email address of the friend to be invited (for invitation via email) and the UR§§§§L of the configurator page (for invitation via link) (Figure 6).

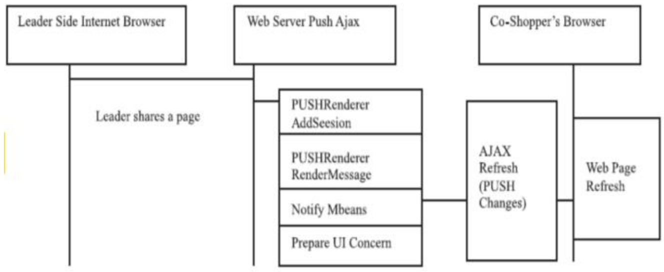


Figure 3. Overview of collaborative co-shopping feature designed by Kumar et al. [37]

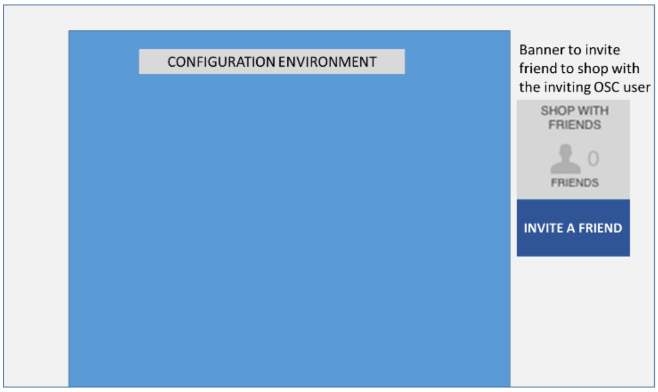


Figure 4. Configuration environment containing an "Invite a friend" button
Source: Our elaboration

Figure 5. OSC user invitation via shared configurator link
Source: Our elaboration

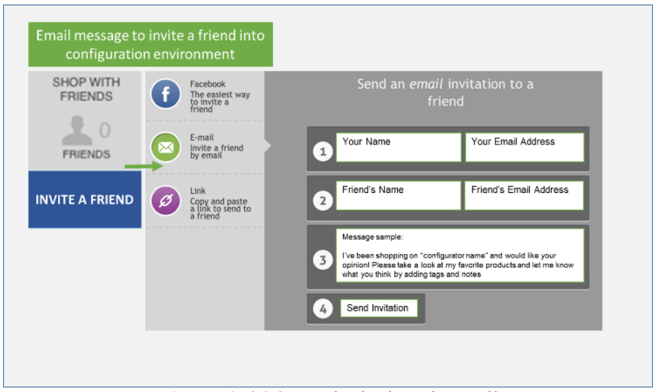


Figure 6. OSC user invitation via email
Source: Our elaboration

To invite Facebook friends, the AJAX request contains Facebook IDs of friends to be invited via Facebook Messenger. The server manipulates the request by pushing the URL on the HTML frames of each friend in an asynchronous manner using PUSH AJAX [37]. Sending the configurator URL via Facebook Messenger allows the user's friend on Facebook to join the configuration environment (Fig. 7).

Facebook has been chosen both because it enables a high number of social relations within its platforms and because it provides multiple ways for online retailers to offer a more social purchase experience by integrating Facebook features either into independent retail sites or (increasingly) by allowing retailers to operate within Facebook itself [1, 38, 39]. Facebook's new features enhance social networking inside and outside of the platform itself. Facebook users can make purchases using their Facebook credentials and can check out without any other form of registration to finalize sales. These trends indicate that the potential power of interaction is now being raised to a new level as the attention and trust of consumers has shifted towards social networks.

All invitations enable a screen sharing option and a chat room where users and friends can interact in real time, as shown in Fig. 8. The initial banner providing the invitation option will then provide users with information about when the friend has joined the configuration environment to start their digital social interactions (DSI).

After logging in, a configurator user and his/her friend(s) can see a user interface that consists of the two frames mentioned above: (a) the URL of the configurator page to be joined, (b) a chat window where they can communicate with each other in real time. As designed by Kumar et al. [37], AJAX PUSH can push a message to a group of clients asynchronously. Collaborative multi-user applications for the web can be easily built using the AJAX PUSH technique.

5 CONCLUSIONS

The interface proposed for users to add social interaction to their customized product purchase via OSCs will trigger social recommendations by allowing users to collect social interactive feedback from trusted sources. Figure 9 shows a simulation of a product configurator supporting the K-interactive feature. While the proposed OSC-SSW connection is not yet deployed by an actual sales configurator, its aim is to provide the following innovative characteristics: (1) OSC users can select who to invite from their contacts, and they are not limited to interacting only with other customers; (2) users are encouraged to invite friends from their social networks to share the configuration environment; and finally, (3) the digital social interaction supported by the proposed feature can be enabled "on demand," implying that OSC users can invite friends when needed during configuration.

The flow of digital social interaction enabled by current OSC-SSW only supports OSC users in sharing their configured product within the configurator website (i.e., where they post their own designs and comment on other designs within a company website or company-led community) or in a social network environment outside the configurator (e.g., on a Facebook profile). The proposed feature is limited to a conceptual speculation and requires study support to determine its software design feasibility. The Facebook invitation feature also requires investigation to address possible policy or company constraints.

E-retailers have difficulty satisfying customers' needs for personal interaction while shopping online [22]. The present study provides design-related insights to enhance users' involvement with social product configuration systems. Notably, the present study is performed from the perspective of scholars who investigate the user experience of configurators connected to social software. Consequently, the proposed idea of interactive features would greatly satisfy OCS users, but the possibility of developing the technology is not yet tested or developed. Development and usability characteristics can be addressed in future research on the feasibility of proposed social interaction features such as Long Poll disadvantages [37] and can explore if models from group recommender systems [8] can provide a more appropriate model to design the K-interaction feature.

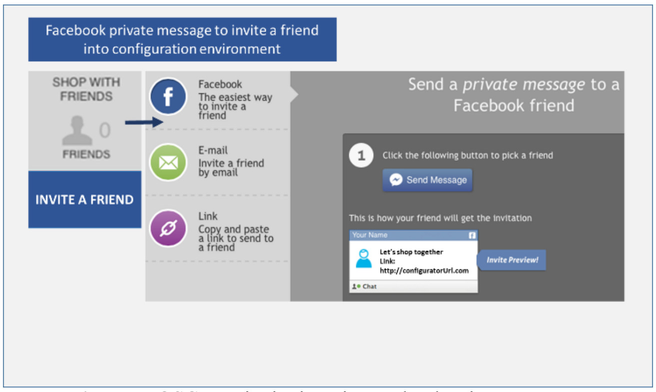


Figure 7. OSC user invitation via Facebook private message
Source: Our elaboration

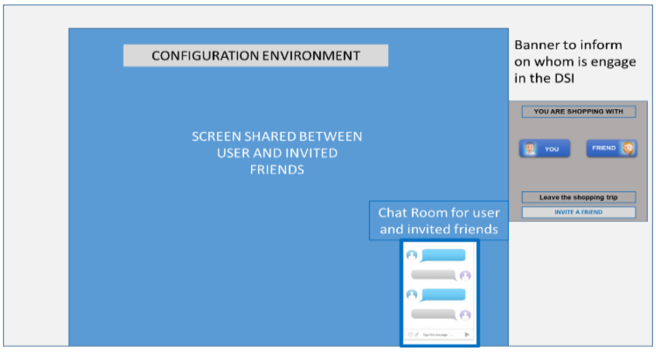


Figure 8. Example of a social interaction interface after user contacts are invited. Source: Our elaboration

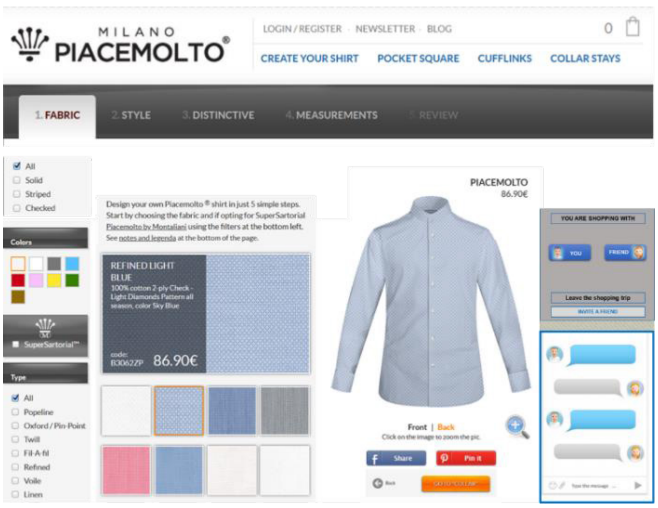


Figure 9. Example of product configurator implemented with the proposed K-interactive feature.
Source: Our elaboration

The present study addresses the user experience in the domain of B2C configurator applications. Further research can extend the investigation to the B2B domain to support end users in online configurations and transactions linked to this process. Research addressing how to improve human-assisted transactions in web-mediated environments is especially relevant during conditions of emergency when social and economic transactions are restricted to an online environment.

REFERENCES

[1] A. J. Kim and K. K. Johnson, "Power of consumers using social media: Examining the influences of brand-related user-generated content on Facebook," *Computers in Human Behavior,* **58**, 98–108, (2016).

[2] C. Hildebrand, G. Häubl, A. Herrmann, and J. R. Landwehr, "When social media can be bad for you: Community feedback stifles consumer creativity and reduces satisfaction with self-designed products," *Information Systems Research,* **24**, 14–29, (2013).

[3] T. Schlager, C. Hildebrand, G. Häubl, N. Franke, and A. Herrmann, "Social product-customization systems: Peer input, conformity, and consumers' evaluation of customized products," *Journal of Management Information Systems,* **35**, 319–349, (2018).

[4] C. P. Moreau and K. B. Herd, "To each his own? How comparisons with others influence consumers' evaluations of their self-designed products," *Journal of Consumer Research,* **36**, 806–819, (2009).

[5] L. B. Jeppesen, "User toolkits for innovation: Consumers support each other," *Journal of Product Innovation Management,* **22**, 347–362, (2005).

[6] N. Franke, P. Keinz, and M. Schreier, "Complementing mass customization toolkits with user communities: How peer input improves customer self-design," *Journal of Product Innovation Management,* **25**, 546–559, (2008).

[7] F. Piller, A. Vossen, and C. Ihl, "From social media to social product development: the impact of social media on co-creation of innovation," *Die Unternehmung-Swiss Journal of Business Research and Practice,* **66**, 7–27, (2011).

[8] A. Felfernig, L. Boratto, M. Stettinger, and M. Tkalčič, *Group recommender systems: An introduction*. Springer, Cham, Switzerland, 2018.

[9] A. Felfernig, L. Boratto, M. Stettinger, and M. Tkalčič, "Personality, Emotions, and Group Dynamics," in *Group Recommender Systems*, ed: Springer, 2018, pp. 157–167.

[10] C. Grosso, C. Forza, and A. Trentin, "Supporting the social dimension of shopping for personalized products through online sales configurators," *Journal of Intelligent Information Systems,* **49**, 9–35, (2017).

[11] C. Grosso and C. Forza, "Users' Social-interaction Needs While Shopping via Online Sales Configurators," *International Journal of Industrial Engineering and Management,* **10**, 139–154, (2019).

[12] Z. Huang and M. Benyoucef, "From e-commerce to social commerce: A close look at design features," *Electronic Commerce Research and Applications,* **12**, 246-259, (2013).

[13] K. Z. Zhang and M. Benyoucef, "Consumer behavior in social commerce: A literature review," *Decision Support Systems,* **86**, 95–108, (2016).

[14] Y. Wang and C. Yu, "Social interaction-based consumer decision-making model in social commerce: The role of word of mouth and observational learning," *International Journal of Information Management,* **37**, 179–189, (2017).

[15] G. P. Moschis and G. A. Churchill Jr, "Consumer socialization: A theoretical and empirical analysis," *Journal of Marketing Research,* **15**, 599–609, (1978).

[16] M. R. Solomon, D. W. Dahl, K. White, J. L. Zaichkowsky, and R. Polegato, *Consumer behavior: Buying, having, and being*. Boston, MA:Pearson, 2017.

[17] A. Bandura and R. H. Walters, *Social learning theory* vol. 1: Prentice-hall Englewood Cliffs, NJ, 1977.

[18] M. K. Lee, N. Shi, C. M. Cheung, K. H. Lim, and C. L. Sia, "Consumer's decision to shop online: The moderating role of positive informational social influence," *Information & Management,* **48**, 185–191, (2011).

[19] M. J. Arnold and K. E. Reynolds, "Hedonic shopping motivations," *Journal of retailing,* **79**, 77–95, (2003).

[20] T. L. Childers and A. R. Rao, "The influence of familial and peer-based reference groups on consumer decisions," *Journal of Consumer Research,* **19**, 198–211, (1992).

[21] H. Kim, K.-S. Suh, and U.-K. Lee, "Effects of collaborative online shopping on shopping experience through social and relational perspectives," *Information & Management,* **50**, 169–180, (2013).

[22] M. B. Kolesar and R. W. Galbraith, "A services-marketing perspective on e-retailing: implications for e-retailers and directions for further research," *Internet Research,***10**, 5, 424-438 (2000).

[23] M. M. Goode and L. C. Harris, "Online behavioural intentions: an empirical investigation of antecedents and moderators," *European Journal of Marketing,* **41**, 5/6, 512-536(2007).

[24] J. Golbeck, *Computing with social trust*, London, Springer Science & Business Media, 2008.

[25] M. J. Metzger, A. J. Flanagin, and R. B. Medders, "Social and heuristic approaches to credibility evaluation online," *Journal of communication,* **60**, 413–439, (2010).

[26] R. D. Lankes, "Credibility on the internet: shifting from authority to reliability," *Journal of Documentation,* vol. 64, pp. 667-686, 2008.

[27] P.-Y. Chen, S.-y. Wu, and J. Yoon, "The impact of online recommendations and consumer feedback on sales," *ICIS 2004 Proceedings,* 58, (2004).

[28] D.-H. Shin, "User experience in social commerce: in friends we trust," *Behaviour & Information Technology,* **32**, 52–67, (2013).

[29] J. Tang, X. Hu, and H. Liu, "Social recommendation: a review," *Social Network Analysis and Mining,* **3**, pp. 1113–1133, (2013).

[30] J. Huang, X. Cheng, J. Guo, H. Shen, and K. Yang, "Social Recommendation with Interpersonal Influence," *Proceedings of the 2010 conference on ECAI 2010: 19th European Conference on Artificial Intelligence*. IOS Press, NLD., 601–606, (2010).

[31] A. Felfernig, E. Teppan, and B. Gula, "Knowledge-based recommender technologies for marketing and sales," *International Journal of Pattern Recognition and Artificial Intelligence,* **21**, 333–354, (Mar 2007).

[32] A. Falkner, A. Felfernig, and A. Haag, "Recommendation technologies for configurable products," *Ai Magazine,* **32**, 99–108, (2011).

[33] J. Tiihonen and A. Felfernig, "Towards recommending configurable offerings," *International Journal of Mass Customisation,* **3**, 389–406, (2010).

[34] H. Ma, D. Zhou, C. Liu, M. R. Lyu, and I. King, "Recommender systems with social regularization," in *Proceedings of the Fourth ACM International Conference on Web Search and Data Mining*, 287–296, (2011).

[35] M. Jamali and M. Ester, "Trustwalker: A random walk model for combining trust-based and item-based recommendation," in *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 397–406, (2009).

[36] J. F. Engel, R. Blackwell, and P. Miniard, *Customer behavior*, Hinsdale, IL: Dryden, 1990.

[37] B. Kumar, K. Abhishek, A. Deepak, and M. Singh, "Implementation of interactive real time online co-shopping using Push AJAX," *Procedia Computer Science,* **89**, 473–482, (2016).

[38] T. Robinson, C. Callahan, K. Boyle, E. Rivera, and J. K. Cho, "I ♥ FB: A Q-Methodology Analysis of Why People 'Like' Facebook," *International Journal of Virtual Communities and Social Networking (IJVCSN),* **9**, 46–61, (2017).

[39] L. Harris and C. Dennis, "Engaging customers on Facebook: Challenges for e-retailers," *Journal of Consumer Behaviour,* **10**, 338–346, (2011).

Complexity of Configurators Relative to Types of Outputs

Sara Shafiee[1] and **Eike Schäffer** and **Lars Hvam**

Abstract.[1] Industrial gained and potential benefits of configurators are explored and researched to great extent in the literature. Moreover, configurators generate most of the needed documents automatically, which can be an effective solution to reduce the number of manual tasks. The generated documents include variety of customized knowledge such as BOM (bills of materials) or 2D and 3D figures. However, the influence of number and type of generated documents on the complexity of configurator has not been addressed in the literature. This paper aims to study the popular generated documents from configurators in an engineering company. Moreover, this study analyses the influence of different types of the generated documents on configurators' complexity. The research method is survey-based followed with interviews where the unit of analysis is based on operating configurators within an engineering company.

1 INTRODUCTION

Configurator projects in engineer to order (ETO) contexts fall into the category of IT projects with high level of complexity [1]. Owing to the increasing concerns about customer needs in the current competitive market, the identification and incorporation of customer requirements into product configuration designs have raised the interest of both researchers and practitioners [1], [2]. Product configurators include a knowledge base with information about product features, product structure, production processes, costs and prices [3], [4], empowering the companies by simulating the work that normally carried out by product experts, such as sales staff and engineers. Configurators are used to support design activities throughout the customization process in which a set of components and connections are pre-defined and constraints are used to prevent infeasible configurations [5]–[7].

Several studies report the gained benefits from the use of product configurators in engineering-oriented companies in the literature [8]. Configurators can bring substantial benefits, such as shorter lead times for generating quotations, fewer errors, increased ability to meet customers' requirements regarding product functionality, use of fewer resources, optimized product designs, co-creation and product innovation, less routine work and improved on-time delivery [9]–[16].

On the other hand, many companies also experience great difficulties in realizing such benefits within reasonable investment on configurators, which in many cases makes them abandon such projects [17]. Some of these challenges can be mentioned such as product modelling [18], documentation in configuration systems [19], system design and development [20][21], configuration process [1], analysis of benefits, risks, failures and impacts [8], [17].

Configurators used to support the engineering processes are considered more complex [19], [22] due to the expected documents and outputs from system. Besides, Configurators are mainly popular due to the automatic generation of the documents such as BOM, CAD (computer-aided design) 2D or 3D drawings, sales summaries, full quotations and others. However, there are no literature to compare the level of complexity in configurators related to the types of generated outputs from the system. Normally, the scope of the configurator is strongly related to the expected outputs from the system.

In this paper, the complexity of the configurator is calculated based on the number of inputs, attributes and constrains coded inside the system. By analysing the complexity in terms of types of the outputs, this research will provide a better understanding of the influence of the outputs and generated documents on the complexity of the configurators. Complexity of configurators influences the performance of the whole system as well as the efforts and investments related to system development and maintenance. Nevertheless, companies sometimes strategically decide to increase complexities to achieve the benefits such as generating more values and documents out of one configurator. This paper therefore aims to provide a more precise understanding on the different output types influencing complexity of configurators by providing answers to the following research questions (RQs):

RQ 1**:** What are the popular generated documents in ETO companies?

RQ 2**:** What are the influences of different types of generated outputs on the complexity of the configurators?

To answers to the RQs, a survey followed with interviews is conducted. This includes analysis based on one company where the unit of analysis is based on operating configurators within the company. This company developed and implemented 159 configurators in operation. The structure of the paper is as follows. Chapter 2 discusses the previous related works, and chapter 3 explains the research method. Chapter 4 highlights the main results of the research, and chapter 5 discusses the results in relation to the RQs and presents the conclusion.

2 RELATED WORKS

This section aims to provide the background for the study. Section 2.1 discusses configurators and automation process, and Section 2.2 provides definition of configurators scoping process. Section 2.3 discusses the complexity of configurators and the available contributions to suggest different solutions for calculating the complexity.

[1] Mechanical Engineering, Technical University of Denmark, email: sashaf@dtu.dk

2.1 Configurators and automation processes

The underlying IT structure of a configurator consists of configuration knowledge representation and reasoning, conflict detection and explanation, and finally an user interface [23]. Configurators can be applied as standalone software, as well as data-integrative and application-integrative systems [24]. Data-integrative configurators can be used to avoid data redundancies and application-integrative configurators allow for communication across different applications (e.g. computer-aided design (CAD) drawings can be generated based on the output of a configurator) [24].

In terms of data integration for configurators, common sources for master data can be found in enterprise resource planning (ERP) systems that often define a production-relevant view of the material. This is required for the assembly process, product data management (PDM) and product lifecycle management (PLM) systems, which are used to maintain production relevant data. Finally, product information management (PIM) systems are used to maintain sales-relevant data [25]. Different configurators can be integrated in terms of, for example, sales and engineering configurators [26]. At a higher level of aggregation, configurators can be integrated into suppliers systems to retrieve the required data from the configuration processes [27].

2.2 Scoping of configurators

Determining the scope of configurators is a knowledge management as one of the main challenge for industries and researchers [6], [28]. Moreover, clarifying the knowledge requirements for the entire project gives the team the opportunity to make intelligent decisions from the early phases of the project [29]. Furthermore, in the early phases of the configuration projects, the scope of the products sheds light on project goals and outputs, process objectives and requirements from involved stakeholders, IT architectural requirements as well as integrations and connectivity of the whole system, etc. [30].

One method of clustering the knowledge in configurators is to determine output knowledge according to stakeholder requirements and sub-categorize them step by step [6], [30]. Listing the sources and resources of the knowledge helps to group the knowledge and delegate the tasks to resources in an efficient manner to both increase the performance and decrease the risks of failures [31]. Depending on the resources, the knowledge might be explicit, and come from the company's internal documentation systems, or tacit, and come from domain experts [32]. However, in majority of the cases, the tacit and explicit knowledge are mixed and has to be clarified, condensed, standardized and visualized. In summary, the scoping and knowledge management process in configurators depends highly on the output documents expected by stakeholders to be generated from the system.

2.3 Complexity of configurators

Industrial companies are interested in calculating the complexities from different configurators and predicting the resource and time consumption required for developing and maintaining the configurator based on the system complexity [19]. The calculations not only clarify the required investments but also the expected gained return on investment from the configurator as well as future beneficial investments in development and maintenance.

Configurators are used to support the product or service configuration process. The configuring process consists of a set of activities that involve gathering information from customers and generating the required product specifications [26], [33]. The product configuration process can be divided into sales and technical configuration processes [34], [35]. The sales configuration process is concerned with identifying products that fulfil customers' needs and determining the main characteristics of the products [34]. The technical (engineering) configuration process, on the other hand, is concerned with generating documentation for the product based on the input gathered during the sales phase [34]. Another dimension of the configuration process is production configuration [35], [36]. Based on the coded data inside the configurator and the expected outcomes, these configurators have different levels of complexities. However, measuring the accurate level of complexity for the configurators is still challenging for companies.

To measure the complexity of configurators, Brown et al. [37] categorize them into three major components; (1) execution complexity, (2) parameter complexity, and (3) memory complexity. Execution complexity covers the complexity involved in performing the configuration actions that make up the configuration procedure and the memory complexity refers to the number of parameters that system manager must remember. In this paper, the parameter complexity is the most important, as it measures the complexity of providing configuration data to the configurator by users and external systems during a configuration procedure [19], [37]. Therefore, it was decided to focus on parameters complexity to determine the complexity of the configurators. In this article, the parameter complexity is determined based attributes and rules included in the configurators.

3 RESEARCH METHOD

The chosen research method for this study is a within company survey followed by interviews. In this study, we analyzed only one company to get an in- depth knowledge about the configuration setup and to compare the complexity of the configurators within the same settings. This single company includes 159 different configurators. The same modelling paradigms are used for all the configurators in the company, which allows comparison of the complexity. The company therefore has an extensive experience from working with configurators. The case company introduced in the study has a world leading position in providing process plants and related equipment for industrial use. The example of the product types could be named as all the small products and technology used in cement plants from the chemical process to mechanical and electrical instruments. The company provides highly engineered products and engineering consultancy for both installation and maintenance. In summary, the case company is chosen because it:

- offers highly engineered and complex products;
- had frequently implemented configurator projects – including projects with frequent and more sporadic maintenance efforts;

- had a database including all the data regarding complexity of configurators, number and types of outputs generated from the configurators; and
- offered a unique level of access to project data.

The questioner was emailed to the company and the interview sessions was planned. In order to get the full understanding of all 159 configurators, the research team was supported through a project manager for one week to gather and evaluate the required data. The data was gathered from both internals systems at the company, documentations, and through survey and interviews, verified by the project manager. The configuration team also gave presentations and knowledge about the development and maintenance of the configurators at the case company.

The configurator were grouped according to specification processes they supported; sales, sales and engineering, and engineering. A limitation of the data is that the majority of the configurators are used to support the engineering processes (75%), and sales and engineering processes (19%) while there are few configurators used to support only sales processes (3%) and finally configurators used to support other processes (such as manufacturing or marketing process) are (2%). Figure 1 illustrates the percentage of different types of configurators at the case company.

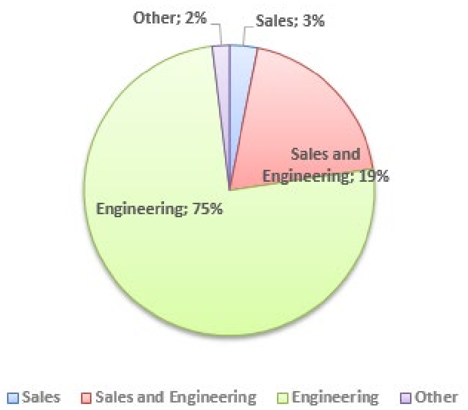


Figure 1. Percentage of types of configuration system in use at the company

4 RESULTS

In this chapter the main result from the survey are presented aligned with both the RQs introduce in the paper.

Within the first section, different types of the outputs generated from different configurators were elaborated at the case company. Some of the configurators generate more than four different documents. The generated documents have a vast range of varieties including sales and production (Prod.) Bills of Materials (BOM), 2D and 3D CAD documents, quotations and etc. The group that has been called 'other' belongs to the documents which has been generated from the other internal home-made IT systems at the case company. The different types of outputs as well as the most popular ones are analyzed (RQ 1). The second section elaborates the influence of different types of generated outputs on the complexity of the configurators (RQ 2).

4.1 Types of generated documents for the configurators

This chapter analyses different types of output documents from configurators at the case company and studies how frequent they were used. First, we extracted different generated outputs at the case company and group them. Second, the number of configurators related to each group of documents has been measured. Figure 2 shows the percentage outputs generated from configurators at the case company.

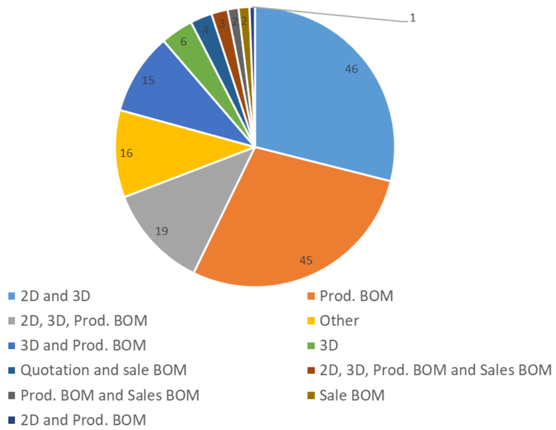


Figure 2. Percentage of different types of outputs generated from configurators at the company

As can be seen in figure 2, there are many configurators generating 2D and 3D CAD documents as well as the production BOM. Based on the numbers on the figure 2, the configurators generating 2D and 3D plus production BOM are in the next level and belong to the configurators that generates three different documents. The figure demonstrates that the most popular documents are 3D and 2D CAD drawings. After CAD drawing documents, production and sales BOM are the most popular documents to be generated automatically from the configurator at the case company.

4.2 Complexity measurement in relation to the types of generated documents

Table 1 illustrates the complexity parameters of the configurators relevant to different types of generated documents. As it is demonstrated in table 1, the total complexity is calculated for different types or mixed of different outputs.

According to table 1, figure 3 is demonstrated to visualize the data in table 1. This data shows that the number of configurators and types of generated documents and the sum of the complexity but we cannot understand the level of complexity for each group of documents. Hence, we calculate the numbers (No.) in table 2.

Table 1. Complexity measurement in relation to the types of generated documents from configurators

Generated documents	Number of configurators	No. Rules	No. Attributes	No. Input Fields	Total complexity
Prod. BOM and Sales BOM	2	1568	2599	627	4793
2D, 3D, Prod. BOM	19	635	1122	180	1937
2D and 3D	46	538	798	215	1551
Other	16	352	454	573	1379
3D	6	483	343	236	1062
Sale BOM	2	240	310	448	998
Prod. BOM	45	334	381	191	906
Quotation and sale BOM	4	210	307	293	810
3D and Prod. BOM	15	278	303	125	706
2D, 3D, Prod. BOM and Sales BOM	3	228	257	146	562
2D and Prod. BOM	1	33	32	27	92

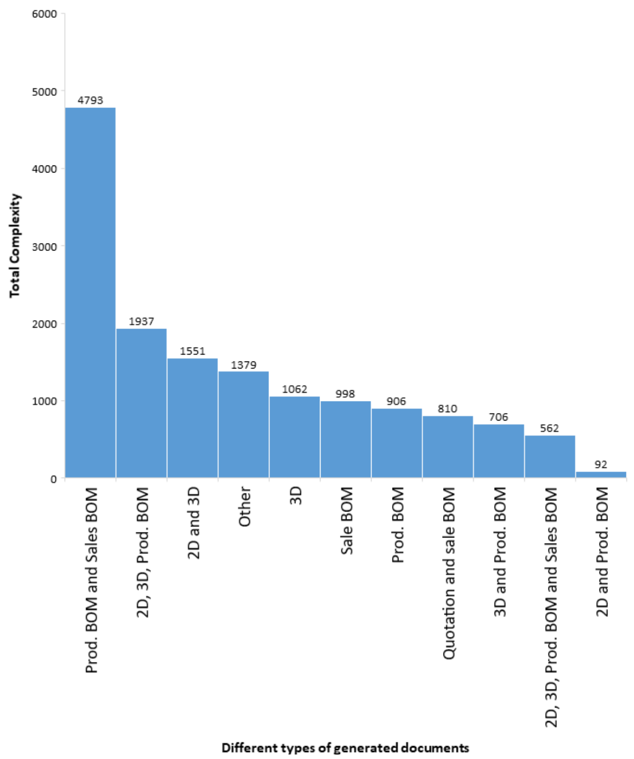


Figure 3. Complexity measurement in relation to the types of generated documents from configurators

Table 2 and accordingly figure 4 demonstrate the average of complexity for each document categories generated from configurators. The reason for generating table 2 and figure 4 is to illustrate the average complexity as cumulative numbers of attributes and rules for different types of documents.

Table 2. The average of complexity per generated documents types for one configurator at the case company

Generated documents	Average of complexity
Prod. BOM and Sales BOM	2396.5
Sale BOM	499
Quotation and sale BOM	202.5
2D, 3D, Prod. BOM and Sales BOM	187.3
3D	177
2D, 3D, Prod. BOM	102
2D and Prod. BOM	92
Other	86.2
3D and Prod. BOM	47.1
2D and 3D	33.7
Prod. BOM	20.1

As demonstrated in figure 4, the average of complexity is the highest when the configurator generates the group of documents of production and sales BOM and in the next level is the Sales BOM. In the third level of the most complex configurators are the ones generating the quotations and sales BOM. Afterwards, the documents generated from CAD such as 2D and 3D documents are located. Thereafter, the numbers shows that BOM documents create the highest amount of complexity in the configurator, while 2D and 3D documents create less complexity in the structure of the configurator. Moreover, 2D and 3D documents and also production and sales BOMs are the most popular documents generated from the configurator at the case company due to the time and resources saved due to this automation.

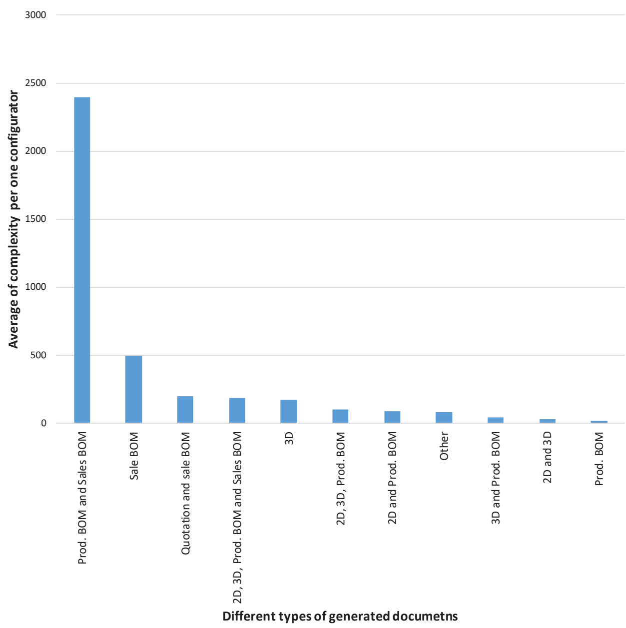


Figure 4. The average of complexity per generated documents types for one configurator at the case company

5 DISSCUSSIONS AND CONCLUSIONS

This study provides insights into the type of generated documents in an engineering company and the influence on the complexity of the configurator. The complexity is analysed based on parameters, which consists of cumulative numbers of attributes and rules. First, we extracted different generated outputs at the case company and group them. Second, the number of configurators related to each group of documents has been calculated. Thereafter, the complexity is calculated based on the number of attributes and rules in the configurator.

In the literature, it is discussed how configurators are used to support the engineering processes are considered more complex [19], [22] due to the expected documents and outputs from system. However, the literature does not explain what are the most popular documents generated from the configurator and how these documents influence the complexity of the configurators. The results provided in the present article aim to contribute to the field of configurators' complexity and the factors influencing them. In this article, we analysed the influence of the generated outputs from the system on configurators' complexity. This is an important topic not only for the research community but also for practitioners. The results show that a difference can be found in relation to the complexity by analysing different types of generated documents.

The first research question in this study aims to identify the most popular and common generated documents. The results demonstrates that 2D, 3D documents and BOMs are the most common generated documents which can be due to the benefits. Normally creating 2D and 3D drawing based on a specific configuration of the system is a time consuming task. Moreover, preparing the bills of material including details and prices based on a specific order is a teady task. Hence, generating these types of documents leads to considerable benefits for the company.

The second research question aims to analyse the relationship between the types of generated documents and the complexity of the configurators. Our analysis shows that the highest is the parameters complexity when the configurator generates the group of documents of production and sales BOM and in the next level is located the Sales BOM. In the third level of the most complex configurators are the ones generating the quotations and sales BOM. Then in lower levels of complexity, the documents generated from CAD such as 2D and 3D documents are located.

The results shows that the most popular are CAD drawing and BOMs while BOMs create the highest level of complexity in the configurator. The reason can be explained technically. While generating CAD documents from configurator, we integrate the system to the CAD and make sure that the names are mapped between the systems. However, creating BOMs requires lots of rules and attributes including the whole list and details of materials that can be millions. Thereafter, the rules for different calculations such as price calculations have to be considered. Therefore, it is rational to conclude that BOMs create very high complexity inside the configurator. Besides, automating the generation of BOMs reduces considerable amount of time and resources from industries.

The results can be very beneficial for strategic planning at ETO companies to decide on scoping of configurators. This research provide a good estimation to compare the complexity of the configurators with different generated outputs.

The result presented in the paper is based on answers from one case company. However, this large case company with 159 different configurator can represent most of the engineering companies. This is thought to provide a valuable insight as by studying one company an in-depth knowledge about the configuration setup could be accessed. More companies will be contacted in the future, which will allow cross functional comparison for the further studies. Considering and analyzing different influencing factors on the complexity level of the configurators and comparing the results can shed lights on potential practical obstacles during development and implementation of the configurators.

REFERENCES

[1] S. Shafiee, "Conceptual modelling for product configuration systems," Technical University of Denmark, Denmark, 2017.

[2] S. Zhao, Q. Zhang, Z. Peng, and Y. Fan, "Integrating customer requirements into customized product configuration design based on Kano's model," *Journal of Intelligent Manufacturing*, vol. 31, no. 3, pp. 597–613, 2019.

[3] C. Forza and F. Salvador, *Product Information Management for Mass Customization*. London: Palgrave Macmillan UK, 2006.

[4] S. Shafiee *et al.*, "Modularisation strategies in the AEC industry : a comparative analysis Modularisation strategies in the AEC industry : a comparative analysis," *Architectural Engineering and Design Management*, vol. 0, no. 0, pp. 1–23, 2020.

[5] A. Felfernig, G. E. Friedrich, and D. Jannach, "UML as domain specific language for the construction of knowledge-based configuration systems," *International Journal of Software Engineering and Knowledge Engineering*, vol. 10, no. 4, pp. 449–469, 2000.

[6] S. Shafiee, K. Kristjansdottir, L. Hvam, and C. Forza, "How to scope configuration projects and manage the knowledge they require," *Journal of Knowledge Management*, vol. 22, no. 5, pp. 982–1014, 2018.

[7] P. Piroozfar, E. R. P. Farr, L. Hvam, D. Robinson, and S. Shafiee, "Configuration platform for customisation of design, manufacturing and assembly processes of building façade systems: A building information modelling perspective," *Automation in Construction*, vol. 106, no. July, p. 102914, 2019.

[8] A. Haug, S. Shafiee, and L. Hvam, "The causes of product configuration project failure," *Computers in Industry*, vol. 108, pp. 121–131, 2019.

[9] A. Haug, L. Hvam, and N. H. Mortensen, "The impact of product configurators on lead times in engineering-oriented companies," *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, vol. 25, no. 02, pp. 197–206, Apr. 2011.

[10] L. Hvam, A. Haug, N. H. Mortensen, and C. Thuesen, "Observed benefits from product configuration systems," *International Journal of Industrial Engineering-Theory Applications and Practice*, vol. 20, no. 5–6, pp. 329–338, 2013.

[11] A. Trentin, E. Perin, and C. Forza, "Product configurator impact on product quality," *International Journal of Production Economics*, vol. 135, no. 2, pp. 850–859, Feb. 2012.

[12] L. L. Zhang, "Product configuration: a review of the state-of-the-art and future research," *International Journal of Production Research*, vol. 52, no. 21, pp. 6381–6398, Aug. 2014.

[13] A. Trentin, E. Perin, and C. Forza, "Overcoming the customization-responsiveness squeeze by using product configurators: Beyond anecdotal evidence," *Computers in Industry*, vol. 62, no. 3, pp. 260–268, Apr. 2011.

[14] Y. Wang, J. Wu, R. Zhang, S. Shafiee, and C. Li, "A 'user-knowledge-product' Co-creation cyberspace model for product innovation," *Complexity*, vol. 2020, no. 1, 2020.

[15] Y. Wang, J. Wu, L. Lin, and S. Shafiee, "An online community-based dynamic customisation model: the trade-off between customer satisfaction and enterprise profit," *International Journal of Production Research*, 2019.

[16] K. Kristjansdottir, S. Shafiee, L. Hvam, M. Bonev, and A. Myrodia, "Return on investment from the use of product configuration systems – a case study," *Computers in Industry*, vol. 100, pp. 57–69, 2018.

[17] A. Haug, S. Shafiee, and L. Hvam, "The costs and benefits of product configuration projects in engineer-to-order companies," *Computers in Industry*, vol. 105, pp. 133–142, 2019.

[18] K. Kristjansdottir, S. Shafiee, H. Hvam, C. Forza, and N. H. Mortensen, "The main challenges for manufacturing companies in implementing and utilizing configurators," *Computers in Industry*, vol. 100, pp. 196–211, 2018.

[19] S. Shafiee, L. Hvam, A. Haug, M. Dam, and K. Kristjansdottir, "The documentation of product configuration systems: A framework and an IT solution," *Advanced Engineering Informatics*, vol. 32, pp. 163–175, 2017.

[20] P. T. Helo, Q. L. Xu, S. J. Kyllönen, and R. J. Jiao, "Integrated Vehicle Configuration System—Connecting the domains of mass customization," *Computers in Industry*, vol. 61, no. 1, pp. 44–52, Jan. 2010.

[21] L. L. Zhang, C. K. M. Lee, and Q. Xu, "Towards product customization: An integrated order fulfillment system," *Computers in Industry*, vol. 61, no. 3, pp. 213–222, Apr. 2010.

[22] L. Hvam, N. H. Mortensen, and J. Riis, *Product customization*. Berlin Heidelberg, Germany: Springer-Verlag, 2008.

[23] A. Felfernig, L. Hotz, C. Bagley, and J. Tiihonen, *Knowledge-based Configuration From Research to Business Cases*. Newnes, NWS, Australia: Morgan Kaufmann, 2014.

[24] T. Blecker, N. Abdelkafi, G. Kreutler, and G. Friedrich, "Product configuration systems: State of the art, conceptualization and extensions," in *Proceedings of the Eight Maghrebian Conference on Software Engineering (MCSEAI 2004),* 2004, pp. 25–36.

[25] T. Krebs, "encoway," in *Knowledge-based configuration: From research to business cases*, A. Felfernig, L. Hotz, C. Bagley, and J. Tihonen, Eds. Morgan Kaufman, 2014, pp. 271–279.

[26] C. Forza and F. Salvador, *Product information management for mass customization*. 2007.

[27] L. Ardissono *et al.*, "A framework for the development of personalized, distributed web-based configuration systems," *AI Magazine*, vol. 24, no. 3, pp. 93–110, 2003.

[28] S. S. Nahrkhalaji, S. Shafiee, M. Shafiee, and L. Hvam, "Challenges of Digital Transformation: The Case of the Non-profit Sector," *IEEE International Conference on Industrial Engineering and Engineering Management*, pp. 1245–1249, 2019.

[29] S. Shafiee, Y. Wautelet, L. Hvam, E. Sandrin, and C. Forza, "Scrum versus Rational Unified Process in facing the main challenges of product configuration systems development," *The Journal of Systems & Software*, vol. 170, p. 110732, 2020.

[30] S. Shafiee, L. Hvam, and M. Bonev, "Scoping a product configuration project for engineer-to-order companies," *International Journal of Industrial Engineering and Management*, vol. 5, no. 4, pp. 207–220, 2014.

[31] J. Tiihonen, T. Soininen, T. Mannisto, and R. Sulonen, *Knowledge Intensive CAD, State of the Practice in Product Configuration - a Survey of 10 Cases in the Finnish Industry*. Springer US, 1996.

[32] I. (1991). Nonaka, *The knowledge creating company*. Harvard Business Review Press, 2008.

[33] C. Forza and F. Salvador, "Managing for variety in the order acquisition and fulfilment process: The contribution of product configuration systems," *International Journal of Production Economics*, vol. 76, no. 1, pp. 87–98, Mar. 2002.

[34] C. Forza and F. Salvador, "Application support to product variety management," *International Journal of Production Research*, vol. 46, no. 3, pp. 817–836, Feb. 2008.

[35] L. L. Zhang, C. K. M. Lee, and P. Akhtar, "Towards customization: Evaluation of integrated sales, product, and production configuration," *International Journal of Production Economics*, vol. 229, p. 107775, 2020.

[36] L. L. Zhang, E. Vareilles, and M. Aldanondo, "Generic bill of functions, materials, and operations for SAP2 configuration," *International Journal of Production Research*, vol. 51, no. 2, pp. 465–478, 2013.

[37] A. B. Brown, A. Keller, and J. L. Hellerstein, "A model of configuration complexity and its application to a change management system Aaron," *IEEE Transactions on Network and Service Management*, vol. 4, no. 1, pp. 13–27, Jun. 2007.

Do product configurators comply with HCI guidelines? A preliminary study

Tony Leclercq[1] and **Claire Deventer**[2] and **Patrick Heymans**[3]

Abstract. While past research on product configuration has focused on knowledge representation and automated reasoning, researchers have paid less attention to the design and evaluation of user experience (UX). For product configurators like for other interactive applications, UX is of paramount importance. This is all the more true since configurators are often primary points of contact between a merchant and its customers, and business-critical assets meant to maximize sales.

Over the years, the HCI (Human-Computer Interaction) community has defined standard guidelines for interactive applications. Yet, previous studies suggested that in practice product configurators often differ from such guidelines. In this preliminary study, we address two main research questions: (1) To what extent do existing configurators deviate from HCI guidelines? (2) Why do those deviations appear?

We first present a synthesis of the main well-established guidelines from the HCI community. We complement these observations with an analysis of the literature to collect more HCI issues and to better understand their origins. Then, we proceed to studying a sample of 50 real-world configurators to observe whether, and to which extent, they comply with the HCI guidelines. We then conclude the paper with directions for future research, emphasizing HCI challenges that are specific to configurators.

This study is part of a broader PhD project which ultimate goal is to define a set of guidelines specifically intended to optimize the UX of sales configurators.

1 INTRODUCTION

Configurators are increasingly popular sales tools [1]. They are often the primary points of contact between a merchant and its customers. As such, they must provide optimal user experience (UX). Given their critical business role, they must be effective in guiding users and be reliable.

In the past, academic research on configuration has mostly focused knowledge representation and reasoning. But configurators are interactive applications where UX is of paramount importance [18].

Over the years, the HCI (Human-Computer Interaction) community has defined standard guidelines for interactive applications [4, 8, 5, 18, 19, 2]. These various guidelines approach UI elaboration from a broad perspective (e.g. the design process, user involvement…) down to fine-grained advice and templates (e.g. use navigation bars, display max 8 items). So doing, they can help develop, for example, an effective navigation system or avoid cognitive overload. But these guidelines remain generic as they are not always relevant for, or directly applicable to, sales configurators. And it has been observed that in practice configurators often deviate from such guidelines [3, 13, 12, 6].

Nevertheless, in the configuration community, interesting UI-related studies have been published. For example, the diagnostic tool of Trentin *et al.* [20] to assess and improve sales configurators based on their perceived benefits by users. More specific works can be found on points addressed by Trentin *et al.* e.g. on flexible navigation, user-friendly space description, side-by-side product comparison [17, 22, 16, 9, 15]. Kamies *et al.* [11] and Valenzuela [21] demonstrate the advantages of attribute-based views for the presentation of items in configurators whereas Jiang *et al.* promote bundle views [10]. Finally, other more specific concerns like impact of the verbosity of price display [7], rich visualisation [16] and user feedback (e.g. user ratings) [22] are addressed in the literature.

In this literature, business and marketing goals of configurators are often taken into account, highlighting desirable characteristics such as trust, self-expressiveness and uniqueness of the product. We concur with these authors that visualizations are ultimately means to obtain such characteristics.

Yet, although these papers have drawn attention on the advantages of various visualizations for the users and the business, they do not explain how to design configuration interfaces in order to systematically offer these advantages.

To narrow this gap, in this paper, we start to address two main research questions: **(1) To what extent do existing configurators deviate from HCI guidelines? (2) Why do those deviations appear?**

In the next sections we present a preliminary study of 50 real-world configurators to observe whether, and to which extent, they comply with HCI guidelines. We also try to better understand the origins of deviations and highlight good practices. Then, we present our results and limitations. Finally, we conclude the paper with directions for future research.

This study is part of a broader PhD project which ultimate goal is to define a set of guidelines specifically intended to optimize the UX of sales configurators.

2 METHODOLOGY

To understand these challenges, we compared and analysed the same type of configurators: car configurators. We focus on their anatomy, good and bad practices from a UI perspective, and try to find how they are related. Then we will use these results to find specific re-

[1] University of Namur, PReCISE research centre - Namur Digital Institute (NaDI), tony.leclercq@unamur.be
[2] University of Namur, Centre for Research on Consumption Leisure (CeRCLe) - Namur Digital Institute (NaDI), claire.deventer@unamur.be
[3] University of Namur, PReCISE research centre - Namur Digital Institute (NaDI), patrick.heymans@unamur.be

sponses. We are aware that these specific responses will at times be conflicting with each other (e.g. choosing between one page with a large number of items and several pages), so we need to be able to identify priorities and make tradeoffs.

In the remainder of this paper we report on our preliminary study and first present its dataset.

2.1 Dataset

To establish our set of sales configurators, we selected car configurators. They are one of the most popular category of configurators according to Cyledge's database [1]. They possess at least three interesting features. Firstly, car configurators contain a high number of options and constraints, making both presentation and reasoning challenging. We thus hope to obtain a broad spectrum of problems and characteristics. Secondly, car configurators are directly aimed at customers where UX is a primary concern. Thirdly, we can find many car configurators which yields a large variety of UIs. In order to obtain 50 configurators, we took car configurators from three countries (US, UK and one french-speaking country) with the choice of the same model car (or equivalent model car between countries, e.g. the three Ford Escape configurators for US, UK and France. If a configurator is the same in two countries, we keep one. In the same way, if a configurator is the same between two brands, one is discarded (e.g. two brands belong to the same corporation and have the same configurator design).

To analyse our car configurators and identify the good and bad practices, we select criteria from different sources chosen for their relevance in the HCI- or configurator-related communities. First, we take the ten heuristics of Nielsen [14] and the eight Golden Rules of Shneiderman [18]. As is usual they are evaluated with Likert scales. We only note if a heuristics or a rule is negatively influenced by something, e.g. one error with a message containing non-understandable technical words violates heuristics about error management. These two first sources (Nielsen and Shneiderman) are also needed to make the link between issues and characteristics of car configurators. Both heuristics and rules are generic for all types of UI. Then, as we work and more specifically on sales configurators, we take further criteria from our pre-study of literature appearing in the introduction section.

Besides the evaluation of the UI of the configurator, we also need to identify sources of issues. For that, we use the criteria defined by Abbasi [3] in his "anatomy" of web configurators.

Finally, a pre-analysis of ten car configurators gave us new measures to evaluate. As a result, the dataset is composed of 177 variables on 50 observations. The data of the evaluation are stored in a Google sheet. Each configurator is evaluated once, except when we add a criterion during the evaluation. In this case, the configurator is evaluated again, but only to the new criteria. The dataset[4] contains information about the analysed configurators, information about UI analysis criteria, and raw data on the evaluation of these criteria.

We present the first observations and discussions in the next sections.

3 FIRST RESULTS

We observe our dataset with descriptive statistics as frequencies or correlation matrices. We divide results into three categories : anatomy (overview), bad practices and good practices.

2.2 Anatomy overview

A summary of the most common characteristics of configurators is visible in Figure 1. The first observation concerns navigation in the UI. We observe that 84% of configurators have a multi-step configuration process (68% of basic multi-step and 16% of hierarchical multi-step) and 16% have a single-step configuration process. Among these multi-step configurators, 64% are open steps (e.g. we can jump from step 1 to 4), 16% have step-by-step navigation, 2% have a mixed process (i.e. first steps must be filled one by one, and free navigation is authorised for the last steps) and 2% are not relevant. During the configuration process, the validity of the configuration is checked interactively (on the fly) in 88% of the cases, and by batch (i.e. upon step change) in 4% of the cases. We notice that no check is made in 8% of configurators because the user can only choose between valid configurations of cars.

Concerning constraints, only 34% have format constraints (principally for email checking). Besides that, we find group, cross-cutting and visibility constraints in, respectively, 100%, 88% and 80% of the cases. Theses constraints generate propagations. We find an absence of notification of propagation for 58% of configurators. The propagations are automatic in 74% of the cases, controlled for 78% (e.g. user confirmation), and guided for 30% (e.g. user needs to choose between two options). 100% of configurators have XOR group constraints (only one item must be selected). No configurator has OR group constraints (at least one item) or intervals group constraints (any number of items between a minimum and a maximum number).

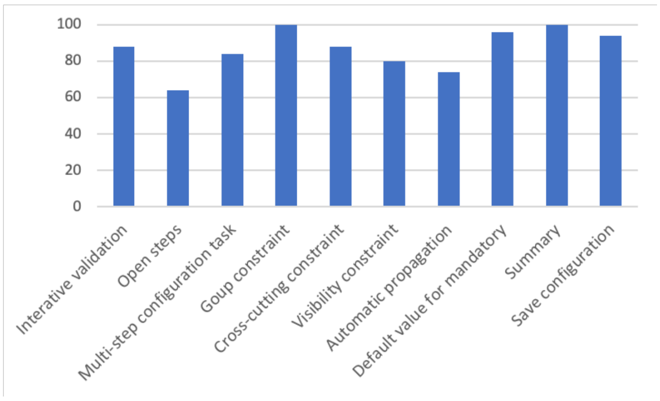


Figure 1. Some common characteristics of car configurators

All the configurators have mandatory options, and 96% of them give a default value to this option. 94% of configurators have optional options. Cloning mechanism (for duplicating components) is absent in all configurators.

We notice a description of the different options in 84% of configurators. When a description is available, 60% of configurators use a dialogue box to provide information, 18% an extensible zone (with a click), 6% a tooltip (i.e. information appears when the mouse is over it) and 6% are plain text explanations.

All configurators propose a summary of the current configuration. The summary is not always visible (92% of the cases) but consulting it does not close the configuration process (98%). Unfortunately, sometimes the summary is not easily accessible (20% of the cases).

[4] the file is available at this address `https://docs.google.com/spreadsheets/d/11BYzDbEWu35nFJKOcrbI7CeMK1zm_Hf1_-Jv67j3ZNY`

8% of car configurators give the possibility to the user to compare several configurations. 66% of configurators load options of a previous configuration in the configurator interface, and 94% save the configuration (28% only offer a save by email or printable display).

The visualisation of the product is a compound of rendered pictures for 68% of configurators, 18% have a full 3D scene, 8% a set of images and 6% just one picture. The controls to navigate in the UI is a navigation bar for 14%, buttons for 16%, and both for 54%. 16% of configurators use another system, e.g. mouse scrolling.

The users have to define less than 10 options in 18% of the cases, between 10 and 20 in 20%, and more than 20 in 62% of the cases. Concerning the number of configuration steps, they are between 1 and 10, but 50% of configurators have between 4 and 7 steps (with a median of 5). The number of versions of a model (commonly called trim) is between 1 and 10, but 50% of configurators have between 3 and 5,75 trims (with a median of 4). Finally, 78% of the car configurators offer option packs in addition to the trim.

3.1 Bad pratices

From Nielsen's viewpoint, 6 of the 10 heuristics are negatively impacted. They are User Control (issue with 90% of configurators), Recognition (80%), Flexibility (96%) and to a lesser extent Visibility (45%), Aesthetics (42%) and Consistency (34%). Equivalent results are observed with the Golden Rules of Shneiderman. 4 out of 8 rules are violated: Shortcut (96%), Memory Load (90%), Yield Closure (68%) and Informative FeedBack (50%). These points have to be confirmed with a user evaluation in a future study, but they are in line with the following observations.

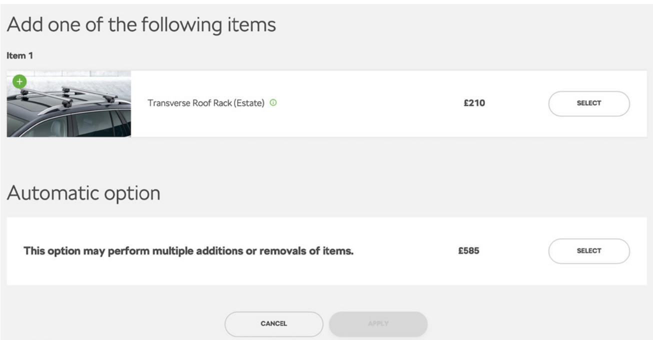


Figure 2. Example of bad practice : with the second option, the user does not know which changes are made

User does not have **feedback information** on his action in 96% of configurators, and 90% of configurators force the user to retain **too much information**.

Most importantly, for an interactive applications, users encounter navigation troubles with an **unknown configuration state** in 96% of car configurators, uses of **irregular navigation system** in 44% and **difficulties to known their location** in the UI (42%) or **achievement of the configuration process** (38%). 14% of configurators that have a multi-step configuration process **do not retain the configuration when browsing backwards**.

As shown in Figure 3, nearly one configuration system out of three (28%) **does not explain** propagations and their result as immutability or availability of the options. Surprisingly, 26% of configurators **do not detect errors** in the configuration. 24% do **not propose a repair mechanism** e.g. to select an **option displayed as unavailable** (present in 30% of car configurators) or to make the needed changes to obtain a valid configuration. 96% of car configurators **do not have an undo option**, and 84% make **invisible propagation**.

In the UI, the user has **too many controls displayed** in 28% of the cases, and needs to **use too many controls for performing an action** in 18% of the cases (e.g. click on several elements of the UI to choose a trim, or select an option).

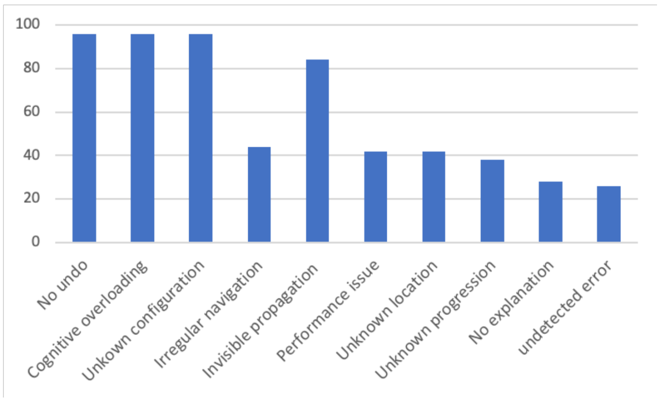


Figure 3. Some common bad practices of car configurators

40% of configurators **are not integrated in the website** (the user has to leaves the brand's main website). When a user chooses to configure a car, she is **redirected to a page containing car configurator links** and needs to select his desired model again in 18% of the cases. Sometimes, the first step is to choose a trim, and the site **redirects the user to a sub-configurator**. This means that the user does not know all available options for a model in 58% of the cases.

30% of configurators do not provide **coherence among options** (e.g. mandatory option are not presented in the same way during the same configuration process). One configurator does not select a **default value for a mandatory option in every case**.

Finally, we encountered **performance issues** in an astonishing 42% of the configurators. They are loading time, configuration loss, or even website crash.

3.2 Good practices

76% of the configurators present to the user an **assisted repair** if some changes are required to select an option. All car configurators offer a summary to the user although they are not always visible (92%) and sometimes not easily accessible (20%). We observe an appropriate response to this problem with a trade-off: some configurators present a **partial display of the summary** (8%).

A challenge is to present to the users all the inaccessible options (e.g. because a change of trim is needed). 12% of configurators give the **possibility to show unavailable options** as shown in Figure 4.

Another difficulty for the designer is to inform the user of her location in the UI as well as her progression in the configuration process. A **realistic, responsive progress/navigation bar** seems to be the right solution but further investigation in the dataset is needed.

When a mandatory option appears, 96% of configurators **select a default option or value**. This default selection possesses two benefits : the user understands when a choice is expected, and it avoids the frustration when a warning or error message appears.

To deal with an high number of options, we observe that designers give the possibility to the user to **reduce the amount of options by filtering them** in 74% of configurators.

Another used mechanism is to easily **change the model of car by transferring** a maximum of **chosen options** in the new model (30%). But for consistency it would be better to change the model of the car like the other options in the configurators.

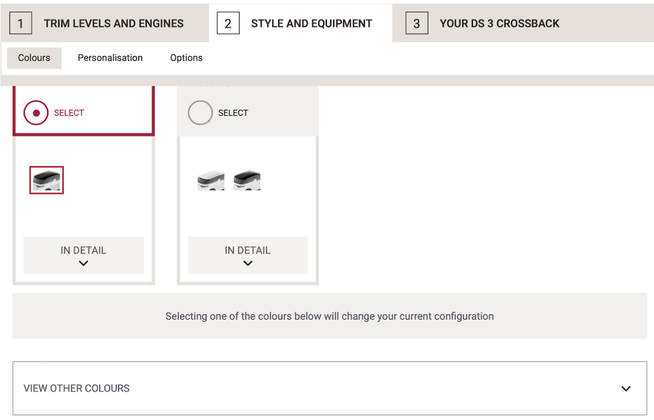


Figure 4. Example of good practice : the possibility to show unavailable options (and select them)

We also notice that some configurators **give advice** to the user (18% of configurators) or **display ratings** based on other customers' selections (8%). A **helpdesk** (16%) or **chatbot** (6%) are available too.

Finally, to avoid being lost in the UI, 6% of configurators **help the user navigate** with tooltips, descriptions or highlighting. To facilitate the car configuration, the users has **side-by-side product comparison** between car models (26%), or more sophisticated comparisons between saved user configurations (8%). And last, configurators display **simple prices without details** in 68% of the cases.

4 DISCUSSION

To find the link between the anatomy of configurators and the bad or good practices, we use first a correlation matrix. Yet, the matrix does not show new or surprising significant relationships between the variables (viz. issues and anatomy). We find links between the unknown configuration state issue and the summary not displayed, or performance issues and the visibility of the status. In any case, some interesting observations have already emerged from the dataset.

The display of the product in configurators can cause performance issues and have a negative impact on user experience (for 66% with 3D rendering and 44% with a set of rendered pictures). For configurators with a set of rendered pictures and performance issues, 93% have more than 20 options. 50% of configurators with 3D rendering and performance issues have more than 20 options. At the opposite, we notice than configurators with just one picture or a set of images do not have performance issues.

For all configurators with performance issues, 50% of configurators have more than 20 options, 20% a number of options between 10 and 20, and 33% less than 10 options.

For the navigation in the UI, if we have mouse scrolling navigation associated with a single-step configuration process and more than 10 option, the user is lost.

Some sources of design issues like information overload are not clearly identified. This may be because most of the configurators have immutable propagations, visibility constraints and a high number of options. These three factors are inevitably involved in cognitive overload. But, we cannot isolate one source of the problem as all configurators have same characteristics. However, we notice that car configurators without cognitive overload issues have all: multi-step configuration process, with all steps accessible, and the possibility to go back without loss of configuration. They all have repair mechanisms for helping users encountering cross-cutting or visibility constraints. However, they have different navigation systems, sizes of visualisation of the product, numbers of options or other characteristics. Therefore, we cannot make general conclusions at this time, and further analysis and research need to be done.

All these observations show that a more in-depth work on the dataset is promising. This first analysis gives tendencies and solutions for avoiding bad practices. It also helps the designer that must deal with the ample configuration space and the occurrence of immutable propagation or visibility constraints, and find a solution. These solutions are maybe already provided by good practices e.g. the possibility to check available options in other possible configurations than the current one. In some cases, the designer must make tradeoffs when responses conflict with each other. This part needs to be further studied, though.

Another limit to this study is that configurators with optional options, OR group constraints or interval group constraints are not covered in the dataset. The same problem appears with the rarity of manual repairs to fix an issue of configuration (e.g. an invalid configuration, an unavailable option). This part needs to be studied too.

5 CONCLUSION AND FUTURE WORKS

Sales configurators are the meeting point between the business and the customers. Special attention is needed in their design. Unfortunately, recent studies show that practitioners still lack the resources for designing high-quality configurator user interfaces.

Accordingly, our long-term objective is to narrow this gap by proposing standard guidelines for designing such interfaces. In this paper, we first summarized existing literature to help designers to develop UIs for sales configurators. Then, we presented a study of 50 real-world configurators. This study gives a first idea of the extent to which existing configurators may deviate from HCI guidelines. It also gives ideas of why deviations appear and highlights first responses to design issues. Currently, we are working in more depth on the dataset to flesh these deviations and responses out. Even if we need to deepen the study, the first results can already help designers build better user interfaces for configurators and spot mistakes to avoid.

Future work is also needed on the identification of priorities and tradeoffs as these specific responses will at times conflict with each other. We aim to reach a global tool to design rich user interfaces with the purpose of improving sales and user satisfaction.

REFERENCES

[1] Cyledge configurator database. `https://www.cyledge.com/de/configurator-database`.
[2] Nielsen norman group. `https://www.nngroup.com/`.
[3] Ebrahim Khalil Abbasi, Arnaud Hubaux, Mathieu Acher, Quentin Boucher, and Patrick Heymans, 'The anatomy of a sales configurator: An empirical study of 111 cases', in *International Conference on Advanced Information Systems Engineering*, pp. 162–177. Springer, (2013).
[4] JM Christian Bastien and Dominique L Scapin, 'Ergonomic criteria for the evaluation of human-computer interfaces', (1993).

[5] Alan Cooper, Robert Reimann, and David Cronin, *About face 3: the essentials of interaction design*, John Wiley & Sons, 2007.

[6] Maxime Cordy and Patrick Heymans, 'Engineering configurators for the retail industry: experience report and challenges ahead', in *Proceedings of the 33rd Annual ACM Symposium on Applied Computing*, pp. 2050–2057, (2018).

[7] Benedict GC Dellaert and Stefan Stremersch, 'Marketing mass-customized products: Striking a balance between utility and complexity', *Journal of marketing research*, **42**(2), 219–227, (2005).

[8] Alan Dix, Alan John Dix, Janet Finlay, Gregory D Abowd, and Russell Beale, *Human-computer interaction*, Pearson Education, 2003.

[9] Alexander Felfernig, Lothar Hotz, Claire Bagley, and Juha Tiihonen, *Knowledge-based configuration: From research to business cases*, Newnes, 2014.

[10] Pingjun Jiang, Siva K Balasubramanian, and Zarrel V Lambert, 'Consumers' value perceptions of e-customization–a model incorporating information framing and product type', *Journal of Consumer Marketing*, (2014).

[11] Arnold Kamis, Marios Koufaris, and Tziporah Stern, 'Using an attribute-based decision support system for user-customized products online: an experimental investigation', *MIs Quarterly*, 159–177, (2008).

[12] Tony Leclercq, Maxime Cordy, Bruno Dumas, and Patrick Heymans, 'On studying bad practices in configuration uis.', in *IUI Workshops*, (2018).

[13] Tony Leclercq, Maxime Cordy, Bruno Dumas, and Patrick Heymans, 'Representing repairs in configuration interfaces: A look at industrial practices.', in *IUI Workshops*, (2018).

[14] Jakob Nielsen, 'Enhancing the explanatory power of usability heuristics', in *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*, pp. 152–158, (1994).

[15] Rick Rabiser, Paul Grünbacher, and Martin Lehofer, 'A qualitative study on user guidance capabilities in product configuration tools', in *2012 Proceedings of the 27th IEEE/ACM International Conference on Automated Software Engineering*, pp. 110–119. IEEE, (2012).

[16] Taylor Randall, Christian Terwiesch, and Karl T Ulrich, 'Principles for user design of customized products', *California Management Review*, **47**(4), 68–85, (2005).

[17] Enrico Sandrin, Alessio Trentin, Chiara Grosso, and Cipriano Forza, 'Enhancing the consumer-perceived benefits of a mass-customized product through its online sales configurator', *Industrial Management & Data Systems*, (2017).

[18] Ben Shneiderman and Catherine Plaisant, *Designing the user interface: strategies for effective human-computer interaction*, Pearson Education India, 2010.

[19] Jenifer Tidwell, *Designing interfaces: Patterns for effective interaction design*, " O'Reilly Media, Inc.", 2010.

[20] Alessio Trentin, Elisa Perin, and Cipriano Forza, 'Sales configurator capabilities to avoid the product variety paradox: Construct development and validation', *Computers in Industry*, **64**(4), 436–447, (2013).

[21] Ana Valenzuela, Ravi Dhar, and Florian Zettelmeyer, 'Contingent response to self-customization procedures: Implications for decision satisfaction and choice', *Journal of Marketing Research*, **46**(6), 754–763, (2009).

[22] Weiquan Wang and Izak Benbasat, 'Attributions of trust in decision support technologies: A study of recommendation agents for e-commerce', *Journal of Management Information Systems*, **24**(4), 249–273, (2008).

Product Configuration Activities in SMEs and their Digitalization: Preliminary Results of a Survey Study

Svetlana Suzic[1] and **Enrico Sandrin**[1] and **Nikola Suzic**[1] and **Cipriano Forza**[1] and **Alessio Trentin**[1]

Abstract. The presence of customization in manufacturing small and medium enterprises (SMEs) is widely known, as is the competitive pressure that even they have to deal with. We also know that there are examples of successful applications of product configurators in SMEs, even in quite small ones. However, we do not know the extent of the presence, in SMEs, of the various product configuration activities or the intensity of their digitalization. The present study offers some preliminary results of research aimed at gaining further insights into product configuration activities in SMEs. Specifically, the present study provides preliminary empirical results gathered in a sample of 18 Italian SMEs. It emerges that configuration activities are frequently present in manufacturing SMEs and that there is high potential for their digitalization.

1 INTRODUCTION

In the last three decades, more and more customers have required products that closely satisfy their specific needs, leading to an increase in product variety and customization by companies (see [1], [2], [3], and [4]). Consequently, even small and medium enterprises (SMEs) in the manufacturing sector must offer product variety and customization to obtain and maintain their competitive position on the current market [5].

Offering product variety and customization implies a number of particular activities (e.g., those of the configuration process) and in general increases the complexity companies have to deal with. The management of product variety and customization is facilitated by the use of some supportive software applications (e.g., product data management [PDM], customer relationship management [CRM], and product configurators) [6]. Implementation of these kinds of software applications is often referred to as digitalization in recent literature (see [7] and [8]). For SMEs, it is often difficult to adopt and subsequently maintain such software applications due to the lack of financial resources and the scarcity of both IT staff and other specialists (see [9] and [10]).

The literature provides few examples of successful implementation of product configurators even in small SMEs (see [11] and [12]). Besides these sparse examples, we do not have information on the extent of the various product configuration activities in SMEs. Moreover, the intensity of digitalization of these configuration activities in SMEs remains unknown. Consequently, researchers do not know whether SMEs are a business context where configurators are diffusely adopted or have the potential to be diffusely adopted. This is important information for researchers because SMEs are a context with specific characteristics; for example, their improvement initiatives towards mass customization are constrained by limited human and financial resources [13]. If configuration activities are heavily diffused across SMEs, then it becomes important to investigate the digitalization of the configuration process and, in particular, the application of configurators because configurators have shown a capability to support great improvements in the configuration process. If the application of configurators in these SMEs is limited, then it is necessary to investigate why. Eventually, this investigation will discover that configurators that better fit SME characteristics need to be developed or that advancements of mass customization enablers that favor the application of configurators, such as part standardization and product modularity, need to be pushed [13]. Altogether, these research efforts could support the development of mass customization implementation guidelines specifically for SMEs and the technological development of configurators more suitable for SMEs.

The present study's objective is to gain more insights into product configuration activities, namely the extent of their presence in SMEs and the intensity of their digitalization. The research plan is to gather information on product configuration practices from a sample of 100 Italian SMEs. Unfortunately, the data collection has been interrupted by the COVID-19 pandemic. For that reason, the present study provides preliminary empirical results gathered in a sample of only 18 Italian SMEs. In considering the figures reported in the present article, the reader is advised that sometimes the reported numbers are percentages and therefore they never refer to the targeted final sample but refer only to the current sample of 18 companies.

Preliminary results show that configuration activities are frequently present in manufacturing SMEs. Also, although the current level of digitalization is not negligible, the companies have shown significant intention to further develop their digitalization, especially in some specific configuration activities.

The rest of the paper is organized into five sections. Section 2 presents a short review of the relevant literature. Section 3 provides details of the method used for gathering information and describes the sample, while section 4 reports the results of the study. Finally, Section 5 discusses the results and provides suggestions for future research.

2 THEORETICAL BACKGROUND

The configuration process is frequently present in companies that offer high product variety. It includes "the set of activities from the collection of information about customer needs to the release of the

[1] Department of Management and Engineering, University of Padova, Italy. Email: suzicsvetlana@gmail.com, enrico.sandrin@unipd.it, nikola.suzic@unipd.it, cipriano.forza@unipd.it, alessio.trentin@unipd.it

product documentation necessary to produce the requested product variant" [6, p. 143]. The configuration process generally includes several configuration activities that refer to three main groups: characteristics specification, component association, and configured product evaluation (see [14] and [15]). Please consider that these activities can be done in different sequences. In addition, there could be several cycles so that both customers and company personnel can go through these (or some of these) activities several times before arriving at a customer order or even before arriving at a quotation.

The first group of configuration activities regards the specification of product characteristics appropriate for a given customer. It includes the communication of options to the customers either in reply to their enquiries or as the proactive initiative of a salesperson or a sales application. The customers select the characteristics that satisfy their needs, and their choices are collected and stored. Product characteristics can be selected with the help of sales personnel or independently by the customers [16]. In both cases, it is necessary not only to collect all the needed product specifications but also to assure they are compatible [11]. This compatibility should ideally be assured while customers are choosing their product characteristics. If that is not possible, it can be done at another time, but if there are incompatibilities, the customer may be called to reconsider his/her choices. Sometimes this specification process leads to one or more product characteristics that are not already predefined by the company. In this case, the offered product space may be enlarged to satisfy the request for the new unforeseen product variant.

The configuration activities of the second group—component association—aim to identify the product components necessary to fulfil the product characteristics chosen by the customer and to establish relationships among these identified components. These configuration activities eventually (e.g., in the case of a new product characteristic) need to identify components that are not already predefined by the company: this is at the borderline of the scope of the configuration notion. Consequently, ad-hoc engineering of new components is required [11].

Finally, the third group—configured product evaluation—concerns evaluating the compatibility of all selected components and goal satisfaction (i.e., whether the configured product satisfies customers' needs, including their time and economic constraints) [14]. Notably, configured product evaluation can lead to changes in the original selection of characteristics and/or ad-hoc engineering. Once the necessary components and the relationships between them have been determined, operative instructions for product variant manufacturing are generated. Actually, the company may find alternatives in terms of components and production sequences that satisfy a given customer request with different impacts on costs and delivery lead times. Price determination is usually included in the configuration process, and it is crucial product evaluation information for the customer. Sometimes, the delivery lead-time is considered or partially considered. Therefore, the configuration process can include the determination of terms of delivery, a description of the product or service (e.g., technical drawings, charts, images, a user manual) and other aspects, some of which are useful for evaluating the product.

Notably, the configuration activities can differ across companies [11], and, as a consequence, the outputs generated by these activities can vary among companies. Specifically, the configuration process may result in all or some of the following outputs: quotation letter with price [11]; product cost [17]; product code (see [12] and [18]); bill of materials (see [17] and [18]); production cycle (see [19] and [20]); technical drawings (see [21] and [22]); product image [23]; or usage manuals [21].

3 METHOD AND SAMPLE DESCRIPTION

3.1 Method

To gather information from SMEs, we developed a questionnaire with several sections, each dedicated to a specific set of issues. This questionnaire starts with the overall company characteristics, such as number of employees. Then it considers the company's main commercial and operative characteristics, such as kind of customers and market response modality. Subsequently, it asks for information on the company's ability to fulfill personalized orders, considering both the performances that define this ability and the enablers that underpin this ability. At this point, the context of interest is clear, so the questionnaire moves to the issue of digitalization. First, it asks for some brief information on the overall digitalization of the company, such as the presence of enterprise resource planning (ERP), commercial presence on the web, customer relationship management (CRM), and use of product data management (PDM)/product lifecycle management (PLM). Finally, it goes into detail about configuration activities and their digitalization. Notably, the present paper focuses on these configuration activities and their digitalization; however, it also uses some information on the overall digitalization of the company and, in order to describe the sample and specify the context, some information on the overall company characteristics and its commercial and operative context. We deliberately limited our analyses to this set of information in order to focus on a specific aim. Obviously, a number of additional interesting questions arose, but in our research process we need first of all to gain an overall picture of the configuration activities in SMEs and their degree of digitalization; after that we will go in deeper analyses.

The questionnaire was designed to be completed by one respondent with overall knowledge of the company during one-on-one meetings with the company's representative. If he/she did not have all the information needed, other informants were contacted by the respondent to collect the needed information.

The questions used to collect the needed information are provided in the tables that report the results of the study (see Tables 2–8). This decision was made to facilitate reading this paper.

The answers provided during the interviews underwent a first check during the same interview. However, this control was a light one since the presence of the interviewer was thought to get answers that were as complete as possible. Challenging the respondent too much would have been counterproductive. We performed a more accurate control later. This check highlighted issues of missing data and some possible issues of coherence between answers. We excluded questions that had observations with possible coherence issues from the present article. However, we included questions with some missing observations. We planned a second interaction with respondents to control the potential coherence issues and issues related to missing information. We will do that when we have more than 50% of the sample on hand. In this way, we will be able to detect more potential coherence issues and avoid going back to the respondents more than once. In addition, when we go back to companies with some preliminary results, this will stimulate collaboration. We are confident that we will be able to perform this second interaction with companies before the Configuration Workshop 2020, at least for the 18 observations used for the present article. During our presentation at the workshop, we will be able to provide reliable statistics on the number of product families and the number of product variants. Currently, these figures are affected by too many non-responses and even though they make sense, we are not sure they correctly describe our sample.

3.2 Sample description

3.2.1 *Company size*

According to the European Commission classification system, all companies in our sample are SMEs (i.e., they have fewer than 250 employees; see Table 1). More precisely, the data reported in Table 1 show that 83% of the companies are defined as small companies (fewer than 50 employees), and 17% are medium companies.

Table 1. Company size

No. of employees	No. of companies	Percentage of companies
1–19	7	39%
20–49	8	44%
50–250	3	17%
TOTAL	18	100%

3.2.2 *Kind of customers and distribution channels*

The companies in our sample sell products equally for final consumer use (business to consumer) and for industrial applications (business to business). The data reported in Table 2 indicate that the turnover derives mostly from direct selling and that less than 20% of the turnover is realized through intermediaries.

Table 2. Turnover split

How is your turnover split?	Mean* (%)
% Products for final customers	**42.0**
% sold directly to final consumers	33.8
% sold through commercial intermediaries	8.1
% Products for business customers	**49.7**
% sold directly to final business clients	43.0
% sold through commercial intermediaries	6.7
% Products made for third parties (not own products/parts)	**8.3**

* The mean is calculated on the 12 SMEs that provided an answer.

3.2.3 *Degree of customization*

Table 3 summarizes the questions and responses regarding degree of customization. The results reported in Table 3 indicate that:

- 93% of the companies receive orders for some functionality to be designed ad hoc. In particular, half of the companies receive more than two-thirds of their orders for functionality to be designed ad hoc.
- 47% of the companies receive orders with options selected. In particular, a quarter of the companies receive 25–50% of their orders for products chosen as a combination of options in the catalog.
- 60% of the companies have no orders for final products already defined in the catalog.

Table 3. Customization degree

Which percentages of customer orders belong to the following categories?	0	1–24	25–49	50–74	75–100	Mean *
	%	%	%	%	%	%
Product orders for which the customer asked for new functionalities that required ad–hoc design	6.7	26.7	13.3	6.7	46.7	55.9
Product orders for which the customer chose by combining predefined options present in the catalog without asking for new functionalities	53.3	13.3	26.7	6.7	0.0	16.1
Product orders for which the customer found the final product already completely defined in the catalog	60.0	6.7	0.0	13.3	20.0	28.0

* The mean is calculated on the 15 SMEs that provided an answer.

The fact that 93% of the sampled SMEs have engineer to order (ETO) orders, whereas only 47% have some configure to order (CTO) orders (only 7% have CTO in more than 50% of their orders and not a single one is CTO for more than 75% of their orders) does not threaten the validity of our sample for studying product configuration practices. This is for two reasons. The first is that an ETO company could redesign its product space to increase the use of CTO. Sometimes, even the introduction of configurators helps to reduce the percentage of ETO orders. Second, the application of a configurator can improve the configuration activities in an ETO process. The notion of partial configurability introduced by Forza and Salvador [11] supports this point. Let us take as an example a request for a machine with one feature to be engineered to order and all other features that are chosen from a predefined list. In this case, a configurator could automatically produce an incomplete bill of material and calculate the cost related to the part of the bill of material that is automatically generated. Obviously, the configuration process cannot be completed automatically; however, the gains in cost, time, and quality can be huge. An example of such a configurator is provided by Forza and Salvador [24].

3.2.4 *Modality of response to customer demand*

We asked for each company its market response modality. Table 4 reports the results from the analyses of the collected answers. These results are quite informative and indicate that:

- 82% of orders go through the technical office. Therefore, compared with Table 3, 72% of orders with new functions or orders with combinations of predefined options and 10% of final products already defined in the catalog pass through the technical office;
- 16% of orders are fulfilled from stock and, therefore, do not involve production. By comparing this with Table 3, it emerges that out of 28% of orders for final products already defined in the catalog, 16% are made on forecast and 10% are made on order.

Table 4. Activities included in order fulfillment

Which percentages of customer orders belong to the following categories?	0	1–24	25–49	50–74	75–100	Mean*
	%	%	%	%	%	%
Customer orders that pass through the technical/R&D department for technical control or design activities	0.0	12.5	6.3	6.3	75.0	81.6
Customer orders fulfilled with products made on sales forecast (and not "to order")	68.8	0.0	18.8	0.0	12.5	15.9

* The mean is calculated on 16 SMEs that provided an answer.

A comparison of figures reported in Table 4 with figures reported in Table 3 shows that both configured and catalog products pass through the technical office. This seems like a suspicious discrepancy, but it is not. While it is justifiable for configured products in the absence of a configurator, it does not seem justifiable for products completely defined from the catalog. However, as reported by Forza and Salvador [7], it could be that all orders pass through the technical office because a note on an order for a standard product could lead to a change in the product and, more specifically, could even lead to an ad hoc engineered product. This is an organizational solution to address the limited individual technical competence of sales personnel and the willingness to deliver quality to the customer.

Furthermore, the fact that 82% of orders go through the technical office while only 56% of orders have new functions indicates that on average 26% of the orders are processed by the technical office. With the use of a configurator with appropriate functionalities, the technical office could be bypassed, leading to gains in cost and answering time. This is a conservative estimate of the potential applicability of configurators in the considered sample because if we consider the possibility of managing partial configurability, the potential of applications is much greater.

4 RESULTS

4.1 Presence of activities

In the questionnaire, 11 configuration activities have been proposed, namely: selecting product characteristics [11, 24], determining price [11], generating the bill of material [11, 17, 18 24], generating the production cycle [11, 19, 20], determining the cost [11, 17], generating technical drawings [11, 21, 22], providing a product image (rendering, photo, sketch) [11, 23], generating a product code [11, 12, 18, 24], providing usage instructions [11, 21], specifying characteristics that are not predefined [11, 24], and identifying components that need ad hoc engineering [11, 24]. The answers provided for each of these 11 configuration activities are summarized in Figure 1. It emerges that each one of them is present, on average, in 80% of the companies that gave a response.

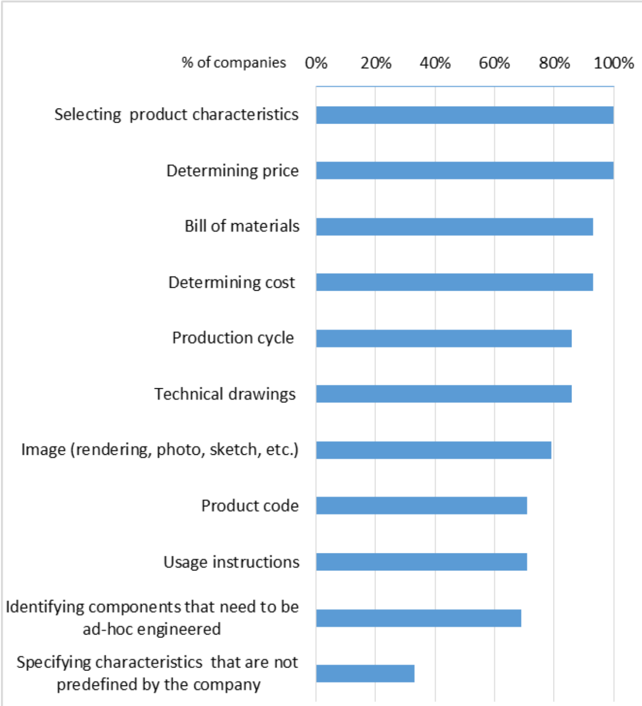


Figure 1. Presence of each configuration activity in SMEs

Figure 1 shows that all participating companies specified that they perform the selection of the product characteristics appropriate for the customer and determine the price during the configuration activity. Furthermore, more than 90% of companies answered that they generate a bill of materials and determine costs during the configuration activity.

We also analyzed the number of configuration activities performed in each SME. All companies indicated that they conduct at least 5 configuration activities, while the median number of activities present for a company is 9 (Figure 2).

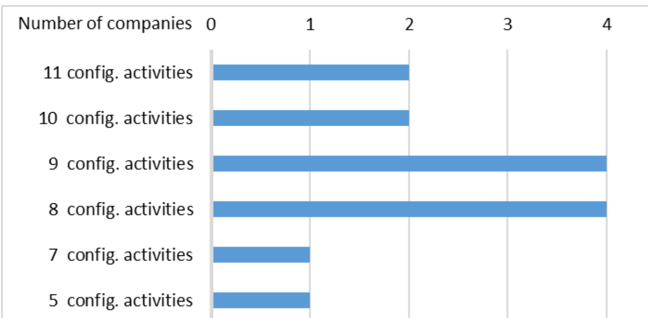


Figure 2. Number of configuration activities per company

The information reported in Figure 2 is important because it is an indicator of the complexity of the configuration process. More configuration activities imply requests for more functionalities in a configurator and greater implementation efforts. One could argue that this is not relevant because the incidence of configured to order orders is not very high. Again, we should consider the possibility of automating partial configurability. Therefore, this information significantly contributes to characterizing the specificity of the configuration context of SMEs.

One further word is needed regarding the activity of selecting the product/service characteristics/functionalities appropriate for the customer. This activity is usually carried out by sales personnel (71%), while 29% of companies indicated that this activity is carried out by the client alone.

4.2 Digitalization level of SMEs

The presence of ERP, MRP, PDM/PLM, and CRM software applications is used to comprehend the overall digitalization status of manufacturing companies that offer customized and/or a high variety of products (Table 5). It was noted that:

- The considered companies have a high presence of ERP (78%), where ERP is defined as a management information system that integrates business processes (i.e., sales, purchases, warehouse management, accounting, etc.). Obviously, these ERPs in most cases are not huge applications such as SAP®. Notwithstanding, their use signals the presence of business process integration supported by software applications: a first step in digital integration has been performed by SMEs.
- The considered companies also tend to have a strong commercial presence on the web (72%). The majority of the companies use their commercial presence on the web to present their products (69%), to collect contacts (54%), to receive requests for offers (34%), to produce offers (31%), and to receive orders (31%). The row data show that the commercial activity on the web differs from one company to another.
- The core of software production management support (MRP) is present in 53% of the sample. Keeping in mind that a number of companies do not need an MRP because they have an extremely limited number of purchasing materials, these findings can be considered a good level of digitalization.
- The adoption of CRM is not negligible (45%). This percentage is even more interesting given that some of these companies work for third parties or with a limited number of customers.
- Finally, the adoption of PDM/PLM is 39%, which is not low considering the complexity of these systems. This percentage shows the willingness of these companies to offer excellent support for product technical data management.

We also explored whether the level of digitalization depends on the size of the company. We saw that companies with less than 20 employees have less digitalization than companies with less than 50 employees, which, in turn, have less digitalization than those with between 50 and 250 employees. On average, digitalization in SMEs with more than 50 employees is double that of SMEs with less than 20 employees. We provide these figures to signal that there is variability in the level of digitalization that should be investigated once the full sample is available.

The presented data indicate, in general, considerable levels of digitalization. Of course, this does not mean that this digitalization is effective, but it does indicate that the considered companies have a significant openness to digitalization.

Table 5. Usage of software in companies

Which of the following software and hardware technologies does your company use?	**Mean% ***
ERP (enterprise resource planning)	77.8
Commercial presence on the web (use of own website or someone else's platform, e.g., Facebook, for commercial activities)	72.2
MRP (materials requirements planning)	52.9
CRM (customer relationship management)	44.4
PDM (product data management) or PLM (product lifecycle management)	38.9

* The mean is calculated on all 18 SMEs because all of them answered each question

4.3 Digitalization of configuration activities

Configuration activities are also significantly digitalized in the participating companies. Figure 3 reports for each activity how many companies declared that it is digitalized and how many declared that it is not digitalized. On average, each configuration activity is digitalized in 58% of SMEs (Table 6 column 3 last row).

When the sample is completed, we will investigate whether the digitalization level of configuration activities varies according to company size or other contingencies. We made a preliminary rough analysis of the distribution of the digitalization of configuration activities across companies. We calculated the percentage of digitalized configuration activities out of the total number of configuration activities present in each company. For these rough analyses, we hereafter report (in the text and not in the tables) numbers that support the possible presence of contingency effects. Data show that 50% of SMEs have digital support for more than two-thirds of their configuration activities, while 20% of SMEs have digital support for less than one-third of their configuration activities. In addition, data show that larger SMEs (>20 employees) report higher digitalization of configuration activities than smaller SMEs (<20 employees). We do not comment further on these data because they need a much deeper investigation (an investigation that considers not only digitalization but also some of the available context variables), which in turn requires a bigger sample.

For each configuration activity, we asked each SME whether it feels the need to improve the digitalization of this activity. Table 6 reports the results of this enquiry in column 4. Despite the fact that the various activities are digitalized, on average, in 58% of the SMEs, the need to improve existing digitalization is perceived, on average, by 48% of SMEs (see Table 6, column 4 last row). It is interesting to note that a number of SMEs felt the need to improve the digitalization of some activities that are already digitalized. This result suggests that in the future, besides greater digitalization, we can also expect to see better digitalization of configuration activities in the SMEs.

To try to gain additional clues about the future level of digitalization for each configuration activity, we calculated a desired level of digitalization (see Table 6 column 5) using the answers regarding the presence of each configuration activity, its actual digitalization, and the declared need to improve its digitalization. A company is counted in the numerator of this percentage when it digitalized the row activity or expressed the need to digitalize this activity. Each company is counted only once in this numerator. The percentage is calculated on the total number of companies that have the row activity. This percentage represents the desired level of digitalization: part of the desire is already satisfied, while part is reasonably expected to be satisfied in the future. Therefore, this percentage also provides an indication of the level of digitalization we could expect in the future. Likely, it refers to a near or at least not too far future since the part not yet digitalized is estimated based on an actually perceived need and not on a generic possibility of digitalizing something. The results of this analysis are that the average level of digitalization of the configuration activities is expected to grow from 58% to 74% if all companies digitalize activities for which they said they feel the need to do so.

Notably, selecting the product/service characteristics/ functionalities appropriate for the customer has a low percentage of digitalization (Table 6). Interestingly, this low percentage is almost the same when the selection activity is carried out by sales personnel (25%) and when it is done by clients on their own (20%; Table 7). However, it is interesting to note that all companies where this activity is carried out by clients expressed a desire to digitalize it (Table 7).

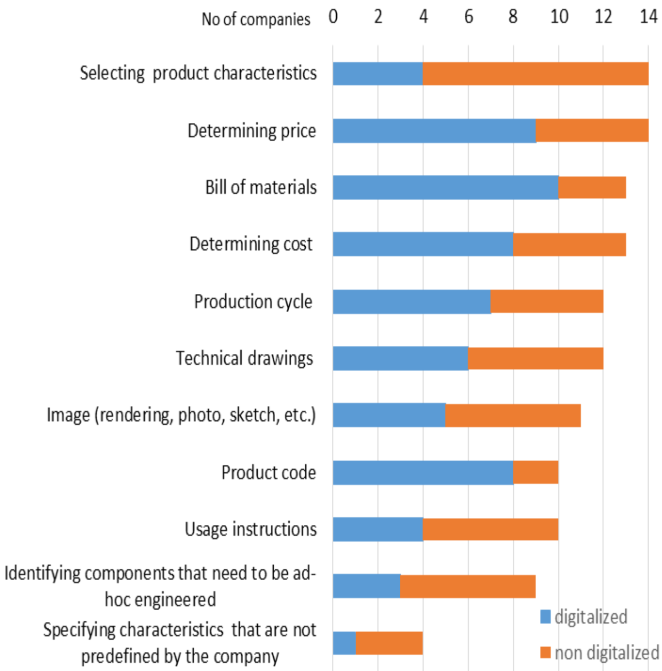


Figure 3. Level of digitalization of configuration activities

Table 6. Presence of activity, level of digitalization, and need for further digitalization

Configuration activity	**Presence of activity (%)**	**Actual digitaliz. (%)**	**Need to improve digitaliz. (%)**	**Desired digitaliz. (%)**	**N tot ***
Selecting the product/ service characteristics/ functionalities appropriate for the customer	100.0	21.4	21.4	42.9	14
Specifying characteristics/ functionalities requested by the customer that are not included in those predefined by the company	33.3	75.0	75.0	75.0	12
Identifying which product components/groups of the bill of materials, if any, need to be ad-hoc engineered	69.2	55.6	77.8	77.8	13
Generating/determining a new product/service configuration requested by the customer:					
its product code	71.4	80.0	40.0	90.0	14
its bill of materials	92.9	76.9	46.1	84.6	14
its production cycle	85.7	58.3	41.7	75.0	14
its price	100.0	69.2	57.1	78.6	14
its cost	92.9	61.5	46.1	69.2	14
its technical drawings	85.7	50.0	33.3	75.0	14
its image (rendering, photo, sketch, etc.)	78.6	45.4	45.4	81.8	14
its usage instructions	71.4	40.0	40.0	60.0	14
Mean	*81.1*	*57.6*	*47.6*	*73.6*	

* N tot. represents the number of SMEs that answered the question for the row configuration activities

Table 7. Presence of activity, level of digitalization, and need for further digitalization

Selecting the product/servic characteristics/ functionalities appropriate for the customer:	**Presence of activity (%)**	**Actual digitaliz. (%)**	**Need to improve digitaliz (%).**	**Desired digitaliz. (%)**	**N tot**
activity carried out by the client alone	28.6	25.0	75.0	100.0	14
activity carried out by salespeople	71.4	20.0	56.0	67.0	14

5 DISCUSSION AND CONCLUSIONS

We considered a small sample of SMEs (see 2.3.1). As far as we know, this is the first effort to describe the product configuration practices in SMEs. Therefore, even contextual information such as the sample description adds to the current knowledge. Half of the SMEs in our sample produce products for industrial applications, while a large portion of the other half produce them for final consumers (see 2.3.2). Almost all companies accept orders for products with new functions that require ad hoc design; half accept orders for products obtained by combining only predefined options, and more than half of the sample has no orders for final products already defined in the catalog (see 3.2.3). In line with this customization strategy, only a small part of their turnover is realized through intermediaries, while more than 80% is realized through direct selling (see 3.2.2). Interestingly, more than 80% of orders pass through the technical office; however, one-third of the companies have at least some orders fulfilled with products made to stock (3.2.4). Given these characteristics, our sample of manufacturing SMEs is characterized by mixed customization strategies with a strong presence of deep customization.

Given the above reported characteristics of the sample, it is not surprising that almost all the provided configuration activities were found in the considered sample (see 4.1). Some activities, in particular, are present in 80–100% of the considered SMEs, namely, selecting product characteristics, determining price, determining cost, producing a bill of materials, determining production cycle, producing technical drawings, and producing an image of the chosen product. In the considered SMEs, the product configuration process is complex, with many outputs. Thus, these manufacturing companies, though small, have to deal with great complexity due to the product customization strategy they adopted. Digitalizing the configuration process in these companies could free up considerable technical resources from the product configuration process during the order definition and/or order fulfillment processes.

Corresponding to the complexity induced by product customization, these SMEs present a digitalization status that is advanced as regards ERP and MRP, commercial presence on the web, CRM, and PDM/PLM (see 4.2). This does not mean that this digitalization is effective, but it does indicate that the considered SMEs are significantly open to digitalization. However, this claim deserves further reflection, since we have yet to deepen the analysis with a full sample. Our current grasp of our sample and data leads us to think that this statement will hold.

Even though the configuration activities are significantly digitalized, the level of digitalization differs vastly across activities (see 4.3). While determining the product code and producing the bill of materials are highly digitalized, other configuration activities, such as selecting product characteristics or producing images or technical drawings, are much less so. In this respect, there is a considerable gap between the current digitalization of the selection of product characteristics and what digitally oriented customers are increasingly requesting. Why this is so? This is a research question suggested by this result.

The gap we have highlighted is likely not unknown to these SMEs since the data show that they feel the need to further digitalize configuration activities (see 4.3). In fact, even though configuration activities are also significantly digitalized, this level of digitalization is expected to grow to 74% if all companies modify the activities for which they believe there is a need. In particular, all companies where the activity of selecting the product/service characteristics/functionalities appropriate for the customer is carried out by clients declared that this activity is to be further digitalized. To signal the strength of this trend, there are some cases that demonstrate the willingness of some companies to digitalize this activity and transfer it to customers, even in cases where it is not currently performed by customers. The fact that 72% of them have a commercial presence on the web and that 44% have a CRM and 53% have an MRP suggests that they have the bases to do it. However, many other things should be considered to determine whether they are mature enough for this step. This is another question for research that this result indicates as timely.

One important issue that emerged is the fact that the considered manufacturing SMEs follow mixed customization strategies with a high presence of deep customization. Almost all companies accept orders for products with new functions that require ad hoc design; half accept orders for products obtained by combining only predefined options, and more than half of the sample does not receive orders for product variants completely defined in the company's catalogs (see 3.2.3). In this context, the notion of partial configurability [11] is a crucial notion. Through this notion, it is possible to bring the benefits deriving from the use of configurators to these SMEs. Without this notion, in many cases, the use of configurators would not be justified, due to the relatively small portion of configured to order orders. The implication for configurators is that functionalities such as those that allow a partial commercial and technical configuration are highly important. It would be a mistake to think that these companies have limited interest in a configurational approach. In all likelihood, it is exactly the contrary. Through the configurational approach [11], they could reconsider their product space and their way of responding to the market. Using configurators that support some fundamental activities, they could increase their awareness about their product space and manage it in a better way.

A second important issue is associated directly with the SMEs' insight into their product space. Interestingly, the questions that yielded the highest number of missing data were those related to the size of the companies' product space. Respondents found it extremely difficult to provide responses for number of families, number of end product variants, and numbers of new end product codes introduced per year. We will go back to most of our sampled SMEs again to get these answers. This is not the first time we have had this experience with these kinds of companies. This issue is problematic for research because the information is important to characterize the context. However, we have also learned that this knowledge is available and is shared in a company when the configurational approach is implemented. From this consideration, we argue that the lack of this information may be used as an indicator of the status of evolution in managing product variety in a company, and, as researchers, we could probably use it in our explanatory models.

Another issue that emerged regards the way the configuration process is split between front and back office. The fact that even part of the orders that are fully defined in the catalogs have to pass through the technical office is something that has already been noticed by Forza and Salvador [11]. However, this is the first time it has emerged from a survey study. This phenomenon deserves further research to understand its causes. It could be that even the introduction of a sales configurator would not solve this issue because without certain individual competencies in the sales personnel, a final check from the technical office is needed. This is only one of the possible conjectures, but it should be enough to call researchers to reflect on it at least a little bit.

The picture that emerges from this partial sample is interesting and encouraging for product configuration researchers. If the same results emerge from a wider and more representative sample, this would indicate that product configuration activities are present in SMEs and are not simpler than those in bigger companies. On the contrary, due to the need to address partial configurability and due to mixed customization strategies, they could be even more difficult. Likewise, resources are more constrained in SMEs, and the volume of activities is lower. It is more difficult to have the required resources to introduce product configurators, and when they are introduced, the gain is likely lower because the smaller size leads to fewer configurations per year. In order to identify ways to overcome these issues, we think the research should consider identifying less expensive product configurators appropriate for SMEs, appropriate implementation processes, contexts that are more appropriate for digitalizing configuration activities, and the possibility of splitting the introduction of configurators into smaller packages that are affordable for SMEs. Researchers of product configuration are called to investigate in these directions.

ACKNOWLEDGEMENTS

We acknowledge financial support from the MC 4.0. Interreg V-A Italia–Austria project. Project ID: ITAT 1057. We thank the anonymous reviewers for their constructive comments.

REFERENCES

[1] B.J. Pine, *Mass Customization: The New Frontier in Business Competition*, Harvard Business Press, Brighton, MA, 1993.

[2] J.H. Gilmore and B.J. Pine, 'The four faces of mass customization', *Harvard Business Review*, **75**, 91–101, (1997).

[3] C. Svensson and A. Barfod, 'Limits and opportunities in mass customization for "build to order" SMEs', *Computers in Industry*, **49**, 77–89, (2002).

[4] D.M. Anderson, *Build-to-Order & Mass Customization: The Ultimate Supply Chain Management and Lean Manufacturing Strategy for Low-Cost On-Demand Production Without Forecasts or Inventory*, CIM Press, Cambria, CA, 2004.

[5] S.B. Taps, T. Ditlev, and K. Nielsen, 'Mass customization in SMEs: literature review and research directions', in *Managing Complexity,* 195–203, Springer, Cham, 2017.

[6] C. Forza and F. Salvador, 'Application support to product variety management', *International Journal of Production Research*, **46**, 817–836, (2008).

[7] J. Hagberg, M. Sundström, and E.Z. Nicklas, 'The digitalization of retailing: An exploratory framework', *International Journal of Retail & Distribution Management*, **44**, 694–712, (2016).

[8] P. Parviainen, M. Tihinen, J. Kääriäinen, and S. Teppola, 'Tackling the digitalization challenge: how to benefit from digitalization in practice', *International Journal of Information Systems and Project Management*, **5**, 63–77, (2017).

[9] Y. Su, W. Liao, Y. Guo, and Q. Ding, 'Key technologies for ASP-based product customization service system for SMEs: a case study', *The International Journal of Advanced Manufacturing Technology*, **42**, 381–397, (2009).

[10] K. Kristjansdottir, S. Shafiee, L. Hvam, C. Forza, and N. H. Mortensen, 'The main challenges for manufacturing companies in implementing and utilizing configurators', *Computers in Industry*, **100**, 196–211, (2018).

[11] C. Forza and F. Salvador, *Product Information Management for Mass Customization: Connecting Customer, Front-Office and Back-Office for Fast and Efficient Customization*, Palgrave Macmillan, New York, 2006.

[12] L. Hvam, M. Bonev, B. Denkena, J. Schürmeyer, and B. Dengler, 'Optimizing the order processing of customized products using product configuration', *Production Engineering*, **5**, 595–604, (2011).

[13] Suzić, N., Forza, C., Trentin, A., & Anišić, Z. 'Implementation guidelines for mass customization: Current characteristics and suggestions for improvement', *Production Planning & Control*, **29** (10), 856–871, (2018).

[14] D. Brown, 'Defining configuration', *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, **12**, 301–305, (1998).

[15] A. Felfernig, G. Friedrich, D. Jannach, M. Stemptner, and M. Zanker, 'Configuration Knowledge Representations for Semantic Web Applications'. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing* **17** (1), 31–50, (2003).

[16] L.L. Zhang, 'Product configuration: A review of the state-of-the-art and future research', *International Journal of Production Research*, **52**, 6381–6398, (2014).

[17] H.E. Tseng, C.C. Chen, and S.H. Chang, 'Applying case-based reasoning for product configuration in mass customization environments', *Expert Systems with Applications*, **29**, 913–925, (2005).

[18] C. Forza and F. Salvador, 'Product configuration and inter-firm co-ordination: An innovative solution from a small manufacturing enterprise', *Computers in Industry*, **49**, 37–46, (2002).

[19] M. Aldanondo, S. Rouge, and M. Reron, 'Expert configurator for concurrent engineering: Cameleon software and model', *Journal of Intelligent Manufacturing*, **11**, 127–134, (2000).

[20] F. Salvador and C. Forza, 'Configuring products to address the customization-responsiveness squeeze: a survey of management issues and opportunities', *International Journal of Production Economics*, **91**, 273–291, (2004).

[21] A. Trentin, E. Perin, and C. Forza, 'Overcoming the customization-responsiveness squeeze by using product configurators: beyond anecdotal evidence', *Computers in Industry*, **62**, 260–268, (2011).

[22] L. Hvam, A. Haug, N.H. Mortensen, and C. Thuesen, 'Observed benefits from product configuration systems', *International Journal of Industrial Engineering-Theory Applications and Practice*, **20**, 329–338, (2013).

[23] J. Tiihonen, M. Heiskala, A. Anderson, and T. Soininen, 'WeCoTin–A practical logic-based sales configurator', *AI Communications*, **26**, 99–131, (2013).

[24] C. Forza and F. Salvador, 'Managing for variety in the order acquisition and fulfilment process: The contribution of product configuration systems', *International Journal of Production Economics,* **76**(1), 87–98, (2002).

Organized by

Università degli Studi di Padova

Graz University of Technology

DTU Technical University of Denmark

Sponsored by

SIEMENS Ingenuity for life