

Faculty of Engineering Technology
Master Thesis Industrial Design Engineering

Configurator type selection improvement through multi criteria decision making

Author

ing. T.D. de Vries

Examination committee

prof.dr. I. Gibson

(chairman)

dr. M.V. Pereira Pessoa

(supervisor)

ir. I.F. Lutters - Weustink

(external member)

Document Number

DPM1669

Date

4-2-2019, 21-1-2020

UNIVERSITY OF TWENTE.

Abstract

Configurators are used to support the customisation of products through storing, retrieving and processing product variant information. However, companies are struggling to select a configurator system that supports their customisable product. The many existing configurator types and the lack of knowledge within companies results in selection of unsuitable configurator systems. The implementation of unsuitable configurator systems causes a troubled configurator performance and an insufficient support of product customisation.

In this thesis, configurator types are investigated in order to develop a method that selects configurator types for companies to prevent the selection of unsuitable configurator systems. This is achieved through first performing a literature study to determine how an ideal configurator can be selected and which configurator types exist.

Literature shows that configurator types should be selected based on their characteristics, which should be in accordance with the characteristics of a particular customisable product of a company. Each configurator type's characteristics and operation are analysed, in order to create selection criteria. The selection advice for configurator types with multiple characteristics can be conflicting with the characteristics of a customisable product. Configurator types can contain both suitable and unsuitable characteristics for the customisable product which results in conflicting configurator type selection advice.

Secondly, Multi criteria decision making (MCDM) is reviewed to determine a selection process for configurator types with conflicting selection advice. The five popular MCDM methods, TOPOSIS, AHP, VIKOR, ELECTRE and PROMETHEE are investigated in the literature study to determine which method can support the selection process of configurator types with conflicting selection advice.

The proposed method is developed through defining shared characteristics for which the configurator types can be selected and through creating a multi criteria selection process for configurator types with conflicting selection advice.

The selection process of the 24 configurator types with non-conflicting selection advice is simplified through using a configurator type as selection reference which represents a set of configurator characteristics. The selection of 9 configurator types through evaluating the 24 existing configurator types with non-conflicting selection criteria is simplified through answering only 4 input questions.

Of the 5 configurator types with conflicting advice, the optimal type is selected through using 2 additional input questions and through performing multi criteria selection. The configurator types with conflicting advice are ranked with the TOPSIS method based on their Euclidean distance to the ideal configurator type's characteristics. The AHP method is used to determine the importance of the characteristics for which the configurator types with conflicting advice are compared in the TOPSIS method.

A descriptive chapter is included to demonstrate how a company can use the proposed method to perform a selection of configurator types. The ranking and weighting performed with the AHP and TOPSIS methods are demonstrated for an existing case.

The validation of the proposed method is done by analysing two existing configurators. In each case, the proposed method is used to create a selection of configurator types which are compared to the analysed configurator types of the existing configurator. The results of the two cases are discussed and used to suggest further improvements of the proposed method. The thesis is concluded through evaluating to which degree the proposed method fulfils the requirements.

Acknowledgements

I would like to thank dr. Marcus V. P. Pessoa, my thesis supervisor, for his support and advice. I give thanks to God for His guidance during my life and support during my study.

Glossary

Decision Maker (DM) searches for the most optimal solution for a multi-objective problem. 19, 26, 30

Mass customization (MC) refers to the marketing of products with large personalization customization of components/features possibilities which creates a lot of product variants in mass production. 9

Multi attribute decision making (MADM) MADM refers to the selection of the most suitable alternative from pre-specified alternatives described in terms of attributes (Gavade, 2014). 26

Multi criteria decision making (MCDM) MCDM refers to making decisions in the presence of multiple, usually conflicting, criteria (Zavadskas et al., 2014). 2, 25

List of figures

1	Configurator type selection based on the characteristics of a customisable product, source by the author	10
2	Methodology implementation, aim and results, based on the DRM framework of Blessing & Chakrabarti (2009)	13
3	Configuration process, retrieved from Forza & Salvador (2008)	15
4	Configuration process, retrieved from Forza & Salvador (2008)	16
5	Bottom-up & top-down approach, <i>source by the author</i>	22
6	AHP: the fundamental scale of absolute numbers, <i>source Saaty (2008)</i>	27
7	Configurator type characteristics, <i>source by the author</i>	31
8	Example plot of configurator type characteristics, <i>source by the author</i>	32
9	Selecting sequence	36
10	Selection of 24 configurator types for non-conflicting selection advice, <i>source by the author</i>	48
11	Selection of 24 configurator types for non-conflicting selection advice, <i>source by the author</i>	62
12	Configurator types selection with non-conflicting selection advice, <i>source by the author</i>	74
13	IKEA: Configurator types selection with non-conflicting selection advice, <i>source by the author</i>	81

List of tables

1	Configurator types selection advice based on their characteristics, <i>source by the author</i>	33
2	Selection based on a few configurator types, <i>source by the author</i>	38
3	Feature-based and technical element type selection criteria, <i>source by the author</i>	41
4	Configurator organisation type selection criteria, <i>source by the author</i>	42
5	Life cycle support type selection criteria, <i>source by the author</i>	43
6	Selection criteria for scope of use type, <i>source by the author</i>	44
7	Interaction nature type selection criteria, <i>source by the author</i>	45
8	Update execution type selection criteria, <i>source by the author</i>	45
9	Integration level type selection criteria, <i>source by the author</i>	46
10	Configurator complexity type selection criteria, <i>source by the author</i>	47
11	Selection criteria for configurator types, <i>source by the author</i>	49
12	Configurator type characteristics with conflicting selection advice, <i>source by the author</i>	50
13	Rule-based reasoning type selection criteria, <i>source by the author</i>	53
14	Case-based reasoning type selection criteria, <i>source by the author</i>	54
15	Logic-based reasoning type selection criteria, <i>source by the author</i>	54
16	Resource-based reasoning type selection criteria, <i>source by the author</i>	55
17	Constraint-based reasoning type selection criteria, <i>source by the author</i>	56
18	Decision matrix of the knowledge based reasoning types selection, <i>source by the author</i>	57
19	Decision matrix with numbers of the reasoning types selection, <i>source by the author</i>	57
20	Decision matrix selection of questions $c_{1a}, c_{2a}, c_{5a}, c_{6a}$, <i>source by the author</i>	63
21	Random index (RI) table (Saaty, 1987)	65
22	Decision matrix table for input of Blue-Lagoon, <i>source by the author</i>	74
23	Blue-Lagoon configurator type comparison, <i>source by the author</i>	76
24	Decision matrix table for input of IKEA, <i>source by the author</i>	82
25	IKEA configurator types comparison, <i>source by the author</i>	83

List of equations

1	MCDM matrix	26
2	AHP: reciprocal matrix of the selection criteria	64
3	AHP: summed reciprocal matrix	64

4	AHP normalised reciprocal matrix	64
5	APH: Principal Eigen vector calculation	65
6	APH: formula of λ_{max} calculation	65
7	APH: λ_{max} determination	65
8	APH: Consistency Index	65
9	APH: Consistency Ratio	65
10	APH: Consistency Index determination	66
11	APH: Consistency Ratio determination	66
12	TOPSIS: complete reasoning matrix	66
13	TOPSIS: normalisation example of decision matrix (Opricovic & Tzeng, 2004) . . .	66
14	TOPSIS: first step to matrix normalisation	66
15	TOPSIS: normalised matrix	67
16	TOPSIS: weighted normalised matrix	67
17	TOPSIS: weighted normalised matrix selection	67
18	TOPSIS: calculation of the ideal and negative-ideal solution Opricovic & Tzeng (2004)	68
19	TOPSIS: ideal and negative ideal determination	68
20	TOPSIS distance to the ideal and negative-ideal solution Opricovic & Tzeng (2004) .	68
21	TOPSIS: example of ideal Euclidean distance calculation	68
22	TOPSIS: determination of the relative closeness to ideal solution	69
23	TOPSIS: example ranking	69
24	AHP: Blue-lagoon reciprocal matrix	75
25	APH: weighing based on the desires of Blue-Lagoon	75
26	TOPSIS: ranking based on the desires of Blue-Lagoon	75
27	APH: weights of the example presented in paragraph 4.1.2	77
28	TOPOSIS: ranking based on weights from the example of paragraph 4.1.2	77
29	AHP: IKEA reciprocal matrix	82
30	APH: weights based on IKEA's desires	83
31	TOPSIS: ranking based on IKEA's desires and input	83

Contents

Abstract	2
Acknowledgements	3
Glossary	4
List of figures	5
List of tables	5
List of equations	6
1 Introduction	9
1.1 Background on configurators	9
1.2 Problem description	10
1.3 Goal	11
1.4 Scope	12
1.5 Methodology	12
2 Literature review	14
2.1 Configuration process	15
2.2 Configurator types	16
2.2.1 Knowledge based reasoning	17
2.2.2 Solution searching	21
2.2.3 Configurator organisation	22
2.2.4 Life cycle support	22
2.2.5 Business strategy	23
2.2.6 Scope of use	23
2.2.7 Interaction nature	24
2.2.8 Update execution	24
2.2.9 Integration level	24
2.2.10 Complexity	25
2.3 Multi criteria decision making	25
2.3.1 AHP	27
2.3.2 TOPSIS	28
2.3.3 VIKOR	29
2.3.4 ELECTRE	29
2.3.5 PROMETHEE	30
2.4 Literature conclusion	30
2.4.1 Configurator type characteristics	30
2.4.2 Selecting strategy	34
3 Method development	36
3.1 Configurator type selection process for non-conflicting advice	37
3.1.1 Determination of (shared) characteristics and questionnaire	39
3.1.2 Solution searching type selection	40
3.1.3 Configurator organisation type selection	41
3.1.4 Life cycle support type selection	42
3.1.5 Business strategy type selection	43
3.1.6 Scope of use type selection	43
3.1.7 Interaction nature type selection	44
3.1.8 Update execution type selection	45
3.1.9 Integration level type selection	45

3.1.10	Complexity type selection	46
3.1.11	Selection process	47
3.2	Configurator type selection process for conflicting advice	50
3.2.1	Reasoning type selection criteria and questionnaire	51
3.2.2	Rule-based type selection	52
3.2.3	Case-based type selection	53
3.2.4	Logic-based type selection	54
3.2.5	Resource-based type selection	55
3.2.6	Constraint-based type selection	55
3.2.7	Reasoning type selection process	56
3.3	Multi criteria decision making process	57
3.3.1	AHP method ratio of importance determination	58
3.3.2	TOPSIS evaluation	58
3.3.3	MCDM method selection	59
4	Method description	60
4.1	Configurator type selection with non-conflicting selection advice	60
4.2	Reasoning type selection	62
4.2.1	Ratio of importance determination	63
4.2.2	Multi criteria selection	66
5	Case study	70
5.1	Reptile enclosure configurator	70
5.1.1	Enclosure configurator evaluation	70
5.1.2	Enclosure configurator configuration selection	72
5.1.3	Enclosure configurator type results comparison	75
5.2	Kitchen furniture configurator	78
5.2.1	Kitchen configurator evaluation	78
5.2.2	Kitchen configurator configuration selection	79
5.2.3	Kitchen configurator type results comparison	83
6	Discussion	85
6.1	Requirements	85
6.2	Selection process	85
6.3	Improvement	86
7	Conclusion	87
7.1	Future work	88
	References	89

1 Introduction

Currently, industry is becoming more competitive where customers want to differentiate themselves through purchasing unique products that fit to their specific wishes (Kristjansdottira et al., 2018). Mass customization (MC) is introduced to cope with the increasing desire of customers to purchase products which fit to their idiosyncratic needs (Haug et al., 2012; Trentin et al., 2014). The difficulty of product customisation roots in the storage, retrieval and processing of product variant information, e.g. documentation to produce the product variant.

1.1 Background on configurators

Configurators are created to allow the customisation of products through a configuration process (Forza & Salvador, 2008; Trentin et al., 2014). The configuration process utilises product customisation by collecting the customer's wishes and identify the corresponding product variant. The configurator supports product customisation through the storage, retrieval and processing of information of customised product-variants (Forza & Salvador, 2008; Trentin et al., 2014).

There are many different types of configurators which can be used to perform the configuring process. The selection of such configurator types for real-world projects/companies are very complex and require substantial domain knowledge (McGuinness & Wright, 1998; Haug et al., 2012). Not all configurator types are suitable to perform the configuration process for specific customisable products.

Configurator types can be classified based on their characteristics. In order to know which configurator types can be selected for a specific product, their characteristics have to be considered. These characteristics can be used to define the selection criteria to determine which configurator types are most suitable for a specific product. In *figure 1* an example is presented of the different aspects that influence the configurator selection.

The selection of configurator types can be conflicting for configurator types with both desired and undesired characteristics. Multi criteria decision making (MCDM) is used to solve selection problems of alternatives with conflicting criteria (Gavade, 2014; Zavadskas et al., 2014). The configuration process can also use MCDM to identify the most suitable configurator types for conflicting criteria.

The configurators have to solve conflicting selection criteria as well, for example when selecting a product variant with the lowest weight ratio, as presented in *figure 1*. A slightly heavier product variant can have a greater performance; the configurator has to make a trade-off which weight ratio is optimal for the given customer requirements.

Knowledge-based reasoning configurator types are used to solve selection problems of product variants with conflicting criteria. The knowledge-based reasoning is the main characteristic of a configurator and includes the database and configuration logic (Blecker et al., 2004). The configurator reasoning configurator types are evaluated in more detail throughout the thesis, since their selection has the most influence on the storage, retrieval and processing of product variant information.

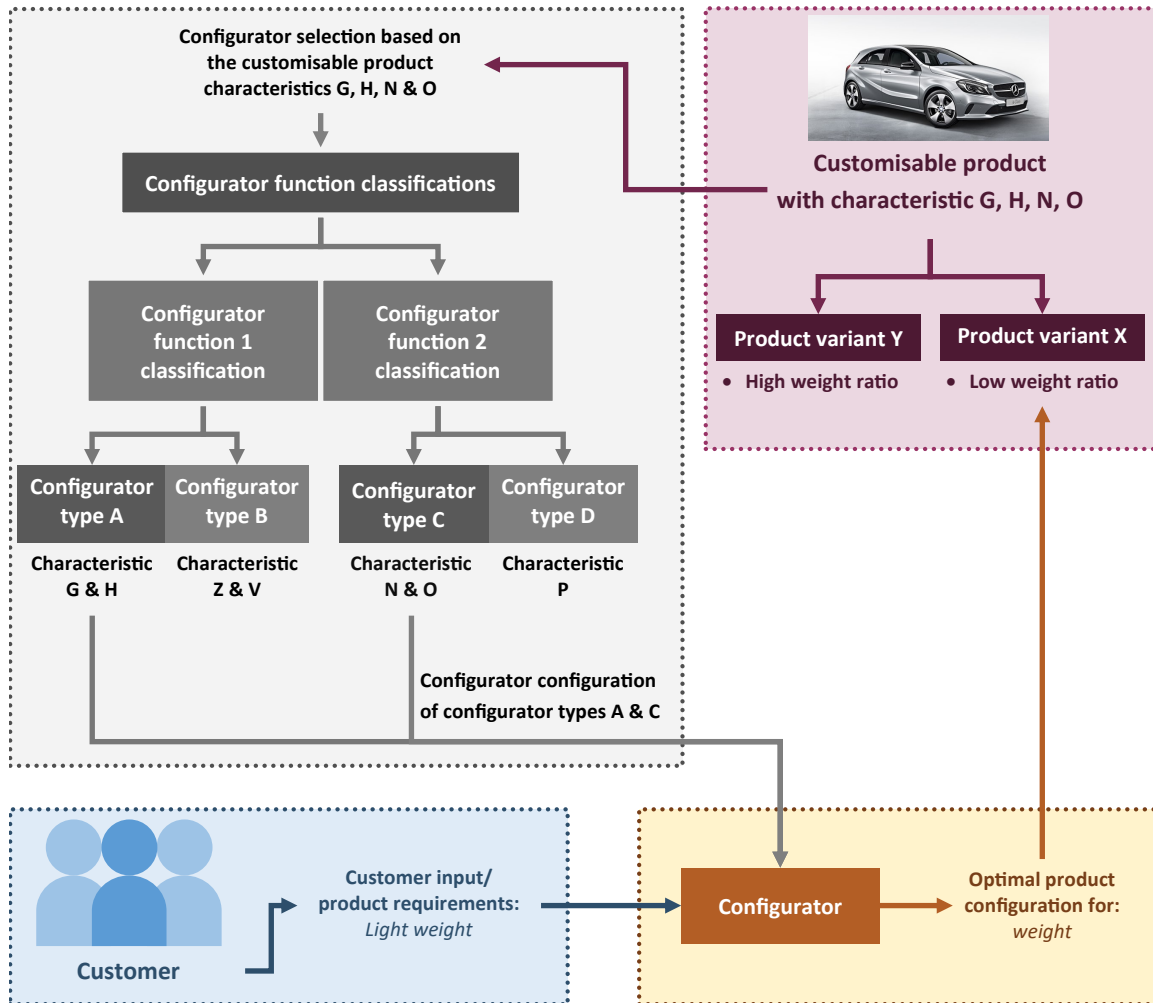


Figure 1: Configurator type selection based on the characteristics of a customisable product, source by the author

1.2 Problem description

Forza & Salvador (2008) described that companies often select unsuitable types of configurator systems due to lack of understanding of the system functions. The selection of not suitable/optimal configurator types/strategies can ultimately lead to diminishing or even losing the benefit of using a configurator to support product customisation (Kristjansdottira et al., 2018). Configurator projects are even abandoned due to the difficulty of implementation and the insufficient support of the company's needs (Haug et al., 2012). The selection of unsuitable configurator types results in a reduced efficiency of the information infrastructure to support product-variant-management/customisability (Forza & Salvador, 2008). To prevent unsuitable implementations of configurator types, the selection of configurator types should be made simpler for companies. The configuration process can be applied to support the selection of configurator types and identify the most suitable types to utilise the customisation of a specific product.

1.3 Goal

The purpose of this master thesis is to enable companies that lack understanding of configurator systems to select a suitable configurator system for their particular customisable product. In the thesis, a method is developed that selects configurator- types based on the specific characteristics of a company's product to utilise product customisation. The main objective of the thesies is to:

- Investigate, develop and validate a method for selecting the optimal configurator configuration based on the characteristics of a particular customisable product.

The main objective is achieved through sub-objectives., tThese sub-objectives are used to give structure in the research procedure of the thesis. The goals of the sub-objectives are to investigate the configuration process, configurator types, selection process and the fulfilment of the proposed method for the main objective.

The configuration process is investigated in order to determine how a configurator can support the storage, retrieval and processing of product variant data.

- Investigation of the configuration process used in configurators to configure a product.
Presented in paragraph 2.1

In order to determine how an ideal configurator system is selected, the configurator types and the characteristics for which they can be selected are analysed.

- Investigation of the different configurator types and their characteristics.
Presented in paragraph 2.2
- Analysis to determine the requirements of an ideal configurator.
- Determination of selection criteria to select the ideal configurator types.
Both presented in the literature review conclusion in paragraph 2.4

Multi criteria decision making is investigated to determine a selection process for configurator types with conflicting selection advice/criteria.

- Investigation of the operation of popular MCDM methods.
Presented in paragraph 2.3
- Review of MCDM methods to use for the selection of configurator types with conflicting selection advice.
Presented in the literature review conclusion in paragraph 2.4

Development of the proposed method to perform the selection of configurator types, based on the analysed configurator types and their (common) characteristics in the literature review.

- Development of the configurator types' selection process based on their (common) characteristics.
Presented in paragraph 3.1 and paragraph 3.2
- Development of a selection process with multi criteria decision making methods for configurator types with conflicting selection advice.
Presented in paragraph 3.3
- Development of questions to obtain the characteristics of a particular customisable product.
Presented in paragraph 3.1 and paragraph 3.2
- Description of the proposed method to explain how companies can use the method.
Presented in chapter 4

The proposed method is validated through a descriptive study.

- Two empirical case studies are performed to validate the proposed method.
Presented in paragraph 5.1 and paragraph 5.2
- Discussion of the results of the empirical case studies.
Presented in chapter 6
- Conclusion of the proposed method and how it fulfils the set objective.
Presented in chapter 7

1.4 Scope

The scope of this thesis is the development and validation of a method that selects a configuration of configurator types based on the in the literature investigated characteristics of a customisable product.

The method selection is restricted to the in the literature found configurator types that support the function of the investigated sales configuration. The sales configuration uses the communication between the customer and the company to come to a sales agreement (Forza & Salvador, 2008; Trentin et al., 2014). The storage, retrieval and processing of the product variant information of the product that the customer wishes to buy and the company agrees to supply is the output of the sales configuration. Therefore, a method is required that selects configurator types that perform the function of a sales configuration to solve the defined problem if paragraph 1.2.

1.5 Methodology

A design research methodology (DRM) described by Blessing & Chakrabarti (2009) is used to fulfil the set objectives. The design research methodology of this thesis is performed with following stages, namely: research clarification, descriptive study I, prescriptive study and descriptive study II. Specific objectives and requirements are made to fulfil the main objective. The methodology is used to have a scientific research approach to substantiate the development of the proposed method. The implementation, aim and results of the four stages are presented in *figure 2* with the corresponding chapters and paragraphs.

For each chapter, the methodology stages are explained with their aim and results, which can be used as readers' guide for the thesis.

Chapter 1: The research clarification is performed to support the assumption that companies have trouble choosing a suitable configurator for their specific customisable product and to define the research goals/objectives. Paragraph 1.1 illustrates that configurators are used for customisation by storage, retrieval and processing of product variant data. The problem description is presented in paragraph 1.2, which encompasses that companies select unsuitable configurator systems that reduce the configurator efficiency in storage, retrieval and processing of product variant information. The thesis goal and the set objective and subobjectives are defined in paragraph 1.3.

Chapter 2: The descriptive study is performed by analysing empirical data in a literature review of the configurator process, configurator types and MCDM methods. The configuration process is reviewed in paragraph 2.1 to create understanding of how this process is used in configurators to support the processing, storage and retrieval of product variant data. The configurator types are analysed in paragraph 2.2 in order to create requirements to select an ideal configurator. The configurator classifications are investigated to determine for which characteristics the configurator types can be selected. In paragraph 2.3, five popular MCDM methods, AHP, VIKOR, ELECTREE, TOPSIS and PROMETHEE, are reviewed to determine which method is most suitable to select configurator types with conflicting selection advice.

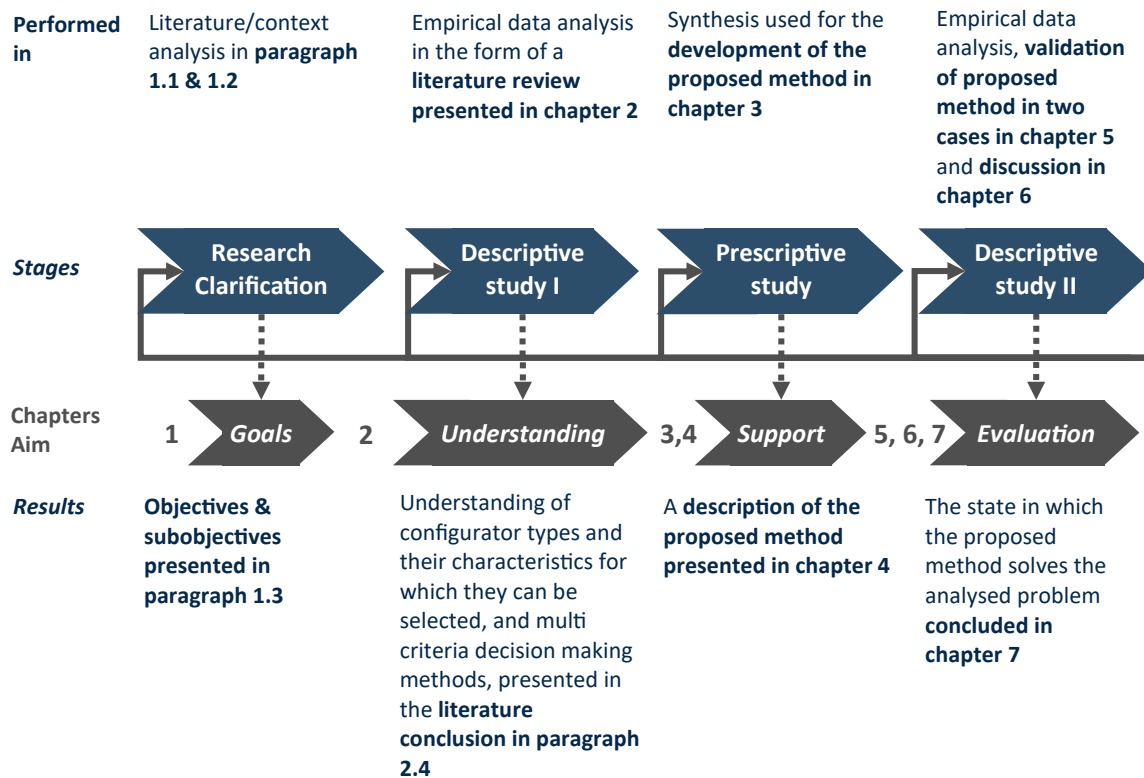


Figure 2: Methodology implementation, aim and results, based on the DRM framework of Blessing & Chakrabarti (2009)

Chapter 3: During the prescriptive study, the proposed method is developed based on the reviewed literature. The analysed characteristics of the literature review are used in paragraph 3.1 and 3.2 to develop selection criteria for configurator types. The configurator types with conflicting selection criteria are reviewed to determine how they can be solved through multi criteria decision making. The AHP method is reviewed in paragraph 3.3 for developing a process to determine weighting, which is required to perform multi criteria decision making. The remaining four MCDM methods are reviewed for performing the selection process of configurator types with conflicting selection advice in the proposed method, from which the TOPSIS method is chosen.

Chapter 4: A description and an example are given of how the proposed method can be used by a company for a particular customisable product.

Chapter 5: In descriptive study II, the proposed method is validated based on its performance through an empirical study of two cases of an existing configurator. The two cases compare an existing product configurator to a configuration made with the proposed method for the same customisable products. The public product configurators used for the validation of the proposed method are retrieved from two different companies. Through the comparison, the proposed method's functioning is validated.

Chapter 6: The functioning and the development of the proposed method is discussed. The proposed method's function is reviewed to determine if the proposed method satisfies the set requirements. The results of the cases are analysed to suggest improvements of the method.

Chapter 7: A conclusion is made to determine to which degree the proposed method fulfils the set objective.

2 Literature review

This chapter investigates the following topics as research foundation: the configuration process in paragraph 2.1, classified configurator types in paragraph 2.2 and multi criteria decision making in paragraph 2.3.

The configuration process is reviewed to understand how configurator types can support the storage, retrieval and processing of product variant information. The configurator process is used in the proposed method to select the optimal configurator types for a specific customisable product. Scopus and Google Scholar are used to find all information presented in the literature review. Papers concerning “mass customisation” are investigated to understand the supporting role of configurators in product customisation. The information storage, retrieval and processing of product variants by configurators, is analysed through reading papers that explain the configuration process. The configurator process is investigated further through searching on its specific sub-processes performed in the “sales configuration” and the “technical configuration”.

Configurator types are reviewed to determine requirements to select an ideal configurator. The classifications of configurator types are analysed in order to know the types that can be configured in the proposed method. The configurator performance for a specific customisable product depends on the accordance of the configurator characteristics with the product. Therefore, the selection of configurator types should be based on their characteristics. The configurator types operation and characteristics are analysed to be later used as selection criteria in the proposed method.

The knowledge-based reasoning configurator types are of greater importance for the configurator performance for customisable products. To guarantee an optimal performing configured configurator, the characteristics and operation of these configurator types are reviewed in greater detail.

To find information about existing configurator types, papers concerning the state of the art of product configuration systems are searched. A paper of Blecker et al. (2004) was found, which described configurator classifications and their configurator types. In an attempt to search for more information on each classification and configurator type their names and the references presented in of the paper of Blecker et al (2004), are used for further investigation. The reasoning configurator types had a lot of search results, however other classifications often had less results and were difficult to find.

The knowledge-based reasoning configurator types have multiple operation characteristics and can have conflicting selection criteria. A configurator type is conflicting when it has both advised and unadvised characteristics. Multi criteria decision making is analysed to be used for the selection of conflicting configurator types in the proposed method. The popular MCDM methods AHP, TOPSIS, VIKOR, ELECTREE and PROMETHEE are reviewed on their operation, strengths and weaknesses. Based on the MCDM methods’ operation, strengths and weaknesses, the most suitable methods are chosen to perform the selection for conflicting configurator types in the proposed method.

Multi criteria decision making is reviewed to select the reasoning configuration, since the functioning of these methods is accessible. The information on the MCDM methods is obtained by searching for “multi criteria decision making methods”. The papers described two types of multi criteria methods, “multi objective decision making” and “multi criteria attribute decision making”. Multi criteria attribute decision making methods is further investigated, since they are used for selection different from the objective decision making which is used for optimisation. Multi criteria selection is most relevant for the proposed method which has to select configurator types based on their shared characteristics.

2.1 Configuration process

The configuration process of customisable products is performed by configurators. The configuration process is analysed to investigate how configurators operate and support mass customisation. Configurators are used to provide guidance for both the sales and production of companies which produce products for mass customisation. The configuration process collects customer needs as input to create the documentation which is needed to produce the requested product variant (Forza & Salvador, 2008; Trentin et al., 2014). The configuration process is divided into two sub-processes, namely the sales configuration and the technical configuration, presented in *figure 3* (Forza & Salvador, 2008). The description of customer wishes of a product to buy and the manufacturer agrees to produce are contained in the sales configuration. The completed sales configuration is handed to the technical configuration. The technical configuration uses the sales configuration to identify how the product variant can be produced and creates documentation for assembly and manufacturing.

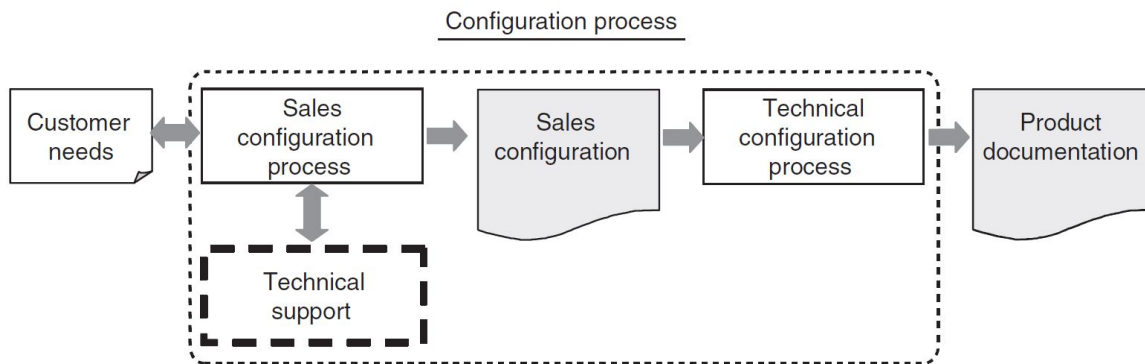


Figure 3: Configuration process, retrieved from Forza & Salvador (2008)

In the sales configuration, the customer and the company communicate to come to a sales agreement (Forza & Salvador, 2008; Trentin et al., 2014). The sales agreement of a product variant is the output of the sales configuration. In the sales configuration, product specifications are checked to be logical and not contradicting with each other, for example if it is producible. The sales configuration is aimed at customer support and is the front-end of the configuration. A graphical user interface is often used in the sales configuration, to guide the customer through the configuration process (Abbasi et al., 2013).

Documentation of the product variant such as structures, bill of materials, drawings and production sequences are the output of the technical configuration (Forza & Salvador, 2008; Trentin et al., 2014). The technical configuration uses the sales configuration as input and does not communicate with the customer. All activities which are needed to generate documentation for the production of product variants are executed in the technical configuration (Forza & Salvador, 2008; Trentin et al., 2014). For example, the processing, storage and retrieval of information during the manufacturing of product variants is performed in the technical configuration. The technical configuration supports the employees which are involved in the production (and delivery) of the product (Forza & Salvador, 2008).

The sales configuration can be different, depending on the customisable product. Two different configurators are presented in *figure 4.A and 4.B*, both represent a different sales configuration. The sales configuration of *figure 4.A* selects a PC based on the game a customer desires to play. In the sales configuration of *figure 4.B* a PC is configured based on the customers input about their desired component specifications. The technical configuration can be the same for both cases, which develops documentation for the manufacturing of the agreed upon product variant in sales configuration.

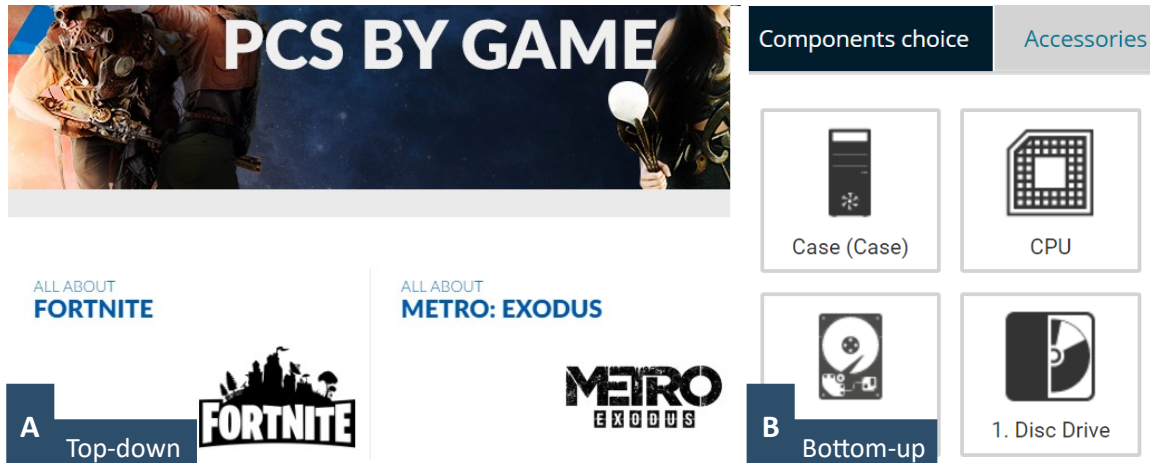


Figure 4: Configuration process, retrieved from Forza & Salvador (2008)

The configuration process of configurator types for specific customisable products is performed through a sales configuration. The sales configuration can be different through selecting different configurator types. The sales configuration requires the input from a company about their customisable product to configure an optimal configuration of configurator types.

2.2 Configurator types

The configurator types are investigated to determine which types can be selected, to find an answer to the question; “how can a good or bad configurator be identified” and to say something about the configurator performance.

An ideal configurator should support customisation through the storage, retrieval and processing of product variant information for a particular customisable product (Forza & Salvador, 2008; Trentin et al., 2014). Configurator type characteristics represent how the configurator stores, retrieves and processes information specific for a customisable product. Therefore, the configurator types need to be selected based on their accordance with the characteristics of the customisable product.

Projects with improper configurator system selections are often abandoned by companies (Haug et al., 2012). A good or bad performing configurator can be identified based on the accordance of the configurator types with the customisable product characteristics for which they are selected. An example of a bad configurator system selection, is when the customisable product changes faster than the configurator maintenance can be performed. The configurator types selection is bad, since the types are not in accordance with the rate in which the customisable product changes.

The knowledge based reasoning configurator types are mainly responsible for the storage, retrieval and processing of configurator data and logic (Blecker et al., 2004). In order to have an ideal configurator, the reasoning type has to be in accordance with the characteristics that represent the company’s strategy for the customisable product.

The requirements for an ideal configurator:

- The configurator types are selected according to the analysed characteristics in the literature
- The selected reasoning configurator type has to be in balance with the company’s strategy for the customisable product

Blecker et al. (2004) classified configurator types based on their function, for example a configuration knowledge classification contains configurator types that perform knowledge based reasoning. The classified configurator types possess different characteristics. For example, a reasoning configurator type can be optimal, suitable or not-suitable for a specific characteristic such as customer guidance.

The different configurator types for each classified function are reviewed in the following paragraphs, in order to obtain their characteristics.

A configurator can be classified as follows(Blecker et al., 2004):

- *Knowledge based reasoning*
- *Solution searching*
- *Organisation*
- *Life cycle support*
- *Business strategy*
- *Update execution*
- *Interaction nature*
- *Scope of use*
- *Integration level*
- *Complexity*

2.2.1 Knowledge based reasoning

The knowledge-based reasoning is the main component of a configurator and includes the configuration logic and the configurator database (Sabin & Wiegel, 1998; Blecker et al., 2004). The configuration logic determines how a configuration can be made through specifying sets of restriction on how components can be combined. The database contains all information about the components and their instances.

When components or specifications of customisable products change, the configurator knowledge and logic have to be updated. The configurator knowledge and logic are updated through knowledge maintenance. Knowledge maintenance is performed through acquisition which encompasses the transfer of knowledge in a configurator (Haug et al., 2012).

Knowledge acquisition can often lead to man made errors, like providing incomplete or incorrect configuration logic such as missing relations between components, contradicting information, or multiple definitions of components (Sabin & Wiegel, 1998; McGuinness & Wright, 1998; Blecker et al., 2004). Configurators represent wrong information or cannot compute a solution through these missing links and information.

Man made errors during knowledge acquisition can be prevented through consistency checking (McGuinness & Wright, 1998). Consistency checking can also be used to prevent contradicting customer choices during configuring. However, not all knowledge-based reasoning configurator types use consistency checking (Sabin & Wiegel, 1998; Juengst & Heinrich, 1998; McGuinness & Wright, 1998; Blecker et al., 2004). The reasoning configurator types differ in their suitability for customisable products that require knowledge acquisition.

The complexity of a customisable product determines the amount of required knowledge acquisition. For example, when a simple product is used, little knowledge acquisition is required to transfer the information of the components and functions of the product. Complex products require more knowledge acquisition to transfer the information of different components, functions and relations between components.

Reasoning configurator types differ in their customer guidance. For some types, prior expert customer product knowledge is required. For example, some configurator types automatically compute a configuration based on customer requirements, without the need for customers to select components (Sabin & Wiegel, 1998; Juengst & Heinrich, 1998). While other reasoning configurator types do not possess automated configuring and require prior product knowledge of customers in order to select components based on their functions and relations.

The knowledge-based reasoning configurator types differ in for the optimisation characteristic, e.g. one reasoning configurator type can perform optimisation based on the required resources (Sabin & Wiegel, 1998; Juengst & Heinrich, 1998). However, not all reasoning configurator types can be used for optimisation (McGuinness & Wright, 1998).

The knowledge-based reasoning of a configurator can be performed with a rule-based, model-based or case-based configurator type (Blecker et al., 2004). To determine which configurator type is optimal for a specific customisable product, the knowledge acquisition, optimisation and customer guidance characteristics are reviewed for each reasoning configurator type.

Rule-based configurator type

The rule-based configurator type uses “if” condition “then” consequence statements and has no separation between configurator knowledge and configuration logic (Sabin & Wiegel, 1998; Blecker et al., 2004). The knowledge of compatibilities between components and the control strategy to compute the solution to a specific configuration problem are both contained in rules. The rules determine the order in which configuration actions are executed. The configuration process executed with a rule-based configurator is difficult to retrace, because the configurator knowledge and configuration logic are not separated.

The rule-based configurator type is not suitable for the configuration of complex products, since it has difficulties with knowledge acquisition, consistency checking and knowledge maintenance (Sabin & Wiegel, 1998; Blecker et al., 2004). The component descriptions, relations and configuring strategy are merged and contained in rules. Complex products require more rules to contain the configurator knowledge and computing strategies. Configurator maintenance of complex products is difficult due to the many rules that have to be checked and adjusted if configurator knowledge is updated. The lack of structure causes many made errors during knowledge acquisition. Consistency checking is also more difficult for complex products, because the descriptions of components are merged with the computing strategy in rules.

A rule-based configurator type is optimal for simple customisable products, which contain little configuring information. The rules are concise and simple when little configurator knowledge and logic are contained to configure simple products.

The suitability of the rule-based configurator type depends on the customisable product complexity. The rule-based configurator type is not suitable for products that have intermediate or high complexity and are optimal for simple products.

Case-based configurator type

The case-based configurator type uses neither deductive or abducting schemes to derive a solution, instead it uses a different approach (Sabin & Wiegel, 1998; Blecker et al., 2004). Case-based reasoning uses previously solved problems that are most corresponding with the current unsolved case and attempts to solve the current case through adapting a previous solution.

The case-based configurator type is suitable if the customisable product consists of a few standard product models and uses model-based adaption methods to modify these products (Sabin & Wiegel, 1998). The case-based configurator type is used for product optimisation based on customer requirements.

The case-based reasoning extracts features of the customer requirements to represent the desired functional roles of the customisable product. Based on the extracted features, the case-based configurator type searches for a case with similar features. The most similar old cases are adapted to solve the new case. Rule-based systems are used to adapt the case to a new situation (Sabin & Wiegel, 1998).

The case-based configurator type is not suitable when there are no previously solved cases or methods to modify standard products. The use of rule-based systems in the case-based reasoning causes knowledge acquisition and maintenance problems for complex products. The case-based reasoning configurator type is suitable for intermediate customisable products with adjustable components.

Model-based knowledge based reasoning configurator types

There are three main types of model-based configurators, namely logic-based, resource-based and constraint-based (Sabin & Wiegel, 1998; Blecker et al., 2004). The resource-based configurator type is based on a producer-consumer model. The most prominent logic-based configurator type is description logic, where formalisms are used for representing and reasoning with knowledge. The constraint-based configurator type uses properties and connection ports to define components and determines how components can be combined.

Description-logic configurator type

The description logic configurator type is the most used logic-based configurator (Sabin & Wiegel, 1998; Blecker et al., 2004). Reasoning with and representation of knowledge is performed through formalisms referred to as description logics. Description logic is based on objects and classes with binary relations and is comparable to object-oriented programming languages (Donini et al., 1998).

The main advantage of the description logic configurator type compared to other configurator types is the ability to provide active completion of knowledge in the form of deductive closure and to detect inconsistencies (McGuinness & Wright, 1998; Sabin & Wiegel, 1998). Description logic configurators provides understanding of the consequences of changes and modifications of a complex system where inconsistencies can be found early on. A global consistency of a state of knowledge is maintained within the description logic configurator through consistency checking. The description logic configurator type is optimal to use for complex customisable products due to consistency checking. Knowledge acquisition and knowledge maintenance are simpler through the global maintained consistency and active consistency checking (Sabin & Wiegel, 1998).

The description logic configurator type does not optimise or use Decision Maker (DM) preferences. Limitations of description logics are found in the probability representation, the preference semantics, the completion support, optimisation and limited expressive power (McGuinness & Wright, 1998).

The description logic configurator type is suitable for customers with little prior product knowledge. The customer is guided through simple questions in the interface. Based on the customer input, the description logic searches for solutions (Sabin & Wiegel, 1998). The description logic configurator is used to allow users to modify their input and observe the consequences of those modifications and to explore the problem space (McGuinness & Wright, 1998).

The description logic configurator type is optimal for complex customisable products and customers with little prior product knowledge. The configurator type is not suitable when optimisation is required.

Resource-based configurator type

The resource-based configurator type characterises each technical entity by the amount of resources it uses, consumes and supplies. The configuration of a resource-based configurator is acceptable when a resources balance is realised (Juengst & Heinrich, 1998; Sabin & Wiegel, 1998).

The resource-based reasoning is performed through domain knowledge and control knowledge (Juengst & Heinrich, 1998). The domain knowledge contains the data of components' demand and supply of resources. The control knowledge affects the processing, optimisation and meta level control. The meta level control is a self-organising algorithm used for finetuning. Optimisation is related to the resources, for example resource costs or supply time can be used as criteria for optimisation.

Component relationships are represented through resource type's behaviour patterns. These behaviour patterns are specified with appropriate mathematical functions (Juengst & Heinrich, 1998). For example, behaviour patterns contain information how resources can be combined and how balances can be checked.

The knowledge is structured in two catalogues, namely the resource catalogue and the component catalogue (Juengst & Heinrich, 1998). The resource catalogue contains knowledge of resource-types and their behaviour. The component catalogue contains knowledge of the component types and their contribution to the resource balance. The knowledge to perform resource balancing is stored in objects which contain the representation of the two catalogues.

Knowledge acquisition and knowledge maintenance are supported in the resource-based configurator through using an organisational structure based on resources (Juengst & Heinrich, 1998; Sabin & Wiegel, 1998). The resource-based configurator type simplifies tasks like component type removal or addition, through the knowledge-base expressed in resources (Juengst & Heinrich, 1998). The resource balancing prevents inconsistencies caused by man-made mistakes. An imbalance will point out which resources are missing, since resources cannot disappear. The resource-based configurator is able to configure complex customisable products without predesigned components through the support of knowledge acquisition and use of resource balancing to prevent inconsistencies.

The resource-based configurator type can be used for different operations, from simple configuration checks to fully automated configuring from requirement specifications (Juengst & Heinrich, 1998). The resource-based configurator type is insufficient in areas where many incalculable decisions are necessary, such as motives to choose configurations based on design aesthetics (Juengst & Heinrich, 1998). The representation of configurations in resources requires customers to have prior expert product knowledge in order to use a resource-based configurator.

Constraint-based configurator type

The constraint-based configurator type defines components by sets of properties and sets of connection points (Sabin & Wiegel, 1998). The configuration must fulfil several functions, called function roles. For each function role, one or more components are identified as key components for providing that role.

The configuration task of implementing functions that are included in components is referred to as constraint-satisfaction problem. How components can be combined to form a configuration is restricted through constraints among components.

There are two different basic classes of strategy which can be used for a constraint satisfaction problem, namely systematic search strategies and repair strategies (Tsang, 1999). The systematic search strategy tries one possible connection or property or function role at a time and makes sure that no constraint is violated until all function roles are satisfied. If a chosen property, connection or function role is not suitable, the just tried option is removed and replaced with an alternative that has not been tried yet, which is called backtracking. The repair strategies try a complete set of options (function roles based on the customer input) initially random. If an option is not suitable it will be replaced with another alternative (repaired) until hopefully all solutions can be found.

Components and their ports are represented through variables and the constraints restrict the way components can be integrated (Sabin & Wiegel, 1998). The formulations of constraints and variables could make the problem significantly easier or more difficult to solve (Tsang, 1999). Knowledge about problem formulation is very important and can be a limiting factor. The constraint-based configurator type is difficult in maintenance and knowledge acquisition Expert knowledge is required to formulate constraints.

The goal of constraint-based reasoning is to build one or more configurations that satisfy the optimisation criteria and desired specifications (Sabin & Wiegel, 1998). The constraint-based configurator optimises configurations based on the function roles of components and their connections.

Customers should have average prior product knowledge to identify one or more components as key components in fulfilling their desired product function (Sabin & Wiegel, 1998). A configuration consists of a set of components and descriptions of the connections among them. The constraint-based configurator can be used for predesigned components, variables and function roles (Sabin & Wiegel, 1998).

2.2.2 Solution searching

The solution searching of a configurator is performed by either a technical elements configurator type or by a feature searching configurator type (Blecker et al., 2004). A configurator uses solution searching to structure the selection process of suitable configurator variants based on the customer wishes (Blecker et al., 2004). The technical element searching configurator type uses a bottom-up approach, in contrast to the feature searching element type that performs a top-down approach.

The solution searching (of a configurator) can be in a bottom-up or top-down approach, or a combination of both (Kott et al., 1992). In the bottom-up approach specific components are selected by the customer to perform their desired function, for which the compatibility of the components is checked (Mittal & Frayman, 1989). The performance of the overall product is not considered in the bottom-up approach. In *figure 5.A* an example is given of a bottom-up approach for the configuration of a bicycle. The customer selects specific components which together perform a function.

The top-down approach selects components based on the customers' requested functions (Mittal & Frayman, 1989). In *figure 5.B* an example is presented of the top-down approach, where customers' requirements are used to select the satisfying optimal components.

In the technical element searching configurator a configuration of available/compatible components is assembled by the customer. The technical element configurator type requires customers to have average to expert prior product knowledge, in order to select a configuration of components. The earlier presented example in *figure 4.B* shows a technical element searching configurator where an expert customer can configure a PC for their specific usage, through selecting the main components of an PC.

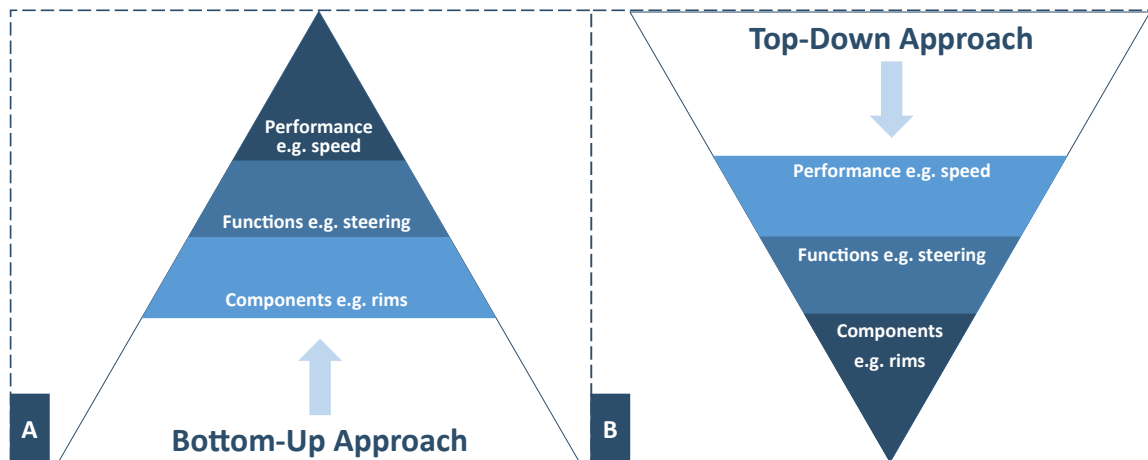


Figure 5: Bottom-up & top-down approach, *source by the author*

The technical element searching configurator type is used for customisable products with existing (predesigned) components which can be selected by the customer.

In the feature searching configurator, the customer gives requirements for which the configurator configures a configuration. The feature searching configurator type enables customers with no or little prior product knowledge to create a product based on their desired functionalities. The earlier presented example in *figure 4.A* shows a feature searching configurator where a customer selects their desired game (functionality) for which a PC is configured. The feature searching configurator type can be used to design/engineer (not predesigned) customisable products based on the customer requirements.

2.2.3 Configurator organisation

The organisation of a configurator is central or distributed (Felfering et al., 2001). The knowledge storage and the manufacturing are performed locally with a central configurator, and distributed for a distributed configurator.

A configurator is distributed when the configuration is manufactured in multiple companies which depend on each other's expertise to complete the production of the configuration. The different companies involved in the production process do not possess the configuration knowledge of each party to guaranty the security privacy of suppliers. The configuration knowledge is transferred through a facilitator agent that collects the (partial) configurations from each party and distributes them among others.

A central configurator is used in companies who manufacture the complete product and possess all the configuration knowledge to configure the product.

2.2.4 Life cycle support

A product can be reconfigured for improvement, to include new or better components, or to replace non-functioning parts. All are forms of product life cycle support. Product life cycle support can be categorized into three cases: a configurator without reconfigurator, a separate configurator and reconfigurator, and an integrated configurator and reconfigurator (Blecker et al., 2004).

A reconfigurator can be used in a company when it is needed to reconfigure produced configurations. For example, for an aircraft with replaced parts it can be required to reconfigure the aircraft with the new parts to simulate its performance.

The reconfiguration can be integrated in the existing configurator or separated, depending on the computing task and the complexity of the configurator and reconfigurator. For example, if the reconfiguration shares a lot of data/knowledge with the configurator, it is more logical to have an integrated reconfigurator. However, a separate reconfigurator is used when the computing load and the system complexity become too high when the configurator and reconfigurator are combined. The separated reconfigurator is suitable for the reconfiguration of complex customisable products.

An integrated configurator can be advantageous to avoid data redundancy. For example, when the product has to be reconfigured often during its lifespan, using an integrated reconfigurator avoids supporting two different systems (the configurator and reconfigurator) with many overlapping functionalities.

The integrated reconfigurator is suitable for the reconfiguration of simple or intermediate customisable products. The computing load with an integrated configurator will not be exceeded for simple and intermediate customisable products.

2.2.5 Business strategy

There are three different main strategies with different requirements on configurators, namely engineer-to-order, assemble-to-order and fabricate-to-order (Blecker et al., 2004). The type of configurator business strategy depends on the company's operation.

The engineer-to-order strategy is used for the engineering of products based on the customer requirements in an order. Designing prosthetics specific to the user is an example of a configurator with an engineer-to-order strategy. The engineer-to-order strategy is used for not predesigned or complex customisable products, which require to design a product to order. The engineer-to-order strategy is used for complex products or components which have to be engineered to order. An engineer-to-order configurator engineers a product based on customer requirements. The customer needs little to no prior product knowledge to state their desired product functionalities.

The fabricate-to-order strategy is used for adjustable products, for example in dimensions. The fabricate-to-order strategy is used for customisable products with intermediate complexity. They are adjusted by the customer to their desired specifications. The customer needs average prior product knowledge in order to understand the consequences of adjusting the customisable product.

The assembly-to-order strategy utilises the assembly of parts which are predesigned/manufactured but not yet assembled. The selection of existing components (such as PC components) by the customer to be assembled is an example of an assemble-to-order configurator. The assembly-to-order strategy is used for predesigned products and requires prior expert product knowledge to select the suitable components. The assembly-to-order strategy uses predesigned products from which the compatibility between components is known. The assembly-to-order products are simple compared to products of the fabricate-to-order and engineer-to-order strategies.

The configurator types 'engineer-to-order' and 'fabricate-to-order' have a higher complexity and require a higher configurator intelligence to compute the engineering, design or adjustments.

2.2.6 Scope of use

The scope of a configurator system can be categorised as single-purpose and general-purpose systems (Blecker et al., 2004).

A general configurator is used for companies with a general configurator process, e.g. the configuring of a customisable product with predesigned or adjustable components. The general configurator is typically used for customisable products with simple to intermediate complexity.

The single-purpose system is used when the configurator process is completely different from other companies and needs to be adjusted based on the (not predesigned) customisable product. For example, when a product is engineered to order, specific (not general) configurator knowledge and logic is used to perform a product design process based on the customisable product. In addition, drawing applications etc. may need to be integrated in the configurator to engineer a product to order. The single-purpose configurator is used for complex customisable products, who require an adjusted configuration process.

2.2.7 Interaction nature

The interaction nature of a configurator can be either online or offline (Blecker et al., 2004). Offline-interaction-nature-configurators have their knowledge for configuration stored offline and work independently from the network. The computing power of a customer's local unit is used for offline configuration. A local unit has more computing power and is suitable to support the configuring of complex products.

Online-interaction-nature-configurators communicate through the web, their knowledge is stored on a central web server or a local unit. The online-interaction-nature is classified into configurators with local data processing and configurators with central data processing (Blecker et al., 2004).

The online-interaction-nature-configurators with local data processing require a configuration application on a customer's local unit that computes the configuration (Blecker et al., 2004). The online-nature-configurator which uses a central unit has continuous communication between the supplier's central unit and the customer's local unit. The local unit is suitable for products with a simple or intermediate product complexity that do not have a high computing load to configure over the internet.

The advantage of an online configurator is the ability to update the configurator's knowledge. The use of a local data processing unit is recommended when the configuration load is high. The local unit can compute the solution without requiring a high-speed internet connection or a high computing power of the supporting server. The central data processing unit is used for complex products that require a high computing load to configure. The use of central data processing is recommended when the configurator content is often changing, all changes are immediately communicated to the customer's local unit.

2.2.8 Update execution

The update execution of the configurator can be performed through either a push or a pull mode, described Blecker et al. (2004). A push mode communicates updates from a supplier's central unit to a customer's local unit, where the local unit has to grant permission to perform the pushed update. The central unit of the supplier contains the configurator's logic. Through the pull mode the local unit retrieves updates if required.

The push mode can be useful if the configurator often needs updates. Then the configurator is updated when a new update is available. The pull update can be favoured when configurator content does not need to be synchronised and the configurator remains useable without content updates. The push mode is logical to select with an online interaction nature, which is selected when configurator content is synchronised. In contradiction, the pull mode is more suitable in combination with an offline interaction nature, when synchronisation updates are not required.

2.2.9 Integration level

The integration level of configurators can be classified as stand-alone, data-integrative and application-integrative (Blecker et al., 2004).

The integration level of a configurator can be explained as the possibility of the configurator to communicate or work with other systems. Standalone configurators do not integrate with other information systems and do not dispose of interfaces to other systems. Data redundancy is avoided by using data-integrative configurators. For example, data integrative-configurators use a certain type of document file to communicate with each other. Application integrative configurators use a whole application to integrate, for example by using a CAD-system integration in the configurator.

The use of a data-integrative configurator has advantage over a standalone configurator when the configurator output is further used in other systems, for example in a separate reconfigurator. To avoid redundant data, the interfaces with other systems are created to support the interchangeability with other information systems.

In case the configurator has to communicate a lot with other information systems, it can be advantageous to implement a complete application in the configurator or vice versa. For example, a simulating application can be integrated to determine the performance of the configuration. An application-integrative system is recommendable for an engineer-to-order strategy where drawings etc. need to be created. The application-integrative system is used for customisable products with a high complexity that are not predesigned.

2.2.10 Complexity

Product configurators can be distinguished based on their design complexity, namely as primitive, interactive or automatic configurators (Blecker et al., 2004). Primitive configurators record configurator decisions made by the customer without checking if it is a valid configuration. The interactive configurators validate the decisions made during configuring and guide the user in making the required decisions. Automatic configurators support the automatic generation of parts or even entire configurations in addition to the functionalities of interactive configurators.

A primitive configurator can be used for applications where the configurator data does not need to be checked (Blecker et al., 2004). For example, during the configuration of a personal postcard the configurator does not need to check the placement of images or text, it simply has to give the options to customise the designs. A customer should have prior expert product knowledge to use a primitive configurator, if the performance of the configured product depends on the selection of compatible components. Since there is no validation or checking in a primitive configurator, the application should either be simple or have expert customers.

An example of the use of an interactive configurator is for configuring a PC: the configurator can guide the user through the needed main components and check if parts fit to each other. The interactive configurator is useful to guide users if customers have average prior product knowledge.

Automatic configurators are required for customers which have no prior product knowledge, for example a configurator which ranks the PC's on the customers preference, such as a gaming pc. The automatic configurator is more detailed and needs more configurator knowledge to automatically derive solutions.

2.3 Multi criteria decision making

Multi criteria decision making (MCDM) techniques are useful tools to help decision maker(s) to select options in the case of discrete problems Roszkowska (2011). During MCDM, multiple criteria are often conflicting (Gavade, 2014; Zavadskas et al., 2014). Multi-criteria decision making is reviewed to select MCDM methods to determine the optimal configurator types for conflicting characteristics. A configurator type is conflicting when it has both advised and not recommended characteristics. The basics of MCDM will be reviewed first, to understand the decision-making process and the different functions the MCDM methods have.

MCDM is a process that enables to make decisions with multiple, often conflicting criteria (Gavade, 2014). The problems of MCDM can be classified into two categories (Gavade, 2014):

- **Multi attribute decision making (MADM):** Through MADM the “most suitable” alternative is selected from pre-specified alternatives described in terms of multiple attributes.
- **Multiple objective decision making (MODM):** Through MODM alternatives are designed which optimise the multiple objectives of the Decision Maker (DM).

This thesis will focus on the MADM methods, since the configurator types already exist, and they only need to be selected based on their characteristics. The following popular MCDM methods are investigated to be used in the proposed method to select configurator types:

- **Analytical Hierarchy Process (AHP)**
- **Technique of Order Preference by Similarity of Ideal Solution (TOPSIS)**
- **ViseKriterijumska Optimizacija Kompromisno Resenje (VIKOR)**
- **Elimination Et Choice Translating Reality (ELECTRE)**
- **Preference Ranking Organization METHods for Enrichment Evaluation (PROMETHEE)**

Multi criteria problems can be represented in a decision matrix (or matrices) (Roszkowska, 2011). Each decision matrix has four main parts, namely: (a) alternatives, (b) attributes, (c) weight or relative importance of each attribute, and (d) measures of performance of alternatives with respect to the attributes. An example of a multi criteria decision making matrix is visible in *equation 1*, the weights are included in the example matrix. Multi criteria decision making matrices can be solved through using one or several MCDM methods.

$$\begin{array}{ccccc}
 & C_1 & C_2 & \dots & C_n \\
 A_1 & \left[\begin{array}{cccc} x_{11}^k & x_{12}^k & \dots & x_{1n}^k \end{array} \right. \\
 A_2 & \left[\begin{array}{cccc} x_{21}^k & x_{22}^k & \dots & x_{2n}^k \end{array} \right. \\
 \dots & \left[\begin{array}{cccc} \dots & \dots & \dots & \dots \end{array} \right. \\
 A_m & \left[\begin{array}{cccc} x_{m1}^k & x_{m2}^k & \dots & x_{mn}^k \end{array} \right.
 \end{array} \quad (1)$$

Equation 1: MCDM matrix, *source by the author*

where:

$A_1, A_2, \dots, A_m$ are possible alternatives from which one is selected,

$C_1, C_2, \dots, C_n$ are the criteria for which the alternative performance is measured,

x_{ij}^k is the k - decision maker rating for alternative A_i with respect to the criterion C_j .

The relative importance of each criterion is given by a set of weights which together sum up to one. Let us denote by $W^k = [w_1^k, w_2^k, \dots, w_n^k]$ a weight vector for decision maker k -, where $w_1^k + w_2^k + \dots + w_n^k = 1$.

Multi-criteria analysis focuses mainly on three types of decision problems: *choice* selects the most appropriate alternative, *ranking* draws a complete order of the alternatives from the best to the worst, and *sorting* selects the best k alternatives from the list Roszkowska (2011). In the following paragraphs the five MCDM methods are analysed on their operation, strengths and weaknesses.

2.3.1 AHP

The presented matrix in *equation 1*, shows criteria for $C_1, C_2, \dots, C_n$ for alternatives $A_1, A_2, \dots, A_m$, in order to solve the MCDM matrix the importance of each criteria is to be determined. The Analytic Hierarchy Process (AHP) method is used to derive ratio scales from paired comparisons and to determine the relative importance of criteria (Saaty, 2008).

The AHP relies on the judgment of experts to derive priority scales, for which the fundamental scale of actual numbers is used, presented in *figure 6*. The scales are used in the AHP method to give weight and prioritize the different attributes/criteria.

<i>Intensity of Importance</i>	<i>Definition</i>	<i>Explanation</i>
1	Equal Importance	Two activities contribute equally to the objective
2	Weak or slight	
3	Moderate importance	Experience and judgement slightly favour one activity over another
4	Moderate plus	
5	Strong importance	Experience and judgement strongly favour one activity over another
6	Strong plus	
7	Very strong or demonstrated importance	An activity is favoured very strongly over another; its dominance demonstrated in practice
8	Very, very strong	
9	Extreme importance	The evidence favouring one activity over another is of the highest possible order of affirmation
Reciprocals of above	If activity i has one of the above non-zero numbers assigned to it when compared with activity j , then j has the reciprocal value when compared with i	A reasonable assumption
1.1–1.9	If the activities are very close	May be difficult to assign the best value but when compared with other contrasting activities the size of the small numbers would not be too noticeable, yet they can still indicate the relative importance of the activities.

Figure 6: AHP: the fundamental scale of absolute numbers, *source Saaty (2008)*

The AHP decomposes a decision-making problem into a system of hierarchies, namely of objectives, attributes (or criteria), and alternatives (Gavade, 2014). Tangible (i.e. objective) and non-tangible (i.e. subjective) attributes can be handled with the AHP method, in particular when subjective judgments of different individuals constitute an important part of the decision process. In order to determine if the AHP method is suitable to select the optimal configurator types for conflicting criteria, the strengths and weaknesses are reviewed based on the work of Gavade (2014).

Strengths

- ☐ The AHP method is more flexible and has a more intuitive appeal to the decision makers compared to other multi criteria methods.
- ☐ The AHP method has the ability to check for inconsistencies.
- ☐ The pairwise comparison form of data input is considered straightforward and convenient.
- ☐ Group decision-making is supported through calculating the geometric mean of the individual pairwise comparisons in the AHP method.

Weaknesses

- The decision problem is decomposed into a number of subsystems. For each system, a substantial number of pairwise comparisons are needed. The pairwise comparisons can become a lengthy task if there are a lot of subsystems.
- The used 9-point scale importance rating is limited and can result in wrong comparison ratios due to the limitation in expression of the relative importance. The 9-point scale importance rating can be made less constricted through using fuzzy numbers for the weighting instead of crisp numbers (precise values), but the scale remains limited.
- It is difficult to assign the 9-point scale of importance and should be executed by experts.

2.3.2 TOPSIS

The TOPSIS method is originally developed by Ching-Lai Hwang and Yoon in 1981. The method finds the solution which is closest to the hypothetically best solution, and farthest from the hypothetically worst solution (Gavade, 2014). The shortest Euclidean distance from the ideal solution and the farthest from the negative ideal solution are selected for the chosen alternative in the TOPSIS method. The (hypothetically) ideal solution corresponds to the maximum values of the attributes of the satisfying solution, the negative ideal solution corresponds to the minimum attributes values in the database.

TOPSIS is a compensatory method which allows trade-offs between criteria, where a poor result in one criterion can be negated by a good result in another criterion (Greene et al., 2011). This provides a more realistic form of modelling than non-compensatory methods.

Criteria of attributes can have different units, for example a lamp can have a criterion about the wattage and a criterion concerning light emitted in lumen. To make it possible to consider all criteria for an alternative, the criteria should be made comparable to another. For example: what is the importance of 40 watt compared to 1000 lumen? To eliminate the units of the criterion functions, vector normalisation is performed in TOPSIS (Opricovic & Tzeng, 2004). In order to determine the relative importance of criteria, weights are needed as input to perform the TOPSIS method.

In order to determine if the TOPSIS method is suitable to select the optimal configurator types for conflicting criteria, the strengths and weaknesses are reviewed according to the work of Gavade (2014) and Roszkowska (2011).

Strengths

- The method is simple and intuitive, based on the consideration of the distance from the ideal solution.
- The input can be any number of criteria and attributes.
- Ease of computation and good computational efficiency.
- It can be visualised.
- The method is intuitive based on the consideration of the distance from the ideal solution

Weaknesses

- Can give unreliable results easily.
- The standard form in TOPSIS is deterministic and does not consider uncertainty in the weightings.

2.3.3 VIKOR

The VIKOR method is similar to the TOPSIS method and is also based on an aggregating function representation "closeness to ideal" (Opricovic & Tzeng, 2004). The VIKOR method was developed to solve MCDM problems with conflicting criteria with different units and is focused on compromising as conflict resolution.

The VIKOR method performs ranking and selecting for a set of alternatives in the presence of conflicting criteria and proposes one or more compromise solutions (Opricovic & Tzeng, 2007). The compromise solution is determined for a maximum "group utility" for "the "majority" and a minimum for an individual regret for the "opponent" (Opricovic & Tzeng, 2004). Similar to the TOPSIS method, the VIKOR method needs weights for criteria to be able to compare them based on their relative importance.

The list of strengths and weaknesses of the VIKOR method is based on the work of Gavade (2014) and Opricovic & Tzeng (2004).

Strengths

- The distinction and balance between maximizing the utility group and minimizing the regret group is performed for the best alternative.
- The VIKOR method calculates an advantage rate and proposes a compromise solution.

Weaknesses

- The rating is quantified as crisp values. Crisp data are in many cases inadequate to model real-life situations, the decision maker must consider imprecise data.

2.3.4 ELECTRE

Elimination et choice translating reality (ELECTRE) method can be applied for three main problems: choosing, ranking and sorting (Roszkowska, 2011; Greene et al., 2011; Gavade, 2014). The ELECTRE method uses thresholds for declaring indifference or preference between two alternatives on a particular criterion (Greene et al., 2011). The ELECTRE method is used for choosing best actions from a given set of actions. There are two main parts to an ELECTRIC application (Roszkowska, 2011; Gavade, 2014):

- The construction of one or several outranking relations, which aims at comparing each pair of actions in a comprehensive way.
- An exploitation procedure that elaborates on the recommendations obtained in the first phase. The recommendations depend on the problem being addressed (choosing, sorting or ranking).

The ELECTRE method also requires weights of relative importance for the criteria in order to compare the actions. The strengths and weaknesses of the ELECTRE method are listed according to Gavade (2014):

Strengths

- Can deal with qualitative and quantitative criteria.

Weaknesses

- It is difficult to understand, because the concordance and discordance matrices are used, which are in fact a quantification of the positive and negative arguments.
- The method requires many steps due to the calculation of the concordance and discordance index.

2.3.5 PROMETHEE

The PROMETHEE method is used for outranking, the method utilizes a function reflecting the degree of advantage of one alternative over another, along with the degree of disadvantage (Olson, 2001; Macharis et al., 2004). The PROMETHEE method uses mutual comparison of each alternative pair with respect to the given criteria. Implementation of the PROMETHEE method requires two types of information according to Macharis et al. (2004):

- Information on the relative importance of the criteria (weighting)
- Information on the decision-makers preference function, which is used to compare the contribution of the alternative in terms of each separate criterion.

This information determines the preference structure of the Decision Maker (DM).

PROMETHEE I is designed to give a partial ranking, as explained by Olson (2001). Partial rankings focus on the best choice, not on a complete ranking. The PROMETHEE II method provides a complete ranking, from best to worst alternative.

The strengths and weaknesses of using the PROMETHEE method are listed according to Gavade (2014):

Strengths

- PROMETHEE accepts (as every outranking method) qualitative and quantitative criteria.

Weaknesses

- PROMETHEE suffers from the rank reversal problem, when introducing a new alternative. This means that other alternatives may change or a set of available alternatives changes.
- PROMETHEE does not really structure a decision problem. In case of many criteria and options it may become difficult in usage.

2.4 Literature conclusion

Concluding from the found literature, there are 10 configurator classifications which contain 29 different configurator types. The configurator types have different characteristics by which they can be selected. Companies have difficulty selecting an overall well-performing configurator due to the many possible configurator types with different characteristics.

Configurator types can have a bad performance for a characteristic such as knowledge maintenance, while others are optimal for knowledge maintenance characteristic as described in the literature review. Therefore, a configurator can have a bad performance when its configurator types characteristics are not selected in accordance with a specific customisable product. The selection of configurator types should be based on the resemblance of the characteristics of a specific customisable product.

2.4.1 Configurator type characteristics

From the literature review, different configurator type characteristics are distinct, for example the knowledge based reasoning configurator types differ in characteristics; optimisation, customer prior product knowledge, product complexity and component type. In addition, the interaction between the customer and the configurator is often used as characteristic to select the interaction nature and the required update execution configurator types. The characteristics of the reviewed configurator types are represented in *figure 7*.

The characteristics are briefly explained in order to understand how the configurator types are selected based on these characteristics. The configurator types' characteristics can be used to determine the configurator type's suitability for specific customisable products. The characteristics can be used to create selection criteria for the configurator types for characteristics of specific customisable products.

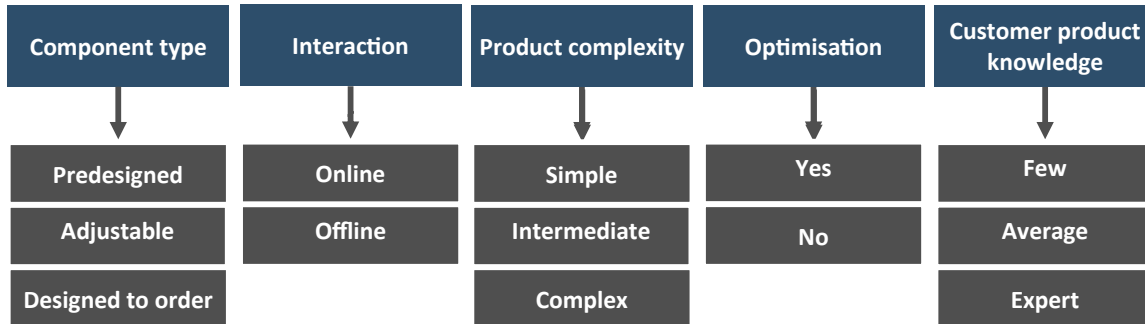


Figure 7: Configurator type characteristics, *source by the author*

Component type

A configurator type can be selected based on the component type characteristics. The used components can be predesigned, adjustable or designed to order. Predesigned means that product components are already fabricated and only have to be assembled to order. Adjustable products contain components that are already fabricated or designed and only need to be adjusted to order, for example changed in dimensions etc. Designed to order component types are not yet designed and have to be engineered based on the order of a customer.

Interaction

Configurator types can be selected based on the “interaction” characteristic, which represents the communication with the customer, both online or offline. An online interaction is used when a customer performs the configuring of the customisable product online. The offline interaction is used when the configuring is performed offline.

Product complexity

The reviewed literature often uses “product complexity” as a characteristic for which configurator types like the knowledge based reasoning types can be distinct. These described differences in suitability for different product complexities are directly used as characteristics. A simple product complexity represents products which do not have many components or are slightly adjustable and therefore requires little configurator knowledge and logic in order to be configured. Intermediate product complexity is used when products have many components which selection has to be checked to prevent that no non-compatible components are used. The intermediate product complexity requires more configurator knowledge and logic to be able to configure a product with many components and check their compatibility. A complex product complexity represents products that have to be engineered/designed by the configurator which does not use fabricated products. The complex products requires expert configurator knowledge and logic to be able to design components from customer requirements.

Optimisation

Not all configurator types are suitable to be used for optimisation, therefore optimisation is used as a characteristic to select configurator types based on their suitability for optimisation. Optimisation is used when a product has to be optimised for the decision maker-preferences/customer- requirements, for example durability or energy consumption etc.

Customer product knowledge

Configurator types can be selected based on their customer guidance or the required “customer prior product knowledge”. The found literature shows that configurator types can be used for different levels of customer prior product knowledge, for example the Element searching configurator type requires customers to have expert prior product knowledge in order to assemble a product based on specific components. A customer with no prior product knowledge requires configurator types which automatically create a configured product based on the customers desires/requirements. When a customer has average prior product knowledge, he possesses knowledge of the product and its component functions. Customers with average product knowledge require still some guidance to configure a complete product. Expert prior product knowledge, is possessed by customers who can configure the products without guidance through knowing the required materials and components to create a well performing complete product.

The ideal configurator types can be determined through selecting the configurator types that are most similar to the characteristic of a specific customisable product. The configurator types can be plotted based on their difference in similar characteristics. An example is presented in *figure 8*. As visible in the figure, the ideal configurator type can be selected based on the distance (or similarity) to the characteristics of a specific customisable product.

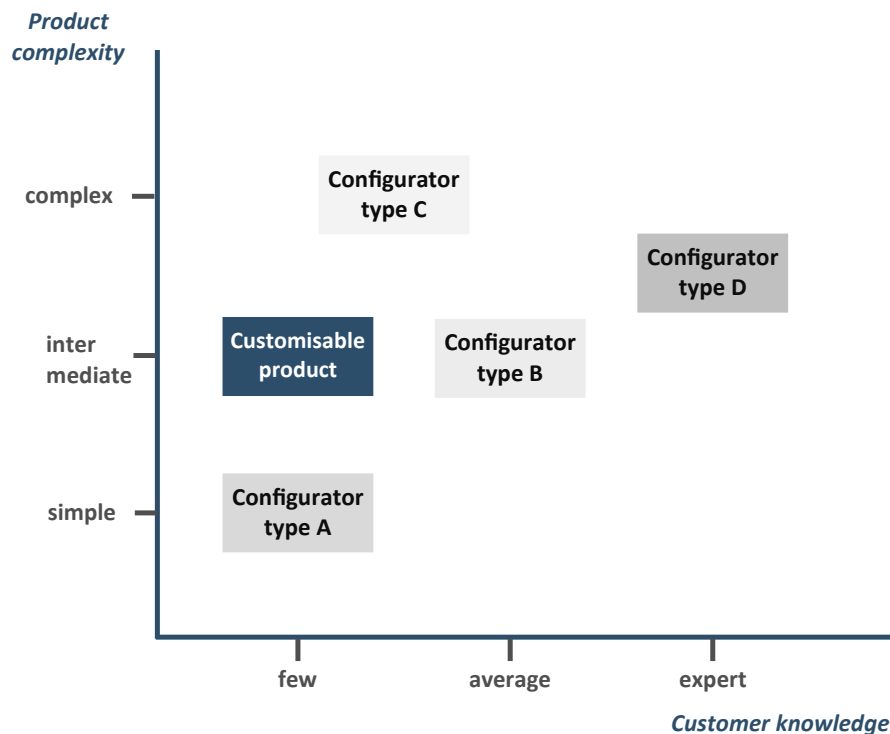


Figure 8: Example plot of configurator type characteristics, *source by the author*

In order to select the most ideal configurator types, their characteristics have to be determined to find the most corresponding type to the characteristics of a specific customisable product. Based on the found literature the characteristics of each configurator type is listed and their selection advice is presented in *table 1*. A plus means that the configurator type is recommended for this characteristic, a minus represents a negative selection advice for that characteristic, an empty block represents a neutral advice. There are several rows which are entirely empty, for these configurator types additional characteristics have to be determined in order to create selection criteria for the proposed method. The cells with shared characteristics between configurator types are marked with a colour to clarify for which characteristics multiple configurator types can be selected.

Classifications (10)	Configurator types (29)	1 Component type			2 Interaction		3 Product complexity			4 Optimisation		5 Customer prior product knowledge		
		Pre-designed	Adjustable	Designed to order	Online	Offline	Simple	Inter-mediate	Complex	Yes	No	Few	Average	Expert
Knowledge based reasoning	Rule based						+	-	-					
	Case based		+	-				+	-	+				
	Description logic			+				+	+	-	+	+		
	Resource based			+				+	+	+		-	-	+
	Constraint based	+	-	-						+		-		
Solution searching	Technical element	+		-								-		+
	Feature searching			+								+		
Organisation	Central													
	Distributed													
Life cycle support	No reconfigurator													
	Integrated configurator						+	+	-					
	Separate configurator						+	+	+					
Business strategy	Assembly to order strategy	+	-	-			+	-	-			-	-	+
	Fabricate to order strategy	-	+	-			+	+	-			-	+	
	Engineer to order strategy	-	-	+				+	+			+	+	
Scope of use	Single purpose	-	-	+			-	-	+					
	General purpose	+	+	-			+	+	-					
Interacting nature	Offline nature				-	+			+					
	Online local processing				+	-			+					
	Online central processing				+	-			-					
Update execution	Push mode				+	-								
	Pull mode				-	+								
Integrational level	Standalone													
	Data integrative													
	Application integrative	-	-	+					+					
Complexity	Primitive configurator											-	-	+
	Interactive configurator											-	+	
	Automatic configurator											+	+	

Table 1: Configurator types selection advice based on their characteristics, *source by the author*

2.4.2 Selecting strategy

In the configurator types selection advice is presented in *table 1* and shared characteristics are visible (colour marked cells). The component type characteristics advice of the application integrative, single purpose and engineer-to-order configurator type are similar. The three different configurator types could be selected based on the selection of one of the three configurator types.

The shared characteristics selection advice can be used to simplify the selection between 29 configurator types and select multiple types based on the selection of a few types. Companies can make fewer errors during the configurator type selection by using a confined/defined configurator type for the selection instead of ambiguous characteristics. Since, a company can have difficulty determining which ambiguous characteristic applies to their customisable product. For each configurator type, shared characteristics are to be determined and a selection sequence of configurator types should be determined in order to select all configurator types based on shared criteria.

By using confined configurator types as selection criteria input, companies only have to pick a few defined configurator types which apply to their customisable product. In addition, the use of configurator types for selection simplification makes it possible to determine multiple required characteristics without asking a company for their customisable product characteristics. For example, by determining the used business strategy (visible in *table 1*) the component type, product complexity and customer prior knowledge are also known. To guarantee companies select the suitable configurator type that are in correspondence with their customisable product, questions have to be developed to determine the required configurator types.

In the presented example configurator type plot in *figure 8* it is visible that the customisable product is close to configurator types A,B and C. To select the configurator type that is optimal for the customisable product the weights of the plotted characteristics have to be determined. For example, configurator type B has the same product complexity characteristic as the customisable product, however the configurator type A has the same customer knowledge characteristic. The choice between configurator type A or B can be performed by determining the importance ratio of the characteristics for which they are plotted. The reviewed AHP method can be used to determine the ratio of importance scales of characteristics in order to determine which configurator type is optimal.

The selection advice of configurator reasoning types can be conflicting, since they can be both optimal and not recommended for characteristics. For example, when optimisation is required for customers with little prior product knowledge, the description logic configurator type is optimal for the customers with little prior product knowledge, but is not recommended for product optimisation.

In order to select configurator types with conflicting selection advice, multi criteria decision making is investigated. Considering the five reviewed popular MCDM methods, all methods require criteria weighting in order to be able to perform multi criteria decision making. The AHP method can be used to perform weighting, therefore the AHP method will be used to determine the ratio of importance of the characteristics.

To perform multi criteria decision making for conflicting selection advice, one method from the remaining four MCDM methods should be selected. The remaining four MCDM methods are evaluated based on the specific requirements for the selection process of the proposed method, in order to select one to perform the MCDM in the proposed method. The selected MCDM method is used in the proposed method to perform the multi criteria decision making for the reasoning configurator types.

Summarised points of action required to develop the proposed method:

- Determined (shared) characteristics for all configurator types

- Develop a selection sequence of configurator types in order to make a selection between the 29 configurator types, based on a few configurator types.
- Develop questions that determine the few input configurator types used for selection of all 29 configurator types
- Determine the ratio of importance of the characteristics with the AHP method
- Select a MCDM method that performs the selection for configurator types with conflicting selection advice

3 Method development

In this chapter a development process is performed to create the proposed method to select the most suitable configurator types for a particular customisable product. The development process of the proposed method is performed by evaluating the presented information in the literature review.

The reviewed sales configurator presented in *paragraph 2.1* of the literature, uses customer requirements/input to find the most suitable product variant/configuration. The proposed method should be developed with similar functionality, where the input of a company's specific customisable product is used to determine a suitable configurator type.

In order to develop the proposed method which selects the suitable configurator types for a specific customisable product, the action points listed in the literature conclusion have to be completed and the results used to develop the proposed method.

First, to be able to create a selecting sequence or selection criteria, the (shared) characteristics for all configurator types have to be determined. In table 1 several configurator types have an empty row, which means they either have other characteristics or their characteristics still need to be determined based on the presented literature. Therefore, all configurator types are evaluated based on available literature to determine their shared criteria and determine a selecting sequence which is needed to select all configurator types based on a few selections, an example of the selecting sequence is presented in *figure 9*.

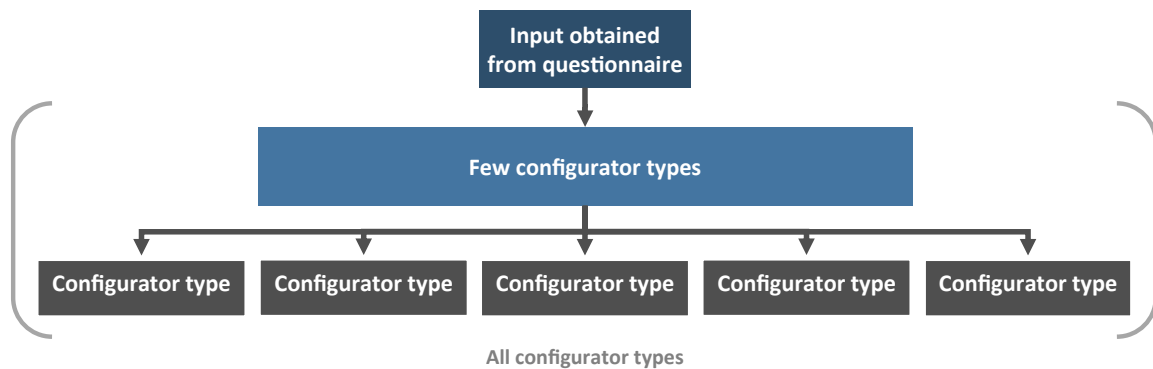


Figure 9: Selecting sequence

In order to obtain the few configurator types that are used for selecting all configurator types with non-conflicting selection advice, input to select the few configurator types in accordance with the characteristics of the customisable product has to be obtained. To obtain this input from a company for their specific customisable product, questions have to be developed whose answers determine which configurator type is required.

Two of the reviewed multi criteria decision making methods are used to create a selection process of the configurator types with conflicting selecting advice. The AHP MCDM method is used to determine the ratio of importance of the characteristics in order to be able to select the optimal configurator type based on multiple different characteristics. The ratio of importance is applied to select the ideal configurator type based on different characteristics, as explained in the literature review conclusion and visualised in *figure 8*. The other four MCDM methods are evaluated, based on the literature, to determine which method is most suitable to perform the multi criteria selection of configurator types with conflicting selection advice.

The development of the proposed method is performed in three steps as described in the following paragraphs:

- 3.1 *Determination of (shared) characteristics for all configurator types with non-conflicting selection advice and formulation of questions to obtain a company's input of their customisable product.*
- 3.2 *Determination of (shared) characteristics and formulation of questions to obtain a company's input of their customisable product for the configurator types with conflicting advice.*
- 3.3 *Development of multi criteria decision making process.*

3.1 Configurator type selection process for non-conflicting advice

Configurator types with non-conflicting advice can simply be selected based on (shared) characteristics of a few configurator types. Since, their selection advice is not in conflict, which means that there is no trade-off needed to determine which characteristic is more important for selection, a trade-off situation of characteristics is presented in *figure 8*.

The configurator types and their shared characteristics as presented in *table 1* of the literature conclusion, can be used for to determine a selection sequence based on a few configurator types. The figure shows that most shared characteristics are represented by the configurator types of the business strategy and interaction nature classification. To clarify this hypothesis, the (shared) characteristics of the online interaction nature and business strategy classification are coloured green and blue in *table 2*.

From *table 2* can be concluded that most of the configurator types can be selected based on the selection of the configurator types of the online interacting nature and business strategy classification. Through using configurator types as selection criteria companies may make less errors in a selection process with ambiguous characteristics. In addition, less input is required from companies, since the business strategy and interaction nature configurator types contain multiple characteristics. Therefore, it is no longer needed to find input to determine which of them applies to specific customisable product for each of these characteristics .

Another advantage of using configurator types for the selection of other configurator types is to no longer require multi criteria decision making. If relations can be found through determining which configurator types have shared characteristics a selection can be based on a co-operative relation. However not all configurator types can be selected based on shared characteristics. The knowledge based reasoning configurator types (marked in red) cannot be selected through the business strategy. Since, the reasoning configurator types have multiple characteristics, which selection advice can conflict with the set of characteristics of the business strategy configurator types. For example, see the red rimmed line around the set of characteristics selection advice of the resource based configurator type. The resource based configurator type has characteristics which are partly represented in the engineer-to-order and the assembly-to-order strategies. Therefore the business strategy cannot be used for the selection of the reasoning configurator types. The multi criteria decision making selection process is (only) needed for the reasoning configurator types. The selection process of the reasoning configurator types with conflicting selection advice is developed in *paragraph 3.2*.

Yet, not all configurator types are expressed in characteristics, for example the organisation classification has an empty row. All available literature of the configurator types are evaluated to determine their (shared) characteristics and confirm the hypothesis; that all (except the reasoning) configurator types can be selected based on the selection of the business strategy and interaction nature configurator type.

Classifications (10)	Configurator types (29)	1 Component type			2 Interaction		3 Product complexity			4 Optimisation		5 Customer prior product knowledge		
		Pre-designed	Adjustable	Designed to order	Online	Offline	Simple	Inter-mediate	Complex	Yes	No	Few	Average	Expert
Knowledge based reasoning	Rule based						+	-	-					
	Case based		+	-				+	-	+				
	Description logic			+				+	+	-	+	+		
	Resource based			+				+	+	+		-	-	+
	Constraint based	+	-	-						+		-		
Solution searching	Technical element	+		-								-		+
	Feature searching			+								+		
Organisation	Central													
	Distributed													
Life cycle support	No reconfigurator													
	Integrated configurator						+	+	-					
	Separate configurator						+	+	+					
Business strategy	Assembly to order strategy	+	-	-			+	-	-			-	-	+
	Fabricate to order strategy	-	+	-			+	+	-			-	+	
	Engineer to order strategy	-	-	+				+	+			+	+	
Scope of use	Single purpose	-	-	+			-	-	+					
	General purpose	+	+	-			+	+	-					
Interacting nature	Offline nature				-	+			+					
	Online local processing				+	-			+					
	Online central processing				+	-			-					
Update execution	Push mode				+	-								
	Pull mode				-	+								
Integrational level	Standalone													
	Data integrative													
	Application integrative	-	-	+					+					
Complexity	Primitive configurator											-	-	+
	Interactive configurator											-	+	
	Automatic configurator											+	+	

Table 2: Selection based on a few configurator types, *source by the author*

3.1.1 Determination of (shared) characteristics and questionnaire

The interaction nature and business strategies are used as selection criteria for other configurator types. The functioning of the business strategy and the interaction nature configurator types are specified to understand how they can be used as selection criteria for other configurator types. Questions are developed to obtain the characteristics of a specific customisable product that determine the selection of the business strategy and interaction nature configurator types.

Interaction nature selection criteria

The offline or online interaction nature is selected based on the need to synchronise configurator content or in other words perform knowledge maintenance. The offline interaction nature is used in configurators that do not require an internet connection to function and do not need to synchronise their configurator content. Configurators with an offline interaction nature are stored on the customers' local unit, or e.g. on a hard drive or CD. Knowledge maintenance or synchronisation of configurator content is not required between versions of a configurator with an offline interaction nature.

Content updates of an offline interaction nature configurator can be performed, e.g. through rebuilding the configurator which is released as a new version. Customers can obtain the rebuilt configurator, e.g. through installing a new version of the configurator. Older versions of the configurator are often still operative. Example given, a CAD program such as SolidWorks can be used offline, even though new versions are released every year, the old versions stay operational. An offline interaction nature can be selected when there is no need to directly synchronise configurator content.

According to the literature review, the online interaction nature is used when knowledge maintenance is often performed. The reason to put such configurator online is to be able to synchronise its content without requiring customers to update the configurator software. These updates have to be performed directly, e.g. to prevent configurations with outdated component prices or relations etc. The selection of the online interaction nature can be based on the need to directly synchronise the configurator content.

Question 1 (c_1) is posed to obtain the interaction nature type, required for a specific customisable product of a company.

c_1 Is it necessary to directly synchronise changes of configurator content?

a Yes, direct synchronisation is required for configuration.

An online nature is recommended, when components, prices, or products are often changed and need to be directly updated in the configurator.

b No, changes of configurator content do often occur or do not need to be directly synchronised.

An offline interaction nature is recommended when it is not required to directly update a configurator when configurator content changes.

Business strategy selection criteria

Most of the configurator types can be selected based on selection of the business strategy. The business strategy can be used to express the complexity of the product and configurator, the data storage quantity and the production process. There are three business strategies as explained in the literature review, namely: assembly-to-order, fabricate-to-order and engineer-to-order.

The assembly-to-order strategy can be selected for simple predesigned products. The assembly-to-order strategy is used for a low configurator complexity, with little configurator data. The configurator data only includes information of the compatibility of predesigned components. Customers need to have in-depth prior product knowledge to pick the configuration which fits their desires for an assembly-to-order strategy.

In contrast, the engineer and fabricate-to-order strategy can be used as selection criteria of reasoning configurator types which are able to configure intermediate or complex products. The fabricate-to-order strategy is used for intermediate products of which dimensions or features can be adjusted/fabricated to order. The fabricate-to-order strategy uses adjustable components, which are adjusted by customers based on their desires. Here, a customer needs average prior product knowledge to determine the required dimensions or specifications of the configuration. The fabricate-to-order strategy requires more data than the assembly-to-order strategy to store the knowledge and strategy to adjust components to order. However, fabricate-to-order requires less data storage than the engineer-to-order strategy, which requires more data to store the knowledge and strategy to automatically design a product.

The engineer-to-order strategy is used for complex configurators and complex customisable products. The products/components are automatically designed/engineered to order, the engineer-to-order strategy does not use predesigned products. There is a lot of data stored to contain the knowledge and strategy to automatically design components. To use the engineer-to-order strategy, customers need less prior product knowledge, since the strategy is to automatically configure the product with customer desires as input.

Question 2 (c_2) is made to determine the type of components of which the customisable product is configured, which in turn determines the business strategy that should be used to support the product configuration.

c_2 Which type of components are used?

a Predesigned components

The assembly-to-order strategy can be used for assembling predesigned components to customers' wishes.

b Adjustable components

The fabricate-to-order strategy is used in configurators where customers can adjust components, for example in dimensions or shape.

c Components are engineered to order

The engineer-to-order strategy is used for configurators that design/engineer products to the customer requirements. For example, design of a prosthesis for a missing limb.

The selection of the configurator types of each (except the reasoning) classification will be further discussed in paragraph 3.1.2 to 3.1.10 to determine if they can be selected based on the interaction nature and business strategy configurator types.

3.1.2 Solution searching type selection

According to *table 1* the two configurator types of solution searching classification are selected based on the customer's prior product knowledge and the complexity of a customisable product.

The technical element searching type can be used when selecting the assembly-to-order strategy. The technical-element searching type is used for predesigned products and customers with an average to expert prior product knowledge, as stated in the literature review. The assembly to order strategy possesses the same characteristics, namely customers with expert prior product knowledge and predesigned components.

The feature-based searching type can be used for customers with no prior product knowledge and products that are adjustable or designed to order. The engineer- or fabricate-to-order business strategy can be used as selection criterion to select the feature-based searching type, since they are also used for components that are designed to order and for customers with average or no prior product knowledge.

The feature-based and technical element searching type can be selected for the selection criteria presented in *table 3*.

	c ₂ Business strategy		
	Assembly to order	Fabricate to order	Engineer to order
Technical element	+	-	-
Feature-based	-	+	+

Table 3: Feature-based and technical element type selection criteria, *source by the author*

3.1.3 Configurator organisation type selection

As mentioned in the literature review, an organisation of a configurator can be either central or distributed. The configurator organisation type determines if configurator knowledge needs to be exchanged with other parties and if the manufacturing is performed in different companies. When the product manufacturing is performed locally (within one company), and the knowledge to configure/manufacture the product is stored in the same company, the configurator is organised centrally. Otherwise, if the product is partly manufactured in different companies with their own knowledge of manufacturing, the configurator organisation is called distributed.

The selection of the central organisation type can be based on the local storage of configuration knowledge and local manufacturing of the product. When a product consists of standard components which are produced elsewhere but do not have to be altered (manufacturing wise) in the configurator, the configurator is still called central. The standard products are then seen as components which are processed within the company.

The assembly-to-order strategy uses predesigned products that only need to be assembled. The configurator configuring knowledge for the assembly-to-order strategy is stored centrally in one company, since the components and products are already produced to be assembled. If a company chooses the assembly-to-order strategy, the configurator organisation will be central. However, the fabricate- and engineer-to-order strategy can also be used with a central organisation. Therefore, the assembly-to-order cannot be used as single selection criterion.

The selection criteria for the organisation types cannot really be more simplified than the selection based on the manufacturing knowledge distribution. The manufacturing of the product can be distributed due to differences in expertise of manufacturing companies. For example, if an architecture company wants to offer customisation of houses, the simulating and construction calculations could be outsourced to other companies who possess this expertise. The configurator knowledge is distributed, as another company has to verify and process the construction of the houses, while the other presents the styling.

The distributed organisation is only needed when the manufacturing processes can be different for different configurations. If the production process does not change for different configurations, then the outsourced manufacturing does not require the information on the specifications of the configured product. Since the outsourced components do not change for different configurations, they can simply be bought without the need to exchange configuration knowledge. Therefore, the distributed organisation of a configurator is only needed when configuration knowledge is exchanged and the manufacturing is performed by multiple companies.

The central organisation of a configurator is used when:

- Manufacturing knowledge and configuration knowledge is kept in one company.

The distributed organisation of a company is used when:

- The configuration knowledge is exchanged and the manufacturing/configuring is performed in multiple companies.

An additional question is created to obtain the input of a company's customisable product to determine the organisation type.

The goal of question 3 (c_3) is to determine the storage for the configuration knowledge and to select the fitting configurator organisation type.

c_3 Does the configurator knowledge need to be exchanged with other companies?

a Yes

Several parties are involved in the manufacturing of the product and each party contributes to the configuration process.

b No

One party performs the configuration process and possess all configuring knowledge.

The configurator organisation type can be selected for the selection criteria presented in *table 4*.

	c_3 Does the configurator knowledge need to be exchanged with other companies?	
	Yes	No
Central	-	+
Distributed	+	-

Table 4: Configurator organisation type selection criteria, *source by the author*

3.1.4 Life cycle support type selection

The life cycle support represents the use of a configurator during a product-life-cycle. Which life cycle support is required depends on the need for a reconfigurator, for example to obtain information or to simulate product performance with new components. There are three options possible, no reconfigurator, a separate reconfigurator and an integrated reconfigurator.

Most companies have no use for reconfiguring, for example when a gift basket is configured with food content. A reconfigurator is only needed when a company wants to simulate adjusted products or retrieve product information of the configured products. For example, when parts are replaced or adjusted in aircraft, a reconfiguration is made to simulate its performance. New or replaced components can have a different performance, which is in such case important to take into account.

When a company needs a reconfigurator, a separate or integrated reconfigurator can be used depending on the reconfiguration task. An integrated reconfigurator is used to avoid data redundancy. However, the integrated reconfigurator adds to the configurator complexity and can result in a high computing load when both configuration and reconfiguration functionalities are supported.

An integrated configurator is recommended to use for the assembly-to-order or the fabricate-to-order business strategy, since the predesigned or adjustable products require less computing power.

The computing load is too high to support non-predesigned products with an engineer-to-order or fabricate-to-order strategy.

The separate reconfigurator can be used to reduce the computing power. The reconfiguration of customisable products that are engineered or fabricated to order is recommended to use with a separate reconfigurator. The choice between the separate and integrated reconfigurator is related to the complexity of the configured product and can be selected based on the business strategy. A separate configurator can be selected for the engineer-to-order strategy, when a reconfigurator is requested.

An additional question is required to obtain a company's need to reconfigure the customisable product. Question 4 (c_4) determines the life cycle support type, through asking if reconfiguration is needed.

c_4 Is product reconfiguration needed?

a No

The configured product does not need to be reconfigured, for example to simulate its new functioning.

b Yes

Reconfiguration is needed to obtain information of the configured product in a later stadium of the product life cycle.

The life cycle support type can be selected for the selection criteria presented in *table 5*.

	c_2 Business strategy			c_4 Is product reconfiguration needed?	
	Assembly to order	Fabricate to order	Engineer to order	Yes	No
No reconfigurator				-	+
Integrated reconfigurator	+	+	-	+	-
Separate reconfigurator	-	-	+	+	-

Table 5: Life cycle support type selection criteria, *source by the author*

3.1.5 Business strategy type selection

As explained in *paragraph 3.1.2*, the business strategy can be selected based on the component types that are used for the customisable product. See *paragraph 3.1.2* for the selection of the business strategy.

3.1.6 Scope of use type selection

The scope of use of a configurator, according to the literature review, depends on the product complexity and the company's operation. A configurator can have a specific or general use. When a company is comparable with other companies which have a configurator, a general configurator can be used. In most cases a general-purpose configurator can be used, which is usable for different companies for diverse product types.

The single-purpose operation is selected for the selection of an engineer-to-order strategy. The engineer-to-order strategy designs products automatically to order. The process is very specific and a design is made specifically to the customer requirements.

For example, when a prosthesis is engineered to order, the engineering process to create a prosthesis is specific and cannot be used for the engineer-to-order process of another product. Therefore, the engineer-to-order strategy, can be used to represent the single-purpose configurator type.

The general-purpose configurators can be selected through the assembly-to-order and the fabricate-to-order strategy. The assembly of components can be performed with a general-purpose configurator, since the assembly process is the same for different products. Only, the configurator knowledge or strategy is different than for other products. However, the presentation of the assembled product can be similar for the assemble-to-order strategy.

The fabricate-to-order strategy is used for adjustable products and can also be represented in a general-purpose configurator. A general-purpose configurator can provide adjusting functionalities, such as entering different dimensions etc. to support the fabricate-to-order strategy. The configuration process of adjustable products is not specific and can be used for different products. The general-purpose configurator type can be selected through the selection of the fabricate-to-order strategy.

The scope of use type can be selected for the selection criteria presented in *table 6*.

	c ₂ Business strategy		
	Assembly to order	Fabricate to order	Engineer to order
General-purpose	+	+	-
Single-purpose	-	-	+

Table 6: Selection criteria for scope of use type, *source by the author*

3.1.7 Interaction nature type selection

As mentioned in *paragraph 3.1.1* the interaction nature of companies can be selected based on the need to synchronise the configurator content. The online interaction nature can be executed with either central or local data processing. The selection of the online interaction nature type depends on the configuration computing load, as mentioned in the literature review.

The advantage of central data processing is to perform the configuration process on a central unit without requiring a customer to download software to perform the corresponding data processing. Central data processing requires a lot of computing power and causes the central server to be slow or to fail when complex products are configured. Therefore, the central data processing type should be used for predesigned or adjustable (simple) products. The selection of the assembly- or fabricate-to-order strategy can be used as selection criterion of the online interaction nature with central data processing.

An online configurator with local data processing is recommended for a high computing load. Since the data processing is performed on the local unit (such as a pc) of the customer which is faster than computing everything on a company's central unit.

The engineer-to-order strategy requires a lot of computing power to automatically create a product design based on customer requirements. Therefore, the selection of the engineer-to-order strategy can be used as selection criterion for the online interaction nature with local data processing.

The interaction nature type can be selected for the selection criteria presented in *table 7*.

	c ₁ Interaction nature		c ₂ Business strategy		
	Online	Offline	Assembly to order	Fabricate to order	Engineer to order
Offline interaction nature	-	+			
Online central data processing	+	-	+	+	-
Online local data processing	+	-			+

Table 7: Interaction nature type selection criteria, *source by the author*

3.1.8 Update execution type selection

The selection of an update execution type for a configurator depends on the need for synchronisation. Therefore, the update execution mode can be selected according to the selection of the interaction nature. As presented in the literature review, the push mode type can be selected when an online interaction nature is used, and continuous synchronisation is desired. A pull mode can be selected when an offline interaction mode is used, and updates are less urgent.

The update execution type can be selected for the selection criteria presented in *table 8*.

	c ₁ Interaction nature	
	Online	Offline
Pull mode update execution	-	+
Push mode update execution	+	-

Table 8: Update execution type selection criteria, *source by the author*

3.1.9 Integration level type selection

The needed communication of the configurator with another system determines which type of integration level is required. For example, when the configurator has to communicate the configuration knowledge to other systems through exporting files. As mentioned in the literature review, the integration level can be distinguished in three different types, namely: standalone, data-integrative and application-integrative.

The application integration type is used when a whole application has to be integrated in the configurator. An application can be required to perform design processes in the configuration process, for example for creating component drawings with a CAD program.

When the engineer-to-order strategy is selected, the configurator should be able to automatically make component drawings to create a product design. Therefore, the integration of a drawing application is recommended to use with the engineer-to-order strategy. The selection of the application integration type can be based on the selection of the engineer-to-order strategy.

The data integrative configurator type can be used to exchange configuration manufacturing information to another system. For example, a data integrative configurator can be used to export the configuration in certain file types, which can be used for the machinery.

The data integrative configurator is recommended for the fabricate-to-order strategy, where products are adjusted to order. The configuration information of adjustments have to be transferred to the production and exported as machinery files to fabricate the configured product. The selection of the fabricate-to-order strategy can be used to select the data integrative integration type.

The standalone configurators do not exchange information to any other information system. The standalone configurator is used when no communication or information exchange to other systems is needed.

The standalone configurator can be selected based on the selection of the assembly-to-order strategy. The configurator does not need to exchange information about the production of the parts, since the parts are predesigned. The assembly-to-order strategy and the standalone configuration only have to document the configuration of predesigned parts.

The integration level type can be selected for the selection criteria presented in *table 9*.

	c ₂ Business strategy		
	Assembly to order	Fabricate to order	Engineer to order
Standalone	+	-	-
Data integrative	-	+	-
Application integrative	-	-	+

Table 9: Integration level type selection criteria, *source by the author*

3.1.10 Complexity type selection

The complexity of a configurator can differ depending on the operation of the configuration process. For example, the configurator has a high complexity when the configuring is done automatically. According to the literature review, the configurator's complexity is distinguished in three different types, namely: primitive, interactive or automatic configurator. The complexity of configuration processes is often dependent on the customer's prior product knowledge, for example the complexity increases when a configurator has to guide the customer or perform the configuration automatically.

Primitive configurators do not check the validity of the configuration and require customers with expert prior product knowledge to configure a product without receiving feedback. The assembly-to-order strategy requires customers with expert prior product knowledge as well, since the customer has to be able to select components which are optimal for their desired operation. Therefore, the selection of the assembly-to-order-strategy can be used as selection criterion of the primitive complexity type.

The interactive complexity type is used when a configurator has to validate the configuration and guide the customer through the configuration process. The interactive complexity type requires customers to have average prior product knowledge to be able to adjust components and to configure the customisable product. The fabricate-to-order strategy requires the same level of customers' prior product knowledge and is also used to guide the customer through configuring adjustable components. Therefore, the selection of the fabricate-to-order strategy can be used as selection criterion of the interactive complexity type.

The automatic configurator selection criteria are similar to the engineer-to-order strategy criteria. The automatic generation and design of parts is performed with an automatic configurator, of which characteristics are shared with the engineer-to-order strategy. Similar to the engineer-to-order strategy, the automatic configurator requires no prior product knowledge, since the configurator performs the configuration automatically for customer requirements. The selection of the automatic configurator can be based on the selection of the engineer-to-order strategy, since their selection is based on shared characteristics.

The configurator complexity type can be selected for the selection criteria presented in *table 10*.

	c ₂ Business strategy		
	Assembly to order	Fabricate to order	Engineer to order
Primitive	+	-	-
Interactive	-	+	-
automatic	-	-	+

Table 10: Configurator complexity type selection criteria, *source by the author*

3.1.11 Selection process

The hypothesis turned out to be correct, most configurator types can be selected based on the selection of the business strategy and interaction nature types. Additional questions are made in order to select all configurator types with non-conflicting selection advice. All (except the reasoning) configurator types can be selected based on the selection the following four configurator types; interaction nature, business strategy, organisation and re-configurator, which is visible in *table 11*. The four configurator types are used as a selection process of all configurator types with non-conflicting selection advice, represented in *figure 10*. The selection process of the configurator types with non-conflicting selection advice can be performed for a specific customisable product based on the obtained input from the created questions, (represented in c_1 , c_2 , c_3 , c_4).

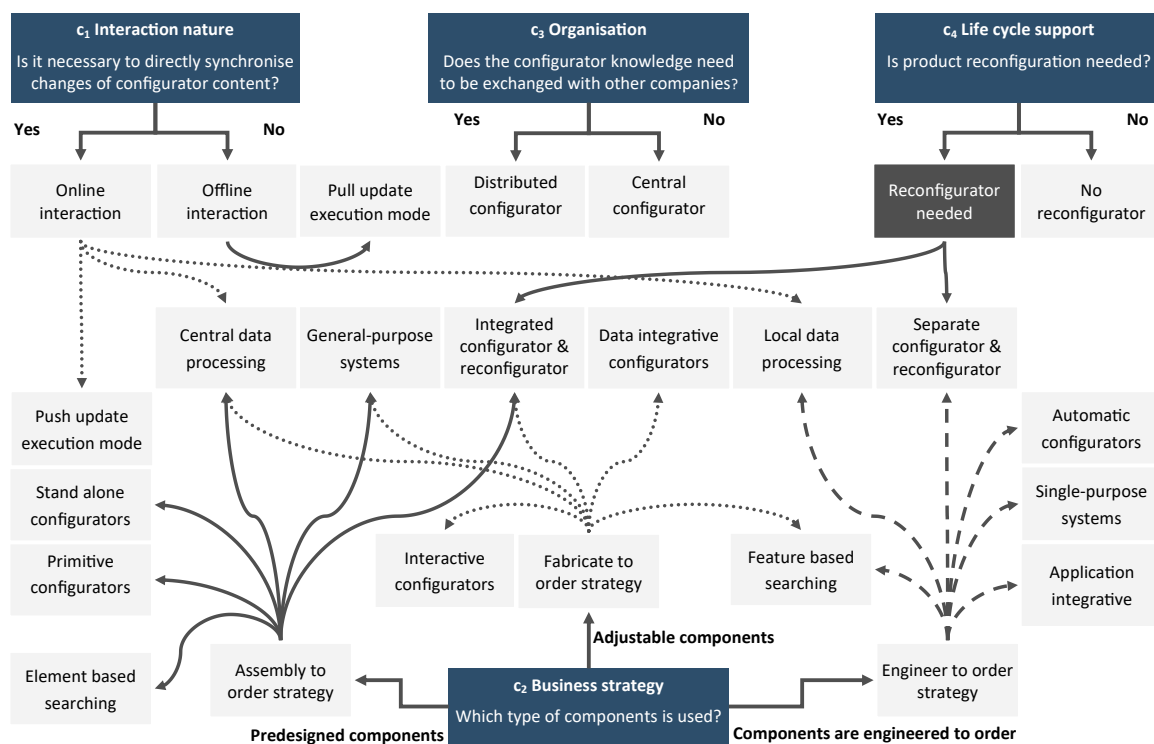


Figure 10: Selection of 24 configurator types for non-conflicting selection advice, *source by the author*

Selection criteria (c ₁ to c ₄)		c ₁ Interaction nature		c ₂ Business strategy			c ₃ Organisation		c ₄ Re-configurator	
Classification	Configurator type	C _{1a} Online nature	C _{1b} Offline nature	C _{2a} Assembly to order	C _{2b} Fabricate to order	C _{2c} Engineer to order	C _{3a} Distributed	C _{3b} Central	C _{4a} Yes	C _{4b} No
Solution searching	Technical element			+	-	-				
	Feature based			-	+	+				
Organisation	Central							+		
	Distributed						+			
Life cycle support	No reconfigurator									+
	Integrated reconfigurator			+	+	-			+	
	Separate reconfigurator			-	-	+			+	
Business strategy	Assembly to order			+						
	Fabricate to order				+					
	Engineer to order					+				
Scope of use	General-purpose			+	+	-				
	Single-purpose			-	-	+				
Interaction nature	Offline nature		+							
	Online central processing	+		+	+	-				
	Online local processing	+				+				
Update execution	Pull mode	-	+							
	Push mode	+	-							
Integrational level	Standalone			+	-	-				
	Data integrative			-	+	-				
	Application integrative			-	-	+				
Complexity	Primitive configurator			+	-	-				
	Interactive configurator				+	-				
	Automatic configurator			-	-	+				

Table 11: Selection criteria for configurator types, *source by the author*

3.2 Configurator type selection process for conflicting advice

Multi criteria decision making is required searching for trade-offs, to select optimal configurator types with conflicting selection advice. As explained in paragraph 3.1, the reasoning types share multiple characteristics which can lead to conflicting selection advice when their characteristics are both in accordance and discordance with the characteristics of the customisable product.

In *table 12* (part of *table 2*) is visible that the reasoning configurator types can be selected for the characteristics; component type, product complexity, optimisation and customer prior product knowledge. As in *paragraph 3.1* is explained the selection process of the configurator types can be simplified through using configurator types to prevent companies from making errors through determine which ambiguous characteristics are in accordance with their specific customisable product.

Classification	Configurator types	(5)	1 Component type			2 Interaction		3 Product complexity			4 Optimisation		5 Customer prior product knowledge		
			Pre-designed	Adjustable	Designed to order	Online	Offline	Simple	Inter-mediate	Complex	Yes	No	Few	Average	Expert
Knowledge based reasoning	Rule based							+	-	-					
	Case based		+	-					+	-	+				
	Description logic			+					+	+	-	+	+		
	Resource based			+					+	+	+		-	-	+
	Constraint based		+	-	-						+		-		

Table 12: Configurator type characteristics with conflicting selection advice, *source by the author*

The business strategy is used in *paragraph 3.1* to replace the characteristics; component type, product complexity and customer prior product knowledge. However, the customer prior product knowledge characteristic cannot be represented by the business strategy configurator types. Since two business strategy configurator types (presented in *table 2*) are both partly represented in the resource based configurator type, which is visible in the rimmed row of the resource based configurator type in *table 12*. Therefore, only the component type and product complexity characteristics can be represented by the business strategy configurator types and the selection based on the customer prior product knowledge characteristic needs to be expressed separately.

The optimisation characteristic is easy to determine for a company's customisable product and is not ambiguous. Therefore, the optimisation characteristic is directly obtained through developing a question which will serve as input for the selection process.

In addition, according to literature, the reasoning configurator types can be characterised for their performance of knowledge maintenance or in other words synchronisation. The need of knowledge maintenance can be determined by the interaction nature type. Therefore, the selection of reasoning configurator types is related to the interaction nature configurator and the possible selection based on the interaction nature is further evaluated.

In *paragraph 3.2.1* questions are developed to obtain input from a company which is used as selection criteria in the selection process of the reasoning configurator type. An MCDM method is selected in *paragraph 3.3.3* to determine the optimal reasoning types for the obtained selection criteria.

The reasoning types are further evaluated (in the paragraphs 3.2.2 to 3.2.6) based on the literature to support the hypotheses that the reasoning configurator types can be selected through the obtained business strategy and interaction nature configurator types and the optimisation and customer prior product knowledge characteristics. In addition, the evaluation is used to express the suitability of the reasoning types for the selection characteristics and configurator types, which is used to develop a decision matrix.

In *paragraph 3.2.7* a decision matrix is developed for the selection of the reasoning type, based on the evaluated reasoning type's suitability for the selection criteria. The decision matrix is used for multi criteria decision making and select the optimal reasoning type for a specific customisable product, which is further evaluated in *paragraph 3.3*.

3.2.1 Reasoning type selection criteria and questionnaire

In order to perform multi criteria decision making, the suitability of the reasoning types for selection criteria is expressed in different degrees during the evaluation, namely recommended, neutrally-recommended and not-recommended. The expressed degrees of suitability are used in the decision matrix to be able to perform multi criteria decision making and select the optimal reasoning configurator type.

The interaction nature and business strategy configurator types are already evaluated in *paragraph 3.1*. Therefore, only the characteristics "optimisation" and "customer prior product knowledge" are further evaluated, which are used to develop questions to obtain a company's input and understand how they can be used for the configurator reasoning type selection.

Product optimisation selection criteria

Product optimisation is used as selection criterion, since the ability to perform optimisation can be of importance for a customisable product. Literature review showed that not all reasoning configurator types can be used for product optimisation, for example the description-logic type is not suitable and the resource-based type is optimal for product optimisation. The selection of the reasoning configurator types can be based on the need to optimise a product. *Table 2* shows that the reasoning configurator types are the only configurator types which are influenced by the product optimisation selection criteria.

The goal of question 5 (c_5) is to determine if the customisable product requires product optimisation.

- c_5 Is the customisable product optimised based on customer requirements, e.g. on weight or price?
 - a Yes
The product has to be optimised to customer requirements.
 - b No
Optimisation is not necessary, the customer can choose a product configuration based on their own preferences.

The configurator types whose performance for product optimisation is not expressed in the literature, receive a neutral selection recommendation for optimisation. A neutral recommendation is given to configurator types that be used for a selection criterion but are neither optimal or not-suitable.

Product optimisation selection criteria

The customer's prior product knowledge is used as additional selection criteria for the reasoning types, since it is not in alignment with the customer prior product knowledge of the business strategies (visible in *table 12*).

The knowledge based reasoning configurator types differ in their suitability to perform customer guidance and require different levels of customer product knowledge. The customers prior product knowledge is expressed in the following three levels, namely “no or little”, “average” or “expert” prior product knowledge.

A customer with no prior product knowledge requires a configurator which automatically creates a configured product, based on the customers desires/requirements. When a customer has average prior product knowledge, they possess knowledge of the product and its component functions. Customers with average product knowledge will require some guidance to configure a complete product. Expert prior product knowledge, is possessed by customers who can configure the products without guidance, through knowledge of the required materials and components, to create a good performing complete product.

Question 6 ($c - 6$) is used to determine the customers prior product knowledge and is based on the customer’s capabilities.

- c_6 Is the customer capable of, or does the customer need to, assemble components into a functioning product without guidance?
- a No, customers have little prior product knowledge and cannot configure a product from components without guidance.
The customer has no prior product knowledge and needs a configurator that guides the customers through the configuring process or automatically performs the configuring based on the customers desires.
 - b No, the customer is able to recognise component functions, but needs guided steps in the configurator to configure a complete product.
The customer has average prior product knowledge and possesses knowledge of the product and its components, and requires a configurator that configures a product in guided steps.
 - c Yes, the customer is able to configure a product to their desires without guidance, by selecting components with specifications which satisfy their needs.
The customer has expert prior product knowledge and can configure the products without guidance through knowing the required materials and components to create a good performing, complete product

All reasoning types are further discussed in the following paragraph 3.2.2 to 3.2.6 in order to determine their selection criteria and suitability for the used selection configurator types and characteristics.

3.2.2 Rule-based type selection

As stated in the literature review, the rule-based type contains both configurator knowledge and the computing strategy in rules. When a customisable product has a lot of product configure strategies and component relations, the required rules to contain this knowledge will become bulky and cause man-made mistakes.

The rule-based type is optimal for predesigned products, since they contain little configurator knowledge. The assembly-to-order strategy represents a simple product complexity with predesigned components and can be used to select the rule-based reasoning configurator type. The rule-based selection is not recommended to be selected for a fabricate or engineer-to-order strategy, which use non-predesigned components.

All rules are interlinked/depending to each other to compute a solution, therefore as mentioned in the literature review, the rule-based reasoning type has difficulties with knowledge maintenance, consistency checking and knowledge acquisition. One change may cause many errors in relating rules, all linking rules have to be checked when a rule is changed.

The rule-based type functions poorly if rules are regularly changed. Only when there is no need to change the computing strategy or knowledge of offered products, or the changes are few and minor, the rule-based type will still be a suitable configurator type. The performance of the rule-based type depends on the need of knowledge maintenance. The rule-based type performs well when there is no need for knowledge maintenance, which causes less possible mistakes made in the knowledge. The offline interaction nature is selected when there is no need for knowledge maintenance. Therefore, the rule-based type can be selected for an online interaction nature. In contrast, the rule-based type is not recommended for an online interaction nature which is selected for a need of knowledge maintenance.

Rule-based type performance for different prior product knowledge of customers and product optimisation is not presented in particular. Therefore, it is assumed that the rule-based type can be used in these situations, however its performance is neither non-recommended or recommended. The rule-based type receives a neutral recommendation to be used in these mentioned situations.

The rule-based reasoning type selection criteria is presented in *table 13*.

		c ₁ Interaction nature		c ₂ Business strategy			c ₅ Optimisation		c ₆ Customer prior product knowledge		
		Online nature	Offline nature	Assembly to order	Fabricate to order	Engineer to order	Yes	No	Few	Average	Expert
Rule-based	a ₁	-	+	+		-					

Table 13: Rule-based reasoning type selection criteria, *source by the author*

3.2.3 Case-based type selection

The in the literature presented case-based type, adapts earlier made configurations to solve/create the requested configuration. Based on the literature reveal it can be concluded that the case-based type are only suitable for adjustable products and old cases to derive a new solution from.

The case-based type can be used for a company with an adjustable standard product, for example a glass company. The configurator knows the standard produced glasses and finds a solution when a requested glass differs from standards through looking at cases which were produced earlier. Standard products can be defined as products which are predefined and can be slightly adjusted, for example glass products in a plates of glass-industry which can be adjusted in dimensions.

Since, the case-based type primary focuses on adjusting products, the type is recommended for the fabricate-to-order strategy. In contrast, the case-based type is not recommended for the engineer-to-order strategy, which engineers products to order.

The case-based type is only recommended when product optimisation is requested, since the reasoning type focuses on the improving of earlier sold configurations.

Rule-based systems are used in the case-based type to adjust old configurations to solve new cases as mentioned in the literature review. For the acquisition of knowledge, these rule-base systems have to be adjusted. The rule-based systems must be updated with information of instances of components which were removed, updated or new. Therefore, the case-based type is not recommended when there is a need for knowledge maintenance. The offline interaction nature can be used as selection criteria of the case-based reasoning type, however the type is not necessarily optimal for an offline interaction nature. The case-based type is not recommended with an online interaction nature which represents a need for knowledge maintenance.

Furthermore, the case-based type is recommended to use with customers with little or no prior product knowledge, since it uses the customer's requirements for configuring, the customers only have to enter their desires. The case-based type receives a neutral recommendation for average and expert prior product knowledge, since the products are simple, so more knowledge of the product is not necessary.

The case-based reasoning type can be selected for the selection criteria presented in *table 14*.

		c ₁ Interaction nature		c ₂ Business strategy			c ₅ Optimisation		c ₆ Customer prior product knowledge		
		Online nature	Offline nature	Assembly to order	Fabricate to order	Engineer to order	Yes	No	Few	Average	Expert
Case-based	a ₂	-			+	-	+	-	+		

Table 14: Case-based reasoning type selection criteria, *source by the author*

3.2.4 Logic-based type selection

The description logic is recommended for complex products and knowledge maintenance, since the description logic is able to upkeep large complex systems and performs optimal with knowledge maintenance through consistency checking. Therefore, the description logic type can be selected for the engineer-to-order strategy and an online interaction nature.

The description logic type receives a neutral recommendation for an offline interaction nature, assembly-to-order and fabrication-to-order strategy. The description logic type is suitable for these criteria, however the description logic is optimal for knowledge maintenance and complex products.

As stated in the literature review, the description logic type is not able to optimise products, and not recommended for configurators that perform product optimisation. This type receives a neutral recommendation if product optimisation is not required.

Customers with no/little prior product knowledge can configure a product with the description logic type by presenting their requirements. As stated in the literature review, the description logic type configures a configuration based on the customer's input. The description logic type receives a neutral recommendation for customers with expert prior product knowledge, since the type's advantage is to be able to guide customers with no prior product knowledge.

The description-logic type can be selected for the presented selection criteria in *table 15*.

		c ₁ Interaction nature		c ₂ Business strategy			c ₅ Optimisation		c ₆ Customer prior product knowledge		
		Online nature	Offline nature	Assembly to order	Fabricate to order	Engineer to order	Yes	No	Few	Average	Expert
Logic-based	a ₃	+				+	-		+		

Table 15: Logic-based reasoning type selection criteria, *source by the author*

3.2.5 Resource-based type selection

The resource-based type is capable of processing complex products through resource-balancing, and can be selected for the engineer-to-order strategy. The assembly-to-order business strategy is also suitable for the resource-based reasoning type. However, the assembly-to-order selection criterion receives a neutral recommendation, since it is not beneficial to perform resource-balancing for pre-designed components.

As explained in the literature reveal, the resource-based type requires expert prior product knowledge to interpret the technical data. The customer should be familiar with the technical specifications, parameters and resources of the product. The resource-based type requires customers to have expert prior product knowledge and is not recommended with little or average prior product knowledge.

The resource-based type is recommended for use in optimisation, since it is capable of optimisation according to the required resources. The type can optimise the configuration through the selection of resources which requires the least resources or the resources which are available or producible within a certain time frame. A neutral recommendation is given for the use of this type if no optimisation is needed, since the type can be used, however it losses the great advantage its optimisation possibilities.

The resource-based type is recommended when there is a need for knowledge maintenance. All part relations are expressed in resources which makes maintenance simpler, as missing links can be retraced through missing resources. In addition, the resource-based type is able to prevent and predict inconsistencies through resource balancing. Therefore, the resource-based type is recommended with the online interaction nature.

The resource-based type can be used for an offline interaction nature as well. However, the type has an advantage in being able to prevent inconsistencies which is optimal for changing configurator content. Therefore, the resource-based type receives a neutral recommendation for the offline interaction nature.

The resource-based reasoning type can be selected for the presented selection criteria in *table 16*.

		c ₁ Interaction nature		c ₂ Business strategy			c ₅ Optimisation		c ₆ Customer prior product knowledge		
		Online nature	Offline nature	Assembly to order	Fabricate to order	Engineer to order	Yes	No	Few	Average	Expert
Resource-based	a ₄	+				+	+		-	-	+

Table 16: Resource-based reasoning type selection criteria, *source by the author*

3.2.6 Constraint-based type selection

In the constraint-based type a constraint-satisfaction-problem is solved through finding satisfying components which perform a certain function role. A customer should have average or expert prior knowledge to give input on which functions the configuration must fulfil in order to solve the constraint-satisfaction-problem. Therefore, the constraint-based type is not suitable for customers with little or prior product knowledge. The type receives a neutral recommendation for customers with expert or average prior knowledge, since the type is suitable, but it has no further advantage to use it.

The formulations of constraints and variables could make the problem significantly easier or more difficult to solve mentioned in the literature review. Expert knowledge is required to add or change the formulations of constraints and variables of the constraint-based type. Therefore, the constraint-based type will be difficult in knowledge maintenance or acquisition since it requires expert knowledge. The type is not recommended to use if the configurator knowledge changes often. The constraint-based type is neutrally recommended for an offline operating nature type, since it is suitable to use if content is not often changing, however the type has no further advantage to use for an offline interaction nature.

The constraint-based type can be used to create and optimise products through using components which fulfil a function role. The type is neutrally recommended in the case where products need optimisation, since it has no further advantage of using the type. For example, the resource-based type has the advantage to optimise for different goals through evaluating resources of different configurations. In addition, the type receives a neutral recommendation if no optimisation is needed, because the type remains useable.

By using existing adjustable components which fulfil a function role, the constrained base type is comparable and is recommended to select with the assembly-to-order strategy. The components and function roles are predesigned, therefore the type is not recommended to be used with the fabricate/engineer-to-order strategy where components are not-predesigned.

The constraint-based reasoning type can be selected for the presented selection criteria in *table 17*.

	c ₁ Interaction nature		c ₂ Business strategy			c ₅ Optimisation		c ₆ Customer prior product knowledge		
	Online nature	Offline nature	Assembly to order	Fabricate to order	Engineer to order	Yes	No	Few	Average	Expert
Constraint-based a ₅	+		+	-	-			-		

Table 17: Constraint-based reasoning type selection criteria, *source by the author*

3.2.7 Reasoning type selection process

The evaluated reasoning types indeed support the hypothesis that all reasoning configurator types can be selected based on the “interaction nature” and “business strategy” configurator types and the “optimisation” and “customer prior product knowledge” characteristics are correct

The suitability of the reasoning types for the selection criteria, can be represented in a decision matrix, as shown in *table 18*. The left column represents the different reasoning types a1 to a5, the first row shows four different selection criteria (c₁, c₂, c₅, c₆). In *table 18* the matrix attributes are expressed with either a plus, minus or an empty space, which is derived from the recommendations. A type which is not recommended for a selection criterion receives the a (-) minus. A neutrally recommended type receives an empty space and a recommended type receives a (+) plus.

Numerical values are required to solve the decision matrix with a MCDM method, therefore the plus, empty space and minus are in that order replaced with the values 3, 1, and 2 as presented in *table 19*. Instead of using negative values only positive values are given, which makes it easier to solve the matrix with an MCDM method.

Selection criteria	c ₁ Interaction nature		c ₂ Business strategy			c ₅ Optimisation		c ₆ Customer prior product knowledge		
	c _{1a} Online nature	c _{1b} Offline nature	c _{2a} Assembly to order	c _{2b} Fabricate to order	c _{2c} Engineer to order	c _{5a} Yes	c _{5b} No	c _{6a} Few	c _{6b} Average	c _{6c} Expert
Reasoning types										
Rule based	a ₁	-	+	+	-					
Case based	a ₂	-		+	-	+	-	+		
Description logic	a ₃	+			+	-		+		
Resource based	a ₄	+			+	+		-	-	+
Constraint based	a ₅	-	+	-	-			-		

Table 18: Decision matrix of the knowledge based reasoning types selection, *source by the author*

Selection criteria	c ₁ Interaction nature		c ₂ Business strategy			c ₅ Optimisation		c ₆ Customer prior product knowledge		
	c _{1a} Online nature	c _{1b} Offline nature	c _{2a} Assembly to order	c _{2b} Fabricate to order	c _{2c} Engineer to order	c _{5a} Yes	c _{5b} No	c _{6a} Few	c _{6b} Average	c _{6c} Expert
Reasoning types										
Rule based	a ₁	1	3	3	2	1	2	2	2	2
Case based	a ₂	1	2	2	3	1	3	1	3	2
Description logic	a ₃	3	2	2	2	3	1	2	3	2
Resource based	a ₄	3	2	2	2	3	2	1	1	3
Constraint based	a ₅	1	2	3	1	1	2	2	1	2

Table 19: Decision matrix with numbers of the reasoning types selection, *source by the author*

The optimal reasoning type is selected based on the highest total value for the selection criteria of a particular customisable product. However, the values cannot simply be added but require weighing to determine for example how important the optimisation selection criteria are compared to the customers prior product knowledge criteria. The AHP method is used to create the weighing in *paragraph 3.3.1* by determining the ratio of importance for each criterion.

An MCDM method is used to solve the matrix for the input of a particular customisable product. The columns that fit the obtained input from the questions, is used in multi criteria decision making.

3.3 Multi criteria decision making process

The AHP method is used to determine the importance ratio of the different selection criteria, which is needed to compare the different criteria attribute values in an MCDM method. The PROMETHEE, TOPSIS, ELECTREE and VIKOR methods are evaluated to determine the most suitable MCDM method to select the optimal configurator reasoning type for a particular customisable product. The analyses of the MCDM methods are based on the usability and simplicity of the selection process to be usable for companies with no prior knowledge of the MCDM method.

3.3.1 AHP method ratio of importance determination

The selection criteria weighing is determined with the AHP method, which prioritisation procedure was developed by Saaty (1987) in 1971.

The AHP method derives ratio scales or in other words criteria weights from paired comparisons. In order to use the AHP method a comparison has to be made through assigning the fundamental scale of actual numbers to the selection criteria, see *figure 7*.

Typically, expert judgment is used to assign the fundamental scale of actual numbers in the paired comparison. For the judgment of the selection criteria however, it should be based on the priorities of a company for their customisable product. For example if the selection criteria are based on the functionalities of the customisable product it is up to a company to decide whether the optimisation of a product is more important than the suitability for customers with no product knowledge. Therefore, the ratio of importance determination with the AHP method in the proposed method is performed by a company of a particular customisable product. In *paragraph 4.2.1* an example is given to derive the ratio of importance with the AHP method.

3.3.2 TOPSIS evaluation

The TOPSIS method is used to rank the alternatives. It is based on the Euclidian distance to the most ideal solution. The method takes the distance from the least ideal and most ideal solutions into account which gives an ideal trade-off solution. The method can give unrealisable results if not used properly, however for the proposed method predefined values are used which prevents usage errors.

TOPSIS ranking can be performed in a few simple steps, which makes it an ideal MCDM method to use for companies with little prior knowledge. The simplicity of the steps makes the TOPSIS method suitable for the selection process, since companies need to be capable to use the method to find optimal configurator types for their company.

VIKOR

The VIKOR method balances between the maximising of the utility group and minimising the regret group for the best alternative, which represents a close-to-ideal solution. A compromise is proposed if the advantage over an alternative is small.

The VIKOR is more complex and requires more steps to compute a solution than the almost similar TOPSIS method. However, the VIKOR method proposes a compromise solution (or more than one), instead of providing only a ranking of advantages rates, if the advantage difference of the alternatives is small.

A Company has more difficulty using the VIKOR method compared to TOPSIS and needs more time to compute a solution due to a larger number of required computing steps. The ability to propose compromised solution(s), is not beneficial for the selection process of configurator types, since the configurator types already exist. Therefore a compromise would require companies to develop new types.

ELECTREE

The ELECTREE method is often used as a filtration process of the alternatives and can be used for choosing, ranking and sorting. The ELECTREE method is difficult to use compared to the other ranking methods, due to the calculation of the concordance and discordance matrices. The ELECTREE method has more functionalities which makes the method time consuming and less intuitive due to larger number of required steps to compute a ranking.

The ELECTREE method provides a more real/precise answer compared to other ranking methods like TOPSIS for example. However, precision is less important because the selection should be based on big differences between usage advantages of the five reasoning types.

PROMETHEE

The PROMETHEE method performs partial ranking and final ranking based on the advantage of alternatives in comparison with each other. Other ranking methods such as TOPSIS perform a partial ranking and are therefore more efficient to use.

The PROMETHEE method is straightforward and uses few steps to come to a final ranking from best to worst alternatives. However, when alternatives are added the PROMETHEE method needs to be re-computed, since other alternatives may change or the available alternatives changes. The lack of structure used in the PROMETHEE method can cause problems when many criteria are used. The lack of structure and trouble when alternatives change is not relevant for the proposed method, since the configurator types do not change and possess few selection criteria.

3.3.3 MCDM method selection

The main priority of the proposed MCDM method is that it needs to be usable for companies with no prior knowledge of the method. Otherwise, the proposed method will not give added value for companies to simplify the selection of configurator types for their customisable product. Therefore, the MCDM method should be simple, intuitive and require few steps to compute a solution.

The precision of the MCDM is less important, since there are only five alternatives from which a selection has to be made. In addition, the precision is not important, because the selection of an optimal type is only based on a large advantage degree compared to other alternatives. When the advantages degree compared to other alternatives is low, the selection is less important, since the types have a similar advantage to be selected.

For this specific use of an MCDM method, the TOPSIS method is the most suitable to perform a selection of reasoning types. The TOPSIS method is intuitive and simple which makes ranking more understandable and easy to perform. Therefore, the TOPSIS method will be used for ranking in the proposed method. In *paragraph 4.2.2* an example is given to show how the TOPSIS can be used to perform the ranking of alternatives with multiple (conflicting) criteria.

4 Method description

The proposed method is based on a sales configuration, where the requirements of the customer are used to determine a configuration which satisfies their wishes. When a company wants to configure a configurator, the proposed method can be used to select the optimal configurator types through answering the 6 created questions presented in *paragraph 4.1*. This chapter describes how the proposed method is used in order to determine the optimal configurator configuration for a specific customisable product. In *paragraph 4.1* an example is given to select the configurator types with non-conflicting selection criteria. The reasoning configurator type selection is explained in *paragraph 4.2*, where firstly the importance ratio is determined using the AHP method and secondly the multi criteria selection is performed with the TOPSIS method.

4.1 Configurator type selection with non-conflicting selection advice

The first step of the proposed method is to obtain the characteristics of a specific customisable product to configure a configurator that suits these characteristics. The 6 listed multiple choice questions below, have to be answered for the specific customisable product.

An example of how to use the proposed method to select the optimal configurator types is given. A configuration is made for answers a) for all 6 input questions ($c_1, \dots, c_6$).

- c_1 Is it necessary to directly synchronise changes of configurator content?
 - a Yes, direct synchronisation is required for configuration.
An online nature is recommended, when components, prices, or products are often changed and need to be directly updated in the configurator.
 - b No, changes of configurator content do often occur or do not need to be directly synchronised.
An offline interaction nature is recommended when it is not required to directly update a configurator when configurator content changes.
- c_2 Which type of components are used?
 - a Predesigned components
The assembly-to-order strategy can be used for assembling predesigned components to customers' wishes.
 - b Adjustable components
The fabricate-to-order strategy is used in configurators where customers can adjust components, for example in dimensions or shape.
 - c Components are engineered to order
The engineer-to-order strategy is used for configurators that design/engineer products to the customer requirements. For example, design of a prosthesis for a missing limb.
- c_3 Does the configurator knowledge need to be exchanged with other companies?
 - a Yes
Several parties are involved in the manufacturing of the product and each party contributes to the configuration process.
 - b No
One party performs the configuration process and possess all configuring knowledge.
- c_4 Is product reconfiguration needed?
 - a No
The configured product does not need to be reconfigured, for example to simulate its new functioning.

- b Yes
Reconfiguration is needed to obtain information of the configured product in a later stadium of the product life cycle.
- c₅ Is the customisable product optimised based on customer requirements, e.g. on weight or price?
 - a Yes
The product has to be optimised to customer requirements.
 - b No
Optimisation is not necessary, the customer can choose a product configuration based on their own preferences.
- c₆ Is the customer capable of, or does the customer need to, assemble components into a functioning product without guidance?
 - a No, customers have little prior product knowledge and cannot configure a product from components without guidance.
The customer has no prior product knowledge and needs a configurator that guides the customers through the configuring process or automatically performs the configuring based on the customers desires.
 - b No, the customer is able to recognise component functions, but needs guided steps in the configurator to configure a complete product.
The customer has average prior product knowledge and possesses knowledge of the product and its components, and requires a configurator that configures a product in guided steps.
 - c Yes, the customer is able to configure a product to their desires without guidance, by selecting components with specifications which satisfy their needs.
The customer has expert prior product knowledge and can configure the products without guidance through knowing the required materials and components to create a good performing, complete product

The configurator types with non-conflicting selection advice can be selected through using the obtained answers of the questions in *in figure 12*.

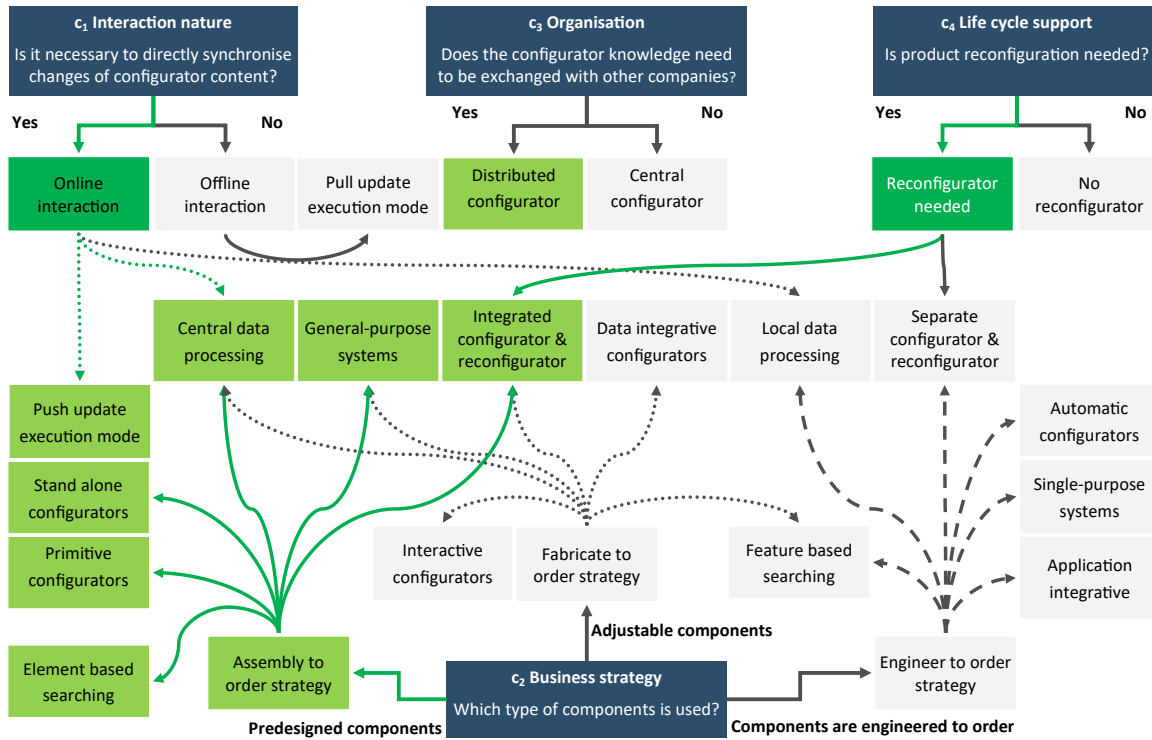


Figure 11: Selection of 24 configurator types for non-conflicting selection advice, *source by the author*

The configurator types with non-conflicting selection advice are marked with light green can be selected based on the a) answers block marked in dark green. The blocks that are selected based on multiple selection criteria (e.g. types with arrows origin from c_1 and c_2) are only selected if both arrows origin from selected answers (represented with green arrows).

4.2 Reasoning type selection

The reasoning configurator types can be ranked based on their suitability, where the highest rank represents the optimal reasoning type for a specific customisable product. An example of the selection of the reasoning types is given, for the answers c_{1a} , c_{2a} , c_{5a} , c_{6a} . A decision matrix can be selected from the table presented in *table 19*, which results in the decision matrix presented in *table 20*.

In order to compare the performance/attributes of the different selection criteria, the ratio of importance of the selection criteria has to be determined. For example, with the ratio of importance the selection criteria can be normalised to compare different criteria such as the importance of the business strategy and the interaction nature.

Selection criteria		C ₁	C ₂	C ₅	C ₆
		Interaction nature	Business strategy	Product optimisation	Customer prior product knowledge
		C _{1a}	C _{2a}	C _{5a}	C _{6a}
		Online interaction nature	Assembly to order	Yes	Few
Reasoning types					
Rule based	a ₁	1	3	2	2
Case based	a ₂	1	2	3	3
Description logic	a ₃	3	2	1	3
Resource based	a ₄	3	2	3	1
Constraint based	a ₅	1	3	2	1

Table 20: Decision matrix selection of questions c_{1a} , c_{2a} , c_{5a} , c_{6a} , *source by the author*

4.2.1 Ratio of importance determination

As presented in the literature review, the ratio of importance determination with the AHP method is performed with expert judgement. However, in the case of the proposed method the expert judgement can be replaced by a company's desires. For example, the selection criteria can be judged based on the desires of a company for their customisable product. When a company finds it more important to have product optimisation than an online interaction nature, the desire can be used in the ratio of importance determination by assigning a higher fundamental value to the product optimisation selection criteria.

In the proposed method a company can use their own weighting based on the fundamental scale of actual numbers, by identifying how important the configurator types and the characteristics are for their specific customisable product. The fundamental scale of actual numbers uses importance intensity values from 1 to 9, presented in *figure 7*.

A company assigns the values through determining paired comparisons of the selection criteria, through relating their importance to one and other. An argumentation and paired comparisons of the selection criteria are made as an example, based on the fundamental values:

Product optimisation may not be required for the specific customisable product. The business strategy is very important in order to select a reasoning type that supports the business strategy's product-complexity and component-type. Therefore, the business strategy receives a "strong plus" value of 6 in relation to the importance of product optimisation. The interaction nature of a company can be "moderate important" to synchronise the product information. The interaction nature receives a fundamental value of 3 in comparison to the product optimisation selection criteria. The business strategy may be (slightly) more important than the selection based on interaction nature, and receive a value of 2 compared to the interaction nature. The selection of configurator types based on the customer prior product knowledge might be "slightly/weakly" more important than the selection based on product optimisation and receives a value of 2. The selection based on the business strategy can be "moderate more important" than the customer prior product knowledge, since the product is simple and customer guidance is not really required. Therefore, the business strategy selection criteria receives a fundamental value of 3 compared to the customer prior product knowledge selection criteria. The selection based on the interaction nature can be determined to be "weakly more important" than the customer prior product knowledge and receive a value of 2. Based on that the product is simple and customer guidance is less important.

All paired comparisons are determined for the selection criteria, and can be listed:

c_1 Interaction nature	3 c_5 Optimisation
c_1 Interaction nature	2 c_6 Customer prior product knowledge
c_2 Business strategy	6 c_5 Optimisation
c_2 Business strategy	2 c_1 Interaction nature
c_2 Business strategy	3 c_6 Customer prior product knowledge
c_6 Customer prior product knowledge	2 c_5 Optimisation

The reciprocal matrix can now be determined based on the paired comparison of the selection criteria's ($c_1, \dots, c_6$) which is visible in *equation 2*. The reciprocal matrix is used to determine the importance ratio of the selection criteria with the AHP method.

$$\begin{array}{c}
 \begin{array}{ccccc}
 & c_{1a} & c_{2a} & c_{5a} & c_{6a} \\
 c_{1a} & \left[\begin{array}{cccc} 1 & \frac{1}{2} & 2 & 3 \end{array} \right. \\
 c_{2a} & \left[\begin{array}{cccc} 2 & 1 & 6 & 3 \end{array} \right. \\
 c_{5a} & \left[\begin{array}{cccc} \frac{1}{2} & \frac{1}{6} & 1 & \frac{1}{2} \end{array} \right. \\
 c_{6a} & \left[\begin{array}{cccc} \frac{1}{3} & \frac{1}{3} & 2 & 1 \end{array} \right.
 \end{array}
 \end{array} \quad (2)$$

The AHP method developed by Saaty (1987) computes priority vectors (or relative weights) through an approximation of the normalised Eigen vector of the reciprocal matrix. The priority value can be computed through normalising each column of the matrix. The first step to normalising is performed through summing each column of the matrix, visible in *equation 3*.

$$\begin{array}{c}
 \begin{array}{ccccc}
 & c_{1a} & c_{2a} & c_{5a} & c_{6a} \\
 c_{1a} & \left[\begin{array}{cccc} 1 & \frac{1}{2} & 2 & 3 \end{array} \right. \\
 c_{2a} & \left[\begin{array}{cccc} 2 & 1 & 6 & 3 \end{array} \right. \\
 c_{5a} & \left[\begin{array}{cccc} \frac{1}{2} & \frac{1}{6} & 1 & \frac{1}{2} \end{array} \right. \\
 c_{6a} & \left[\begin{array}{cccc} \frac{1}{3} & \frac{1}{3} & 2 & 1 \end{array} \right. \\
 sum & 3\frac{5}{6} & 2 & 11 & 7\frac{1}{2}
 \end{array}
 \end{array} \quad (3)$$

The reciprocal matrix is normalised through dividing each element with the sum of its column, visible in *equation 4*.

$$\begin{array}{c}
 \begin{array}{ccccc}
 & c_1 & c_2 & c_5 & c_6 \\
 c_1 & \left[\begin{array}{cccc} \frac{6}{23} & \frac{1}{4} & \frac{2}{11} & \frac{6}{15} \end{array} \right. \\
 c_2 & \left[\begin{array}{cccc} \frac{12}{23} & \frac{1}{2} & \frac{6}{11} & \frac{6}{15} \end{array} \right. \\
 c_5 & \left[\begin{array}{cccc} \frac{3}{23} & \frac{1}{12} & \frac{1}{11} & \frac{1}{15} \end{array} \right. \\
 c_6 & \left[\begin{array}{cccc} \frac{2}{23} & \frac{1}{6} & \frac{2}{11} & \frac{2}{15} \end{array} \right. \\
 normalised & 1 & 1 & 1 & 1
 \end{array}
 \end{array} \quad (4)$$

The normalised principal Eigen vector is obtained through averaging across the rows, visible in *equation 5*. Another name of the normalised principal Eigen vector is the priority vector, which shows the relative weights of the (case specific) selection criteria.

$$W = \frac{1}{4} * \begin{bmatrix} \frac{6}{23} & \frac{1}{4} & \frac{2}{11} & \frac{6}{15} \\ \frac{12}{23} & \frac{1}{2} & \frac{6}{11} & \frac{6}{15} \\ \frac{3}{23} & \frac{1}{12} & \frac{1}{11} & \frac{1}{15} \\ \frac{2}{23} & \frac{1}{6} & \frac{2}{11} & \frac{2}{15} \end{bmatrix} = \begin{bmatrix} 0,2732 \\ 0,4918 \\ 0,0928 \\ 0,1422 \end{bmatrix} \quad (5)$$

In order to know if the created selection criteria usage based judgment is reliable, its consistency can be checked. The matrix is consistent if lambda-max is equal to n the number of elements being compared (Saaty, 2008). The consistency of the reciprocal matrix can be checked with the principal Eigen value. The principal Eigen value can be determined through multiplying the Eigen vector products of *equation 5* with the summed values of the matrix columns of *equation 3*. The formula to determine the principal Eigen value is presented in *equation 6* (Saaty, 1987), the principal Eigen value is calculated in *equation 7*.

$$\sum_{i=1}^n \sum_{j=1}^n a_{ij} w_j = \sum_{j=1}^n \left(\sum_{i=1}^n a_{ij} \right) w_j = \sum_{i=1}^n \lambda_{max} w_i = \lambda_{max} \quad (6)$$

$$\lambda_{max} = 3 \frac{5}{6} * (0,2732) + 2 * (0,4918) + 11 * (0,0928) + 7 \frac{1}{2} * (0,1422) = 4,1184 \quad (7)$$

The calculation in equation 6 show that lambda-max is equal to the number of elements which are being compared and therefore the reciprocal matrix of the replaced expert judgement is consistent. The degree of the consistency can be determined to know how consistent the reciprocal matrix is, represented the Consistency Index (CI) in *equation 8*. The consistency is sufficient if it is less or equal to the threshold of 10% (Saaty, 2008).

$$CI = \frac{\lambda_{max} - n}{n - 1} \quad (8)$$

The Consistency Ratio (CR) is determined through comparing CI with the Random Consistency Index (RI), visible in *equation 9*. The Random Consistency Index (RI) is presented in *table 21*, with n representing the number of compared elements. The RI value of 0.90 is found for a n of 4 compared elements.

$$CR = \frac{CI}{RI} \quad (9)$$

n	1	2	3	4	5	6	7	8	9	10
RI	0	0	0.58	0.90	1.12	1.24	1.32	1.41	1.45	1.49

Table 21: Random index (*RI*) table (Saaty, 1987)

The CI & CR are calculated in *equation 10* and *11*, the consistency degree of the selection criteria usage values is 4,4%, which is less than the threshold of 10%, therefore the comparison values are consistent.

$$CI = \frac{4,1184 - 4}{4 - 1} = \frac{0.1184}{3} = 0.03947 \quad (10)$$

$$CR = \frac{0.03947}{0.90} = 0.04385 = 4,4\% \quad (11)$$

The selection criteria comparison values are consistent and therefore the determined importance ratio/weights visible in *equation 5* can be used in the example of the proposed method.

4.2.2 Multi criteria selection

The TOPSIS method is used to rank the optimal configurator types to select the most suitable type to use in the configurator for a specific customisable product. The first step of the ranking process with TOPSIS is the normalisation of the reciprocal matrix. The reciprocal matrix presented in *equation 12*, is made of the decision matrix represented in *table 19*.

$$\begin{array}{c} \begin{array}{c} c_{1a} \quad c_{1b} \quad c_{2a} \quad c_{2b} \quad c_{2c} \quad c_{5a} \quad c_{5b} \quad c_{6a} \quad c_{6b} \quad c_{6c} \\ a_1 \left[\begin{array}{ccccccccc} 1 & 3 & 3 & 2 & 1 & 2 & 2 & 2 & 2 & 2 \end{array} \right] \\ a_2 \left[\begin{array}{ccccccccc} 1 & 2 & 2 & 3 & 1 & 3 & 1 & 3 & 2 & 2 \end{array} \right] \\ a_3 \left[\begin{array}{ccccccccc} 3 & 2 & 2 & 2 & 3 & 1 & 2 & 3 & 2 & 2 \end{array} \right] \\ a_4 \left[\begin{array}{ccccccccc} 3 & 2 & 2 & 2 & 3 & 3 & 2 & 1 & 1 & 3 \end{array} \right] \\ a_5 \left[\begin{array}{ccccccccc} 1 & 2 & 3 & 1 & 1 & 2 & 2 & 1 & 2 & 2 \end{array} \right] \end{array} \end{array} \quad (12)$$

Normalisation is required to be able to compare criteria with incongruous data. In the TOPSIS method, normalisation is performed by using the formula presented in *equation 13* (Opricovic & Tzeng, 2004).

$$r_{ij} = f_{ij} / \sqrt{\sum_{j=1}^J f_{ij}^2}, \quad (13)$$

$$j = 1, \dots, J; \quad i = 1, \dots, n.$$

The normalisation of the reciprocal matrix is calculated in *equation 14* and *15*.

$$\begin{array}{c} \begin{array}{c} c_{1a} \quad c_{1b} \quad c_{2a} \quad c_{2b} \quad c_{2c} \quad c_{5a} \quad c_{5b} \quad c_{6a} \quad c_{6b} \quad c_{6c} \\ a_1 \left[\begin{array}{ccccccccc} 1 & 3 & 3 & 2 & 1 & 2 & 2 & 2 & 2 & 2 \end{array} \right] \\ a_2 \left[\begin{array}{ccccccccc} 1 & 2 & 2 & 3 & 1 & 3 & 1 & 3 & 2 & 2 \end{array} \right] \\ a_3 \left[\begin{array}{ccccccccc} 3 & 2 & 2 & 2 & 3 & 1 & 2 & 3 & 2 & 2 \end{array} \right] \\ a_4 \left[\begin{array}{ccccccccc} 3 & 2 & 2 & 2 & 3 & 3 & 2 & 1 & 1 & 3 \end{array} \right] \\ a_5 \left[\begin{array}{ccccccccc} 1 & 2 & 3 & 1 & 1 & 2 & 2 & 1 & 2 & 2 \end{array} \right] \end{array} \end{array} \quad (14)$$

$$\sqrt{\sum x_{ij}^2} \quad 4.58 \quad 5 \quad 5.48 \quad 4.69 \quad 4.58 \quad 5.20 \quad 4.12 \quad 4.90 \quad 4.12 \quad 5$$

$$\begin{array}{c}
\begin{array}{c}
a_1 \\
a_2 \\
a_3 \\
a_4 \\
a_5
\end{array}
\begin{array}{c}
c_{1a} \quad c_{1b} \quad c_{2a} \quad c_{2b} \quad c_{2c} \quad c_{5a} \quad c_{5b} \quad c_{6a} \quad c_{6b} \quad c_{6c} \\
\left[\begin{array}{ccccccccc}
\frac{1}{4.58} & \frac{3}{5} & \frac{3}{5.48} & \frac{2}{4.69} & \frac{1}{4.58} & \frac{2}{5.20} & \frac{2}{4.12} & \frac{2}{4.90} & \frac{2}{4.12} & \frac{2}{5} \\
\frac{1}{4.58} & \frac{2}{5} & \frac{3}{5.48} & \frac{3}{4.69} & \frac{1}{4.58} & \frac{3}{5.20} & \frac{1}{4.12} & \frac{3}{4.90} & \frac{2}{4.12} & \frac{2}{5} \\
\frac{3}{4.58} & \frac{2}{5} & \frac{2}{5.48} & \frac{2}{4.69} & \frac{3}{4.58} & \frac{1}{5.20} & \frac{2}{4.12} & \frac{3}{4.90} & \frac{2}{4.12} & \frac{2}{5} \\
\frac{3}{4.58} & \frac{2}{5} & \frac{2}{5.48} & \frac{2}{4.69} & \frac{3}{4.58} & \frac{3}{5.20} & \frac{2}{4.12} & \frac{1}{4.90} & \frac{1}{4.12} & \frac{3}{5} \\
\frac{1}{4.58} & \frac{2}{5} & \frac{3}{5.48} & \frac{1}{4.69} & \frac{1}{4.58} & \frac{2}{5.20} & \frac{2}{4.12} & \frac{1}{4.90} & \frac{2}{4.12} & \frac{2}{5}
\end{array} \right]
\end{array}
\end{array}
\quad (15)$$

$$\frac{x_{ij}}{\sqrt{\sum x_{ij}^2}}$$

The selection criteria of the normalised matrix of *equation 15* are still not comparable. In order to compare the different selection criteria, the determined weights using the AHP method in *equation 5*, have to be multiplied by the attributes of each selection criteria. The weighted matrix is visible in *equation 16*.

$$\begin{array}{c}
\begin{array}{c}
a_1 \\
a_2 \\
a_3 \\
a_4 \\
a_5
\end{array}
\begin{array}{c}
c_{1a} \quad c_{1b} \quad c_{2a} \quad c_{2b} \quad c_{2c} \quad c_{5a} \quad c_{5b} \quad c_{6a} \quad c_{6b} \quad c_{6c} \\
\left[\begin{array}{ccccccccc}
0.06 & 0.16 & 0.27 & 0.21 & 0.11 & 0.04 & 0.05 & 0.06 & 0.07 & 0.06 \\
0.06 & 0.11 & 0.18 & 0.31 & 0.11 & 0.05 & 0.02 & 0.09 & 0.07 & 0.06 \\
0.18 & 0.11 & 0.18 & 0.21 & 0.32 & 0.02 & 0.05 & 0.09 & 0.07 & 0.06 \\
0.18 & 0.11 & 0.18 & 0.21 & 0.32 & 0.05 & 0.05 & 0.03 & 0.03 & 0.09 \\
0.06 & 0.11 & 0.27 & 0.10 & 0.11 & 0.04 & 0.05 & 0.03 & 0.07 & 0.06
\end{array} \right]
\end{array}
\end{array}
\quad (16)$$

$$\begin{array}{c}
\text{used} \\
\text{AHP} \\
\text{weights}
\end{array}
\begin{array}{c}
c_{1a} \quad c_{1b} \quad c_{2a} \quad c_{2b} \quad c_{2c} \quad c_{5a} \quad c_{5b} \quad c_{6a} \quad c_{6b} \quad c_{6c} \\
\begin{array}{ccccccccc}
0.27 & 0.27 & 0.49 & 0.49 & 0.49 & 0.09 & 0.09 & 0.14 & 0.14 & 0.14
\end{array}
\end{array}$$

The weighted normalised matrix presented in *equation 16* will be different when other judgment values are used in the AHP method, and should be calculated by multiplying the matrix in *equation 15* with the determined weights. The columns that represent the answers of the 6 multiple choice questions presented in *paragraph 4.1*, represent the example's specific customisable product. In *equation 17* the weighted normalised matrix is presented by only the example's 6 question answers.

$$\begin{array}{c}
\begin{array}{c}
a_1 \\
a_2 \\
a_3 \\
a_4 \\
a_5
\end{array}
\begin{array}{c}
c_{1a} \quad c_{2a} \quad c_{5a} \quad c_{6a} \\
\left[\begin{array}{cccc}
0.06 & 0.27 & 0.04 & 0.06 \\
0.06 & 0.18 & 0.05 & 0.09 \\
0.18 & 0.18 & 0.02 & 0.09 \\
0.18 & 0.18 & 0.05 & 0.03 \\
0.06 & 0.27 & 0.04 & 0.03
\end{array} \right]
\end{array}
\end{array}
\quad (17)$$

$$\begin{array}{c}
\text{used} \\
\text{AHP} \\
\text{weights}
\end{array}
\begin{array}{c}
c_{1a} \quad c_{2a} \quad c_{5a} \quad c_{6a} \\
\begin{array}{cccc}
0.27 & 0.49 & 0.09 & 0.14
\end{array}
\end{array}$$

The next step of the TOPSIS method is to determine the ideal and negative ideal solutions with the presented formulas in *equation 18* (Opricovic & Tzeng, 2004), where I' is associated with benefit criteria and I'' is associated with cost criteria.

$$\begin{aligned}
v_j^* &= \{v_1^*, \dots, v_n^*\} \\
&= \{ \max_j v_{ij} | i \in I' \}, (\min_j v_{ij} | i \in I'') \}, \\
v_j^- &= \{v_1^-, \dots, v_n^-\} \\
&= \{ (\min_j v_{ij} | i \in I'), (\max_j v_{ij} | i \in I'') \}
\end{aligned} \tag{18}$$

In *equation 19* the negative ideal and ideal solutions are defined for the weighted normalised matrix presented in *equation 17*.

$$\begin{array}{ccccc}
& c_{1a} & c_{2a} & c_{5a} & c_{6a} \\
a_1 & \left[0.06 & 0.27 & 0.04 & 0.06 \right] \\
a_2 & \left[0.06 & 0.18 & 0.05 & 0.09 \right] \\
a_3 & \left[0.18 & 0.18 & 0.02 & 0.09 \right] \\
a_4 & \left[0.18 & 0.18 & 0.05 & 0.03 \right] \\
a_5 & \left[0.06 & 0.27 & 0.04 & 0.03 \right] \\
v_j^* & 0.18 & 0.27 & 0.05 & 0.09 \\
v_j^- & 0.06 & 0.18 & 0.02 & 0.03
\end{array} \tag{19}$$

The next step is the calculation of the Euclidean distance to the ideal solution of each reasoning alternative/type. In *equation 20* the formula is presented to calculate the Euclidean distance for the ideal and negative ideal solutions of each type (Opricovic & Tzeng, 2004).

$$\begin{aligned}
D_j^* &= \sqrt{\sum_{i=1}^n (v_{ij} - v_j^*)^2}, \quad j = 1, \dots, J. \\
D_j^- &= \sqrt{\sum_{i=1}^n (v_{ij} - v_j^-)^2}, \quad j = 1, \dots, J.
\end{aligned} \tag{20}$$

An example of the Euclidean distance to the ideal and negative ideal solution calculation is presented for the first reasoning alternative a_1 , in *equation 21*.

$$\begin{aligned}
a_1 \quad D_j^* &= \sqrt{\sum_{i=1}^n (v_{ij} - v_i^*)^2} = ((0.06 - 0.18)^2 + (0.27 - 0.27)^2 + (0.04 - 0.05)^2 + \\
&\quad (0.06 - 0.09)^2)^{0.5} = 0.124 \\
a_1 \quad D_j^- &= \sqrt{\sum_{i=1}^n (v_{ij} - v_j^-)^2} = ((0.06 - 0.06)^2 + (0.27 - 0.18)^2 + (0.04 - 0.02)^2 + \\
&\quad (0.06 - 0.03)^2)^{0.5} = 0.096
\end{aligned} \tag{21}$$

The last step to complete the ranking of reasoning configurator types with the TOPSIS method is the determination of the relative closeness to the ideal solution. The formula and calculation of the relative closeness to the ideal solution of a_1 are presented in *equation 22* (Opricovic & Tzeng, 2004).

$$a_1 \quad C_j^* = \frac{D_j^-}{D_j^* + D_j^-} = \frac{0.096}{0.124 + 0.096} = 0.436 \quad (22)$$

Lastly, the relative closeness to the ideal solution is calculated for each reasoning type (alternative a_1 to a_5) which is presented in *equation 23*. The description logic reasoning type has the highest value of relative closeness to the ideal solution and is the optimal configurator type for the given answers and used weighting.

$c_{1a}, c_{2a}, c_{5a}, c_{6a}$		D_j^*	D_j^-	C_j^*	$rank$
<i>Rule – based</i>	a_1	0.124	0.096	0.436	3
<i>Cased – based</i>	a_2	0.149	0.068	0.314	5
<i>Description – logic</i>	a_3	0.097	0.133	0.578	1
<i>Resource – based</i>	a_4	0.107	0.124	0.538	2
<i>Constraint – based</i>	a_5	0.134	0.092	0.406	4

(23)

The configurator types with non-conflicting and conflicting selection advice are now selected and a configurator configuration is made based on the characteristics of a specific customisable product.

5 Case study

The goal of the case chapter is to validate the proposed method. The proposed method is valid when it delivers a suitable configurator configuration for a specific company. As stated earlier in the introduction, companies often select the unsuitable configurator types due to a lack of knowledge. The selection of not suitable configurator types give errors in storage, retrieval and processing of configuration information.

Two existing configurators from two different companies are evaluated to determine their configurator types. To validate the proposed method a configuration will be made for each customisable product based on answering the 6 presented questions in paragraph 4.1. The configured configurator types with the proposed method are compared with the determined configurator types of evaluated existing configurators. When the configurator configuration, made with the proposed method, is similar to the evaluated configurator, the method can be assumed to be valid.

The configurator types of an existing configurator can easily be determined, since the characteristics and the used configurator knowledge and logic structure are known. These existing configurators are already a success/operational. The companies that want to develop a configurator do not have this information and will have difficulty selecting the right configurator types. This difficulty is tackled in the case by using the proposed method in paragraph 5.1.2 and 5.2.2.

The first case presented in paragraph 5.1, is oriented at a configurator used in a company that sells reptile enclosures. The configurator configures enclosures based on the type of animal inhabitant and customer inputs while validating if the enclosure is producible. More detailed information concerning this configurator is obtained, since the author contributed to the development of the company's configurator during an internship.

The second case presented in paragraph 5.2, examines a configurator that configures kitchen furniture in the company IKEA. Customers can configure their kitchen by, for example, adding closets or an oven and change dimensions, materials and patterns.

5.1 Reptile enclosure configurator

The company Blue-Lagoon builds enclosures for reptiles, amphibians and rodents. Customers can create their pet enclosure on the website of Blue-Lagoon using a configurator. Each type of animal requires different enclosure conditions. For example, different dimensions, lighting, humidity, temperature and ground substrate. The customer selects the type of animal for which they want to design an enclosure. The configurator calculates and shows the price during the configuring process, including material and production cost. The configurator gives advice based on the selected type of animal and the producibility, such as for example the type of lighting that is required.

The configurator of the Blue-Lagoon is first evaluated in paragraph 5.1.1 to determine which configurator types are used. Then, in paragraph 5.1.2, the proposed method is used to configure a configuration for the customisable enclosure by answering the 6 input questions. Finally, the configuration of the proposed method is compared with the found configurator types during the evaluation to determine if the proposed method is valid.

5.1.1 Enclosure configurator evaluation

The configurator classifications are all presented to evaluate which configurator types apply to the customisable enclosure of the company.

Solution searching

The configurator performs the configuring process through proposing detailed elements to the customer who can select their preferred elements. For example, the type of glass used (polished or not) is an element which the customer has to choose. The solution searching type is an element searching type, in which detailed elements are selected by the customer which together form the configuration.

- Element searching

Organisation

The production of the customisable enclosure is performed in one company, the configuring knowledge is contained in one company. Therefore, the organisation of the Blue-Lagoon is “central”, since the product knowledge and production are both stored and produced in one central place.

- Central configurator organisation

Life cycle support

The configurator does not provide an option to reconfigure a previously created configuration. The configurator of the Blue-Lagoon is not used for reconfiguration so a “configurator without reconfiguration” is used.

- Configurator without reconfiguration

Business strategy

The customisable enclosure is fabricated-to-order, since multiple materials, dimensions and backgrounds etc. are fabricated-to-order. The configurator uses both predesigned as adjustable products. The business strategy of the Blue-lagoon is “fabricate-to-order”.

- Fabricated-to-order business strategy

Scope of use

The configurator is made for a “particular industry”, since the configurator is quite complex. The Blue-Lagoon configurator shows a life rendering of the customisable enclosure. To support this render preview a configurator had to be designed specifically to the customisable product. In addition, this company produces part of the components and offers to install predesigned products, such as lighting systems. Considering, specific product representation and production the configurator is particularly built for the customisable enclosure, therefore, the configurator has a single-purpose system.

- Single-purpose system

Interacting nature

The configurator is placed online and the data of the customisable product is stored on an online server. The products and components are often updated in the configurator, therefore the online configurator is used to fulfil the need of update synchronisation. The interacting nature of the Blue-Lagoon configurator is an online nature with “central data processing”. The central data processing is usable, since the computing load is manageable for the updates of product prices or predesigned components of the fabricate-to-order strategy.

- Online nature with central data processing

Update execution

The components of the product require updates often in prices etc., therefore the configurator requires a continuous synchronisation of the configurator knowledge. The configurator knowledge, is stored online on a central server. Therefore, the update execution type of the Blue-lagoon configurator is a “push-update” which can be used for continuous pushing of updated information.

- Push-mode

Integration level

The configurator exchanges information with an Enterprise Resource Planning (ERP) data-base and the final configuration can be exported to a file format readable in MS-Excel. The configurator is designed to exchange information in the JavaScript Object Notation (JSON) data format between the database and the configurator. The final configuration can be exported in a comma-separated values (CSV) file which can be read and used in MS-Excel to print production work lists. The integration level type of the Blue-Lagoon configurator is a “data integrative configurator”, since the Blue-Lagoon configurator is designed to exchange configuring information in a certain file format.

- Data integrative configurator

Complexity

The configurator of the Blue-Lagoon guides the customer through configuration steps. The configuration is performed in 13 steps in which the entire enclosure is designed. For each step the customer has to choose their desired component or material. For example, in one step the ventilation maze is selected, the customer can choose a maze for which the configurator determines if the option is possible with the previously selected components. In addition, the configurator gives advice about which is more suitable for the specific animal inhabitant. For example, a fine ventilation maze is advised for dart frogs to prevent fruit flies (their food) to escape the enclosure. The configurator complexity type is an “interactive configurator” type, since the configurator guides the customer through the configuration process and checks if the configuration is valid/producing/functional.

- Interactive configurator

Configurator reasoning

The configurator reasoning is not performed by one of the reviewed reasoning types. The configurator-reasoning is performed through object-oriented programming, instead of a specific configurator type.

Object oriented programming is used in the configurator to retrieve, store and process the configuration information. As stated in the literature review, object oriented programming is similar to the description-logic configurator type (Donini et al., 1998). Therefore, the description logic reasoning type is used to represent the configurator reasoning of the configurator of the Blue-Lagoon.

In addition, the description logic is the optimal reasoning type for the Blue-Lagoon, since it is optimal to use for synchronisation of configurator knowledge and thus for an online interaction nature. Since the configurator is used on an online platform the need for synchronisation of configurator knowledge is present. Furthermore, the description logic is optimal for customers with little prior product knowledge, which is required for the customers of Blue-Lagoon, who need guidance in the configuring process.

- Description logic

5.1.2 Enclosure configurator configuration selection

To use the proposed method to configure an optimal configurator, the 6 characteristics input questions have to be answered for the customisable enclosure. The questions are answered below based on accessible knowledge of the company. The proposed method is used to select the non-conflicting and conflicting configurator types based on the answers provided.

An online nature is required, often components or prices change and need to be directly updated in the configurator.

c_1 Is it necessary to directly synchronise changes of configurator content?

- a Yes, direct synchronisation is required for configuration.

The fabricate-to-order strategy is used to enable customers to adjust components, in dimensions or shape.

c_2 Which type of components are used?

b Adjustable components

The configuration is produced and its data is contained in one company.

c_3 Does the configurator knowledge need to be exchanged with other companies?

b No

The product is not reconfigured there is no simulation provided of the product performance.

c_4 Is product reconfiguration needed?

a No

Optimisation is not necessary since the customer can choose a product configuration based on their own preferences.

c_5 Is the customisable product optimised based on customer requirements, e.g. on weight or price?

b No

The customer prior knowledge differs a lot per customer of the Blue-Lagoon. In most cases the customer has prior knowledge before he starts configuring their pet's enclosure. However, the prior knowledge is not in-depth and most customers still need guidance to know which features are needed for their pet.

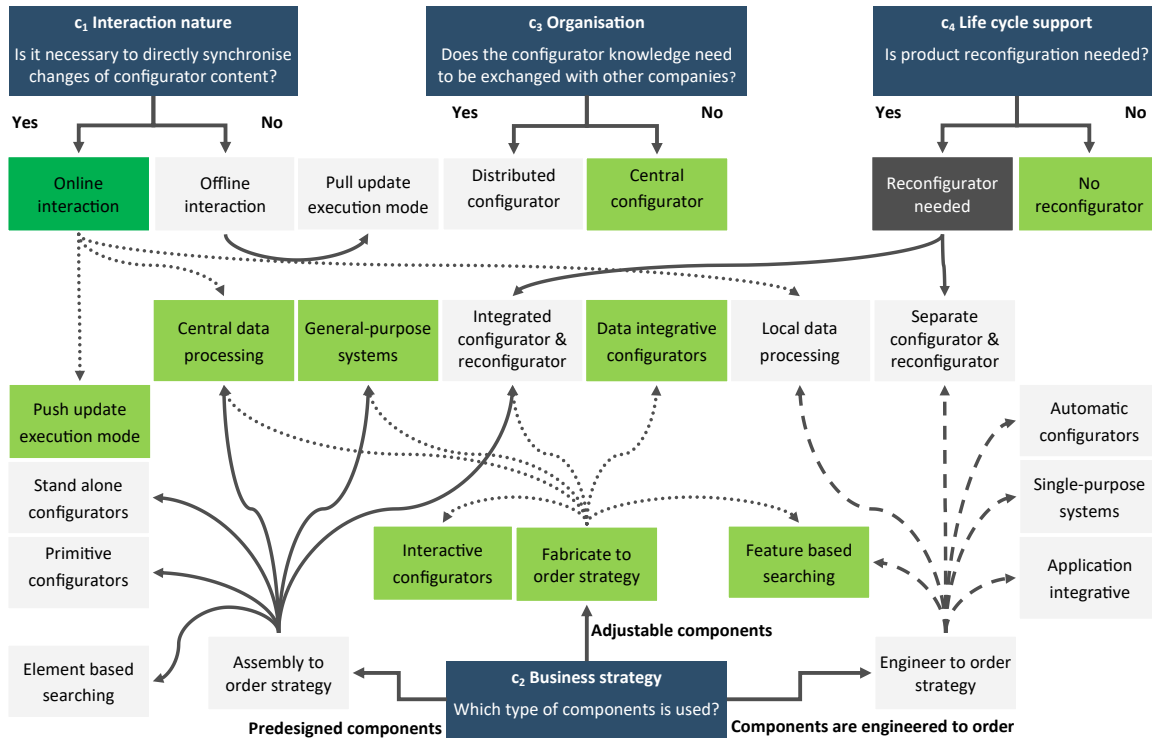
c_6 Is the customer capable of, or does the customer need to, assemble components into a functioning product without guidance?

b No, the customer is able to recognise component functions, but needs guided steps in the configurator to configure a complete product.

Configurator type selection with non-conflicting selection advice

The configurator types with non-conflicting selection advice are selected based on the 6 input questions. As is visible in *figure 12* the configurator types with non-conflicting selection advice are:

- ☐ Feature based searching
- ☐ Central configurator organisation
- ☐ Configurator without reconfiguration
- ☐ Fabricate-to-order business strategy
- ☐ General-purpose system
- ☐ Online interaction nature, with central data processing
- ☐ Push update execution
- ☐ Data integrative configurator
- ☐ Interactive configurator

Figure 12: Configurator types selection with non-conflicting selection advice, *source by the author*

Configurator type selection with conflicting selection advice

The optimal reasoning configurator type is selected with the proposed method through performing multi criteria selection. The decision matrix table presented in *table 22* is used for the TOPSIS method to rank the optimal reasoning types.

Selection criteria		Interaction nature	Business strategy	Product optimisation	Customer product knowledge
		C _{1a}	C _{2b}	C _{5b}	C _{6b}
		Online nature	Fabricate to order	No	Average
Reasoning types					
Rule based	a ₁	1	2	2	2
Case based	a ₂	1	3	1	2
Description logic	a ₃	3	2	2	2
Resource based	a ₄	3	2	2	1
Constraint based	a ₅	1	1	2	2

Table 22: Decision matrix table for input of Blue-Lagoon, *source by the author*

The selection criteria weighing is determined using the AHP method based on the desires of Blue-lagoon and the judgment scales presented in *figure 7*.

The Blue-lagoon's main interest is to provide an online configurator which can be linked with their online enterprise resource planning (ERP) platform. Product optimisation is of the least interest, since the company is more oriented at letting the customer select their product preferences.

Therefore, the interaction nature selection criterion receives a 7 (judgement scale of strongly favoured) compared to the product optimisation criterion. The customer guidance is important in order to prevent customers from asking a lot of questions, or being incapable of configuring, and receives the value 5 of strong importance compared to the optimisation selection criteria.

The business strategy is important to enable customers to adjust their enclosures and components to their desires, in dimensions layout etc. Business strategy receives a value of 5 which represents a strong importance compared to the optimisation selection criteria.

The comparison matrix of the selection criteria based on the desires of Blue-Lagoon is presented in *equation 24*. The presented comparison matrix has a consistency of 0,6 % and is therefore consistent to use in further calculations.

$$\begin{array}{c}
 c_{1a} \quad c_{2b} \quad c_{5b} \quad c_{6b} \\
 c_{1a} \begin{bmatrix} 1 & 2 & 7 & 2 \\ \frac{1}{2} & 1 & 5 & 1 \\ \frac{1}{7} & \frac{1}{5} & 1 & \frac{1}{5} \\ \frac{1}{2} & 1 & 5 & 1 \end{bmatrix} \\
 c_{2b} \\
 c_{5b} \\
 c_{6b}
 \end{array} \quad (24)$$

The AHP weighting determined for the created comparison matrix is presented in *equation 25*. The selection criteria weighting based on the desires of Blue-Lagoon are used to perform multi criteria selection with the TOPSIS method.

$$W = \begin{bmatrix} \text{Interaction nature} & c_{1a} \\ \text{Business strategy} & c_{2b} \\ \text{Optimisation} & c_{5b} \\ \text{Customer knowledge} & c_{6b} \end{bmatrix} = \begin{bmatrix} 0,45 \\ 0,25 \\ 0,05 \\ 0,25 \end{bmatrix} \quad (25)$$

The TOPSIS method is used to rank the reasoning type to select the most optimal type. The ranking is presented in *equation 26*. The description logic has the highest relative closeness value and is ranked as most suitable reasoning type for the customisable product characteristics and desires of Blue-Lagoon.

$$\begin{array}{c}
 C_j^* \quad rank \\
 Rule - based \quad a_1 \quad 0.28 \quad 4 \\
 Cased - based \quad a_2 \quad 0.38 \quad 3 \\
 Description - logic \quad a_3 \quad 0.80 \quad 1 \\
 Resource - based \quad a_4 \quad 0.72 \quad 2 \\
 Constraint - based \quad a_5 \quad 0.22 \quad 5
 \end{array} \quad (26)$$

5.1.3 Enclosure configurator type results comparison

The evaluated configurator types of the company Blue-Lagoon are almost similar to the determined configurator types with the proposed method. However, 2 out of 10 configurator classifications differ in configurator type as is visible in *table 23*.

Configurator classification	Evaluated configurator types	Proposed method types selection
1. Knowledge based reasoning	Description logic	Description logic
<u>2. Solution searching</u>	<u>Element based searching</u>	<u>Feature based</u>
3. Configurator organising	Central organising	Central organising
4. Life cycle support	Configurator without reconfiguring	Configurator without reconfiguring
5. Business strategy	Fabricate-to-order strategy	Fabricate-to-order strategy
<u>6. Scope of use</u>	<u>Single-purpose</u>	<u>General-purpose</u>
7. Interaction nature	Online nature, central processing	Online nature, central processing
8. Update execution	Push update execution	Push update execution
9. Integrational level	Data integrative	Data integrative
10. Complexity	Interactive configurator	Interactive configurator

Table 23: Blue-Lagoon configurator type comparison, *source by the author*

Dissimilarities

The configurator type of the scope of use and solution searching classification are not in accordance with the evaluated configurator types and with the proposed method selected configurator types.

The scope of use evaluated configurator type is based on the desire of Blue-Lagoon to have a configurator that shows real life renderings of the product and is connected to their ERP data-base. Such a configurator needs to be built specifically for this single-purpose. This difference would normally not be present when a configurator is made for a company with a fabricate-to-order strategy as explained in *paragraph 3.1.6*.

The other dissimilarity is the solution search classification, which is performed with an element based searching type instead of the proposed method selected feature searching type. This difference is caused by use of different component types, since the configurator is used for both predesigned and adjustable components. Therefore, business strategy is a mix of assemble-to-order and fabricate-to-order strategies which selection is explained in the literature in *paragraph 2.2.5*. The element searching type is selected for an assemble-to-order strategy which is more in agreement with the operation of Blue-Lagoon, as customers choose specific elements in the configurator. The feature based searching type is selected for a fabricate-to-order strategy, where adjustable components are used and a configuration is made based on customer requirements.

The difference in similarity for the solution search type may be prevented by using the optimisation criteria for the selection of the feature based searching type, since, feature based searching is oriented at configuring a configuration based on the customer requirements, therefore the type searches for the optimal (optimised) configuration. If the optimisation selection criterion was used in this particular case the feature based type would not have been selected as there was no desire for optimisation.

Similarities

Eight of the ten configurator classifications have a similar configurator type. In the selected configurator organisation, life cycle support, business strategy and interaction nature type are also in agreement with the evaluated configurator types. The similarity is logical since the types all have an input question directed on their selection, visible in questions c_1, c_2, c_3, c_4 in *paragraph 4.1*.

In addition, the similarity of the update execution is self-evident as well since it has a logical link to the interaction nature as explained in *paragraph 3.1.8*.

The integration level type which is in the proposed method linked to the business strategy is similar to the evaluated configurator type. The link mentioned in *paragraph 3.1.9*, to select a file format exchanging system based on the selection of the fabricate-to-order business strategy, satisfies in the case of the Blue-Lagoon.

The selection of the configurator complexity based on the shared customer prior knowledge of the business strategies is sufficient in the case of the Blue-Lagoon. The evaluated interactive configurator type is also selected through using the proposed method based on the business-strategy that is in alignment with a similar customer prior knowledge as explained in *paragraph 3.1.10*.

The reasoning type is similar which is most important to the overall performance of the configurator. Therefore, the performance of the overall configurator will be satisfactory if the proposed method succeeds in picking the optimal reasoning configurator type.

The selection of the reasoning type appears to depend a lot on the determined weighing specific to a company's desires for a customisable product. For example, the ranking of *equation 26* and *28* differ a lot, whereas the selection criteria is similar only the used weighing is different. Therefore, the determination of selection criteria weighting is very important to the selection process. The ranking, visible in *equation 28*, is completely different because of the used weighing of the example in *paragraph 4.2.1* (presented in *equation 27*) compared to using the weights of *equation 25*.

$$W = \begin{bmatrix} \text{Interaction nature} & c_1 \\ \text{Business strategy} & c_2 \\ \text{Optimisation} & c_5 \\ \text{Customer knowledge} & c_6 \end{bmatrix} = \begin{bmatrix} 0,27 \\ 0,49 \\ 0,09 \\ 0,14 \end{bmatrix} \quad (27)$$

		C_j^*	rank
Rule – based	a_1	0.42	4
Cased – based	a_2	0.64	1
Description – logic	a_3	0.61	2
Resource – based	a_4	0.59	3
Constraint – based	a_5	0.15	5

(28)

Case conclusion

The proposed method shows a promising selection performance based on a company's customisable product. The selection of the reasoning type appeared to be highly influenceable by the weighing determined based on the company's desires. The selection of the feature based type may need to be adjusted to perform a more precise selection of configurator types using the proposed method. The selection of the feature based type could either be improved by adding the optimisation selection criteria or by assigning the fabricate-to-order strategy to the element searching type. The impact of these changes is probably not be that big, since a company will notice if the selected search strategy is not desirable, since the difference between the searching types is visible in the user interface. However, the selection is preferred as it is as precise as possible, therefore changes should be made if the method is further improved in future work

5.2 Kitchen furniture configurator

IKEA is a company which sells household items and furniture. The company IKEA has multiple configurators to plan or design your furniture. This case focuses on the kitchen configurator of IKEA. The IKEA kitchen planner, allows customers to visualise and design their own kitchen online. The customers can select predefined products to assemble their ideal kitchen. In addition, customers can adjust predefined products, for example the kitchen counter, cut-outs for a sink or induction plate, colour and material.

The kitchen configurator types are evaluated in paragraph 5.2.1 and a configuration of configurator types is made with the proposed method in paragraph 5.2.2. The results from the configurator evaluation and the configurator selection with the proposed method are compared in paragraph 5.2.3

5.2.1 Kitchen configurator evaluation

As performed earlier for the first case, the configurator types are determined for each configurator classification through evaluating the existing configurator of IKEA.

Solution searching

The customers of IKEA can search for furniture products to include in their kitchen configuration. The search of other products is element based, where specific components are searched instead of a configuration based on customer requirements, which is feature based searching.

- ☐ Element searching

Organisation

The company IKEA operates worldwide and produces and buys a lot of components in different places in the world. However, the configurator knowledge and the assembly of the final product is contained in the company IKEA. Therefore, the company has a central configurator organisation.

- ☐ Central configurator organisation

Life cycle support

The kitchen configurator of IKEA can save configurations to continue designing in another time. Reconfiguring is however not supported, since there is no further use for reconfiguration, when a configuration can be stored.

- ☐ Configurator without reconfiguration

Business strategy

In the configurator it is possible to adjust predefined products, often these adjustments are colour or pattern. Production of these components often remains the same, and is only adjusted in finishing. For some products the dimensions can be changed, for example strips to close gaps between closets etc. However, these products are also predefined and eventually sawn to desired dimensions. Therefore, the configurator of the IKEA kitchen uses an assembly-to-order strategy, where the predefined products are composed to order.

- ☐ Assembly-to-order strategy

Scope of use

IKEA is a large company and has many furniture configurators, therefore they designed a general-purpose configurator which they can use for all of their customisable products. For example, IKEA uses the same configurator for the design of kitchen furniture, living space furniture, closet configuration and chair configuration etc.

- ☐ General purpose-configurator

Interacting nature

The configurator of IKEA is presented online to their customers, and has an online interaction nature. The data is stored and configuring calculations are stored and performed on a central server. The customer local unit only has to read the data from the central server. The configuring speed is depending on the computing power to process the configuration and the internet connection of the customer to perceive the configuring results.

- Online interaction nature with central data processing

Update execution

The update execution is a push update, whenever configuration information changes this information/data is automatically pushed to the online configurator.

- Push mode

Integrational level

The configurator does not really export more than assembly information of predesigned products, therefore, the configurator is stand alone and does not require specific communication-systems to other systems.

- Standalone configurator

Complexity

The configurator does not check for the validity of the configuration and gives a warning to the customer that IKEA is not responsible for improper configurations made using the kitchen configurator. Furthermore, the configurator gives advice, however this advice is static and does not check the validity of the configuration. The advice is often presented as a warning for the specifications which have to be considered for using a specific product. Therefore, the kitchen configurator has a primitive operation, where the customer has to understand the product to consider the validity of the configuration.

- Primitive configurator

Configurator reasoning

The configurator requires a reasoning configurator type, which is optimal to synchronise product changes, since IKEA continuously designs and uploads products. Furthermore, the configurator reasoning, should be able to manage a lot of different products and different categories and requires consistency checking. The IKEA configurator does not optimise their products based on their performance or preferences, therefore no optimisation is required. Based on these notations the description logic reasoning type is most suitable to fulfil these requirements, and is probably used in the IKEA kitchen design configurator.

- Description logic

5.2.2 Kitchen configurator configuration selection

The 6 input characteristics for the proposed method are again determined through answering the created questions for a customisable product. The questions are answered based on the available information of the company IKEA. The answers are used in the proposed method to select configurator types with non-conflicting and conflicting selection advice.

The IKEA configurator is used online which requires an online interaction nature. IKEA produces many new products which are introduced in configurators online. Therefore, the configurator knowledge is updated often and requires synchronisation of configurator data.

- c_1 Is it necessary to directly synchronise changes of configurator content?

- a Yes, direct synchronisation is required for configuration.

The IKEA uses predefined components which are assembled to order.

- c_2 Which type of components are used?

- b Predesigned components

The configuration is produced and its data is contained within the company IKEA.

- c_3 Does the configurator knowledge need to be exchanged with other companies?

- b No

The product is not reconfigured and there is no simulation provided of the product performance.

- c_4 Is product reconfiguration needed?

- a No

Optimisation is not necessary the customer can choose a product configuration based on their own preferences.

- c_5 Is the customisable product optimised based on customer requirements, e.g. on weight or price?

- b No

Customers of IKEA differ a lot in age and intelligence which results in many different levels of customer prior product knowledge. To enable all customers of IKEA to use the kitchen configurator is quite difficult. Even if a configurator is created that configures an automatic design based on requirements, the elderly can have problems using the configurator. IKEA offers customers to configure their product in a local IKEA store, where employees can help or perform the configuring based on the customer desires.

The configurator should support the majority of customers and have customer guidance based on customers with no or little prior product knowledge.

- c_6 Is the customer capable of, or does the customer need to, assemble components into a functioning product without guidance?

- a No, customers have few prior product knowledge and cannot configure a product from components without guidance.

Configurator type selection with non-conflicting selection advice

The configurator types with non-conflicting selection advice are selected based on the 6 input questions. As is visible in *figure 13* the configurator types with non-conflicting selection advice are:

- ☐ Technical element based searching
- ☐ Central configurator organisation
- ☐ Configurator without reconfiguration

- Assembly-to-order business strategy
- General-purpose system
- Online interaction nature, with central data processing
- Push update execution
- Standalone configurator
- Primitive configurator

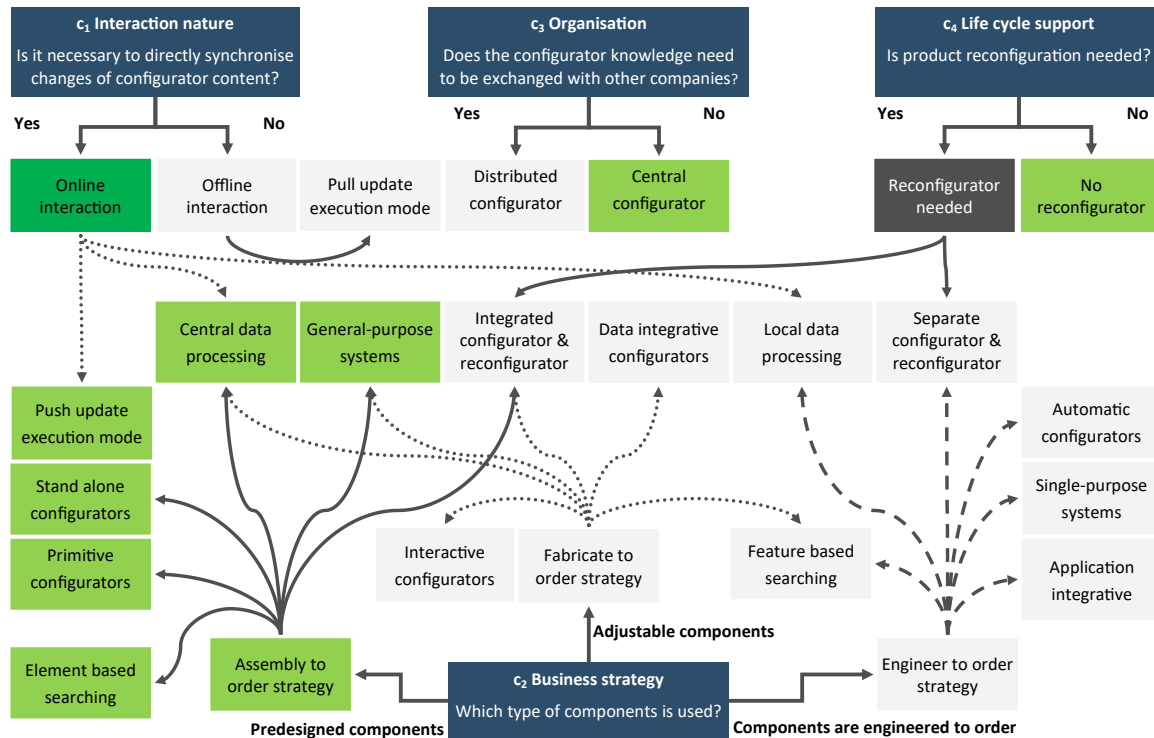


Figure 13: IKEA: Configurator types selection with non-conflicting selection advice, *source by the author*

Configurator type selection with conflicting selection advice

The reasoning configurator type is selected by the proposed method through performing multi criteria selection. The decision matrix table for IKEA is presented in *table 24* is used for the TOPSIS method to rank the optimal reasoning types.

Selection criteria weighing is determined by the AHP method based on the desires of IKEA and the judgment scales presented in *figure 7*.

The IKEA company aims at customers with different skills and prior product knowledge to use a configurator, for example people of all ages and computer skill levels should be able to use the kitchen configurator.

All products used in the configurator are predefined, therefore product optimisation is not relevant since the products are already fabricated. The guidance for customers with few prior product knowledge is more important and relevant than the need for optimisation of the product.

Selection criteria		Interaction nature	Business strategy	Product optimisation	Customer product knowledge
		c _{1a}	c _{2a}	c _{5b}	c _{6a}
Reasoning types		Online nature	Assembly to order	No	No to few
Rule based	a ₁	1	3	2	2
Case based	a ₂	1	2	1	3
Description logic	a ₃	3	2	2	3
Resource based	a ₄	3	2	2	1
Constraint based	a ₅	1	3	2	1

Table 24: Decision matrix table for input of IKEA, *source by the author*

The products offered in the configurator are predesigned and are assembled to order in the configurator. An assembled to order configurator is often used for customers with in depth product knowledge to be capable to assemble a functional product to their desires. However, customer guidance and support of the majority of IKEA customers is more important for the company. Therefore, the selection of a configurator for customers with little prior product knowledge, is more important to IKEA than the selection of a configurator based on the business strategy.

IKEA creates many products which requires the configurator to be updated with new products. Therefore, the online interaction nature is of equal importance as the selection based on customer guidance and thus more important than the business strategy and the need for product optimisation.

The following relations are based on the desires of the company IKEA:

- The selection based on customer prior product knowledge is assumed to be of equal importance as the online interaction nature.
- The online interaction nature is assumed to be 6 times more important than the need for optimisation.
- The online interaction nature is assumed to be 2 times more important than the business strategy
- The business strategy is assumed to be 4 times more important than the need for optimisation.

The assumed desires of IKEA are represented in the decision matrix in *equation 29*. The presented comparison matrix has a consistency of 0,4 % and is therefore consistent to use in further calculations.

$$\begin{array}{c}
 \begin{array}{c} c_{1a} \\ c_{2a} \\ c_{5b} \\ c_{6a} \end{array}
 \begin{bmatrix}
 1 & 2 & 6 & 1 \\
 \frac{1}{2} & 1 & 4 & \frac{1}{2} \\
 \frac{1}{6} & \frac{1}{4} & 1 & \frac{1}{6} \\
 1 & 2 & 6 & 1
 \end{bmatrix}
 \end{array} \quad (29)$$

The AHP weighting determined for the created comparison matrix is presented in *equation 30*. The selection criteria weighting based on the desires of IKEA are used to perform multi criteria selection with the TOPSIS method.

$$W = \begin{bmatrix} \text{Interaction nature} & c_{1a} \\ \text{Business strategy} & c_{2a} \\ \text{Optimisation} & c_{5b} \\ \text{Customer knowledge} & c_{6a} \end{bmatrix} = \begin{bmatrix} 0,37 \\ 0,20 \\ 0,06 \\ 0,37 \end{bmatrix} \quad (30)$$

The ranking based on the desires of IKEA and the answers of the 6 input questions are visible in *equation 31*. The description logic is ranked as optimal reasoning type for the customisable product characteristics and desires of IKEA.

$$\begin{array}{llll} & & C_j^* & rank \\ \text{Rule – based} & a_1 & 0.32 & 4 \\ \text{Cased – based} & a_2 & 0.48 & 3 \\ \text{Description – logic} & a_3 & 0.86 & 1 \\ \text{Resource – based} & a_4 & 0.51 & 2 \\ \text{Constraint – based} & a_5 & 0.15 & 5 \end{array} \quad (31)$$

5.2.3 Kitchen configurator type results comparison

The selected configurator types of the proposed method based on the desires of IKEA and 6 input questions are identical to the evaluated configurator types of IKEA, as is presented in *table 25*.

Configurator classification	Evaluated configurator types	Proposed method types selection
1. Knowledge based reasoning	Description logic	Description logic
2. Solution searching	Element based searching	Element based searching
3. Configurator organising	Central organising	Central organising
4. Life cycle support	Configurator without reconfiguring	Configurator without reconfiguring
5. Business strategy	Assembly-to-order strategy	Assembly-to-order strategy
6. Scope of use	General-purpose	General-purpose
7. Interaction nature	Online nature, central processing	Online nature, central processing
8. Update execution	Push update execution	Push update execution
9. Integrational level	Standalone configurator	Standalone configurator
10. Complexity	Primitive configurator	Primitive configurator

Table 25: IKEA configurator types comparison, *source by the author*

Similarities

The similarities of the evaluated and selected configurator types are obtained through the distinct operation of the configurator of IKEA.

The functionalities of the kitchen configurator of IKEA are exactly represented in the existing configurator types, there is no compromised configurator type. Therefore, the selected configurator types for IKEA with the proposed method are completely corresponding to the evaluated functionalities of the existing IKEA configurator.

The description logic configurator reasoning type is ranked as most optimal type and is evaluated as configurator type of the existing configurator as well. The selection of the configurator reasoning is mainly influenced by the predetermined desires of the company IKEA and the input of the six questions. The determination of the desires of IKEA are assumed and give the highest possibility of errors of different outcomes. Normally, the desires for the performance of the configurator are made by the company itself to prevent a possible error of assumption of which desires are most important for the company.

Case conclusion

The results of the selection of configurator types for an IKEA kitchen configurator, are very accurate since the selection is completely similar to the evaluated configurator types of the existing configurator. The similar outcome is a result of the distinct configurator of IKEA, which functionality does not require compromised solutions and uses an existing general configurator system. The result of the proposed method is highest when the desired configurator is aligned with existing configurator types.

As previously concluded in the first case, the selection of the reasoning configurator type is highly influenced by the determination of the desires for a configurator for a particular customisable product. The prevent assumption errors this desires are normally determined by the company of the customisable product.

6 Discussion

The results of the two cases presented in chapter 5 are discussed, to determine if the proposed method satisfies the requirements to select optimal configurator types to store, retrieve and process product variant information as well as possibly for a specific customisable product. The proposed methods' selection process is reviewed to determine if all selection criteria make sense, and if the results of the two cases are logical. Based on the results of the cases, improvements of the proposed method are suggested to increase the level in which it performs the configurator type selection.

6.1 Requirements

The goal of the development of the proposed method is to select the optimal configurator configuration for the characteristics of a customisable product. Through the selection of an optimal configurator to support the characteristics of the customisable product, the configurator is able to support the storage, retrieval and processing of the customisable product information.

The proposed method meets this requirement through selection of the optimal configurator types for a particular customisable product. In both cases the proposed methods selected an almost similar configurator configuration which is used in the two existing companies. The selection of the most important configurator reasoning type is similar to the used reasoning in both companies. Therefore, the goal of the proposed method can be considered to be fulfilled, the proposed method has proven to be valid in both performed cases. However, from the two cases it became clear that a company does not always select the optimal configurator types to support the customisable product. A company often has a desire for additional configurator functionalities which result in a different evaluated configurator configuration from the selected configuration with the proposed method. For example, the single-purpose system contrary to the general-purpose system is selected by Blue-Lagoon, because the company wanted additional functionality of product rendering built in the configurator, with an fabricate-to-order strategy.

6.2 Selection process

The selection configurator types with non-conflicting and conflicting selection advice were, in both cases, almost entirely similar to the existing configurator configuration of configurator types. The differences between the selected and evaluated configurator types were either caused by company desires to pick a different configurator type, or the characteristics of the customisable product which were a mix of used selection criteria. For example, the customisable product of Blue-Lagoon used both characteristics of the assemble-to-order strategy and the fabricate-to-order strategy, since their components are both predesigned as adjustable to order.

The questionnaire of the proposed method is simply answered by analysing the customisable product of a company. The online or offline interaction nature is evident, since the customisable product is either offered on an online or offline platform. The business strategy can be determined quickly as well, through determining which type of components are used in the customisable product, i.e. predesigned. However, in the case of the Blue-Lagoon, products can have a mixed business strategy, with both predesigned and adjustable products. In addition, the use of a reconfigurator and a configurator organisation types are evident as well and are more a question of filtering an out of the ordinary configurator specification.

The customer prior product knowledge is easily determined through looking at the customer guidance and the degree in which a configurator automatically configures a configuration. Additionally, the product optimisation requirement can easily be recognised through searching for the possibility to list configurations based on the fulfilment of the set preferences.

Based on the results of the two cases, the defined selection of element searching types could be improved. It appeared that the element based searching type is selectable for companies with a fabricate-to-order strategy, instead of only in an assembly-to-order strategy. This difference is actually more logical, since the fabricate-to-order strategy uses adjustable products. Therefore customers are still composing their own product, instead of the feature based searching type, where the customer presents their requirements, for which the configurator searches for the optimal configuration.

The selection of the configurator reasoning type turned out to be highly influenceable by the weights determined with the AHP method. Therefore, the desires of a company for the customisable product are important in order to determine the specific weighting. As presented in *equation 26* and *28*, the ranking of the reasoning type differs greatly lot for the selection criteria weighting based on company desires.

6.3 Improvement

The selection of the reasoning type appeared to be highly influenceable by the weighting based on the desires of the company for the customisable product. In addition, the desires of a company often result in a different configurator type selection. Therefore, to improve the proposed method the method should determine the desires of a company for their customisable product functionality in a more guided process. By improving the determination of a company desires, less errors can occur in the weighing determination which will result in a more precise selection of the optimal reasoning configurator type. If the weighing determination is not done properly, ranking and selection of the optimal reasoning type will be incorrect. An incorrect reasoning selection will cause problems in the support of the storage, retrieval and processing of product variant information. Since, the reasoning type is the main component of a configurator, which contains the configurator knowledge and logic.

7 Conclusion

Configurators are used to support customisation of products, and utilise the processing of large product variant data. Companies are often inclined to develop a configurator based on their ideas & desires, without knowing if such configurator can be realised or if the realisation of their ideal configurator has any downsides. Configurator projects are often abandoned, because companies often lack the understanding of the functioning of configurator systems and select incompatible configurator types (Haug et al. 2012).

Unsuitable configurator type selections by companies, result in a troubled functioning of storage, retrieval and processing of product variant information. Configurator reasoning has the biggest influence on the configurator performance and thus in the storage, retrieval and processing of product variant data. The selection of the configurator reasoning type is often conflicting, since there are multiple performance characteristics which can be in accordance or in discordance with the desired customisable product characteristics.

The objective of this thesis is to develop a method that selects suitable configurator types for companies based on their specific customisable product, in order to provide an optimal configurator selection which supports the storage, retrieval and processing of product variant information. In order to select the ideal configurator types their selection should be based on their characteristics which have to be in accordance with the characteristics of the customisable product.

The proposed method fulfils the set objective since it selects configurator types based on their shared characteristics. The proposed method obtains input from a company through questions and uses the answers to select a set of configurator types which are used to select all other configurator types, except the reasoning types.

Companies can use the proposed method to configure a configurator configuration without requiring in-depth knowledge of the configurator types. By using the proposed method the optimal performance of the storage, retrieval and processing of product variant data can be achieved for a specific customisable product.

The reasoning type's selection advice can be conflicting, since it has multiple characteristics which can be both in accordance and discordance with the customisable product characteristics. In addition, multiple reasoning types have similar characteristics. In order to determine which reasoning type is more suitable, trade-offs have to be made to which characteristic is more important for a customisable product. In order to fulfil the set objective to select the ideal configurator, the method has to be able to select configurator types with conflicting selection advice.

The proposed method is capable of selecting the ideal configurator by using multi criteria selection for configurator reasoning types with conflicting selection advice. The proposed method uses the TOPSIS multi criteria decision making method to determine which reasoning type is optimal based to their Euclidean distance to the ideal characteristics that are in accordance with the customisable product. The AHP method is used to determine the weighing of the selection criteria which is needed for the TOPSIS method in order to determine which characteristic is of more importance for the customisable product.

To validate the proposed method two empirical case studies are performed. Based on the results the proposed method is judged to be valid and improvements are suggested to optimise the function of the method for configurator type selection. In conclusion, the proposed method is definitely a good indicator to see which configurator types are optimal for desired customisable product characteristics.

7.1 Future work

The proposed method is able to select optimal configurator types based on customisable product characteristics. However, companies have often diverging desires which cause to have a different configuration than the optimal configuration which is selected with the proposed method. For example, the configurator of Blue-Lagoon has a single-purpose system instead of general-purpose-system, because they want additional functionalities built in their configurator such as live rendering of the fabricate-to-order product.

The accuracy of the selection of the reasoning type which is most important for the configurator performance, is highly depending on the determined weighting based on company desires. Therefore, the determination of the company desires has to be performed with great precision, this process suggested to be further improved to prevent selection errors in the proposed method.

The ability of the proposed method to select alternatives for conflicting criteria can be applied in other real-life selection problems. The determination of selection criteria weighing of the AHP method and multi criteria selection performed the TOPSIS method are applicable to other selection problems. The combined added value of performing multi criteria selection with TOPSIS and AHP is quick and simple in operation. However, the combined use of the two MCDM methods is only suitable if the ranking does not have to be precise, and is performed properly to prevent errors.

References

- Abbasi, K. E., Hubaux, A., Acher, M., Boucher, Q., & Heymans, P. (2013). The anatomy of a sales configurator: An empirical study of 111 cases. In: *Salinesi C., Norrie M.C., Pastor Ó. (eds) Advanced Information Systems Engineering. CAiSE 2013, volume 7908*, 162–177.
- Blecker, T., Abdelkafi, N., Kreuter, G., & Friedrich, G. (2004). Product configuration systems: State-of-the-art, conceptualization and extensions. *Conference on Software Engineering and Artificial Intelligence (MCSEAI 2004)*, 25–36.
- Blessing, L. T. M. & Chakrabarti, A. (2009). *DRM, a Design Research Methodology*. Springer.
- Donini, F. M., Lenzerini, M., Nardi, D., Nutt, W., & Schaerf, A. (1998). An epistemic operator for description logics. *Artificial Intelligence*, 100(1-2), 225–274.
- Felfering, A., Friedrich, G., Jannach, D., & Zanker, M. (2001). Towards distributed configuration. *KI '01 Proceedings of the Joint German/Austrian Conference on AI: Advances in Artificial Intelligence*, 198–212.
- Forza, C. & Salvador, C. (2008). Application support to product variety management. *International Journal of Production Research*, 46(3), 817–836.
- Gavade, K. R. (2014). Multi-criteria decision making: An overview of different selection problems and methods. *(IJCSIT) International Journal of Computer Science and Information Technologies*, 5(4), 5643–5646.
- Greene, R., Devillers, R., Luther, J., & G. Eddy, B. (2011). Gis-based multiple-criteria decision analysis. *Geography Compass*, 5/6(6), 412–432.
- Haug, A., Hvamb, L., & Mortensen, H. N. (2012). Definition and evaluation of product configurator development strategies. *Computers in Industry*, 63, 471–481.
- Juengst, E. W. & Heinrich, M. (1998). Using resource balancing to configure modular systems. *IEEE Intelligent Systems and their Applications*, 13, 50–58.
- Kott, A., Agin, G., & Fawcett, D. (1992). Configuration tree solver: A technology for automated design and configuration. *Mechanical Design*, 114(1), 187–195.
- Kristjansdottira, K., Shafieea, S., Hvama, L., Forzab, C., & H. Mortensen, N. (2018). The main challenges for manufacturing companies in implementing and utilizing configurators. *Computers in Industry*, 100, 196–211.
- Macharis, C., Springael, J., De Brucker, K., & Verbeke, A. (2004). Promethee and ahp: The design of operational synergies in multicriteria analysis.: Strengthening promethee with ideas of ahp. *European Journal of Operational Research*, 153(2), 307–317.
- McGuinness, L. D. & Wright, R. J. (1998). Conceptual modelling for configuration: A description logic-based approach. 12, 333–344.
- Mittal, S. & Frayman, F. (1989). Towards a generic model of configuraton tasks. *IJCAI'89 Proceedings of the 11th international joint conference on Artificial intelligence*, 2, 1395–1401.
- Olson, D. L. (2001). Comparison of three multicriteria methods to predict known outcomes. *European Journal of Operational Research*, 130(3), 576–587.
- Opricovic, S. & Tzeng, G.-H. (2004). Compromise solution by mcdm methods: A comparative analysis of vikor and topsis. *European Journal of Operational Research*, 156(2), 445–455.
- Opricovic, S. & Tzeng, G.-H. (2007). Extended vikor method in comparison with outranking methods. *European Journal of Operational Research*, 178(2), 514–529.

- Roszkowska, E. (2011). Multi-criteria decision making models by applying the topsis method to crisp and interval data. *Computer Science*.
- Saaty, L. T. (2008). Decision making with the analytic hierarchy process. *International Journal of Services Sciences*, 1(1), 83–98.
- Saaty, R. (1987). The analytic hierarchy process—what it is and how it is used. *Mathematical Modelling*, 9(3-5), 161–176.
- Sabin, D. & Wiegel, R. (1998). Product configuration frameworks—a survey. *IEEE Intelligent Systems and their Applications*, 13, 42–49.
- Trentin, A., Perin, E., & Forza, C. (2014). Increasing the consumer-perceived benefits of a mass-customization experience through sales-configurator capabilities. *Computers in Industry*, 65, 693–705.
- Tsang, E. (1999). A glimpse of constraint satisfaction. *Artificial Intelligence*, 13(3), 215–227.
- Zavadskas, K. E., Turkis, Z., & Kildienė, S. (2014). State of art surveys of overviews on mcdm/madm methods. *Technological and economic development of economy*, 20(1), 165–179.