



HAL
open science

A knowledge representation framework for the design and the evaluation of a product variety

Antonio Giovannini

► **To cite this version:**

Antonio Giovannini. A knowledge representation framework for the design and the evaluation of a product variety. Computer Aided Engineering. Université de Lorraine, 2015. English. NNT: 2015LORR0002 . tel-01109715v2

HAL Id: tel-01109715

<https://hal.science/tel-01109715v2>

Submitted on 26 Jan 2015

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution - NonCommercial - NoDerivatives 4.0
International License

THESE

Présentée pour l'obtention du titre de
Docteur de l'Université de Lorraine
en Automatique, Traitement du Signal et des Images, Génie Informatique
par

Antonio Giovannini

A knowledge representation framework for the design and the evaluation of a product variety

Soutenue publiquement le 16/1/2015 devant le jury composé de :

<i>PRESIDENT :</i>	C. SALINESI	PROFESSEUR A L'UNIVERSITE PARIS 1 PANTHEON-SORBONNE
<i>RAPPORTEURS :</i>	M. ALDANONDO	PROFESSEUR A L'ECOLE DES MINES D'ALBI-CARMAUX
	M. TRAJANOVIC	PROFESSEUR A UNIVERSITY OF NIS, SERBIE
<i>EXAMINATEURS :</i>	M. DASSISTI	PROFESSEUR A POLITECNICO DI BARI, ITALIE
	L. PIERREL	INGENIEUR, TRANE, GROUPE INGERSOLL RAND
	H. PANETTO	PROFESSEUR A L'UNIVERSITE DE LORRAINE (DIRECTEUR DE LA THESE)
	A. AUBRY	MAITRE DE CONFERENCES A L'UNIVERSITE DE LORRAINE (CO-DIRECTEUR)
	H. BRIL-EL HAOUZI	MAITRE DE CONFERENCES A L'UNIVERSITE DE LORRAINE (CO-ENCADRANTE)



anything said is said BY an observer

Maturana, H.R., Varela, F.J., 1980. Cognitive Function in General, in: Autopoiesis and Cognition, Boston Studies in the Philosophy of Science. Springer Netherlands.

anything said is said TO an observer

Von Foerster, H., 2003. Understanding Understanding: Essays on Cybernetics and Cognition. Springer.

the only thing that meaning and intuition have in common is their obscurity

Chomsky, N., 1975. The Logical Structure of Linguistic Theory. Springer US.

l'ingegno, che è stato dato all'uomo per conoscere, ossia per fare

Vico, G., 2005. De antiquissima Italorum sapientia: con traduzione italiana a fronte. Ed. di Storia e Letteratura.

interpréter l'objet non pas en lui-même, mais par son comportement

Le Moigne, J.-L., 1994. La théorie du système général: théorie de la modélisation. Presses Universitaires de France-PUF.

*la structure (c'est-à-dire l'ordre des parties) et la fonction (l'ordre des processus) sont
peut-être la même chose.*

Von Bertalanffy, L., 1956. General systems theory.

*la connaissance consiste essentiellement, en effet, non pas seulement à acquérir et à
accumuler des informations, mais encore et surtout [...] à les organiser et à les régler
par des systèmes d'autocontrôles orientés vers des adaptations, autrement dit vers
la solution de problèmes.*

Piaget, J., Buscaglia, M., 1992. Biologie et connaissance. Delachaux et Niestlé.

Acknowledgements

I would like to thank all the people who were more or less casually involved in the path to the construction of this work. All the discussions, debates and disputes I had during these years made this thesis and myself richer.

I would like to spend some more words to thank

- Hervé Panetto, Alexis Aubry and Hind El Haouzi for the opportunity they gave me and for the precious support during these years at CRAN and Trane and, generally, during my experience in France;
- Ludovic Pierrel to have accepted to replace my former enterprise tutor and to have guided me at Trane during this adventure;
- Professor Michele Dassisti for the inspiring discussions had during these years;
- Professors Aldanondo and Trajanovic, who accepted to be the official reviewers of the thesis;
- Camille Salinesi to have accepted to be part of the jury;
- all the colleagues and the members of CRAN for the interesting discussions,
 - special thanks to Gerard Morel for the interesting remarks about system theory,
 - special “sorry” to all the PhD and master students that I *bored* with my long remarks and argumentations;
- all the colleagues that I had the luck to meet at Trane during my stay in France,
 - special thanks to Sebastien Estublier for the essential help provided for the case study,
 - as above, special “sorry” to all.

I would also give my special attention to Professor Didier Wolf, who welcomed me during my stay in France, at CRAN facilities.

Sorry to the ones I forgot.

Contents

CONTENTS	6
CHAPTER 1 INTRODUCTION	9
1 - PREFACE.....	9
2 - GENERAL INTRODUCTION	9
3 - INTRODUCTION TO THE PROBLEM STATEMENT.....	12
4 - ANALYSIS OF THE INDUSTRIAL ISSUES: THE TRANE SCENARIO.....	12
5 - CONTEXT AND RESEARCH QUESTIONS: PV ALTERNATIVES GENERATION AND EVALUATION.....	21
6 - CONCLUSIONS.....	24
CHAPTER 2 A DESIGN KNOWLEDGE REPRESENTATION FRAMEWORK FOR PRODUCT VARIETY (PV) ALTERNATIVES GENERATION	27
1 - LITERATURE ABOUT PV DESIGN APPROACHES: LINKING THE VARIETY OF TWO DOMAINS	28
1.1 - <i>Linking Real Customers with Customer Profiles</i>	30
1.2 - <i>Linking Customer Profiles with Product Variants</i>	33
1.3 - <i>Linking Product Variants with Process Variants</i>	35
1.4 - <i>The gap in the literature: the lack of an unambiguous knowledge representation</i>	36
2 - DESIGN KNOWLEDGE REPRESENTATION	39
2.1 - <i>Schools of Knowledge Representation</i>	39
2.2 - <i>Anti-logicism, logicism and the ambiguity of the representation</i>	40
3 - OUR APPROACH: DESIGN KR FRAMEWORK FOR PV ALTERNATIVES GENERATION	43
3.1 - <i>Design KR framework</i>	43
3.1.1 - Conceptual model.....	43
3.1.2 - A short instantiation of the experience concept.....	45
3.1.3 - How the experience instances can be retrieved and connected.....	48
3.1.4 - An algorithm to prove the non-ambiguity of the representation.....	49
3.2 - <i>From design KR to PV design: the generation of the PV alternatives</i>	50
3.2.1 - The discretisation of the domains.....	51
3.2.2 - Testing the customer-process couples feasibility	52
3.3 - <i>Expected effects on the question RQ1 and on the industrial issues</i>	54
4 - CONCLUSIONS.....	55
CHAPTER 3 EVALUATING THE PV ALTERNATIVES	56
1 - OVERVIEW ON PV EVALUATION CRITERIA.....	57
1.1 - <i>Index based on the customer requirements</i>	58
1.2 - <i>Indexes based on the product features</i>	58
1.3 - <i>Indexes and methods based on the process features</i>	61

1.4 - <i>PV design with index implementations</i>	63
1.5 - <i>The gap in the literature: the relation variety/cost</i>	64
2 - THE IMPACT OF THE VARIETY ON THE MANUFACTURING COSTS: <i>LOT STREAMING</i>	66
3 - OUR APPROACH: PV ALTERNATIVES EVALUATION	68
3.1 - <i>About the relation between the variety and the costs</i>	68
3.2 - <i>An approximation of the waiting time</i>	72
3.3 - <i>The implementation of the new criterion for the PV evaluation</i>	74
4 - CONCLUSIONS.....	76
CHAPTER 4 CASE STUDY: WATER COIL ASSEMBLY	77
1 - THE MULTI-DOMAIN KR FRAMEWORK: A CASE STUDY	77
1.1 - <i>The water coil knowledge represented in CATIA V5</i>	78
1.2 - <i>The design of the water coil variety</i>	82
1.3 - <i>The unambiguity of the water coil knowledge</i>	88
1.3.1 - The two experiences represent the same measurements.....	88
1.3.2 - The two experiences shares the representation of some measurements.....	89
2 - WATER COIL VARIETY ALTERNATIVES: GENERATION AND EVALUATION.....	90
2.1 - <i>The step one: the structure of the data file and the constraints in Lingo14</i>	91
2.2 - <i>The step two: the interface with Lingo</i>	93
2.3 - <i>The step three: the generation of the feasibility report file</i>	95
2.4 - <i>The step four: the optimisation model generation</i>	96
2.5 - <i>The two last steps: the solution of the optimisation model</i>	97
2.6 - <i>Modification of the models to reduce the execution time of the solver</i>	98
3 - CONCLUSIONS.....	99
CONCLUSIONS: LIMITS AND PERSPECTIVES OF THE PROPOSALS.....	101
1 - GENERAL CONCLUSIONS	101
2 - RESEARCH CONCLUSIONS.....	102
2.1 - <i>First research question (RQ1)</i>	102
2.1.1 - Limits of the proposals	103
2.2 - <i>Second research question (RQ2)</i>	103
2.2.1 - Limits of the proposals	104
2.3 - <i>Perspectives and possible improvements</i>	104
3 - INDUSTRIAL CONCLUSIONS	105
3.1 - <i>First industrial issue (II1)</i>	105
3.2 - <i>Second industrial issue (II2)</i>	105
3.3 - <i>Third industrial issue (II3)</i>	106
3.4 - <i>Limits and possible improvements</i>	107

LIST OF ACRONYMS..... 108

LIST OF FIGURES 109

LIST OF TABLES..... 111

PERSONAL REFERENCES..... 112

REFERENCES..... 113

ANNEX 1: LINGO SYNTAX 120

ANNEX 2: ASSEMBLY SYSTEM FOR THE ANALYSIS OF THE VARIETY/COST RELATION .. 122

ANNEX 3: WATER COIL VARIETY GENERATION MODEL..... 131

ANNEX 4: WATER COIL PV OPTIMISATION MODEL..... 141

Chapter 1 Introduction

1 - Preface

This PhD thesis is the result of the **collaboration between the Trane company and the Research Centre for Automatic Control (CRAN)**, Joint Research Unit of the University of Lorraine and the CNRS. This thesis is included in the CIFRE (Convention Industrielle de Formation par la REcherche) framework, i.e. the work is co-founded by Trane.

During the three years of works on this thesis, the **thesis-related Trane context** has been characterized by a project about the development of a new customizable product. This product was designed by an engineering team in the south of Europe and had to be manufactured in more than one Trane manufacturing plant, not only in Europe. The project represented two main challenges for Trane: the **product knowledge transfer** and the **customer and manufacturing heterogeneity harmonization**. Actually, these two project challenges became the motivation of the enterprise-university collaboration and so the origin of this thesis.

2 - General introduction

Nowadays, in more and more market segments, **high demanded volumes** (*mass-*) and a **large variety of requirements** (*-customisation*) make the *engineer-to-order* approach and *make-to-stock* approach ineffective.

Indeed, the *engineer-to-order* approach is usually adopted when the customers have highly **specific requirements**. Therefore, at each order entry, new requirements are formalized and a new engineering stage should be performed. This strategy is plausible when **low volumes** of the products (i.e. of different products) are required, i.e. inversely, too many engineering stages should be performed.

Instead, the *make-to-stock* approach is usually adopted when the customer **requirements are homogeneous**. Therefore, at each order entry, the customers specify only the volume of the product they need. On the engineering side, all the work is already performed. Indeed, the products are expected to be already in stock before the order entry takes place. This strategy naturally fit when **high volumes** and **low variety** of the products are demanded.

Therefore, none of the above approaches can fit the *mass-customisation* scenario: i.e. high volumes and high customer requirement variety. Indeed, when the target of an enterprise is a customer segment with these characteristics, a *make-to-order* strategy is naturally adopted: the enterprise propose to the customer a set of already engineered products that, at the order entry, should be only manufactured. In order to adopt this strategy, enterprises need a different approach to the product design: the **design for variety** (ElMaraghy et al. 2013). By designing a set of products, namely a product variety (PV), in a single design stage, the enterprises can guarantee to the customers a good level of customisation and an appropriate responsiveness to high demanded volumes.

This thesis highlights a fundamental need in the approaches for the design of a PV:

*the need of a method to design the variety (i.e. the set) and the products variants (i.e. the members of the set) **at the same time.***

This point is essential when the target of the enterprise is a customer set expressing a high degree of requirement variety. Indeed, in this case, it is not possible to design individually the variants before starting with the variety design stage, i.e. as for *engineer-to-order*, too many engineering stage should be required. Unfortunately, the individual design is the main requirement of all the reviewed methods for the PV design. And, in the reviewed methods, this is due to a lack of design knowledge representation, required to connect in an unambiguous way 1) the customer requirements to the product that should meet its requirements and 2) the products to the manufacturing process that is able to manufacture them. The 1) ambiguity of the customer profiles definition and 2) the lack of design knowledge hinder the current methods in literature to design a PV without an initial definition of the individual variants.

In order to cope with this lack, we propose, in this work, an unambiguous knowledge representation framework, a method to deploy it and a criterion to design and evaluate PV alternatives.

The dissertation will follow the structure presented in Fig. 1.1. In the first chapter (i.e. the following sections of the current one), the problem statement is discussed. In the second and third chapter, the research questions are treated. Both the chapters have the same structure: 1) literature to find existing answers for the addressed research question and identification of a gap in the literature; 2) literature to build our proposal; 3) our proposal. In the fourth chapter, a case study is presented. The detailed structure is the following.

- In this **first chapter**, the industrial issues (IIs in Fig. 1.1) are discussed. For each issue, the enterprise scenario is modelled and an analysis is performed to identify the context of the problems, i.e. the PV design. Starting from the analysis of the issues, in the second part of the chapter, the research context is detailed. The definitions of *mass customization* and so of the *design for variety* allow to deal with the issues from a research viewpoint. From this contextualization, the two research questions are identified: the first requiring a variety model including the knowledge from the customer, product and process domains; the second requiring a criterion to evaluate each PV alternative generated with the variety model.
- In the **second chapter**, the first research question is treated. An overview on works about the PV design is provided: the focus is on how the methods in literature allow to model the link between 1) the real customers and the customer profiles, 2) the customer profiles and the product variants and 3) the product variants and the process variants. An analysis of this literature allows to identify a gap about the lack of *unambiguous design knowledge representation (KR)* in the method for the PV design. This lack does not allow to provide an answer to the first research question. After an overview on works about the knowledge representation, by transposing the principles of the **anti-logicist** KR approach, our **framework**

for design KR is presented. In the last part of the chapter, the *application of this KR framework* on the identified PV design lack is showed. The result of the application is a set of designed PV alternatives.

- In the *third chapter*, the first research question is treated. An overview on works about the PV design is provided: in this chapter, the focus is on how the methods in literature allow the evaluation of PV alternatives. The review of the research papers shows also in this case a gap: *no-methods propose a criterion to take into account the manufacturing costs other than the direct one*. As shown by studying *the nature of the variety/cost relation*, this lack can lead to wrong evaluations. A *lot-streaming*-based mathematical model allows to analyse this relation. On the basis of the result of this analysis, we propose *a PV evaluation criterion* that considers the effect of the process variety on the manufacturing costs.
- In the *fourth chapter*, the approaches to answer the two identified research questions RQ1 and RQ2 are applied on the industrial issues: a case study is presented. The formalisation of the knowledge relative to a *water coil* variety is shown. The instantiation of our framework is performed on a CAD software. Then a simulation of knowledge retrieval for a PV design stage is described. After a demonstration of the unambiguity of the KR, the application of the formalised design knowledge for the generation of PV alternatives is shown. The developed prototype is based on a Java tool that is interfaced with a mathematical solver to provide the PV alternatives and compare them to define the best PV.

At the end of the dissertation, a conclusion sums up the main points of the work. The impacts and the limits of the proposed solutions on the industrial issues and research questions are discussed. Moreover a discussion is presented on the perspectives about the research community and about the industrial application of the proposed approaches.

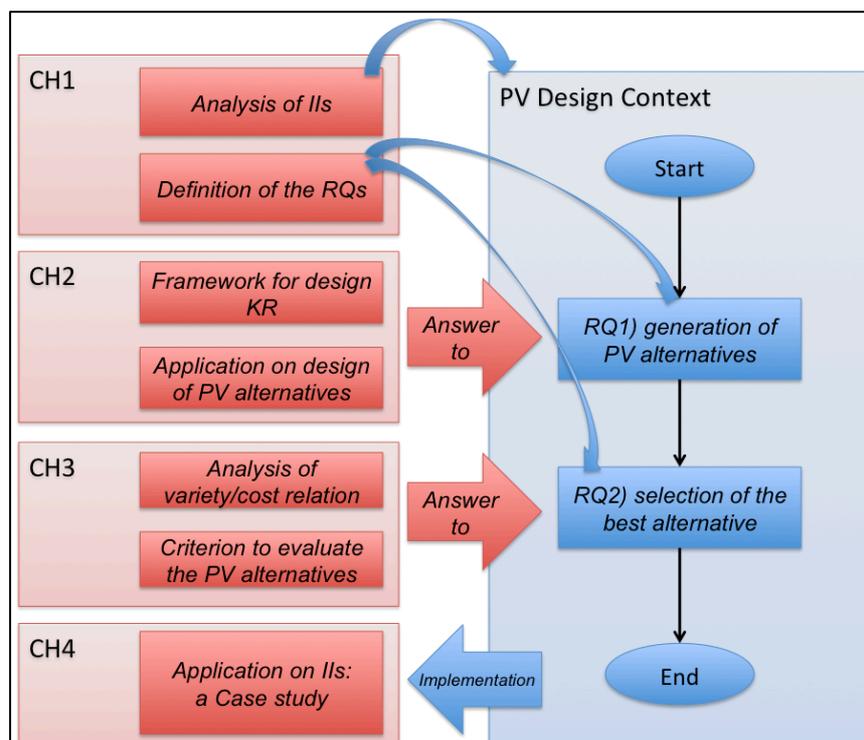


Fig. 1.1 - structure of the thesis.

3 - Introduction to the problem statement

In the reminder of this first chapter of this dissertation, the motivation of the work is discussed. The structure of the chapter is the following (Fig. 1.2):

- the section 4 is dedicated to the analysis of the industrial issues (IIs): the enterprise is briefly introduced; an overview on the Trane products and services is given; then three industrial issues are discussed;
- the aim of the section 5 is to identify the research context in which these issues are treated and then to identify the research questions (RQs) from these industrial concerns.

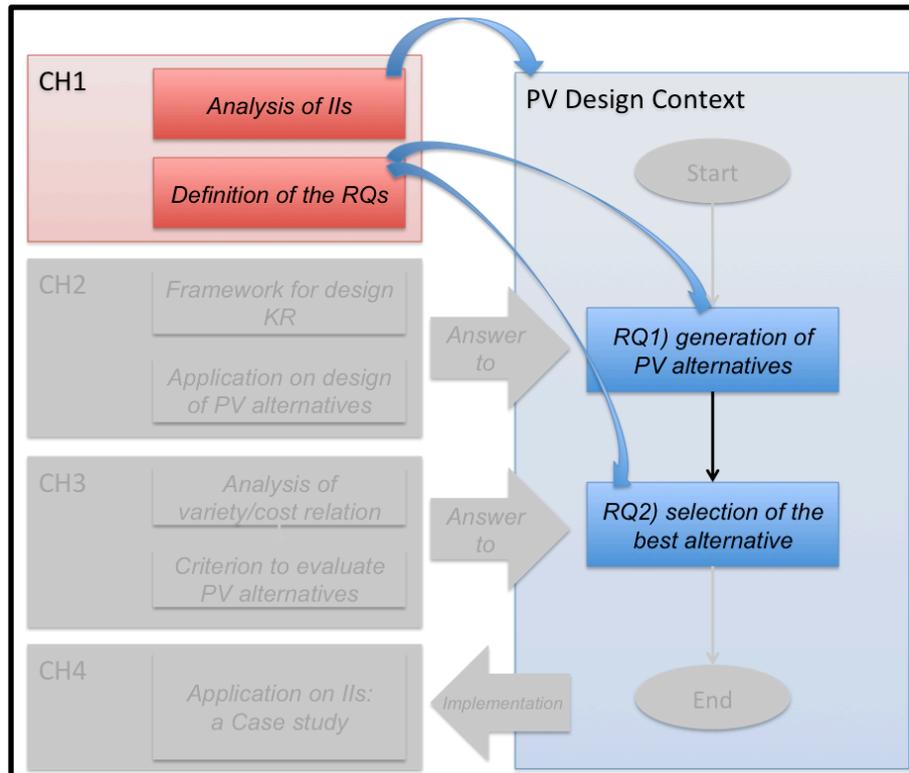


Fig. 1.2 - role of the chapter in the thesis structure.

4 - Analysis of the industrial issues: the Trane scenario

Trane is an industrial company that designs, manufactures and commercialises Heating, Ventilation and Air Conditioning (HVAC) systems and provides HVAC related services. Currently Trane is a business of Ingersoll Rand. In the commercial segment (i.e. hotels, universities, hospitals, restaurants and so on), Trane revenue is about \$ 5-6 billions. The global presence of the company is organised in 4 geographic areas (Fig. 1.3): North America, Latin America, EMEIA (Europe, Middle East, India and Africa), and Asia Pacific. In these areas, Trane has more than 20 manufacturing locations, 250 service locations and about 400 business locations in more than 100 countries.

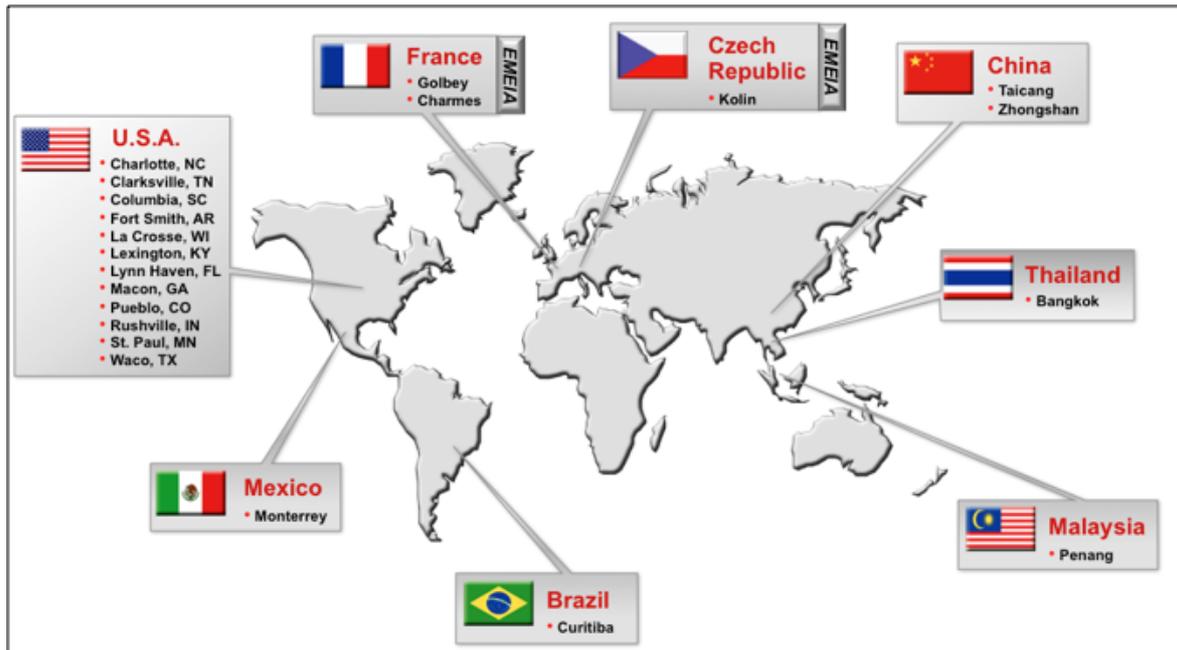


Fig. 1.3 - Trane manufacturing plants over the world.

Trane **products and services** can be grouped in 6 main categories (Fig. 1.4):

- *chillers*: these machines are used to remove heat from the water; the removed heat can be released to the air (i.e. air-cooled chillers) or to the water (i.e. water-cooled chillers); some convertible machines can also heat the water; the range of power goes from 13 kW to 14 MW; *chillers* can be equipped with scroll, screw or centrifugal compressors;
- *airside*: the machines in this category use the water coming from the *chillers* to treat the air that has to be distributed into the building; *airside* machines humidifies, dehumidifies, cools, heats and filters the air; larger machines (e.g. *AHUs* - *Air-Handling Units*) are for centralised control and terminal units (e.g. *fan coils*) are for distributed control of the airflow properties; Trane *AHUs* can handle airflows from 1000 to 250.000 m³/h;
- *unitary*: machines in this group merge the *chillers* and the *AHUs* functions in one single product; anyway, these machines provide lower power than *chillers* (i.e. the biggest provides up to 165 kW) and less refined air treatment option compared to the *airside* units;
- *controls*: this category includes all HVAC control systems for single units (e.g. *fan coils*, air-handling units) and the software for the centralised control of all building systems;
- *services and parts* and *energy solutions*: Trane provides services such as assistance for guaranteed performances, part supply, predictive maintenance, compressor restoring, remote system monitoring, etc.

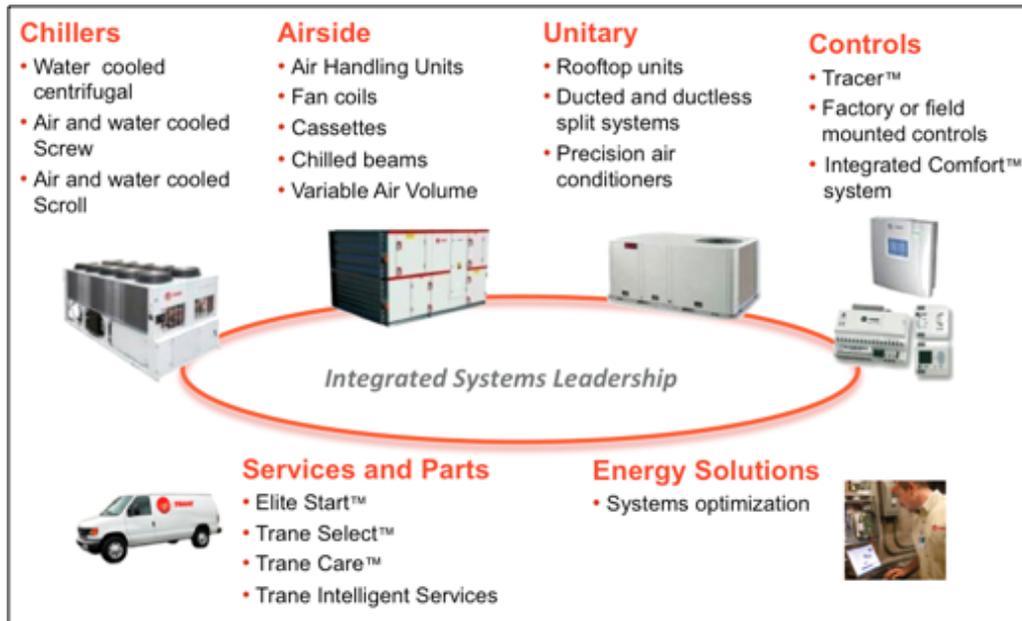


Fig. 1.4 - Trane products and services

From the largest chiller to the smallest fan-coil, each one of the Trane products takes part to the air-handling system of a building. In this thesis, the water coil family representing one of the fan coil unit components (airside category) will be the core of most examples and especially the object of the test case in the application chapter. The main function of a water coil is to transfer the heat from the water to the air and/or vice versa. A further description of the coils is presented later in this dissertation. The choice of the **water coil** for the examples is due to three reasons:

1. the simple physical structure of the *water coil* makes easier and more understandable the presented examples; this feature does not impact the size of the treated case study; however, an extensive discussion on the size problem is presented at the end of the fourth chapter, when the impacts of the proposals are discussed;
2. the component is anyway complex; indeed, as in (p. 19 in (Von Bertalanffy 1956)), the complexity is due to the intensity and the non-linearity of the interactions between the system features;
3. the *water coil* represents a good compromise between the complexity of the component and its versatility; indeed, the water coil is used wherever a water-air heat transfer is needed, e.g. air-handling units.

The versatility of most of Trane products components is required by the commercial market that represents the Trane target. Indeed, **Trane customers have highly specific requirements**. Usually enterprises cope with this requirement variety by deploying an *Engineer-To-Order* (ETO) strategy, i.e. engineers design a brand new product at each order entry. Sectors that typically adopt this strategy are characterized by complex (and expensive) products and low volumes. Because of the required high volumes, the ETO cannot completely fit the Trane required responsiveness.

This high variety/high volumes scenario impacts the Trane design process. Indeed, the usual design stage at Trane results in a set of products with certain similarities that are addressed to a given set of customers. This *Product Variety* (PV) is a sort of menu or catalogue in which customers with different

requirements should find the product (i.e. the *product variant*) that meets their expectations. Selecting a variant in the variety, the customer is ordering an already engineered product. In this way the PV allows Trane to avoid an engineering stage for each new customer (i.e. for the customer considered during the PV design).

The customer variety and the consequent PV design impact the **order entry** process. Indeed, during each order entry, the customer has to express his specific needs (for the scope of this dissertation, the eventual differences between the meanings of the terms “requirement” and “need” are not influent) and a product to meet this needs has to be selected (Fig. 1.5).

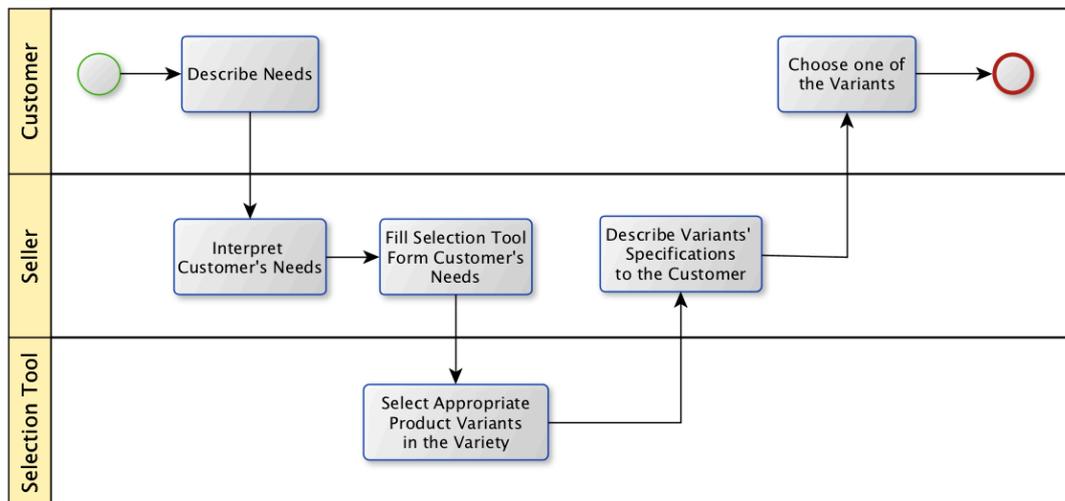


Fig. 1.5 - Trane product selection process on BPMN

The main stages of this selection process are the following.

- A customer describes his requirements and a seller interprets these requirements in order to link the customer with a customer profile described by a software selection tool. In each selection process, this interpretation provides the link between a real customer (set A in Fig. 1.6) and the customer profiles (set B in Fig. 1.6). This link is expected to be formal, i.e. each customer has a correspondent customer profile. Instead, in the Trane process, an ambiguity during this interpretation can happen: **two sellers who interpret the needs of the same customer can link his needs with two different customer profiles.**
- Once the profile has been defined, the selection tool (in Fig. 1.7 a screenshot of the tool for the fan coils selection is shown) performs a research of the “right” product that can meet the customer profile requirements. The task performed by the selection tool provides the links between the customer profiles (set B in Fig. 1.6) and the Trane products (set C in Fig. 1.6). The term “right” is used because, once the customer profile is defined, the profile is expected to be linked with the best product (economic considerations included) the company can provide to meet the profile-related needs. At Trane, **the selection tool can propose more than one solution for the customer without specifying which is the best one.**

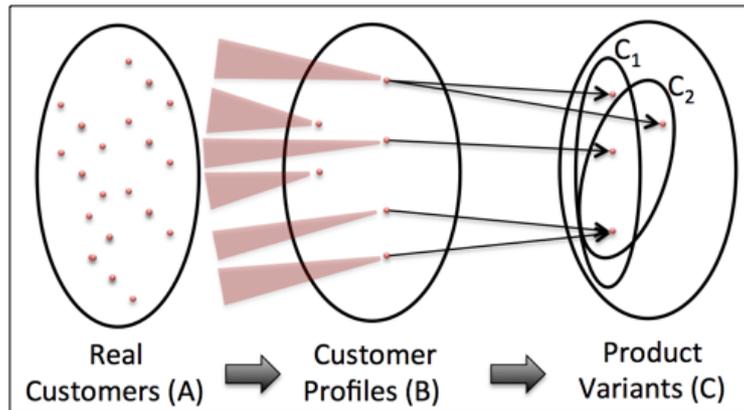


Fig. 1.6 - Trane product selection: building links from the real customers to the product variants.

For the sake of example, let us consider the selection of the appropriate *fan coil* to meet the customer's needs. In Fig. 1.7, the selection tool for the *fan coil* variety is shown. The most complex part to be selected is the *water coil*. In order to make the tool performing a research among the *fan coil* variants, the seller has to characterise the customer profile as room temperature, humidity, water temperatures and so on. The tool produces several resulting *coils*. Then the customer should choose one of them according price, size, weight, noise level, power and other performances.

By focusing on the Trane product selection process, the first industrial issue (II1) can be described as follows:

II1 - *If a customer describes its requirements to more than one seller then more than one different resulting product can be proposed to the customer (for the same needs). Are the proposed product variants equally good for the customer? Is there a best alternative for the customer? Is there a best alternative for Trane? And so, given a set of customers, how to design a PV that represents the best way to cover the customers' needs, i.e. the best subset C in Fig. 1.6?*

Solving this industrial issue, for the simple instantiation of figure 1.6, means choosing between C_1 and C_2 .

As shown in Fig. 1.3, Trane is present in four continents with its manufacturing plants. Often, the same customer profiles can be defined as targets for **more than one manufacturing plant**. This represents a **second source of variety**, this time on the manufacturing side. Indeed, the same customer profile can be linked with a product manufactured in two manufacturing side with their own manufacturing processes (e.g. drills or punching machines to manufacture a hole in a sheet metal) and constraints (e.g. cost and availability of raw materials).

During past years **two strategies** were adopted to deal with this manufacturing heterogeneity:

1. a *global design strategy* (Fig. 1.8): imposing a common design for each plant except for local legal and manufacturing feasibility constraints, e.g. regulation and norms on electric equipment, punching tool availability, local machine capacities and so on;

2. a local design strategy (Fig. 1.9): engineers manage locally product features (e.g. coil sizes) and manufacturing process features (e.g. punching tools); product and process engineers perform a sort of improvement loop until both agree on the PV design.

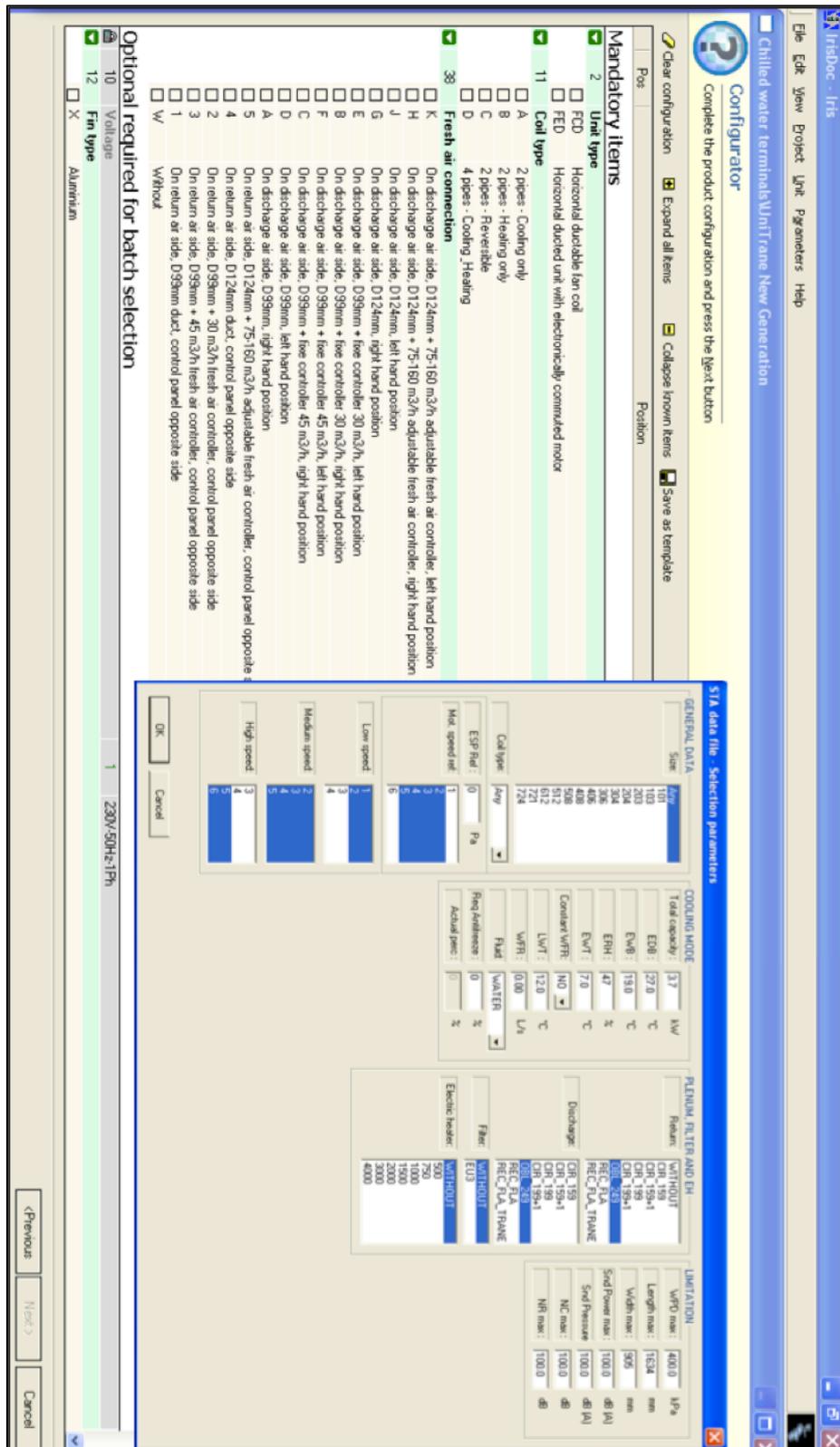


Fig. 1.7 - Trane product selection: selection tool screenshot

The aim of each one of these strategies is the harmonization of the manufacturing process heterogeneity with the customer profiles homogeneity. Starting from the experiences on the past PV design projects the following limitations can be highlighted.

Concerning the **first strategy**, on one hand it minimizes the product engineering effort but on the other hand **it reduces the degrees of freedom of the process engineers by imposing a common product design**. Indeed, process engineers should propose product changes that 1) optimise the usage of the local plant features (e.g. equipment, resources) and 2) preserve the satisfaction of the customer requirements. Unfortunately, this first strategy cannot take into account these optimisation opportunities because the product features, even if not directly impacting the customer requirements, have to be the same for each manufacturing plant.

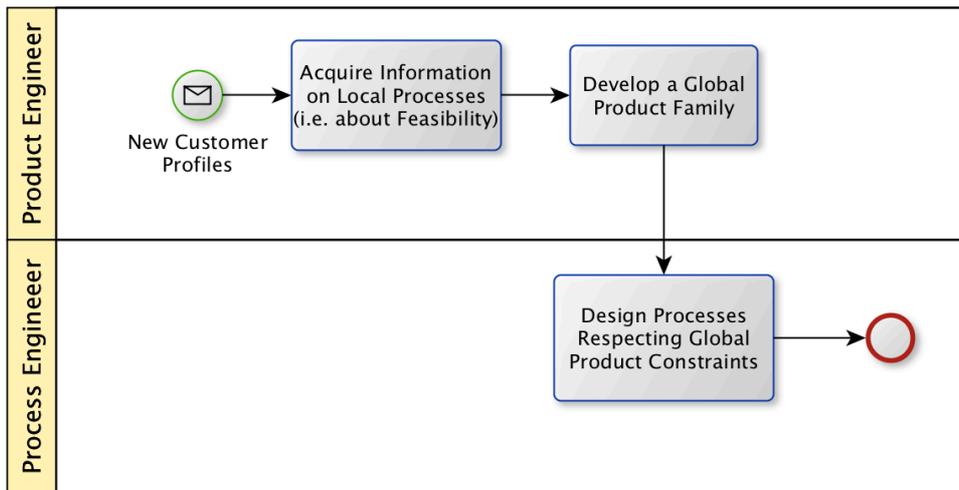


Fig. 1.8 –Trane design and development process according global strategy 1.

The aim of the **second strategy** is to exploit all these optimization opportunities and so to let the process engineers design an optimal use of the plant resources. In this case, the same product can have different features that depend on the plant in which the product has to be manufactured. The manufacturing processes in each plant can be optimal because the constraints on the production are only related to the required product performances to meet the customer requirements, i.e. no constraints about the way to get these performances. On the other hand, in this case **the reduction of the product design efforts is not considered**, i.e. product designers have to work separately for each manufacturing plant. This means that different design stages to cope with the same customer requirements are performed.

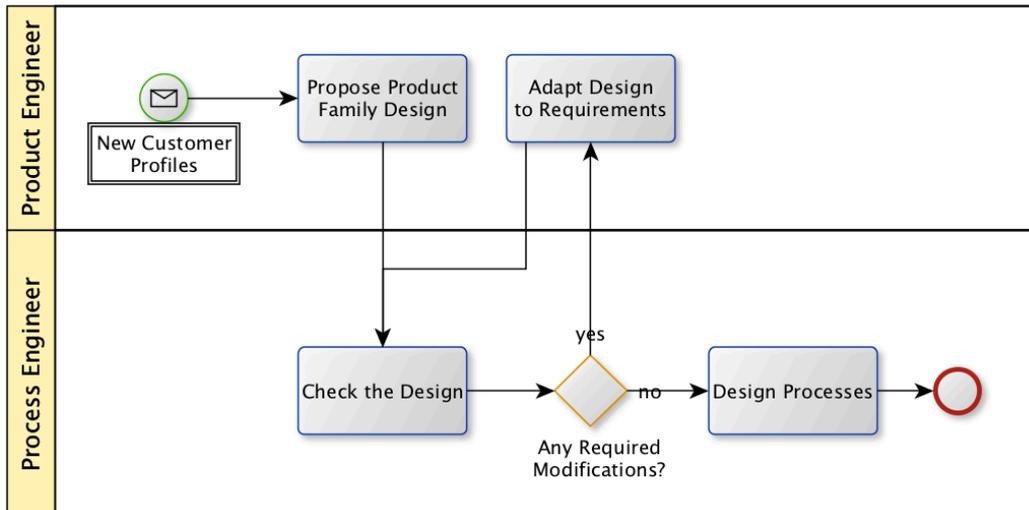


Fig. 1.9 –Trane design and development process according strategy 2.

To sum up, both strategies tries to find a compromise between the minimization of the PV design efforts and the maximization of the process engineer degrees of freedom. The issues related with both the strategies can be summarized with the following industrial issue:

II2 - Is it possible to conjugate design effort and local manufacturing cost reduction? In other words, how to link the product variety C and process variety D (Fig. 1.10) in order to impose to D, only the constraints that guarantee the satisfaction of the customer requirements? And consequently, how to find the best process variety, i.e., the best subset of D coming from different manufacturing plants?

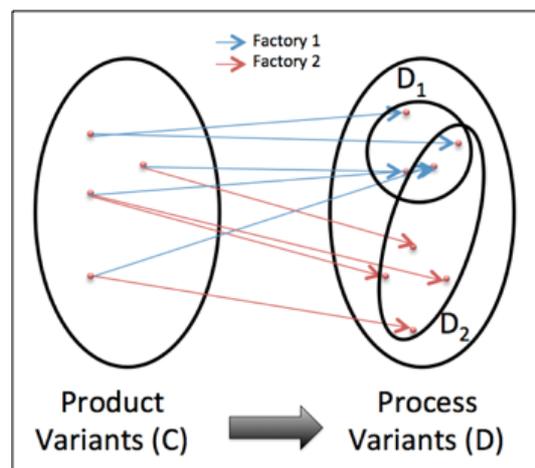


Fig. 1.10 - Trane design process: building links from the product variants to the process variants

Solving this industrial issue, for the simple instantiation of figure 1.10, means choosing between C_1 and C_2 .

With its PV design stage, Trane addresses a really wide market segments. The definition of these segments is done on the basis of the forecasted customer's demand for each profile. The same definition determines the only target for the PV design. Therefore, even if many efforts are done to

fulfil a large set of customers, there are some customers with requirements that were not considered in the PV design stage, i.e. requirements related to profiles with an expected low demand.

In most sectors, e.g. automotive, the customer can only select one of the proposed product variants in the PV. No-further customisations are allowed after the end of the PV design stage. About this point, Trane represents a particular case, also in its own sector. If, during the selection process (Fig. 1.5), the tool performs a research without finding any variants meeting the customer's needs, Trane accepts anyway the order. This acceptance is temporary. Indeed, the feasibility verifications, the design and the manufacturing cost estimations have to be performed.

This situation represents a **special order** (Fig. 1.11), i.e. the customer does not find a product fulfilling his needs in the already designed PV. In some cases, especially for smaller required modifications (e.g. particular room humidity and coil water conditions), the seller has enough experience to formalise these new needs and send them to the engineer team. For non-common request (e.g. a different shape for the *fan coil* case), the customer can discuss its requirements directly with the Trane engineers.

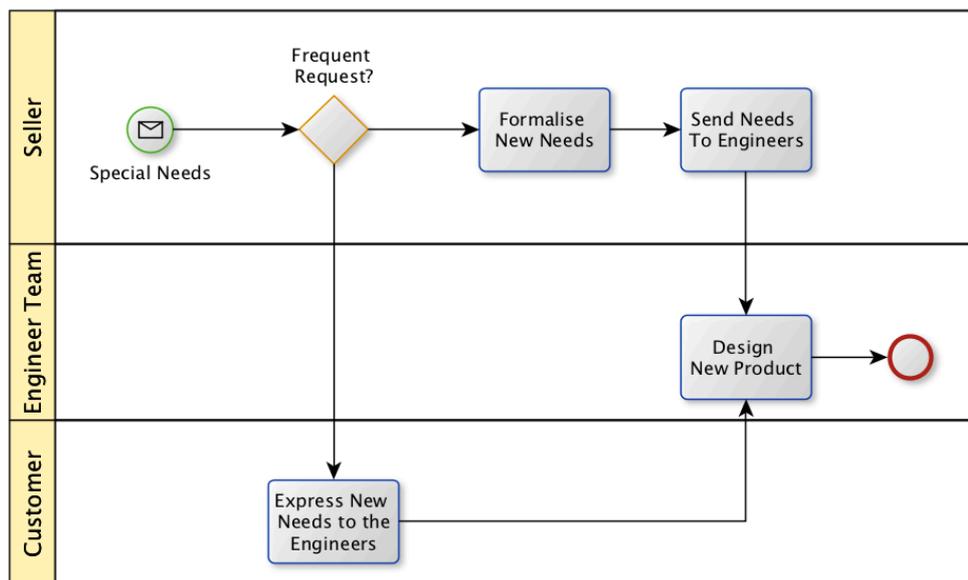


Fig. 1.11 – Trane order entry for special orders

By accepting the order, a new design stage is performed. The aim of this process is to generate a new product variant that fulfils the special requirements of the customer. During this phase, engineers stress on the reuse of the work performed during the PV design stage. All this new design stage strives for the maximization of the commonalities between the new required product variant and the ones in the designed PV. This design knowledge reuse is relative to both the product and the manufacturing process features. But, at Trane, **even if the same knowledge used during the PV design is also needed for the design of the special variant, a new design stage is performed.**

A third industrial issue can be defined to summarize the concerns about the special order process:

I13 - At Trane, the special design stages can represent 20% of the orders of certain product families. What is the right way to link the main design stage (i.e. related to the PV) with the secondary ones (i.e. for special orders)? Can a PV model adapt (Fig. 1.12) the product offer to the new customer requirements? If yes, how to model it?

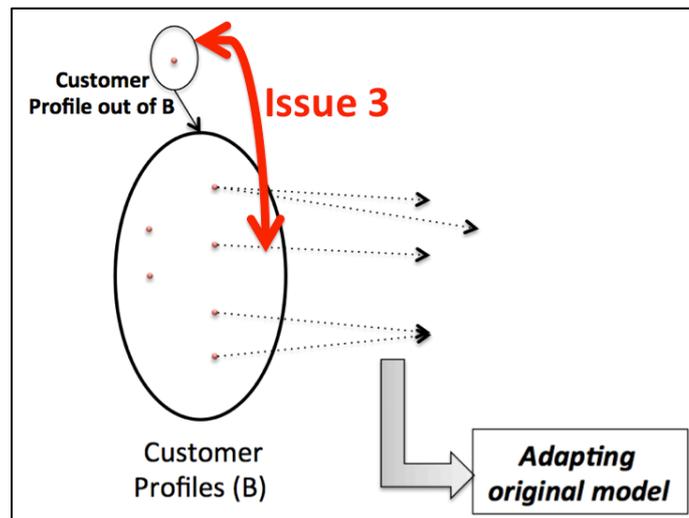


Fig. 1.12 – Trane special order issue: stress on the knowledge reuse

To sum up, from the order entry process until the design stage, in this section, an overview and an analysis of how Trane cope with the variety has been presented. Three points have emerged. These issues represent improvable points of the Trane above discussed processes. In the next section, a contextualization of these issues in a research field is performed. This should allow to recognize the research field in which solutions that fit the Trane requirements can be retrieved.

5 - Context and research questions: PV alternatives generation and evaluation

What if technology made it possible for every one of the five thousand shirts to be customized while on the assembly line [...] produced just as quickly as the five thousand identical shirts, yet at not greater expense? Starting from these words Davis in (S. M. Davis 1987) introduced for the first time the concept of Mass Customisation (MC) for a shirt manufacturer. The two-sided nature of the MC concept can be better identified in (Tseng and Jiao 2001) or (Blecker et al. 2004): ideally MC is a business approach that want to fulfil individual customer's needs having an organizational efficiency of a mass production company. Trane, as discussed in the first industrial issue, has high volumes (mass-) and a variety of customer needs to be met (-customisation). Therefore the Trane business can be considered in the MC context.

In order to follow the MC ambitions, an enterprise should be able both to provide a high level of product customization and to react with a reasonable time-to-market to cope with the high required

volumes. Neither an engineer-to-order (ETO) approach nor a make-to-stock (MTS) approach can deal with this double requirement. The former approach can guarantee good levels of customisation but also leads to a large increase of the time-to-market. The latter approach can guarantee excellent time-to-market but also leads to inappropriate levels of customization. Therefore, the right approach is at the middle, i.e. make-to-order (MTO): enterprises design a variety of products and at the order entry only the manufacturing time impacts the time-to-market. In other words, the big deal is 1) to compress the duration of the engineering stage at each new customer order and 2) to propose to the customers a good level of customisation. As at Trane (see the section 1.1), one of the most common solutions is to design a Product Variety (PV) in place of lots of single products.

In the literature the “PV” concept can be associated to different terms, e.g. product family (Hong et al. 2008), configurable product (Aldanondo and Vareilles 2008) or product line (Pohl, Bockle, and Van Der Linden 2005; Mazo et al. 2012). The small differences between these terms are out of the scope of this work. Therefore, in the remainder of this thesis, we will use the following definition given by (ElMaraghy et al. 2013) because it is the most general one.

*A **variety** and a **variant** are respectively defined as a number or collection of different things of a particular class of the same general kind and an instance of a class that exhibits usually slight differences from the common type or norm (ElMaraghy et al. 2013).*

As discussed in the industrial issues, the PV design implies the modelling and the connection of knowledge (Fig. 1.13) coming from the customer, the product and the process domains (in this thesis, the term *domain* is used to define a set of pieces of knowledge): customer-product in the first issue; product-process in the second issue; *new* customer-customer in the third issue.

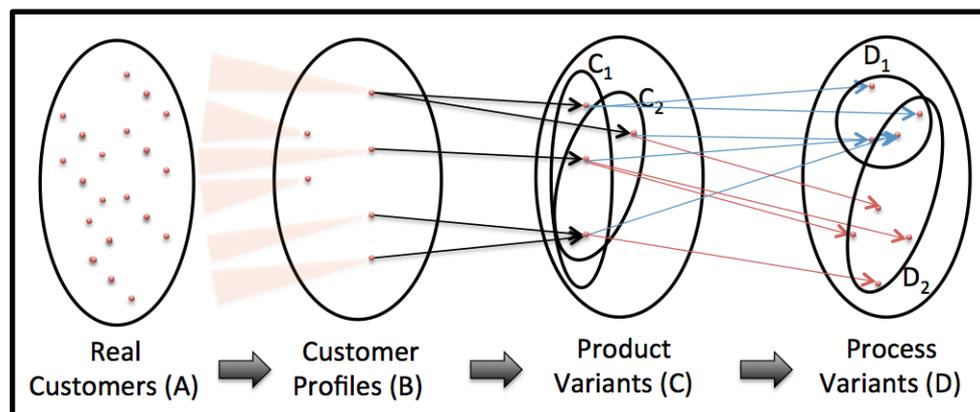


Fig. 1.13 - Links between domains involved in the PV design stage.

1. The **first issue** concerns the links between a variety of customers and the PV: if the customer-PV connections are not formalised then the product variant selection can be fuzzy (i.e. same customer and two sellers results in two variants) or far from the optimal one.
2. The **second issue** considers the connections of the PV with the process variety: if the model of the relations between the PV and the process variety is not formalised then the selection of

the process variety is performed by increasing the product designer efforts or by reducing the process engineer degrees of freedom.

3. The **third issue** considers the integration of a new customer variant into an existing customer variety: if a *special* customer is not integrated in the model of the customer variety then a new design stage is needed.

The variety modelling issues related to the industrial issues are shown in Fig. 1.14. At the top of the figure, the first and the second issues are modelled as connections between the real customer set (i.e. the target of the PV) and the manufacturing process set. Both the first issue and the second issue are about the lack of a formal model of the connection between the variety of the customers, the products and the processes. The first issue shows some lacks about the formalisation of the links between the customer domain (set A and B) and the product domain (set C). The effect is a quiet fuzzy product selection. On the other hand, the second issue is about the connections between the PV (set C) and the process variety (set D). The effect is the choice of some unbalanced product/process design strategies. The third industrial issue can be solved by the same solution to be developed for the first two issues: if a mapping is possible between varieties expressed in different domains then a mapping is possible between variety coming from the same domain (Fig. 1.14), i.e. the customer domain.

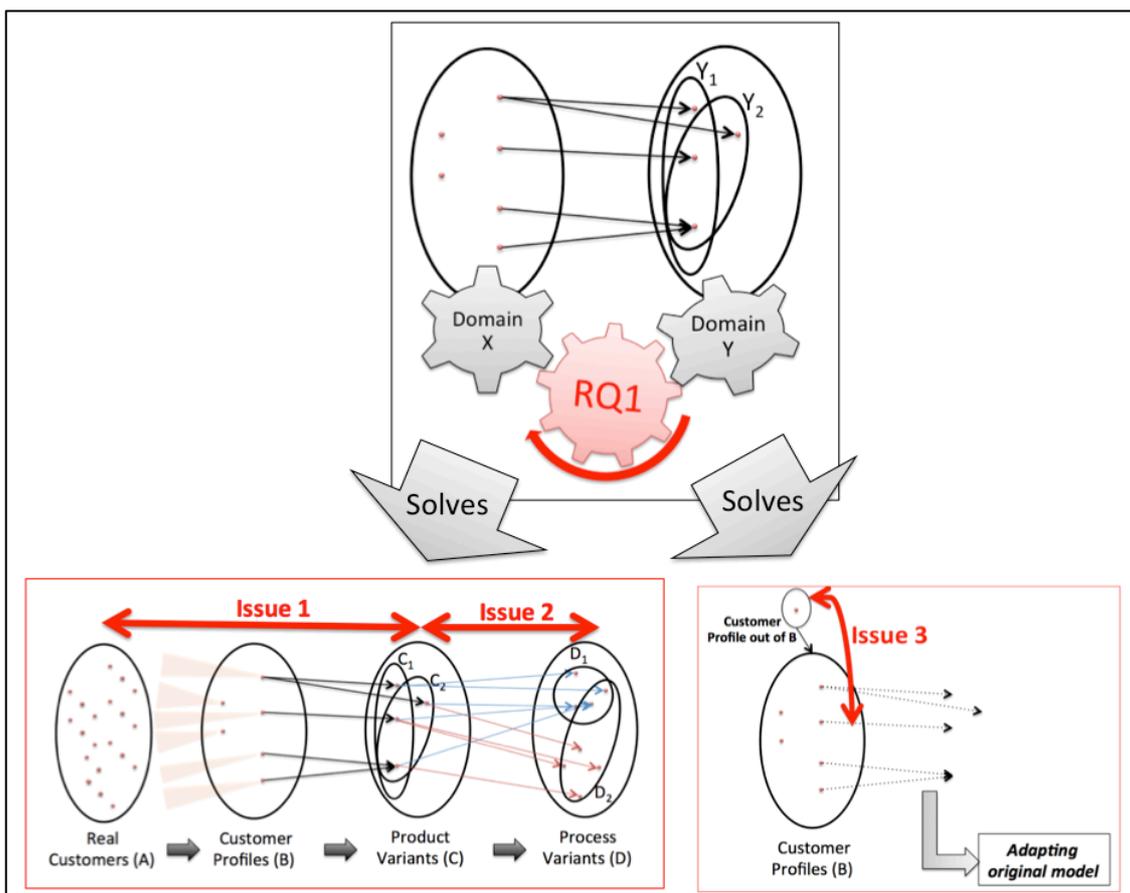


Fig. 1.14 - Representation of the industrial issues as mapping problem between variety representations: from different domains (1st and 2nd issues) or the same domain (3rd issue).

From this analysis of the relations between the industrial issues, the first research question can be formulated as follows:

RQ1 - How to model the variety in order to support a PV design process that at the same time takes into account the knowledge that relates the customer profiles, the product variants and the manufacturing process variants?

By answering the first research question, a formalisation of the links between customers and products should be possible. These relationships should allow to have a complete control of the product selection (II1), i.e. for the same customer requirements only one variant should be proposed. Building the same mapping until the manufacturing processes should allow to consider product, process varieties and their impacts on the customer set requirements in the same model. Therefore it should be possible to constrain the manufacturing processes only on the product features that cannot be modified to meet the performances required by the customers. Doing so, each PV should emerge from the customer and the manufacturing constraint. This should allow both to minimise the product design effort and to maximise the local cost reduction opportunities (II2). Since a model to formalise the variety from different domains, the integration of a new customer in the older PV model (II3) should become possible. The aim is to understand if the formalised knowledge is enough to meet the new customer requirements or some new piece of knowledge has to be added to fulfil them (i.e. the new customer has no corresponding product variants).

Once it is possible to connect variety model coming from different domains, starting from a given customer set a designer should be able to propose process alternatives (e.g. D_1 and D_2) that can meet the requirements of the customers. Once it is possible to generate these process variety alternatives, the deployment of some criteria to compare them is needed. Now, let us introduce the second research question:

RQ2 - How to compare different variety alternatives, i.e. process varieties that allow to manufacture PVs able to meet the same customer variety?

Here the objective is to find one or more criteria to compare the alternatives generated by answering to the first research question. These criteria should guarantee a sound approach to establish which is the best product variant for a given customer (II1 and II3) and which is the best process variant for a given product variant (II2 and II3). Notice that the criteria are not expected to optimise individually the variants. Indeed, the solution at the question RQ2 should provide a metric to evaluate the variety as a whole.

6 - Conclusions

In this first chapter the discussion about the industrial issues has been presented (Fig. 1.15). After a description of the scenarios and a contextualisation of the industrial issues, in the last section the two research questions that will be treated in this dissertation have been formulated.

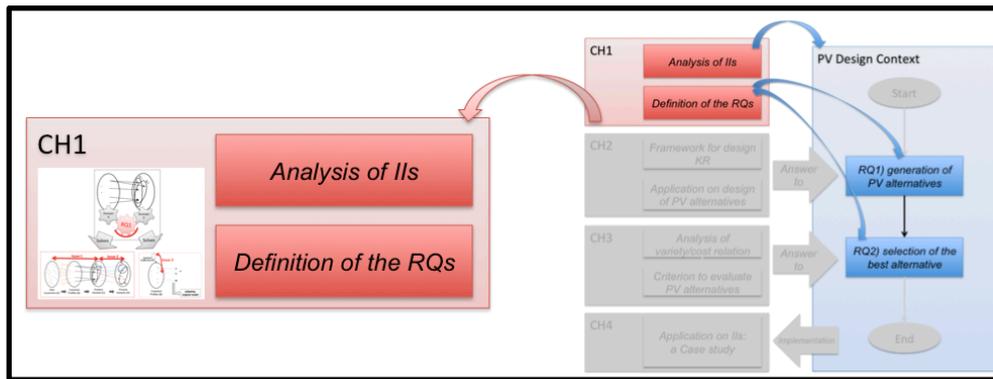


Fig. 1.15 - results of the first chapter.

Both the questions (Fig. 1.16) concern aspects of the PV design modelling: 1) the first question copes with the generation of a model to support a PV design that takes into account at the same time the customers, the products and the manufacturing processes. 2) The second question deals with the development of a criterion (or more criteria) to evaluate the PV alternatives generated during the PV design.

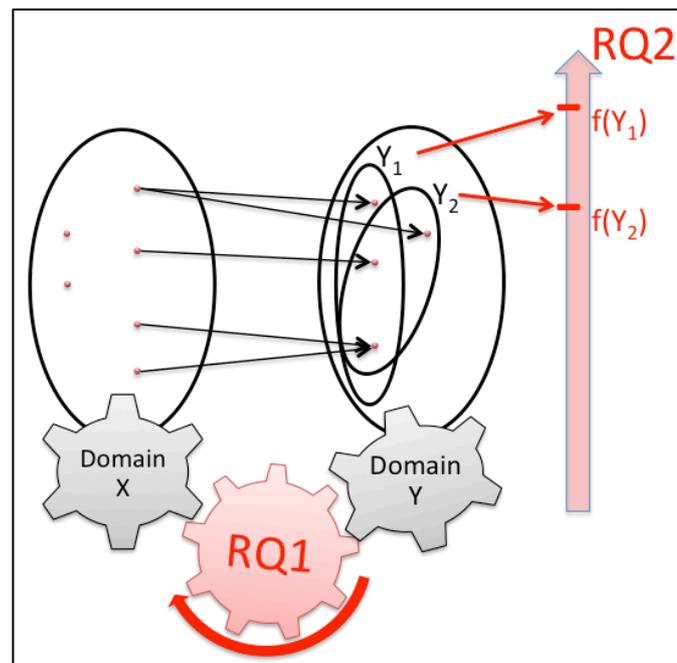


Fig. 1.16 - Illustration of the research questions

In the following two chapters the research questions are treated sequentially. According to the different viewpoints, **in each chapter the research works about the PV design are analysed differently.**

- In the chapter 2, the focus is on how the methods in the literature cope with 1) the customer profile formalization and 2) the representation of the links between the real customers, the customer profiles, the product variants and the process variants. By identifying a gap in the reviewed works, a framework for the design knowledge representation is proposed. Finally the implementation of this framework on the PV alternative generations is discussed.
- In the chapter 3, the focus is on how the methods in the literature cope with the evaluation of the PV alternatives. The analysis of the nature of the variety/costs relations will allow the

identification of a gap in the works in literature, From the result of this analysis, the development of a more comprehensive criterion to evaluate the PV alternatives is presented. Finally, in the chapter 4, the framework and the criterion are tested on the design of the variety of a HVAC system component to assess the impacts on the industrial issues.

Chapter 2 A design knowledge representation framework for Product Variety (PV) alternatives generation

The aim of this chapter is to review the works about the PV (*Product Variety*) design and to propose an answer to the first research question:

RQ1 - *How to model the variety in order to support a PV design process that at the same time takes into account the design knowledge that relates the customer profiles, the product variants and the manufacturing process variants?*

In the **first section** of this chapter, a literature review about the PV design is presented: the focus here is on how the presented methods link the variety representations. In other words, the issues treated in the papers below are classified on the basis of the links between the varieties (real customers, customer profiles, product variants and process variants) that are involved in the PV design stage. At the end of this section a discussion about the applicability of the methods in the literature is presented.

After the identification of a gap in the literature, the aim of the **second section** is to find works in literature that can help to bridge this gap. An overview on works about the Knowledge Representation (KR) is detailed. The purpose is to select and use certain KR principles as the foundations of the solution to answer the question RQ1.

In the **third section** of the chapter, our proposal to answer the question RQ1 (Fig. 2.1) is detailed. A simple instantiation is shown and the impact of the deployment of the proposals on the question RQ1 and on the industrial issues is discussed.

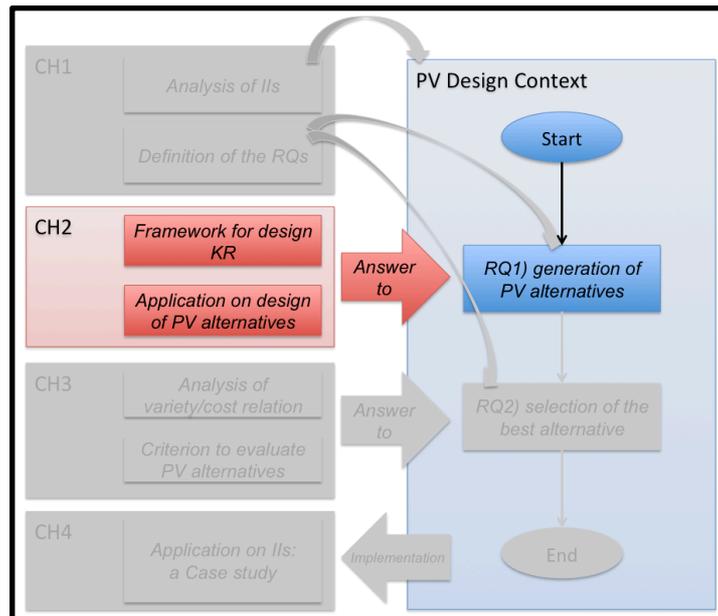


Fig. 2.1 - role of the chapter in the thesis structure.

1 - Literature about PV design approaches: linking the variety of two domains

In this section an overview of the method for the PV design is presented. The literature about PV design represents more than 20 years of works on all the aspects of the variety design and development process. Generally a **PV design approach can be classified according to three features** (Simpson, Siddique, and Jiao 2006; Fujita and Yoshida 2004): the type of *variety representation*, the *decomposition of the design stage* and the *input of the design stage*.

During the PV design stage, the **1) variety representation** (Simpson, Siddique, and Jiao 2006) can be based either on a module selection (i.e. *module-based design*) or on the value definitions of a set of product related parameters (i.e. *scale-based design*). For instance, let us consider the AHU example. If the variety is characterized by the presence/absence of one or more components (e.g. humidification module, heat recovery module) then the design is *module-based*. If the variety is characterized by the values of some particular attributes of the AHU system (e.g. weight, power, materials) then the design is *scale-based*.

An approach to the PV design process can be classified according to the **2) decomposition of the design stage** (Fujita and Yoshida 2004). Indeed, the PV design stage can be performed in *one step* or *two steps*. In the first case, the two steps are performed simultaneously. In both cases, the output of the PV design stage is the definition of the optimal non-shared modules or parameter values for the PV. At the end of the PV design stage, the optimal set of product variants (i.e. the PV) is obtained. When a two-steps approach is performed an intermediate stage is required. This stage is called *Product Platform (PP) design*. The output of this stage of the PV design process is the optimal set of modules or attributes values that should be shared between the variants in the PV, i.e. the optimal PP. A PP is a set of modules or parameter values that is shared between the variants in the PV. If all the variants in a PV share the same modules or attributes values then the PV is defined as a *single-platform PV*:

e.g. all the laptops share the same keyboard and trackpad; all the air conditioning units share the same sheet metal case; all the water coils share the same tube thickness. A PV can be based on more than one platforms (*multi-platform PV*): all the laptops share the same keyboard X, the 30% of these laptops share the trackpad A, the 70% share the trackpad B; the laptop family is based on two platforms, the first one is represented by the keyboard X and the trackpad A and the second one by the keyboard X and the trackpad B. If we consider the *water coil* example: all the coils share the same aluminium for their fins; the 50% of the coils share the fin profile A and the remainder of the variants share a fin profile B; the difference about the fin profile determines the two platforms which the PV is based on. A PP can be used for more than one PV.

Finally, a PV design approach can be classified according to the type of **3) input of the design stage** (Simpson, Siddique, and Jiao 2006):

- *Bottom-up PV design*: this approach of PV design requires that a set of individual products has been already designed; the aim is then to maximize the commonalities and to minimize the negative effects on customer requirements, i.e. the commonality maximization alters the product performances and so it can alter the effects on customer requirements;
- *Top-down PV design*: this approach of PV design does not require an already designed set of products; the design stage starts with a given set of customer requirements related to customer profiles; then the product variants are specifically developed for the customer set, i.e. not designed separately and then modified as for the *bottom-up* approach.

The approaches reviewed in the following subsections are analysed with **the focus on which domains links is considered** (Fig. 2.2).

1. *Real customers to customer profiles (section 1.1)*: the customer requirements have to be translated into product expected behaviours, i.e. performances. The formalization of this link includes all topics about the customer involvement into the PV design. Usually this class of issues deals with the analysis of the voice of the customer (VOC) and the customer clustering. These analyses are usually followed by the formulation of the relations between the expected performances and the customer demand, i.e. how the customer's purchase decision is impacted by the performances provided by a variant in the PV.
2. *Customer profiles to the product variants (section 1.2)*: it is about the transformation of performance requirements into product design variables, i.e. parameters or modules about the physical representation of the product variants. Concerns in this stage are called PV design issues. Most of the works about this class of issues includes the management of variant commonalities in the PV design.
3. *Product variants to the variants of manufacturing processes (section 1.3)*: The third stage is about the impacts of the PV on the manufacturing processes and the supply chain. Topics about the link between the product and the process domain are defined as *back-end* issues. Most of works on these issues are about manufacturing and assembly processes design and supply chain design.

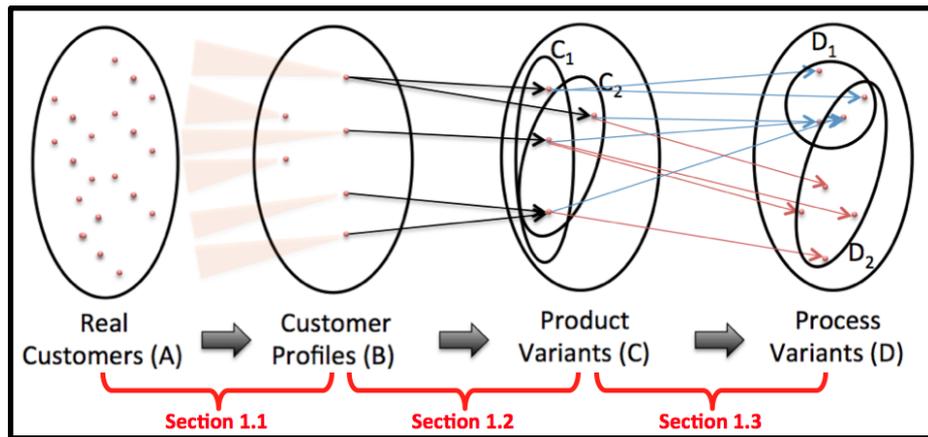


Fig. 2.2 - Domain connections to be reviewed and section numbers of the reviews.

In the last part of this section (**section 1.4**), a discussion about the applicability of these approaches to answer the question RQ1 question is reported.

1.1 - Linking Real Customers with Customer Profiles

Concerning the first link, most of works in the literature use different techniques such as data mining (Fayyad, Piatetsky-Shapiro, and Smyth 1996), fuzzy logic (Slowinski 1998), conjoint analysis (Green and Srinivasan 1978), and quality function deployment (QFD - (Akao 2004)) to analyse the real customer requirements and interpret them. More than the others, some of these techniques (e.g. QFD) are naturally used for the customer requirement transformation. The others have been transposed for the use in the PV design stage (e.g. data mining). The aim of these methods is to assess how the product performances and/or features impact the customer's purchase decision (i.e. the forecasted customer demand) and so the expected profit.

These techniques can be applied during each PV design approach: modular and scale based (e.g. modules can be used in place of product features); *top-down/bottom-up* and one-/two-steps because the decisions about the product platform or variety are not made at this stage (i.e. here the aim is to define the targets of the design stage).

Method based on conjoint analysis

Conjoint analysis allows to represent the customer requirements as relative importances given to a product feature or performances. Roughly, this characterisation allows to approximate the customer satisfaction as a customer utility function (U),

$$U = \sum_{i=1}^N I_i x_i$$

where I is the importance of the feature or performance i ; x is the value of the feature or performance i ; N is the number of features and/or performances.

For **instance**, in (Ferguson, Olewnik, and Cormier 2011), the use of the conjoint analysis for the design and development of a MP3 player is investigated. Here the aim is to provide a method to

connect the marketing data coming from the conjoint analysis to the engineering domain. A customer segmentation and an attribute prioritisation for each segment allow the authors to determine where (i.e. which product feature) a customisation is required.

The two main **comments** that can be expressed about the conjoint analysis are the followings.

- If the customer expresses its preferences as product features, he is performing a product engineer task (at the right of Fig. 2.3), i.e. the variety that is represented is in the domain of the product and so it is not modelling the customer profiles.
- if the performances (seen as expected effects of the product on the customer environment) are considered, then the first point is not still valid. But, anyway, 1) if the representation of the preferences aims to identify which is the performance to be maximized in all the variants, then the variety is still not represented: the maximisation is valid for each customer and so it is not representing the variety. Moreover, 2) let us consider
 - the utility as proportional to the price that the customer is willing to pay
 - and that the product that maximizes the utility costs more than the price that the customer is willing to pay for that level of performances.

Also in this case, the maximization limits the representation of the variety, i.e. all the product performances that can correspond to a utility are not represented.

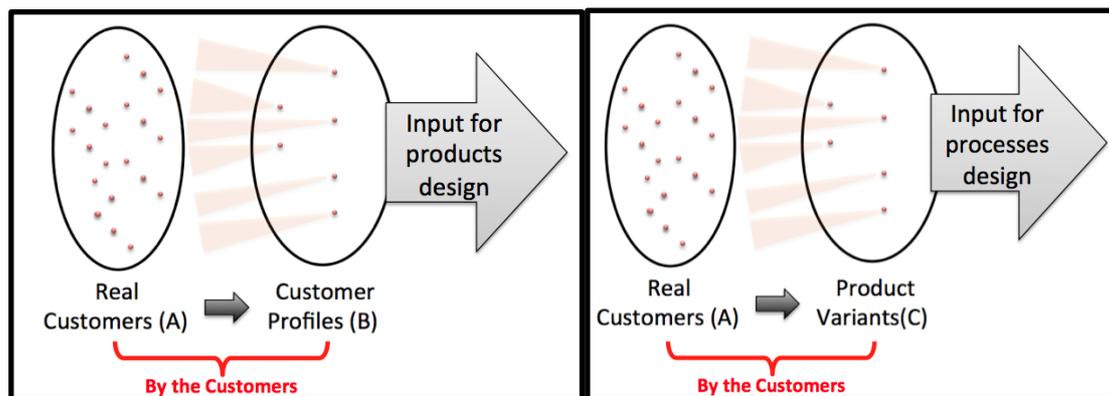


Fig. 2.3 - the different representations of the variety performed by the customer.

Method based on data mining

Data mining is a technique that allows to identify some patterns between variables, starting from a certain amount of data. Customers' data can also be collected from the internet like in (Tucker and Kim 2008): the authors develop a system for connecting the market demand analysis with the PP and the PV design. Data mining can be used to analyse the raw data from the Internet. The aim of the analysis (i.e. cleaning, transformation, pattern discovery and evaluation) is to build mathematical relations about the effects of the product features on the customer demand. This technique allows the calculation of the willingness to pay of the customers for given features. Moreover expected profits can be evaluated on the basis of the selected product feature set. The latter will represent the target in the engineering level. At the engineering level the architecture is related with the design constraints and the costs in order to find the best PV to be developed.

The same first **observation** done for the conjoint analysis can be expressed for the data mining: if the customers talk about the product, the variety represented is still about the product domain and about the customer profiles.

Method based on QFD

The QFD is a matrix to represent connections of the voice-of-customer with the engineering view of the product-related requirements.

The **use of the QFD** for connecting marketing information to the engineering domain has been investigated in (Kazemzadeh et al. 2009) and in (Jariri and Zegordi 2008). In (Kazemzadeh et al. 2009), authors propose a five-steps procedure to integrate marketing and QFD for PV design. 1) A first conjoint analysis is performed for prioritising the customer requirements and to bridge the gap with the designers. 2) A two-steps clustering procedure is applied to identify homogeneous set of customers. 3) A conjoint analysis is then performed for each cluster. 4) The traditional QFD is applied to each segment and to all customers. 5) They propose three criteria to assess which one between a general product or a product for each segment represents the best choice. Authors in (Jariri and Zegordi 2008) merge the QFD use with a mathematical modelling to integrate the customer requirements in the PV design. The first QFD table, i.e. customer requirements to technical requirements, offers a vision of the relations between the customer satisfaction and the product functions. The second QFD table, i.e. technical requirements to component characteristics, provides different alternatives to achieve the required functions. Also in this case, the mathematical problem aims to maximise the customer utility (characterisation of the customer's will of purchase) and to minimise the cost of the PV.

Also in this case, the same **observation** for the techniques reviewed so far is valid. Moreover, the clustering of the customers allows to provide a better vision of the variety compared to the simple conjoint analysis. The issue is about the number of QFD stages that is equal to the number of the variants to design.

Method based on fuzzy logics

Many other works deploy customer analysis using fuzzy logic to take into account reasoning that is not fixed and exact, i.e. approximate. Indeed, fuzzy logics, if compared to Boolean (i.e. preference expressed as "true" or "false") techniques, are more expressive because allow to represent a partial truth, e.g. "80% true". Examples of how fuzzy logic can be used for customer analysis in PV design are the following (Barajas and Agard 2009):

- customer clustering (using *fuzzy clustering*), to evaluate similarities among customer requirements;
- satisfaction level (using *fuzzy goal programming*), to assess the fulfilment level of the engineering design requirements; fuzzy logic is used to provide a better representation of the natural language;
- prioritisation of customer demands (using *fuzzy inference*), to fit the fuzziness of the voice of the customer;
- fuzzy QFD (using *fuzzy numbers*), to improve the definition of the links between customer requirements and the engineering characterisation of the product.

An example of application is in (Ostrosi et al. 2012). Here the authors propose an approach based on *fuzzy logic* for supporting the collaborative and distributed design for variety. The process has three main steps: 1) a mapping between requirements, functional and physical views of the PV is performed; 2) supported by a multi-agent system, the customer and the engineers find a sort of compromise solution that should satisfy the customer requirements and respect the engineering constraints.

Our **observations** about the fuzzy logics are the followings: 1) they can be used along with the other techniques to provide a more expressiveness to the representation; 2) as for other techniques, the issue is still related to which are the features or performances that should characterise the customer profile.

Others

In (Sullivan, Ferguson, and Donndelinger 2011), the authors compare the customer requirement expressiveness of two different techniques, i.e. continuous and discrete representations of the requirement values. An assessment of the ability to capture the expressiveness of the customer domain is performed. Our **observation** about the techniques reviewed in the paper is still about which features or performances should characterise the customer profile.

General Observations

To sum up, the reviewed techniques for the representation of the customer variety (i.e. the customer profiles definition) have to main issues:

- the risk of letting the customer talk about the product variety,
- the risk of focusing on the performances maximization.

Both the points can lead to a limitation to the product engineer degrees of freedom.

1.2 - Linking Customer Profiles with Product Variants

The links between the customer profiles and the product variants mainly cope with mathematical modelling and the algorithm development. The **general mathematical model** for p products is the following (as formalised in (Z. Liu, San Wong, and Lee 2011)):

$$\begin{aligned}
 &\text{Find } X = (x_1^1, \dots, x_1^p, x_2^1, \dots, x_2^p, \dots, x_n^1, \dots, x_n^p) \\
 &\text{Max. } \left\{ \sum f^P(X), f^C(X) \right\}, \\
 &\text{s.t. } g_j(X) \leq 0, \quad j = 1, \dots, k \\
 &\quad X_L \leq X \leq X_U, \quad X = [X^d, X^c]^T
 \end{aligned}$$

where X is a set that describes the n design parameters for the p products; for each product p in the PV, $f^p(X)$ are functions about the product performances (P), $f^c(X)$ are functions about the variant commonalities (C); $g_j(X)$ represent the k design constraints; X_L and X_U are the properties lower (L) and upper (U) bounds. The design variables can be continuous (X^c) or discrete (X^d).

The mathematical problem is quite complex to be solved by an exact algorithm. Therefore most of the works deal with the development of **heuristics**. In (Khajavirad, Michalek, and Simpson 2009), the

authors propose a multi-objective genetic algorithm to optimise the PV. The design approach in the paper is a simultaneous approach. In (Qu et al. 2010), the authors propose a two-steps PV design based on graph representations of product components and a genetic algorithm to find the best values of the product variants parameters. Graphs are used to represent the variants components and the relationships between them. Once all the variants are represented, a genetic algorithm maximises the number of common components between the variants. In (Li and Huang, 2009), the authors propose a multi-objective evolutionary algorithm to maximise the PV commonalities at product, module, component and parameter levels. An experiment to test different algorithms for large instances of the PV design problem is also performed and discussed in (Belloni et al. 2008). The authors propose also an algorithm to solve a problem in which the number of possible variants in the PV as fixed input.

Even if the mathematical problem is pretty complex, there are works that deal with **exact solutions and/or approximated formulation of the problems**. In (Chowdhury, Messac, and Khire 2011), the authors propose a Mixed-Integer Non-Linear Programming (MINLP) model to solve the PP design (first step of a two-steps approach). In (Fellini et al. 2005; Fellini et al. 2004), the authors address problems of larger size by relaxing the combinatorial problem into a continuous one. Also in these cases the aims of the method is the PP design. Complementarily, authors in (Khajavirad and Michalek 2008; Khajavirad and Michalek 2009) deal with the second stage of a two-steps design approach. The authors test two different methods to achieve the global optimum. In both the cases, the PP design has already been performed.

Observations

The aim of most of the reviewed mathematical problems is only to manage a compromise: the commonality maximisation by preserving the performance levels and a certain degree of variety. Indeed: on one hand the commonalisation is considered related to the cost minimisation; on the other hand, an excess of similarity is considered as related to a reduction of the customer demand.

The point here is that all the variables of the models are either related to the customer or related to the products, i.e. **there is not a method that models the links between customer profiles and product variants**. Indeed, there are two cases:

- the models represent *bottom-up* approaches (independently from how many stages or if modular or not) to the PV design; therefore, the problems of this group have as input a set of already designed variants; the costs and the customer demand are both related to the features of the designed variants; the variables and so the decisions are all made in the product domain; therefore the links have not been formalised; this is possible if the customer requirements are represented as product features (at the right of the Fig. 2.4);
- the models can also represent a *top-down* (independently from how many stages or if modular or not) approaches to the PV design; in this case the variants are not already designed; the variables and so the decisions are all about the customer expected, i.e. what can define a customer profile ; this means that the solutions of the models gives only the target of the product variants that should be designed during another stage (at the left of the Fig. 2.4).

Finally in both cases, the variants have been already designed or are still to be designed. None of the works gives a method to design the product variety starting from a set of customer profiles.

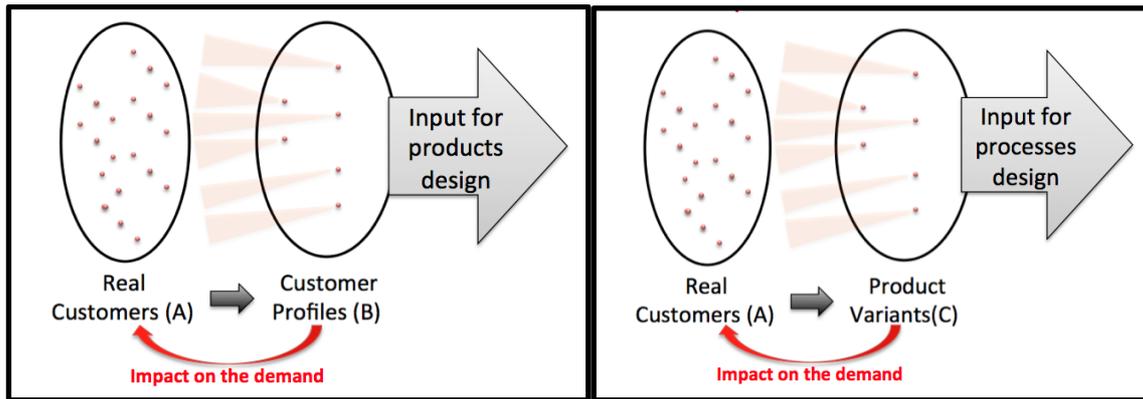


Fig. 2.4 - the actual results of the papers reviewed in the section 1.2.

1.3 - Linking Product Variants with Process Variants

Linking the process variety with the product variety deal with the integration of manufacturing processes, assembly processes and supply chain constraints into PV design stage. Most of related works fall into the operational research field. Since the high number of different technique used in the literature, the works are here presented in a table (Tab. 2.1).

Tab. 2.1 - main characteristics of the papers about the manufacturing domain integration in the PV design.

	PV design approach	Focus and techniques
(Siddique and Wilmes 2007)	Second step of two-steps approach. Module-based.	Respect of the assembly constraints.
(F. Gao, Zhang, and Xiao 2013)	First step: PP design. Scale- + module-based.	Machining and assembly constraints. Fuzzy logics.
(Z. Liu, San Wong, and Lee 2011)	Second step: <i>multi-platform</i> approach. Scale-based.	Product and process commonalities. Genetic algorithm.
(Michalek et al. 2006)	Simultaneous PV design. Scale-based.	Marketing, product, manufacturing costs and constraints. Problem decomposition.
(Fujita, Amaya, and Akai 2013)	Simultaneous PV design. Module-based.	Transportation costs. Genetic algorithm.
(Shahzad and Hadj-Hamou 2013)	First step: PP design. Module-based.	Production and supply chain costs.
(Das, 2011)	The PV is already designed.	Supply chain uncertainty: e.g. quality of supplied parts, inventory stock-outs.

(Agard and Penz 2009)	Simultaneous PV design. Module-based.	Assembly times. Simulated annealing.
(Khalaf, Agard, and Penz 2010b)	Simultaneous PV design. Module-based.	Supply chain impacts: e.g. assembly times, transportation costs. Tabu search.
(Khalaf, Agard, and Penz 2010a)	Two-steps PV design: 1)which module to which facility; 2)best solution for each facility. Module-based.	Supply chain impacts: e.g. assembly times, transportation costs.
(Sandborn, Prabhakar, and Eriksson 2008)	Simultaneous PV design. Module-based.	End-of-life of product components, i.e. maintenance costs.
(Lamothe, Hadj-Hamou, and Aldanondo 2006)	Simultaneous PV design. Module-based.	Supply chain impacts: e.g. assembly times, transportation costs.
(Williams et al. 2007)	First step: PP design. Scale-based.	Product and process commonalities.
(Zacharias and Yassine 2008)	First step: PP design. Module-based.	Project budget. Variant assembly process.

A main **observation** can be done about the works in the table: ***the links with the process variety are not built in these methods***. Actually, each one of the model in the table has as variables the product attributes and/or modules. The computation is only about how this variables impact the supply chain and/or assembly costs. Even the last ones can be assessed directly on the basis of the product features. ***There is not a method that design the best process variety depending on the product variety*** defined in the same stage or in a previous one.

1.4 - The gap in the literature: the lack of an unambiguous knowledge representation

After the review of the researches on PV design, in this section, the aim is to identify the gap between the methods in the literature and a method to answer the first research question:

RQ1 - *How to model the variety in order to support a PV design process that at the same time takes into account the design knowledge that relates the customer profiles, the product variants and the manufacturing process variants?*

How the works in literature consider the variety in the domains involved in the PV design and development is shown in Tab. 2.2. Here an X has been associated with a reviewed work if:

- the proposal deals with the raw data coming from the real customers, e.g. from the internet, the voice-of-customer, surveys, etc.; in these cases, an X in the first column has been associated to the work;

- the proposal deals with the impacts of the decisions about the product variants on the process and supply chain costs; in these cases, an X in the third column has been associated to the work;
- the sign X in the second column has been attributed to the proposal that deal with the impact of either the expected performances (at the left of the Fig. 2.4) or the product variant features (at the right of the Fig. 2.4) on the customer demand.
- As observed in the previous sections each links built in the reviewed works can present an issue.

Real Customer to Customer Profiles: here the issue is about the representation of the customer variety. None of the previous works cope with a proper representation of the variety: **1) the customer variety should be about performances that are not to be maximised but only represented; 2) the customer variety should be expressed in the customer domain and not in the product domain**, i.e. the customer should talk about the product only if strictly required to minimise the constraints on the product engineers. A more detailed description of these points can be found in (Giovannini et al. 2014). Here the authors present a state-of-the art about the impact of these points on the configuration systems (Tab. 2.3): works about configuration system design are classified for the way they model the variety on the customer side, i.e. the system interface used by the customer. Even looking at the tools that are used to interface the customer to the designed PV shows how the proper representation of the variety in the customer domain (i.e. profiles) is an open problem.

Tab. 2.2 - reviewed references and domain connections.

	Real Customers to Customer Profiles	Customer Profiles to Product Variants	Product Variants to Process Variants
(Sullivan,2011), (Ferguson,2011)	X		
(Khajavirad,2008), (Khajavirad,2009a), (Khajavirad,2009b), (Qu,2010), (Li,2009), (Chowdhury,2011), (Fellini, 2004), (Fellini,2005), (Belloni,2008)		X	
(Das,2012)			X
(Tucker,2008),(Jariri,2008), (Ostrosi,2012), (kazemzadeh,2009)	X	X	
(Zacharias,2008), (Siddique,2007), (Liu,2011), (Gao,2013), (Sandborn,2008), (Michalek,2006), (Williams,2007)		X	X

Tab. 2.3 - how the variety is represented in the works about configuration system design (from (Giovannini et al. 2014))

	Customer needs	Functional requirements	Design Parameters	Deployed configuration method
(Aldanondo and Vareilles 2008; Mazo et al. 2012; Yang and Dong 2012)			x	Constraint-based approach
(Du, Jiao, and Tseng 2001; Janitza et al. 2003; Qin and Wei 2010)		x		Mapping approach
(P. Gao et al. 2003; X. B. Liu et al. 2003)		x		Ontology mapping approach
(Helo et al. 2010)	x			Ontology mapping approach
(Hong et al. 2008)		x		Genetic programming
(Inakoshi et al. 2001)			x	Case-based reasoning & constraint-based approach
(B. Li et al. 2006)		x		Genetic algorithm
(Wang et al. 2011)			x	Ontology mapping
(Xie, Henderson, and Kernahan 2005)		x		Constraint-based approach
(Zhou, Lin, and Liu 2008)		x		Constraint-based approach & genetic algorithm
(Zhu et al. 2008)		x		Fuzzy decision making approach

Customer Profiles to Product Variants: the observations done about the works that deal with this link concern the absolute absence of a formalization of this link. Indeed, in the reviewed proposals, the decisions (i.e. commonality and expected demand maximisation) are made about either the customer profiles or the product variants. Here **a formalisation of how a product variant has to be in order to meet the customer requirements is missing**, i.e. the knowledge about the product design is missing.

Product Variants to Process Variants: the observations done about the works that deal with this link concern the absolute absence of process definition. Indeed, in the reviewed proposals, the decisions (i.e. process commonality, supply chain and assembly costs) are still based on the product variants features and do not concern the features of the processes that should manufacture the products. Here **a formalisation of how a process variant has to be in order to manufacture a required product variant is missing**, i.e. the knowledge about the manufacturing processes is missing.

To sum up, in order to have a PV design approach that allows to answer the question RQ1, the main issue is about the **lack of design knowledge**. Notice that, even if the customer expresses his needs as product features, the knowledge about how to realise the product should still be missing. Moreover, even if the design knowledge has been represented, connecting representations coming from different domains means linking models built by different modellers (e.g. customers, engineers) at different moments. The cultural gap between the modellers represents a challenge for this knowledge connection (Zdravković and Trajanović 2009; Whitman and Panetto 2006). Therefore, in order to provide an efficient mechanism to connect these models, the representation of these pieces of knowledge have to be *unambiguous*. Literally, **unambiguous** is defined as *not open to more than one*

*interpretation*¹. If the model of the knowledge is not open to more than one interpretation, then it is possible to be sure that the intentions of the modeller (i.e. what the modeller wants to represent) cannot be misunderstood by another modeller, even if the modellers are experts coming from the different domains. In summary, if the connection of more pieces of knowledge is required then the Knowledge representation (KR) has to be not open to more than one interpretation, i.e. *unambiguous*. This requirement should be met in order to provide a model that represents design knowledge from different domains and so it is essential to provide an answer to the question RQ1.

In summary,

the gap identified in this section should be expressed as a lack of unambiguous knowledge representation to support the PV design and so to answer the question RQ1.

In order to bridge this gap, a short literature about the two schools of the KR is presented in the next section.

2 - Design knowledge representation (KR)

This second section represents a sort of bridge between the literature on PV design and our approach to answer the first research question. Here a quick overview on methods to represent knowledge is discussed. The aim is to find the foundations of our framework that is supposed to provide a method to represent the design knowledge coming from different domains.

2.1 - Schools of Knowledge Representation

KR is one of the problematic of the Artificial Intelligence. Works on KR can be divided in two sorts of schools of thoughts: *logicist* and *anti-logicist*. The term *logicist* comes from mathematics: the *logicism* believes that all mathematics can be reduced to logic (Russell and Whitehead 1925). In KR, a *logicist* believes that knowledge is best represented using formal logics (pp. 67-74 in (Van Harmelen, Lifschitz, and Porter 2008)), such as First Order Logic, Description Logic (Protégé² is based on it), Second Order Logic and so on.

Since a state of the art in KR is out of scope of this thesis, here only a discussion about the arguments of both these two schools of thoughts is presented (from pp. 67-74 in (Van Harmelen, Lifschitz, and Porter 2008)). The debate about the usage of the logic for KR is mostly around the following four points:

1. *reasoning about exceptions*. The modelling and the reasoning process about exceptions can be a really tough task for logic-based KR. For the sake of an example, *if x is bird and is not a penguin then x can fly*. There are two issues about the formalisation of the exceptions: 1) they can be too many; 2) if we know that y is a bird, that is not enough to infer that y can fly. In

¹ From <http://www.oxforddictionaries.com/definition/english/>

² <http://protege.stanford.edu/>

order to cope with this critic, *logician* developed *non-monotonic* reasoning. The main idea about this way of reasoning can be synthesized as follows “in the absence of any information to the contrary, assume...”.

2. *reasoning based on logics is too expensive*. According to the *anti-logician*, reasoning about several thousands or millions of axioms is simply not possible. Indeed, *logician* researches are mainly focused on this point. A lot of efforts in the last years have been done to improve the reasoning performances. Some good results have been obtained under restrictive assumptions.
3. *other approaches do it better*. An *anti-logician* critic is based on the existence of other methods that can represent knowledge better than logics. The *logicians state* that logic remains the best way to represent knowledge for complex application and other approaches can be better for the solution of easier problem, e.g. learning with a restricted vocabulary.
4. *representing all the knowledge is infeasible*. Even if there have been successful implementations of the *logician* approach, some doubts about this critics still remain:
 - a. it is difficult to become aware of all the implicit knowledge and then make it explicit;
 - b. there is some apparently obvious knowledge that is really difficult to express in any language (E. Davis 1998);
 - c. there are no efficient theory about the reasoning about the absence of knowledge;
 - d. a conceptual model describing a domain is a prerequisite and represents something extremely difficult to build;
 - e. the mapping problem is a quite hard issue because concepts do not always match easily.

Besides these considerations, both the *logician* and the *anti-logician* have proved their validity in different applications e.g. robotics, multi-agent systems, decision support systems. In the next section, the aim is to explain why the use of *logics* for our problem is not the best solution.

2.2 - *Anti-logicism, logicism and the ambiguity of the representation*

The first works proposing an alternative to the *logics* for KR were published in mid-80s: the works of Agre and Chapman on a computer program called Pengi (Agre and Chapman 1987), the Brooks' subsumption architecture (Brooks 1991), the Kaelbling's intelligent reactive systems (Kaelbling 1987), are examples of this approach to KR. The main ideas behind these solutions are the following:

- intelligent behaviour can be obtained without representation and reasoning about symbols, e.g. natural language (Brooks 1991);
- intelligence emerges from the interaction between the system and the environment (Brooks 1991);
- perception directly triggers reactions (Fig. 2.5), i.e. no reasoning intervenes between them (Agre and Chapman 1987);
- systems react without having an explicit model of the world (Agre and Chapman 1987);

- things in the world are not represented objectively, i.e. independently of the system's purpose, but rather they are represented in relation with the system nature and its projects (Agre and Chapman 1987).

In summary, anti-logicists do not formalise explicit plans to be executed by a system. They give the system a way to interact with the environment (e.g. sonar, motors), a set of skills (i.e. how to avoid an object, how to explore an environment) and a set of purposes (e.g. do not hit objects, explore). On the basis of the perceptions, one or more skills are applicable. A prioritization system decides which one have to be applied at a certain time. Even some commercial products like iRobot vacuum cleaner³ are based on this approach.

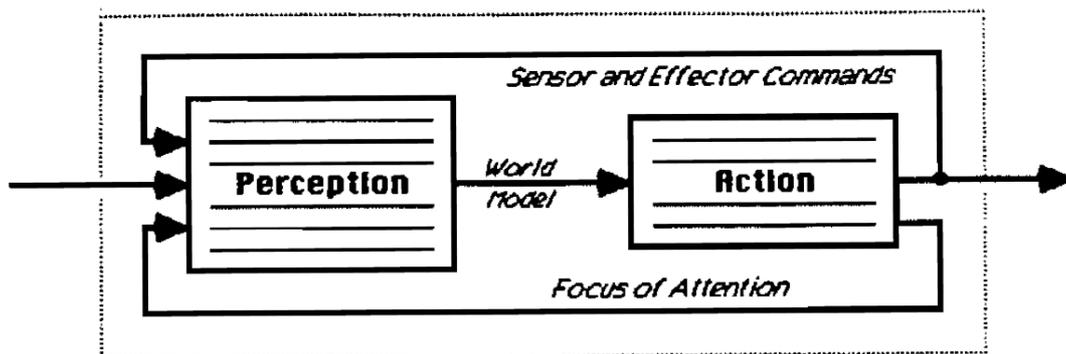


Fig. 2.5 - part of the reactive system architecture in (Kaelbling 1987).

Now, how these works can help solve the first research question? As explained above, the most important feature that our KR approach needs to have is the unique interpretation, i.e. the representation needs to be *unambiguous*. The *anti-logicist* architectures for robots and multi-agent systems provide this feature by avoiding abstract reasoning: in order to use the represented knowledge (*showing intelligence* in (Brooks 1991)), no-abstraction is needed (Fig. 2.6). Indeed, each abstract concept instantiation requires the user of the model to bridge the gap between the perceptions and the concepts. Since different users can bridge the gap in different ways, the abstract concept instantiation leads to a high risk of ambiguity in the KR. For this reason, the *logicist* KR approach cannot meet the requirement of *unambiguity*.

³ <http://www.irobot.com/About-iRobot>

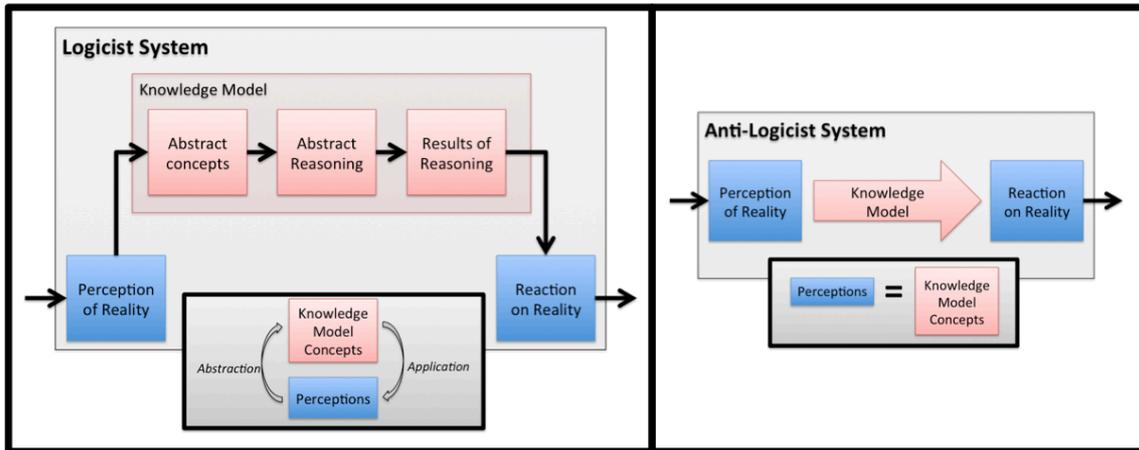


Fig. 2.6 - representation of how the knowledge is used in a logicist and anti-logicist-based systems.

In *anti-logicist* systems, the systems react on the basis of some perceptions of the world collected by sensors. Knowledge is about the *perception-reaction* relations, i.e. the description about how to alter the *system-environment* interaction in order to achieve a system goal. Both perceptions and reactions are directly connected to sensors. In other words, there is no conceptual definition of the knowledge that should be instantiated to infer an appropriate system reaction that should orient the system toward its goals.

Doing so, *anti-logicist* works show that the use of the represented knowledge is possible without abstract reasoning (Fig. 2.6). Therefore also a design KR without interpretations depending on the personal experience should be possible.

In order to build an anti-logicist KR framework, the following transposition of concepts is required (Tab. 2.4). In a framework for the unambiguous design KR, the knowledge models should be rooted in the reality by means of the perceptions. For *anti-logicist* systems, perceptions are sensor inputs. For design knowledge modellers, perceptions should be *measurements* (e.g. temperature, humidity). For *anti-logicist* systems, the perception-reaction relations allow to orient the system towards its goals. For design knowledge modellers, the relations between measurements should allow to describe how the represented knowledge can meet certain requirements. For *anti-logicist* systems, this approach allows intelligence to emerge from the system-environment interactions. In this framework, this approach should allow the PV to emerge from some *what* (i.e. customer requirements) - *how* (i.e. engineering knowledge to meet the requirements) relations. Every modeller represents a knowledge that does not require any abstraction (and so interpretation) to be used. Therefore even if the represented knowledge is from more domains, a correct use of this knowledge can be performed.

In the next section our framework based on the concept of *measurement* is proposed. In the first part of the section the concepts of the meta-model and the relations between them are detailed. A short instantiation is shown. Finally an algorithm that should prove the unambiguity of the representation is proposed. In the second part of the section, the KR approach is implemented on the PV design issue. Here the potential impacts on the industrial issues are discussed.

Tab. 2.4 - transposition of concepts.

Anti-logicist systems	Anti-logicist design-KR
Sensors and actuators	Measurements
Perception-Reaction models	Requirements-engineering solution models
Intelligence without abstraction	Use of the knowledge without abstraction

3 - Our approach: design KR framework for PV alternatives generation

In this section, the aim is to present a KR framework that should allow an unambiguous representation of the design knowledge coming from different domains. The representation of this knowledge should be unambiguous to allow customers and engineers to use without ambiguities the knowledge models. The transposition of the anti-logicist KR principles allows to highlight the fundamental role of the concept of *measurement*. In the following sub-section, the semantics of *measurement* and the related concept at the base of our KR framework are presented.

3.1 - Design KR framework

3.1.1 - Conceptual model

In this framework, the *measurement* is the concept that allows to avoid the abstraction and so the ambiguity of the KR (as the *perception* concept does for the *anti-logicists*).

The conceptual model (represented as a UML class diagram) showing the basic concepts of this framework is presented in Fig. 2.7. The semantics of the main concepts in our framework is the following.

- A *measurement* is the characterisation of the act of perceiving, by mean of an instrument of measurement, a certain measure in a certain environment at a certain time. For instance, the temperature of a certain volume of air at a certain time is a *measurement*.
- Each measurement is composed by a *space*, a *time* and a *shape* characterisation (*STS*). For instance, the temperature can be the *shape* of the measurement, the volume of air can be the *space* of the measurement and the *time* represent *when* the temperature is measured. Each one of these three concepts are *properties*.
- Each *property* is involved in one or more *transformations*: a *transformation* is a mathematical relation between a set of *properties*. For instance, the relation between the temperature and the time in which it is measured is the *transformation* $T=f(t)$, where t is the time of the *measurement*, T is the temperature and f is the mathematical relation that links the two *properties*.
- A set of mathematical relations between the properties of a set of *measurements* is an *experience*. An *experience* is a sort of report of an experiment: here the setup (*measurements* that are constant during all the observations) and the variables observed during the

experiment are all formalised as mathematical relations between *measurements*. For the sake of an example, let us consider an experiment to test the validity of the ideal gas law $PV=nRT$ (an instance of *transformation*). The instance of the *experience* has to capture the mathematical relations between all the STSs of the *measurements* (e.g. the pressure, the temperature, the volume, the considered gas) that describe the environment conditions in which the respect of the law is guaranteed. In other words, the *experience* instance describes a law and when this law is applicable. All this is formalised by *measurements* and *transformations*.

- Each *property* has to be detailed by a *range of values*, a *UOM* (unit of measure) and a *tolerance* of the *measurement*. For instance, the temperature can be observed in Celsius degrees (*UOM*) and from 25 to 50 (*range of value* for the validity of the experiment law), with a *tolerance* of ± 0.1 .

The main idea behind this KR framework is that every atom of knowledge is an *experience* and every *experience* should have been perceived. If the *experience* has been perceived in terms of instruments of measurement and time-space of measurements then no ambiguity can exist between the *experiences* (as for perception for *anti-logicist*)⁴. The result of the mathematical regression to approximate the mathematical relation between the observed factors of the observed system is the main instance (i.e. the observed law) of the *transformations* in the *experiences*. The other instances of *transformation* should characterise the applicability of the observed law. This applicability is described by the *UOMs*, the *ranges of values* observed and the *tolerances* (error of the instrument) of every *property* of every *measurement*.

Notice that even the objects of the experiment are characterised by *measurement* instances. For instance a copper cable should be described as a cylinder (*space*) in which we can measure some *shapes* that should define the interaction of the copper with the rest of the observed system, e.g. the conductivity of the copper if we are observing the behaviour of an electric power system. This method allows to formalise all objects (that are usually formalised as abstract concept by means of the natural language) as *measurements* that are directly connected with the system behaviour and so with knowledge expressed by the instances of *experience*. This provides a mean to avoid any abstraction to connect the perception of the reality (described by the *measurements*) with the knowledge about a system behaviour (described by the *transformations* in each *experience*). As explained above, if there is no need of instantiate abstract concepts then every ambiguity should be avoided.

Even if it is hard to imagine a common customer describing his requirements with this level of detail, it is important to remember the kind of product that represent the target of this method. As discussed in the first chapter, the industrial issues do not concern make-to-stock products (e.g. foods, mobile phones). In engineer-to-order or make-to-order markets, customers have a deep understanding of what they need. Nonetheless, even for requirements hard to imagine as measurable because strictly

⁴ The measurements are considered as perceptions therefore as the only way to perceive the reality: when two experiences describes the same system but with different instrument of measurements then a further knowledge is required to represent and map the difference between the two ways of perceive the reality.

related to the human perceptions, there exist works that attempt mathematical formalisations of physical-physiological relations (Chauvet 1993; Lieber et al. 2013).

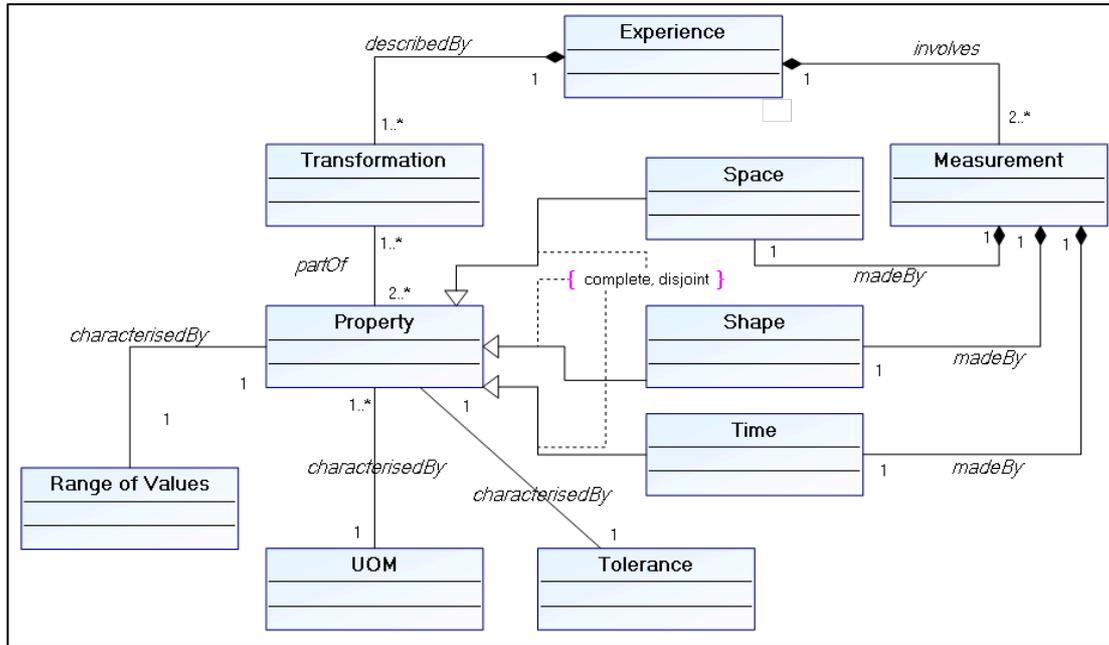


Fig. 2.7 - Conceptual model of the framework

3.1.2 - A short instantiation of the experience concept

For the sake of an example, let us consider a simple instantiation: our **experience** is the representation of the heat exchange between a certain volume of air and the flow of water in a bare tube water coil. Here a volume of water flows into the coil to cool an airstream. In order to instantiate the model, the *space* characterisation of the *measurements* is required. This can be performed by using a CAD software: parameters are added to the CAD files and linked to the volumes or surfaces that should provide the representation of *where* the *measurement* is performed. Notice that, in the case of dynamic behaviour, the *shapes* (i.e. variables such as temperature, pressure and so on) and the *spaces* (i.e. CAD volumes) can be expressed as function of a *time* variable.

In this dissertation, the software used to illustrate the framework instantiation is CATIA V5 by Dassault Systems⁵. The details about the usage of the software can be found in the fourth chapter, before the application of the proposed framework on the case study.

The representation in CATIA of the bare tube water coil system is shown in Fig. 2.8. The represented behaviour is a sensible heat transfer as in (ASHRAE 2012).

⁵ <http://www.3ds.com/products-services/catia/welcome/>

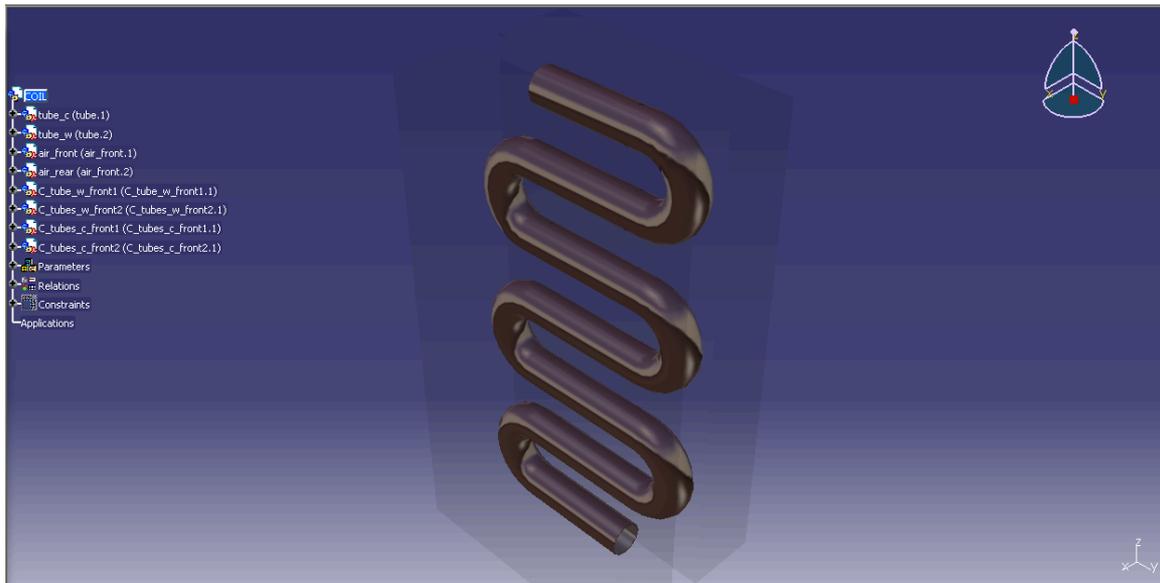


Fig. 2.8 - representation of the bare tube coil experience on CATIA V5.

The instances of the **measurement** class are:

- the temperatures of the air before and after the heat exchange, T_{A1} and T_{A2} ;
- the temperatures of the water before and after the passage in the coil, T_{R1} , and T_{R2} ;
- the speed of the air, V_A ;
- the density and the specific heat capacity of the air and the water, RO_A , RO_W , C_P , C_R ;
- the film coefficient of the heat transfer between the air and the external coil surface, F_A ;
- the film coefficient of the heat transfer between the water and the internal coil surface, F_R .

The relative **shapes** are the temperatures, the speed, the density and the heat transfer coefficients. All measurements are related to specific volumes or surfaces (the **space** characterisation). For instance in Fig. 2.9, the volume related to the C_P , RO_A and T_{A1} measurements is highlighted. The link of the measurement and the volumes in the CAD is performed adding a specific parameter and associating it to the measurement of the relative volume or surface. For instance, the parameter $vol_{T_{A2}}$ and the *formula.155* in Fig. 2.10 associate the T_{A2} measurement with the volume in which the T_{A2} value is measured.

The observed system is stationary so all measurements are **time**-invariant. The associations of the **measurements** to the volumes provide a characterisation of the properties related to the volume that are involved in the system behaviour. For instance, the association of the C_P and RO_A with the air volumes characterise the property of the air that are involved in the behaviour of the system, i.e. the sensible heat transfer.

Therefore the abstract concept of *air* is formalised by mean of the STSs of the C_P and the RO_A . In this way this framework provides a representation of the object without using the natural language. The abstract concepts that should define an object are replaced by the **shapes** of the **measurements** in the volume occupied by the object in a certain time. The **shapes** that define the object are the properties that characterise the interactions between the objects in the represented system. In other words, objects are represented by mean of the **properties** that are involved in the system behaviour described by the **transformations**. Since everything is defined by measurements, no abstractions are

needed to connect a certain measured reality with a modelled system behaviour. Therefore no ambiguity is possible when the formalised knowledge is retrieved and interpreted for designing a new system. Some more details about how experiences connect each others can be found in the next section.

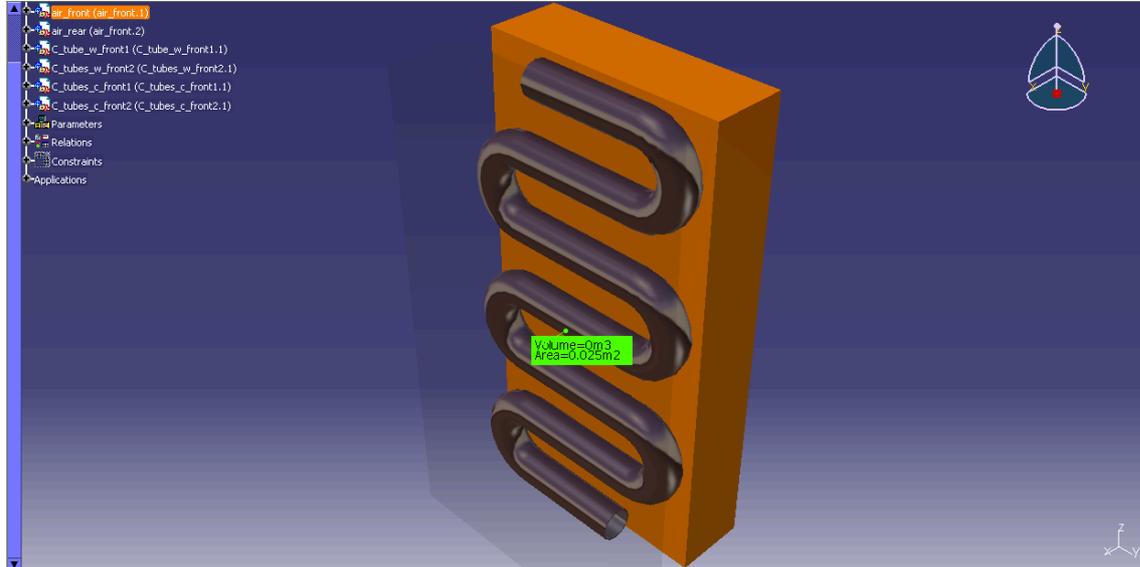


Fig. 2.9 - representation of the bare tube coil: the volume highlighted is related to the C_P, RO_A and T_A1 measurements.

The **transformations** between the STSs of the measurements are shown in Fig. 2.10. The mathematical relations include a group of properties not related to the above cited measurement:

- Q_T is the heat exchange rate;
- W_A is the air mass flow;
- W_R is the water mass flow;
- DELTA_T is the mean temperature difference between the airstream and the water;
- U_0 is the overall coefficient of heat transfer for sensible cooling (without dehumidification).

These properties are derived by mathematical relations between the above measurements, e.g. the DELTA_T is calculable from the water and air temperature values. The other not cited properties in the transformations are:

- A_A that is the front surface of the coil;
- A_0 is the external surface of the coil;
- A_I is the internal surface of the coil.

These three properties are directly related and calculated on the basis of the characterisation of the coil geometry in the CAD file.

Notice that, actually also F_A and F_R can be calculated on the basis of the coil geometry. But the detailed instantiation is out of the scope of this section. A more comprehensive instantiation can be found in the fourth chapter.

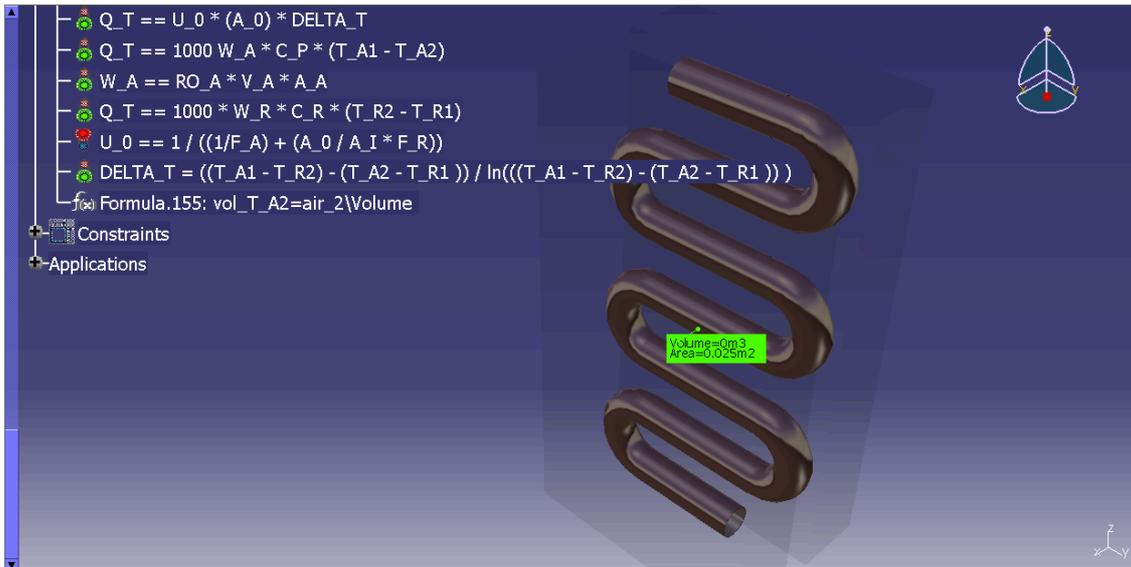


Fig. 2.10 - representation of the bare tube coil: the transformations in CATIA V5.

3.1.3 - How the experience instances can be retrieved and connected

Since all the KR is based on the definition of the *measurements*, the *experiences* have to be retrieved and connected on the basis of the *measurement* STS. Since this definition guarantees the non-ambiguity of the representation, the connection should not be subject to more than one interpretation of the knowledge user, i.e. human (e.g. customers, engineers) or machine.

Customer-to-Product

An experience can be retrieved by defining a set of requirements as *measurement* STSs. Each property has to be defined by *UOM*, *range of values* and *tolerance*. The requirements constrain the *experience* compatibility.

For instance the bare tube coil experience can be retrieved if a customer define a certain temperature value and tolerance to be measured in a given volume. The customer should be asked for other *properties* values: this should characterise the constraint that have to be respected by the designer, e.g. the volume occupied by the coil. For instance, the volume can be related to other measurements (e.g. RO_A, C_P) that characterise the presence of the air in the volume. All the remaining properties become degrees of freedom for the engineer.

In order to have the compatibility of the *experience* with the requirements, the customer defined temperature must have a value included in the *range of values* described by the *experience*. Moreover the *tolerance* in the experience needs to be equal or more accurate than the one of the temperature requirement.

Product-to-Process

If the coil is not available (i.e. in stock), an experience to manufacture the coil is required. Therefore the *measurements* related to the engineer degrees of freedom become the new requirements for the researched coil manufacturing *experience*. As for the air, the coil is a volume characterised by certain measurements, e.g. the thermal conductivity that impacts the values of the F_R and F_A and so the thermal exchange. The manufacturing *experience* is represented as an alteration of these *measurement* in the time, i.e. the deformation of the copper bended by the appropriate machine.

In summary the knowledge retrieval is performed on the basis of a STS correspondence, e.g. a temperature or a thermal conductivity at a certain *time* in a certain *space*.

3.1.4 - An algorithm to prove the non-ambiguity of the representation

Since the represented knowledge is expected to be unambiguous, in this section we build an algorithm to compare two different experiences only on the basis of the syntax of the model. This should demonstrate that even to map the relations between two pieces of knowledge, no-interpretation of the representation is needed.

The algorithm should allow to understand, **by syntax comparisons (i.e. no-required abstraction)**, if either 1) one experience includes the other or 2) the experiences are *inconsistent* or 3) the experiences are not related. Therefore this algorithm should allow to map the relations between different pieces of knowledge and/or different knowledge bases.

In this dissertation, this algorithm is used to show how the comparison between two different experiences can be performed without human interpretation and so it can be automated. The aim is not to present an algorithm for the automatic mapping. The actual purpose is to prove the non-ambiguity of the representation by developing a mapping process completely based on the syntax of the model, i.e. the values of the STSs.

Before the presentation of the algorithm, let us spend some few words on the meaning of the term *inconsistency* in our framework. Indeed, theoretically a set of experiences cannot be inconsistent (as for logic-based representations) because someone collected and analysed the data at the base of the *experience* representation, i.e. *someone observed the represented system behaviour during an experiment*. In this framework, an *inconsistency* is anyway possible but it is related to the implicit hypothesis done when the knowledge is represented as an *experience*, i.e. the *experience* represents all and the only conditions to guarantee that the represented system behaviour is respected. Actually, in order to define an instance of experience, for each *property*, *ranges of values* and *tolerances* have to be specified. Suppose that two *experiences* are compatible, i.e. STSs of each *measurement* are compatible. Suppose now that, even if they describe the same system, i.e. same STSs, the transformations are not the same. This is as if two observation of the same system have been done and the results are completely different. These two *experiences* are considered *inconsistent*. Actually a third experiment should be run to understand the reason of this *inconsistence*. A possible explanation is that in the design of the experiment, one influential factor has been omitted.

In Fig. 2.11, a flowchart representing the algorithm to compare a couple of *experiences* is shown. The most important points of the algorithm are the following:

- *experiences* can have all the STS compatible for each *measurement* (at least on a certain range of values, e.g. experience 1 has X from 10 to 20 and experience 2 to has X from 15 to 22 then experiences are compatible from 15 to 20); in this case if the *transformations* are equivalent (same values for all corresponding properties of the two experiences) then the *experiences* represent the same piece of knowledge (conclusion 2); otherwise there is an inconsistency (conclusion 1 - as described above);

- *experiences* have to be STS compatible, i.e. all *UOM*, *tolerances* and *ranges of values* for *shapes*, *times* and *spaces* for each measurement needs to be compatible; STS incompatibility means that *experiences* are representing non-related pieces of knowledge, i.e. they represent two different systems (conclusion 3);
- the *experiences* share only certain common *measurements*;
 - if the transformations are never equivalent, experiences represent two different pieces of knowledge or they are not consistent (conclusion 4); in order to verify which conclusion is the most appropriate, a further experience observing the Non-Common *Measurements* (NCM) should be performed;
 - if the transformations are always equivalent then the experiences represent the same piece of knowledge (conclusion 5); moreover the NCM are not related to the other measurement in the experiences;
 - if the transformations are equivalent only for certain property values then experiences are equivalent only for those values of the NCM (conclusion 6).

To conclude, the algorithm performs a comparison between two *experiences* only on the basis of the STSs: i.e. volumes, the evolutions of the volumes in the *time*, the *shapes* of the *measurements*. Each one of these comparisons is based on the syntax. Since it is possible to perform this inference only on the basis of the syntax (i.e. the values of *tolerances*, *UOMs* and *range of values*) then the proposed algorithm does not instantiate abstract concept. Therefore it shows how the pieces of knowledge formalised with our KR framework is non-ambiguous. The presented algorithm is applied to a case study in the 4th chapter.

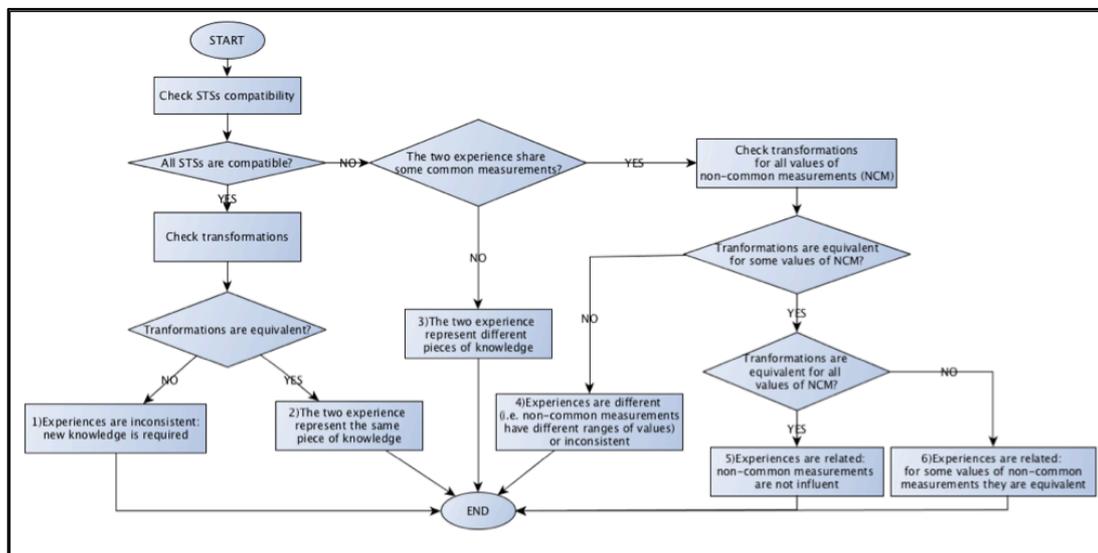


Fig. 2.11 - flowchart representing the algorithm presented in the current section.

3.2 - From design KR to PV design: the generation of the PV alternatives

Once the framework allows an unambiguous representation of the knowledge, it is possible to build a model that integrates design knowledge from different domains. Doing so, a *top-down* PV design should be possible.

Consider for instance a customer that constrains the *experience* applicability imposing values of some *properties*. A customer variety can be considered by defining *ranges of values* in place of simple values for the constrained *properties*. The *ranges of values* of the remaining *properties* represents the degrees of freedom for the engineers. As for the *scale-based* PV design presented in literature (see definitions above), here the varieties are represented as *ranges of values* of some *properties*.

The difference between the *scale-based* PV design is that in our approach not all the represented variants are feasible, i.e. the design of the variants have still to be performed. The feasibility is described by the design knowledge represented as *transformations*. Therefore, in order to understand which process can manufacture the product that can meet a particular customer variant, the feasibility relations between the customer and the processes need to be built (Fig. 2.12).

A set of feasible customer-process couples represents a PV alternative.

Finally, the aim of this section is to show how an *experience* or a properly connected set of *experiences* can be employed to design a PV, i.e. connecting a customer set with alternatives of process variety (Fig. 2.12).

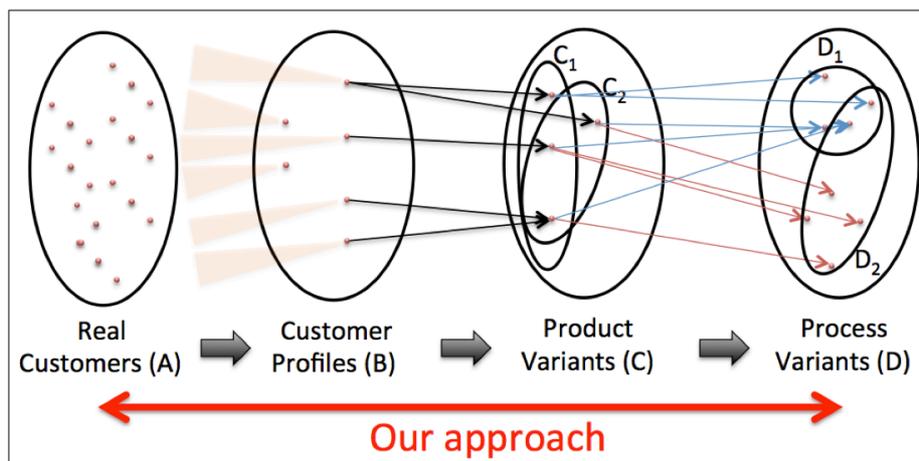


Fig. 2.12 - the connections between variety models coming from different domains provided by the proposed framework.

In order to apply *experiences* to the PV design, two steps are required. 1) The discretisation of the customer domain and the process domain: If the customer variety is represented by a *range of values* for the temperature and the humidity, then a discretisation stage is needed to define the individual customer variant. Once the customer and process variants are individually defined, 2) the *transformations* that links the properties defining the customers and the processes are used to determine the feasibility of each *customer-process* couple. A couple is feasible when a process is able to manufacture a product that can meet the customer requirements.

3.2.1 - The discretisation of the domains

Once the set of measurements and the set of transformations that describe the system are available, a model to design a PV taking into account customers, products and processes is available. Since the

represented knowledge is about product-process design, by using the model it is possible to infer if a particular manufacturing process can realise a product that can meet the requirements represented as a customer profile. The customer and the process varieties are represented by *properties* so they are characterised by a *range of values*, a *UOM* and a *tolerance*. Customer-related properties are the property values constrained by the customer requirements. Process-related properties represent the degrees of freedom that engineers have to manufacture the products that should meet the customer requirements.

The presence of a value for the *tolerance* allows the discretisation of the customer variety and process variety. Doing so a specific customer profile or a specific manufacturing process can be identified in the definition of the customer variety and the process variety. For instance, let us consider that only the property x is constrained by the customer variety. The property x has values from 10 to 15 with a tolerance of ± 1 . From the discretisation of the customer variety six different customers are identified:

- $9 < x_1 < 11$
- $10 < x_2 < 12$
- $11 < x_3 < 13$
- $12 < x_4 < 14$
- $13 < x_5 < 15$
- $14 < x_6 < 16$

Since a variety is represented by a set of *properties*, i.e. a vector, then one variant is a set of intervals like the above ones. For the sake of an example, let us consider a customer variety characterised by some room temperature (x_1) and humidity (x_2) values. Temperature values go from 24 to 25 °C with a tolerance of ± 0.1 °C. Humidity values go from 47% to 53% with a tolerance of ± 1 %. As described above, the discretisation results in eleven temperature ranges and seven humidity ranges. A customer variant is one of the combination of these ranges, e.g. a customer X_i is related to $24.2 < x_1 < 24.4$ and $48 < x_2 < 50$. In this example, customer variants are seventy-seven.

3.2.2 - Testing the customer-process couples feasibility

Once the customer and the process variety are discretised, a test to verify the feasibility of a process solution for a customer variant needs to be performed. This test is a mathematical problem in which the constraints are represented by the *transformations* involved in the connection of *measurements* in the domain of the customer (i.e. requirements) with *measurements* concerning manufacturing processes (i.e. degrees of freedom of the production). A customer-process feasibility test is required for each *customer-process* couple. In order to structure a customer-process feasibility test, a Mixed Integer Non Linear Problem (MINLP). Notice that, MINLP is required to provide flexibility to the formalisation of design knowledge, i.e. the expressiveness of the framework should be not restricted to linear relations between continuous variables. The general model is the following:

$$\text{find } \bar{x}, \bar{y}, \bar{k}$$

st.

$$L_1 < \bar{x} < U_1$$

$$L_2 < \bar{y} < U_2$$

$$L_3 < \bar{k} < U_3$$

$$f_j(\bar{x}, \bar{y}, \bar{k}) = 0, j \in [1, P]$$

$$g_w(\bar{x}, \bar{y}, \bar{k}) \leq 0, w \in [1, Q]$$

where:

- the P functions f and the Q functions g are the linear and/or non-linear constraints representing the *transformations* (equations in Fig. 2.10 for the bare tube coil example),
- the vector x has dimensions representing the *properties* related to the customer variety (e.g. the air temperature to be kept constant in the above example),
- the vector y has dimensions representing the *properties* related to the process variety (e.g. the bending force to apply on the copper tube to deform it),
- the vector k has dimensions representing the *properties* involved in the *transformations* but not belonging neither to the customer variety nor to the process variety (e.g. the coil geometry can be completely defined constraining the impact on customer requirements and the parameters of the coil manufacturing process),
- L_i and U_i are the vectors with the lower and upper bounds for each dimension of x, y, k (i.e. for x and y the bounds are calculated using the tolerances as in the previous section, for k bounds represent the range of values that the properties can assume and they remain invariant during the feasibility tests of all the couples).

Variables can be integer or real.

Each model can be solved by an appropriate MINLP solver. The solver has to verify that a solution respecting all the constraints and the properties ranges exists. If a solution exists then the considered process variant is able to manufacture a product that meet the requirements of the considered customer variant.

To sum up, the customer-process feasibility tests allow to make the links between customer profiles and manufacturing process emerge (Fig. 2.10). Starting from a customer variety, these links should be used to understand how many groups of processes can manufacture a PV able to meet the requirements of all or a part of the customer variants. No decision about which is the best are made at this stage. The result of this stage is only a list of feasible *customer-process* couples that can be grouped to generate the PV alternatives.

3.3 - *Expected effects on the question RQ1 and on the industrial issues*

The presented framework provides a method to build the customer-process feasibility links between the customer and the processes. Moreover the definitions of customers, products and processes are unambiguous. Consequently, the framework provides a method to support a PV design that takes into account at the same time the customer, the product and the manufacturing domain (i.e. an answer to the question RQ1). The presented method allows an approach to PV design that is:

- **top-down**, i.e. the variants (i.e. customer-process couples) are designed at the same time of the variety alternatives (i.e. each group of customer-process couples);
- **scale-based**, i.e. the variety of the customers, products and processes is only characterized as properties;
- **one-step (simultaneous)**, i.e. the invariant and the variant property values of the PV alternatives are defined during the same stage (i.e. the customer-process couples feasibility).

On the other hand, the expected impacts of the proposed framework on the industrial issues are the following.

1. The first industrial issue concerns the correspondences between the real customers and the product variants. 1) If the customer profile definition is based on the *measurements* then this definition is unambiguous and so every seller should always connect the same customer to the same profile. This is possible because the real customer and the sellers can interpret in a unique way the customer profile representations. 2) Since also the relations between customer profiles and product variants are based on the *measurement properties*, the product variant selection should have a result completely independent from the seller interpretation of the customer requirements.
2. The second industrial issue deals with the relations between the product variety and the process variety. Since the connections between experience is non ambiguous, it is possible to formalise a model that integrates the product design knowledge also with the knowledge about the manufacturing process design. This model allows to make decisions considering the customer and the manufacturing process constraints. The PV emerges from the interaction of these two categories of constraints. Letting the product emerge produces two different results: 1) product design efforts are minimised by formalising a knowledge independent from the manufacturing process; 2) all the manufacturing plant heterogeneity can be applied at the same time on the product knowledge, e.g. the same customer profile should result in two different product variants if the requirements are met by two different manufacturing plants.
3. The third industrial issue treat the reuse of the formalised knowledge when a new customer profile is added. Since the KR is not ambiguous, the knowledge is fully reusable only if it is retrieved on the basis of the *measurement properties*. This property of the KR should allow the engineers: 1) to avoid supplementary design efforts because a new design stage should be performed only if a new knowledge is required, i.e. knowledge base extensions; 2) to integrate the older product and process design decisions with the new customer requests.

4 - Conclusions

In this chapter, an approach to answer the first research question,

RQ1 - How to model the variety in order to support a PV design process that at the same time takes into account the design knowledge that relates the customer profiles, the product variants and the manufacturing process variants?

has been proposed (Fig. 2.13). Firstly, a gap in the PV design literature has been identified. The reviewed works do not allow a *top-down* PV design stage. The need of an unambiguous formalisation of the design knowledge has been expressed in order to cope with this issue. Secondly, the fundamentals of the framework have been found in the KR literature. An overview on the two KR schools has allowed the identification of some principles applicable to answer the first research question. Since the *anti-logicist* vision provides solutions for the KR that guarantee no-multiple interpretations, these principles has been transposed to the variety design problem. Finally a framework for the design KR has been presented. The framework should provide a modelling approach able to generate variety model with 1) an unambiguous definition of the customer profiles and 2) unambiguous links between the customer profiles, the product variants and the process variants. These models should be used to generate PV alternatives PVs. A PV alternative is a set of processes able to manufacture a set of products to meet the requirements of a set of customer profiles.

Now these PV alternatives must be evaluated according to a criterion able to consider the PV as a whole. This is the goal of next chapter. Then, in the fourth chapter an implementation of the framework on a prototype is shown. Finally, in the same chapter, a test case is used to test the feasibility of the approach.

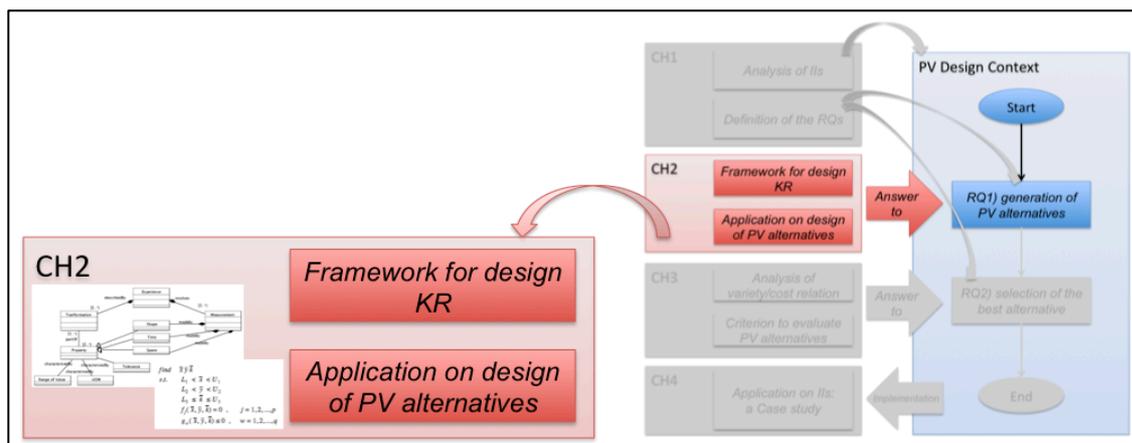


Fig. 2.13 - results of the second chapter.

Chapter 3 Evaluating the PV alternatives

This third chapter aims to review the literature and to propose an approach for comparing the alternatives of PV (Fig. 3.1) generated during the deployment of the approach proposed in the chapter 2. This chapter is divided into three main sections.

The **first section** is dedicated to an overview of the research works about the PV design. In this chapter, the focus is on how authors compare the alternatives of PV (in the second chapter the focus was on the PV modelling approach). A discussion about the reviewed methods aims to identify a gap between the current literature and the answer to the question RQ2 (*How to compare different variety alternatives, i.e. process varieties that allow to manufacture PVs able to meet the same customer variety?*).

Once the gap has been identified, the **second section** of this chapter aims to detail the nature of this gap and introduces the motivation of the proposed approach. An overview of works about the *lot-streaming* will allow to build a mathematical model to be used to analyse the relation between variety and manufacturing costs.

On the basis of this analysis, in the **third section** of this chapter, a new criterion to compare PV alternatives is proposed. A mathematical formulation is presented. Finally, the integration with the output of the feasibility tests (see 2nd chapter) is detailed.

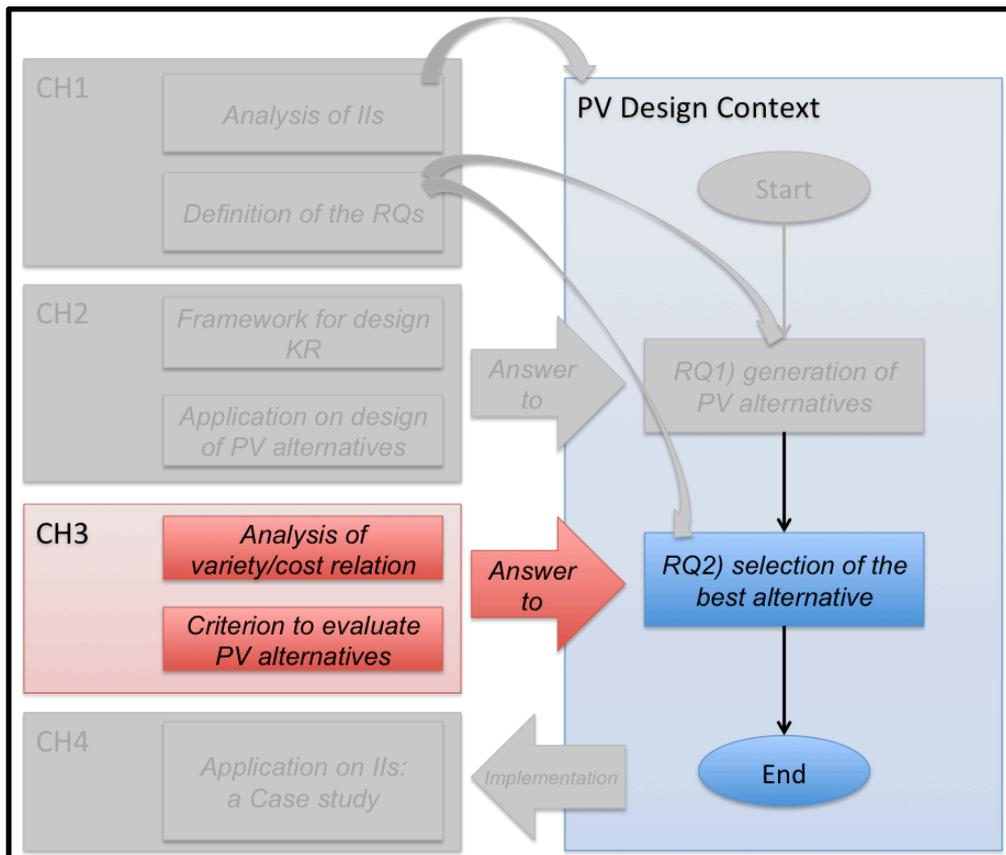


Fig. 3.1 - role of the chapter in the thesis structure.

1 - Overview on PV evaluation criteria

In this section, an overview is provided on works about the criteria used for the PV evaluation and/or optimisation. The aim is to highlight the parameters on which the authors base the PV evaluation.

In the second chapter, the most of the reviewed works on PV modelling already contained criteria to perform the optimisation. But in the chapter 2, the focus was on how they formalise the link between the customer, the product and the process variety. In this chapter the focus is on how they compare two PV alternatives. At the beginning of this section (1.1, 1.2, 1.3), we discuss the works that focus only on the criteria. Then, at the end of the section (1.4), we integrate the works reviewed in the second chapter that contain also the criteria for the PV evaluation and optimisation.

One of the most comprehensive review about product variety is (Fixson 2007). The focus of this review is on the viewpoint of the variety assessment: the author shows how methods that deal with variety can focus on features about the products and the processes. Actually, most of the works are about the first two categories and the papers deal with more than one viewpoint, e.g. product and process at the same time.

As discussed in the previous chapter, the design for variety is usually approached maximising the commonality and the modularity in the PV. The effects of this approach on the design stage are the following (Fixson 2007):

- On customer demands: here there is a trade-off between the potential cost savings, i.e. lower prices, and the revenue generation potential.
- On manufacturing: the impacts are about reduced process complexity, increased economy of scale, fixed cost distribution, assembly costs.
- On the product design: here there is a trade-off between platforming costs and scale economies via design reuse opportunities.
- On the supply chain: the implications deal with the supply network complexity, time horizons, procurements times, the service level and so on.
- On the product use: variety plays an important role on spare optimisation under failure regimes, e.g. maintenance cost minimisation.
- On the delivery time: commonality can contribute to process complexity reduction and so to process parallelisation; moreover the commonality should impact positively the set-up times.

After recalling the major effects involved during the design phase, the following sections will present an overview about quantitative criteria for PV design based on the product and the processes. The section is organised as follows: 1.1) overview on indexes based on product features; 1.2) overview on indexes based on process features; 1.3) examples of implementation of the presented index for the PV design; 1.4) discussion about the applicability of the approaches in literature to answer the question RQ2.

1.1 - Index based on the customer requirements

The Generational Variety Index (GVI) represents the only index reviewed here that consider the customer requirements. The index is presented in (Martin and Ishii 2000). The index considers the evolutions of the PVs. The GVI computation is based on two QFD matrices (Fig. 3.2), i.e. customer requirements to engineering metrics (EM) and EM to components. An assessment about the changes in EM in the future generations of the PV is required. The GVI is calculated as the sum of the expected changes in EMs of all the components used in the PV.

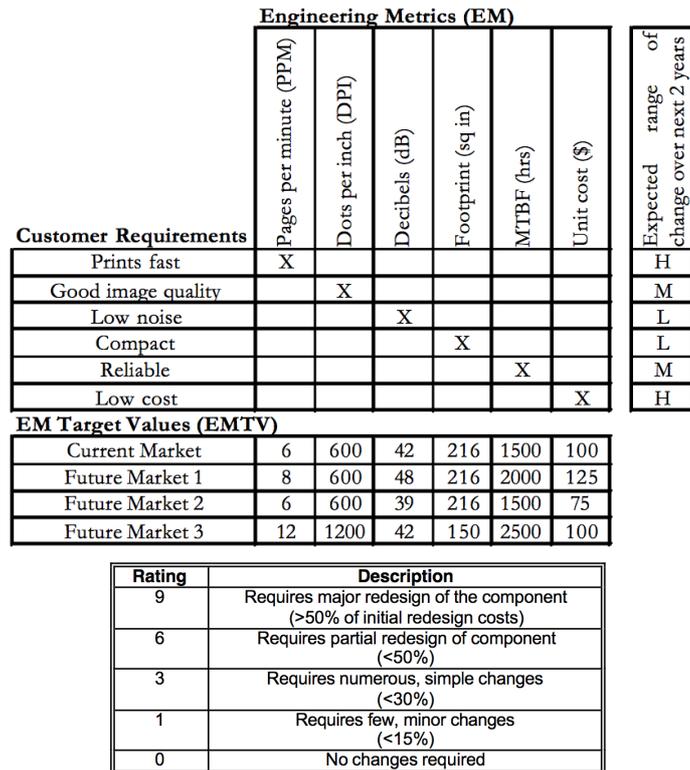


Fig. 3.2 - examples of QFD matrix for the GVI computation, from (Martin,2000)

On the basis of the observations on the ambiguity of the customer needs representation, the **observation** that can be done on this index concerns the **ambiguity** of the customer representation, here performed with the natural language.

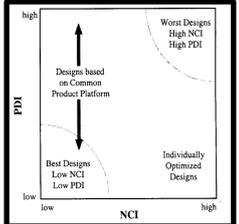
1.2 - Indexes based on the product features

In this section an overview of the indexes for the PV evaluation based on product features is presented (Tab. 3.1). The following parameters are used in more than one criterion: *D* is the number of different components; *I* is the number of product variants; *N* is number of design variables for the PV.

Tab. 3.1a - index for PV evaluation based on product features.

	Index	Variables	Comments
Degree of Commonality Index (DCI), (Collier 1981)	$DCI = \frac{\sum_{j=1}^D \Phi_j}{D}$	Φ_j is the number of immediate parents of a component j , i.e. the number of times j is used.	The index is based on the hierarchical structure of the variants in the PV. The higher is better.
Total Constant Commonality Index (TCCI), (Wacker and Treleven 1986)	$TCCI = 1 - \frac{D-1}{\sum_{j=1}^D \Phi_j - 1}$		from 0 to 1. The higher is better.
Percent Commonality Index (%C), (Siddique, Rosen, and Wang 1998)	$C_c = \frac{100 * c_{-c}}{c_{-c} + u_{-c}}$ $C_{cn} = \frac{100 * c_{-cn}}{c_{-cn} + u_{-cn}}$ $C_{al} = \frac{100 * c_{-al}}{c_{-al} + u_{-al}}$ $C_{aw} = \frac{100 * c_{-aw}}{c_{-aw} + u_{-aw}}$ $\%C = I_c * C_c + I_{cn} * C_{cn} + I_{al} * C_{al} + I_{aw} * C_{aw}$	components (if common c_{-c} , if unique u_{-c}), component-component connections (if common c_{-cn} , if unique u_{-cn}), assembly loading (if common c_{-al} , if unique u_{-al}), assembly workstation (if common c_{-aw}), I_x is the weight corresponding to X .	from 0 to 100 because the sum of the weights is equal to 1. The weights should be selected on the basis of the importance of the factors.
Component Part Commonality Index (CI ^(c)), (Jiao and Tseng 2000)	$CI^{(c)} = \frac{\sum_{j=1}^D \left[P_j \sum_{i=1}^l \Phi_{ij} \sum_{i=1}^l (V_i Q_{ij}) \right]}{\sum_{j=1}^d \left[P_j \sum_{i=1}^m (V_i Q_{ij}) \right]}$	P_j is the price or the internal cost of the component j , m is the number of variants in the PV, V_i is the volume of the variant i , Φ_{ij} is the number of component j used for the product variant i ; Q_{ij} represents the number of components j are required for the product variant i .	
Non-Commonality Index (NCI), (Simpson, Seepersad, and Mistree 2001)	$NCI = \sum_{n=1}^N w_{1n} \frac{\sum_{i=1}^I \left(\frac{\sum_{i=1}^I x_{in}}{i} \right) - x_{in}}{r_n}$	where x_{in} is the value of the design variable n for the product variant i ; r_n is the difference between the upper and lower bounds for the design variable n ; w_{1n} is the weight about the relative difficulty to change the value of the design variable n .	

Tab. 3.1b - index for PV evaluation based on product features.

	Index	Variables	Comments
Performance Deviation Index (PDI), (Simpson, Seepersad, and Mistree 2001)	$PDI = \sum_{i=1}^I w_{2i} \sum_{n=1}^N w_{3n} \left(\frac{ x_{in}^* - x_m }{x_m^*} \right)$	where x_{in}^* is the target value for the design variable n for the product variant i ; w_{2i} is the weight about the importance of the product variant i for the end users; w_{3n} is the weight about the importance of a design variables n for the end user.	low values for both indexes correspond to best PV designs, i.e. low performance deviations and high commonality level. 
Commonality Index (CI), (Martin and Ishii 1997)	$CI = 1 - \frac{D - \max Q_i}{\sum_{i=1}^I Q_i - \max Q_i}$	where Q_i is number of components in the variant i .	from 0 to 1. higher values of the CI are favorable because more product variants have been designed with less unique components.
Commonality Diversity Index (CDI), (Alizon, Shooter, and Simpson 2009)	$CDI_{cd} = \frac{1}{G_{fd}} \sum_{k=1}^{G_{fd}} \left(1 - \frac{n_{com_div_{fdk}}}{\max div_{fdk}} \right)$ $CDI_{F_f} = \frac{1}{D_f} \sum_{d=1}^{D_f} CDI_{cd}$ $CDI_{pv} = \frac{1}{F} \sum_{f=1}^F CDI_{F_f}$	where G_{fd} is number of component subsets (related to the number of PP for the PV); D_f is number of generic components for the function f ; F is number of functions for the PV.	components that should be common (i.e. not allowed diversities) and components that should be different (i.e. not allowed commonalities) are compared per couples. allows to estimate the distance of an existing PV from an ideal set of variants that respect an established level of commonality.
Singular Value Modularity Index (SMI), (Höltkä-Otto and de Weck 2007)	$SMI = \frac{1}{D} \arg \min_{\alpha} \sum_{d=1}^D \left \frac{\sigma_d}{\sigma_1} - e^{-(d-1)/\alpha} \right $	where σ_d is the square root of the eigenvalues of $DSM^T DSM$.	It measures the modularity of the considered system. It is based on the design structure matrix (DSM - matrix describing the strengths of connections between product components).
Non-Zero fraction (NZF), (Höltkä-Otto and de Weck 2007)	$NZF = \frac{\sum_{d=1}^D \sum_{e=1}^D DSM_{de}}{D(D-1)}$		It measures how much the system is connected: a value of 1 means that all components are connected to all other components; a much lower value means that the system is sparsely connected relative to its size.

The only index not included in the Tab. X1 is presented in (Martin and Ishii 2000) and it is matrix-based: the Coupling Index (CPI). The CPI measures the strength of coupling between components of the variants. The CPI is based on the degree of coupling of the components in the PV. Two components are considered as coupled if a change on one component can require the second component to change. In order to compute the index a sensitivity value (Fig. X2) needs to be assigned to each flow that is exchanged by components in the variants of the PV. The CPI is equal to the sum of these sensitivities.

Components receiving specifications	Components supplying specifications					
	PCA / Firmware		Print Cartridge		Carriage Sub-Assembly	
		Sensitivity		Sensitivity		Sensitivity
PCA / Firmware			Resistance	9	Motor speed	3
			# of nozzles	3	Resistance	6
			Nozzle pitch	3	Torque	1
			# of inks	3		
			Firing rate	3		
			Ink viscosity	3		
			Drying time	3		
Print Cartridge	Voltage	6			X dimension	6
	Firmware	1			Y dimension	6
					Z dimension	6
Carriage Sub-Assembly	Voltage	6	Weight	3		
	Firmware	1	X dimension	6		
			Y dimension	6		
			Z dimension	6		

Fig. 3.3 - examples of matrix for the CPI computation, from (Martin,2000).

The main **observation** on the index for the PV evaluation based on the product features is that them all are based on a simple hypothesis:

if the degree of product variety is minimised (i.e. the commonalities are maximised) then the cost of the PV is minimised too.

1.3 - Indexes and methods based on the process features

In this section an overview of the indexes for the PV evaluation based on process features is presented (Tab. 3.2). The following parameters are used in more than one criterion: D is the number of different components; I is the number of product variants.

Tab. 3.2 - index for PV evaluation based on process features.

	Index	Variables	Comments
Component Part Commonality Index (CMC), (Thevenot et al. 2007)	$CMC = \frac{\sum_{d=1}^D n_d (C_i^{\max} - C_i) \prod_{x=1}^4 f_{xn}}{\sum_{d=1}^D n_d (C_i^{\max} - C_i^{\min}) \prod_{x=1}^4 f_{xn}}$	where n_d is the number of product variants in the PV that use the component d ; the four f_{xi} represent the ratios of the number of variants with the same size and shape (f_{1i}), material (f_{2i}), manufacturing processes (f_{3i}) and assembly schemes (f_{4i}) that share d to n_d ; the four f^{max}_{xi} represent the ratios of the number of variants with the same size and shape (f^{max}_{1i}), material (f^{max}_{2i}), manufacturing processes (f^{max}_{3i}) and assembly schemes (f^{max}_{4i}) that share d to the possible number of variants that could have shared d ; C_i , C_i^{\min} , C_i^{\max} are the current, minimum and maximum total cost of d .	It gives a characterisation of the size, shape, manufacturing process, assembly scheme, initial cost and production value.
Differentiation Index (DI), (Martin and Ishii 1997)	$DI = \frac{\sum_{p=1}^P d_p v_p a_p}{Pd_1 I \sum_{p=1}^P a_p}$	where v_i is the number of variants exiting process p , P represents the number of processes, I is the number of offered product variants, d_p is the average time from process p to the sale, d_1 is the average time from beginning of production to sale and a_p is the value added process p .	
Setup Index (SI), (Martin and Ishii 1997)	$SI = \frac{\sum_{p=1}^P v_p c_p}{\sum_{i=1}^I C_i}$	where v_p is the number of different variants exiting the process p , c_p is the cost of setup at process p and C_i is the cost of material, labor, etc. of the product variant i .	
Product Line Commonality Index (PLI), (Kota, Sethuraman, and Miller 2000)	$PCI = \frac{\sum_{d=1}^D \mu_d f_{1d} f_{2d} f_{3d} - \sum_{d=1}^D 1 / \mu_d^2}{D * I - \sum_{d=1}^D 1 / \mu_d^2} * 100$	where n_d is the number of variants that share the component d , the f_{1i}, f_{2i}, f_{3i} are the ratios between the actual and the ideal component sharing according three viewpoints: shape and size (f_{1i}), the manufacturing and materials (f_{2i}), assembly schemes (f_{3i}).	It includes a characterisation of size, shape, manufacturing process and assembly scheme factors. The index values go from 0 (i.e. no commonalities or no compatibility about shape, materials, etc.) to 100.

The only approach for PV evaluation not included in the table above is presented in (Park and Simpson 2007). Here, the authors use the Activity-Based Costing (ABC) method to allocate indirect costs to the parts of the PV. ABC allows to capture the product resource consumption. The method is based on three guidelines:

- **direct-costs** (*variable costs that can be directly attributed to a particular job or operation. Direct material and direct labour are traditionally considered direct costs.*⁶) are traced to the product. Costs including material, labour and other resources consumed only by one product are allocated.
- **indirect-costs** (*Costs that are not directly incurred by a particular job or operation. Certain utility costs, such as plant heating, are often indirect.*⁷) allocated to multi-purpose resources, cost centres (e.g. major production machines, human resources) and activities.
- activity cost drivers allow to relate the activities to the products that make use of them, e.g. the setup time relate the setup activity (and the related costs) the to the products that employ it.

In the PV design, the allocation of the activity costs on the PV components provides a more comprehensive criterion to make decisions especially about product and process commonalities.

As in the previous section, the main **observation** on the index for the PV evaluation based on the process features is that them all are based on a simple hypothesis:

if the degree of process variety is minimised (i.e. the commonalities are maximised) then the cost of the PV is minimised too.

We will demonstrate in the section 2 - that making this hypothesis can degrade the performance of the global system.

1.4 - PV design with index implementations

An example of PV design supported by the use of two presented indexes can be found in (Thevenot and Simpson 2007) and (Ye et al. 2009). Here the authors provide a method for PV design based on the CDI and the CMC. The combined use of both the indexes allow to cover PV evaluation problems involving information about multi-platform, product functions, value variety, manufacturing processes, assembly schemes, component cost. The PV design is divided in a two-step method: 1) the preliminary project in which the CDI is used to maximise function commonalities; 2) the detailed studies in which both the indexes are used to ensure that functions and related components preserve an appropriate level of commonality. In (Ye et al. 2009) the authors pair the CDI and the CMC with an third factor that should take into account the impact of company strategic factors on the PV commonality. According the selected factors the third factor helps compare the company with the competitors.

After a review of 6 different indexes (DCI, TCCI, PCI, %C, CI, CI[©]), the authors in (Thevenot and Simpson 2006) propose a four-step procedure for PV design: 1) the focus of the company (i.e. number

⁶ <http://www.apics.org/industry-content-research/publications/apics-dictionary>

⁷ <http://www.apics.org/industry-content-research/publications/apics-dictionary>

of common components, non-differentiating components, number of common connections and assembly, cost of the components) determines which is the best index to use; 2) computation of the TCCI and the DCI to have a general idea about the PV quality; 3) computation of focus-specific index; 4) redesign of the PV on the basis of the index results.

Two matrix-based approaches for product variant clustering are presented in (Ye and Gershenson 2008) and (Rojas Arciniegas and Kim 2010). In (Rojas Arciniegas and Kim 2010) the focus of the optimisation is on the component arrangement in the products. The authors develop an automatic method to select the PP and design the PV. The method is based on the DSM and on a function-component matrix. The optimisation of the component arrangement in the products allows to find the PP that facilitate the module sharing between the variants in the PV. The authors in (Ye and Gershenson 2008) merge market analysis and conceptual engineering knowledge to identify the appropriate commonality and variety trade-off. A matrix based design tool help engineers to cluster product attributes and to verify the attribute's occurrences across the market segment.

1.5 - *The gap in the literature: the relation variety/cost*

As shown in the literature review, most of works in PV design focus on commonality maximisation. Indexes deal with commonalities at product, function and component level. Works coping with process similarities are less common. Finally criteria based on process costs are really rare, especially when the allocation of the indirect costs are considered (the same critic can be found already in the 1997 in (Martin and Ishii 1997)).

A classification of the evaluation criteria is given in Tab. 3.3. The table includes the works about the PV modelling already reviewed in the previous chapter.

- **Customer** - The only index discussed in this chapter that takes into account a customer view is the GVI: this index considers how the PV is adaptable to the evolution of the customer requirements. Instead customers are mostly represented in PV optimisation models (discussed in the previous chapter) as customer demand related to some expected product performances. In the first case, the **ambiguity of the customer requirement** representation makes not feasible the estimation of the PV evolutions. In the second case, the representations of a *tolerance* (as in the section 3 of the previous chapter) associated to each performance should allow to understand the limits of the customer acceptability for a given product variant feature (i.e. module and/or attribute). Therefore **there should be no-need to have a customer expected performance to be optimised**.
- **Product** - Product-based criteria are mainly about the commonality maximisation between functions, modules and interfaces. As for expected performances in the previous point, when product design variables are considered, usually it is for minimising the performance deviations. Also in this case, **there should be no-need to have a product feature to be optimised**.
- **Process** - There are less works that consider the production processes for the PV assessment. Usually the processes are considered to compare the way two or more product modules are manufactured or assembled. These comparisons should allow the module

commonalisations and so the reduction of the degree of variety in the PV. **Only few works focus on the manufacturing costs.**

Tab. 3.3 - criteria for variety assessment.

	Customer Profiles	Product Variants	Process Variants
(Ye and Gershenson 2008), (Tucker and Kim 2008; Kazemzadeh et al. 2009; Jariri and Zegordi 2008)	X		
(Collier 1981; Wacker and Treleven 1986; Jiao and Tseng 2000; Martin and Ishii 1997; Hölltä-Otto and de Weck 2007), (Siddique, Rosen, and Wang 1998), (Alizon, Shooter, and Simpson 2009; Rojas Arciniegas and Kim 2010), (Martin and Ishii 2000), (Zacharias and Yassine 2008)		X	
(Martin and Ishii 1997; Siddique and Wilmes 2007)			X
(Simpson, Seepersad, and Mistree 2001; Khajavirad and Michalek 2009; Khajavirad, Michalek, and Simpson 2009; Qu et al. 2010; Fellini et al. 2004; Fellini et al. 2005), (L. Li and Huang 2009)	X	X	
(Park and Simpson 2007; Fujita, Amaya, and Akai 2013; Shahzad and Hadj-Hamou 2013; Agard and Penz 2009; Lamothe, Hadj-Hamou, and Aldanondo 2006; Khalaf, Agard, and Penz 2010a; Khalaf, Agard, and Penz 2010b), (Michalek et al. 2006), (Williams et al. 2007), (Kota, Sethuraman, and Miller 2000; Thevenot et al. 2007; Thevenot and Simpson 2007; F. Gao, Zhang, and Xiao 2013)		X	X

Indeed, methods that deal with a real estimation of costs are rare. In the reviewed works, 1) the *direct* costs (i.e. the costs that depends on the process and resources that are required to manufacture the product variants, e.g. materials, machining time) are the only considered and 2) the degree of commonality (or, inversely, the degree of variety) is the main criterion employed to determine the impact of the PV on the costs. But,

*in our vision, the direct costs and the degree of commonality are not enough to model the impacts of a PV on the **manufacturing costs**. In other words, two PVs with the same direct costs can have different total manufacturing costs and these costs can be not proportional to the degree of variety of the two PVs. If applied in this case, the reviewed criteria should provide a non-optimal result.*

In order to support this position, in the next section, a review of the principle of the *lot streaming* problem allows to present the fundamentals to model the relations between the production system type, the lot size, the lot number and the manufacturing time and costs. On the basis of the works on *lot streaming*, a model is proposed in the third section of the chapter: here **two PVs with the same direct costs and two different degree of variety are related to their manufacturing costs**. The

aim is to show that **the PV with a lower degree of variety can be also the one with the lowest related manufacturing costs.**

2 - The impact of the variety on the manufacturing costs: *lot streaming*

In order to show that the minimisation of the degree of variety (or maximisation of commonality) of a PV is not enough to approximate the manufacturing costs, we want to demonstrate that an optimal *lot streaming* can compensate the presence of a too high degree of variety. Since the *lot streaming* problem allows to model the links between the variety and the costs and durations of the manufacturing processes, in this section the fundamentals of the *lot streaming* models are presented.

The *lot streaming* is a technique in which the production lots that have to be processed are split into sub-lots and overlapping operations are performed on the different manufacturing stages (Chang * and Chiu 2005). Doing so, an optimal *lot streaming* can reduce the production duration, the cycle time and average the work-in-process inventory (Cheng, Mukherjee, and Sarin 2013).

Indeed, in (Jacobs and Bragg 1988), the authors show how the time earned with an optimal lot split strategy can largely compensate the time lost for extra setups due to process variety. To clarify the meaning of extra setup, let us consider two different lots (i.e. A and B) on a one-machine production system. The machine requires different setups for the two lots. Therefore if each of the lot is split in two equal sub-lots and the production planning is A-B-A-B, then three setup times are required to finalise the production. Since without splitting the lots only one setup would be required, two of the three required setups are extra. This means that commonality and process costs cannot characterise completely the production time and so the production costs related to a set of processes.

Works on *lot streaming* can be classified according the production system, the number of products, the flexibility, etc. A comprehensive review can be found in (Cheng, Mukherjee, and Sarin 2013). The authors classify the studies on *lot streaming* according nine dimensions:

- Machine configuration refers to the arrangement of the machines: flow shop (i.e. each type of job goes through the same machines), job shop (i.e. each different job can have different machines on which it have to be processed), open shop (i.e. there are no constraints job-machine), parallel machines, hybrid systems and assembly systems. Moreover each configuration can be multi-stage.
- Number of production types: single products or multi-products.
- Sub-lot type: consistent (i.e. the size of the sub-lot remains constant over machines), equal (i.e. all the sub-lot have the same size) and variable.
- Idling: no-idling (i.e. a lot have to be processed immediately after the completion of the previous one), intermittent idle (i.e. an idle time is admitted).
- Number of sub-lots: this number can be fixed a priori or determined by the solution.
- Sub-lot size: the size of the sub-lot can be considered as continuous or discrete.

- Setups: lot-attached setup (i.e. the setup can be started only at the arrival of the lot at the machine), lot-detached (i.e. the setup can be performed before the arrival of the lot to be processed).
- Transfer time or removal time: both refer to the time required to move a sub-lot from a machine to another; during a transfer time, the machine is not available; on the contrary, during a removal time, the machine can start the next sub-lot.
- Objective function: studies about *lot streaming* optimise time (e.g. process durations, mean flow time) or costs (e.g. production, inventory).

According to three reviews of papers about *lot streaming*, (Chang * and Chiu 2005; Sarin and Jaiprakash 2007; Cheng, Mukherjee, and Sarin 2013), most of the studies concern the flow shop and especially a single-product flow shop. The complexity of the problem increases in the job shop and in the assembly system cases: most of works about these two production systems approximate the lot-streaming problem considering only one product and/or one single lot and/or a fixed number of lots etc.

Our aim is to show that the minimisation of the degree of variety of a PV is not enough to approximate the manufacturing costs.

In order to support this thesis, we should model a situation in which the PV with the highest degree of variety corresponds to the lowest manufacturing costs. The model to be employed should have to following requirements.

- The two PV alternatives have equal *direct* costs.
- The model has to constrain as less as possible the optimal *lot streaming* solution. Therefore the sub-lot sizes and the sub-lot numbers should not be fixed a priori.
- Obviously, the model considers more than one product variant.
- The model represents an *assembly system*. Indeed:
 - the *flow shop* can be considered as a special case of an *assembly system* (Fig. 3.4) in which
 - at the first stage, one machine A has a process times much higher than the other machines at the same stage and
 - at the second stage, the assembly workstation needs only one component from the machine A;
 - *open* and *job shops* models are out of the scope because they model more than one production flow; since in our case we want to deal with equal *direct* costs and the production flow impacts them, the model of a single flow is considered;
 - the *parallel machines* system can be seen as an extension of the above systems; it has the same behaviour but it allows always lower waiting time because the availability of the workstations at the same stage is higher.

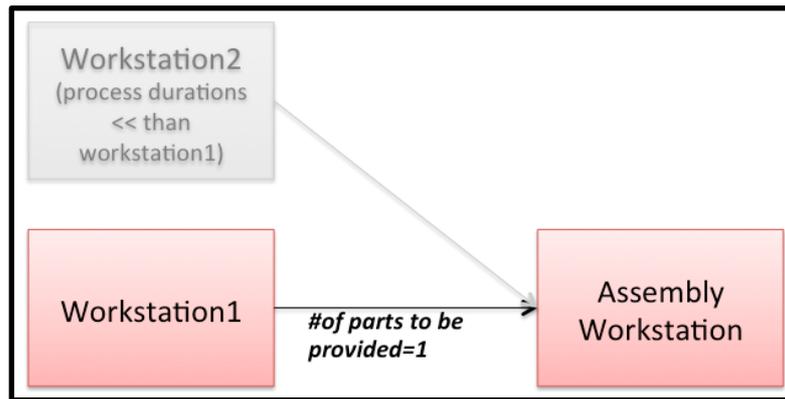


Fig. 3.4 - scenario in which an assembly system can be modelled as a flow shop.

Works on *assembly systems* are rare (Cheng, Mukherjee, and Sarin 2013). The models in literature have constraints on sub-lot sizes (e.g. equal size in (Ruben and Mahmoodi 1998; Yao and Sarin 2014), number of product (e.g. single-product in (Moily 1986; Sarin, Yao, and Trietsch 2011; Chan, Wong, and Chan 2008; Wong, Chan, and Chan 2009)). Since a discussion based on the global optimal values is required, in the next section a model for an assembly system, multi-product, flexible on sub-lot size is formalised. The following discussion on results should provide 1) a validation of the need of a more comprehensive PV evaluation criterion (considering the effects of the sub-lot strategy and the operations overlapping) and 2) some more details about how to model this criterion.

3 - Our approach: PV alternatives evaluation

In this section our approach to a more comprehensive assessment of the variety-cost relation is presented. In the first part, an *assembly system lot-streaming* model is presented to compare two PVs with the same *direct* costs. On the basis of the analysis of the results of the PVs evaluation, a new criterion is proposed. Finally the new criterion is implemented to perform the evaluations between customer-process couples generated from the solution of the mathematical model formalised in the chapter 2.

3.1 - About the relation between the variety and the costs

In this section, the aim is to show that the relation between the degree of variety and the expected manufacturing costs should describe also the how the lots are split and the operations overlapped. For this experience, we test, on an exact algorithm, the costs of two process varieties (PRV), i.e. the process varieties related to some PVs. The model does not fix values of the lot size and the lot number but preserve the treatability (i.e. the algorithm that has to solve the model finds a global optimum). The features of the modelled *assembly system* (Fig. 3.5) are the following.

- It is a two stages assembly system.
- The assembly process involves only two components, X1 manufactured on P1 and X2 manufactured on P2.
- At the first stage, only the manufacturing time of X1 is considered. X2 is considered as always available, i.e. the manufacturing time is largely shorter than the one related to X1.

- The lot number is fixed.
- Each workstation has a setup times (Tab. 3.4).
- The lot size is not constrained and is discrete.
- Idle times are allowed.
- The setups are attached.

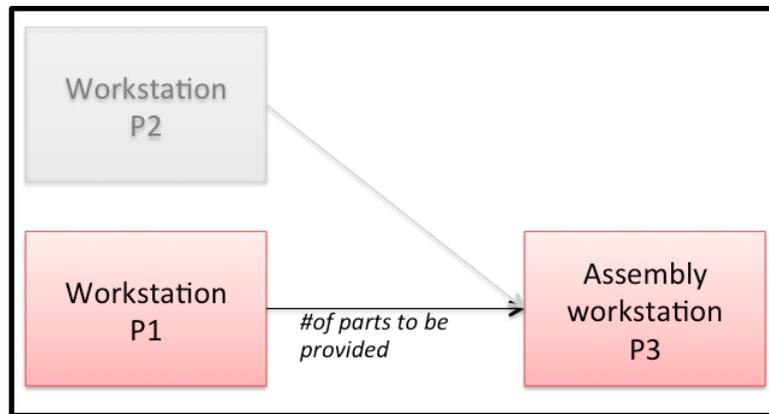


Fig. 3.5 - modelled assembly system.

The two PRVs have the following features.

- The first one has three product variants, A, B and C. The machining (P1) and assembly times (P3) are shown in the Tab. 3.5. The numbers in italic are the quantity to be processed.
- The second PRV has two product variants, D and E. The machining (P1) and assembly times (P3) are shown in the Tab. 3.6. The numbers in italic are the quantity to be processed.
- The two PRVs require the same span of time to be manufactured (i.e. 1280 units of time - Tab. 3.6). If the process costs does not depend on which product is manufactured, the direct process costs are equal for the two mixes.

Tab. 3.4 - data about the working stations.

	setup time	max number of sub-lots
P1	10	4
P3	3	4

Tab. 3.5 - data about the first PRV.

	A	B	C	total time = 1280
P1	100 (2)	100 (2)	150 (1)	550
P3	1 (230)	1 (200)	2 (150)	730

Tab. 3.6 - data about the second PRV.

	D	E	total time = 1280
P1	200 (2)	150 (4)	1000
P3	2 (40)	2 (100)	280

The mathematical formulation of the model with the MINLP is the following.

$$e_{i_x-1}^x \leq s_{i_x}^x \quad \forall i_x \in [2, \dots, R_x], j \in [1, \dots, V], x \in \{P1, P3\} \quad (1)$$

$$s_{i_x}^x \leq e_{i_x}^x \quad \forall i_x \in [1, \dots, R_x], j \in [1, \dots, V], x \in \{P1, P3\} \quad (2)$$

$$e_{i_x}^x - s_{i_x}^x = \sum_j q_{i_x j}^x t_{xj} z_{i_x j}^x + set_x(x) \quad \forall i_x \in [1, \dots, R_x], j \in [1, \dots, V], x \in \{P1, P3\} \quad (3)$$

$$q_{i_x j}^x, q_{jT}^x, q_{jA}^x \in N^+, z_{i_x j}^x \in \{0,1\} \quad \forall i_x \in [1, \dots, R_x], j \in [1, \dots, V], x \in \{P1, P3\} \quad (4)$$

$$q_{i_x j}^x = 0 \rightarrow z_{i_x j}^x = 0 \quad \forall i_x \in [1, \dots, R_x], j \in [1, \dots, V], x \in \{P1, P3\} \quad (5)$$

$$q_{i_x j}^x \neq 0 \rightarrow z_{i_x j}^x = 1 \quad \forall i_x \in [1, \dots, R_x], j \in [1, \dots, V], x \in \{P1, P3\} \quad (6)$$

$$z_{i_x j}^x, z_{(i_x-1)j}^x, x_1 \neq x_2 \rightarrow set_x(x) = set(x) \quad \forall i_x \in [1, \dots, R_x], j \in [1, \dots, V], x \in \{P1, P3\} \quad (7)$$

$$z_{i_x j}^x, z_{(i_x-1)j}^x, x_1 = x_2 \rightarrow set_x(x) = 0 \quad \forall i_x \in [1, \dots, R_x], j \in [1, \dots, V], x \in \{P1, P3\} \quad (8)$$

$$\sum_i q_{i_x j}^x = q_{jT}^x \quad \forall i_x \in [1, \dots, R_x], j \in [1, \dots, V], x \in \{P1, P3\} \quad (9)$$

$$\sum_i q_{i_{P3} j}^{P3} = 1 \quad \forall i_{P3} \in [1, \dots, R_{P3}], j \in [1, \dots, V] \quad (10)$$

$$\sum_j z_{i_x j}^x \leq 1 \quad \forall i_x \in [1, \dots, R_x], j \in [1, \dots, V], x \in \{P1, P3\} \quad (11)$$

$$z_{i_{P3} j}^{P3} A_{i_{P3} j}^{P3} \leq s_i^{P3} \quad \forall i_{P3} \in [1, \dots, R_{P3}], j \in [1, \dots, V] \quad (12)$$

$$A_{i_{P3} j}^{P3} = \min_{i_{P1}} (e_{i_{P1}}^{P1} u_{i_{P3} i_{P1} j}) \quad \forall i_{P1} \in [1, \dots, R_{P1}], i_{P3} \in [1, \dots, R_{P3}], j \in [1, \dots, V] \quad (13)$$

$$(Aq_{i_{P1} j}^{P1} - \sum_{k=1}^{i_{P3}} q_{kj}) \leq 0 \rightarrow u_{i_{P3} i_{P1} j} = 1 \quad \forall i_{P1} \in [1, \dots, R_{P1}], i_{P3} \in [1, \dots, R_{P3}], j \in [1, \dots, V] \quad (14)$$

$$(Aq_{i_{P1} j}^{P1} - \sum_{k=1}^{i_{P3}} q_{kj}) > 0 \rightarrow u_{i_{P3} i_{P1} j} = M \quad \forall i_{P1} \in [1, \dots, R_{P1}], i_{P3} \in [1, \dots, R_{P3}], j \in [1, \dots, V] \quad (15)$$

$$Aq_{i_{P1} j}^{P1} = \left\lfloor \sum_{p=1}^{i_{P1}} \left(\frac{q_{pj}}{q_{jA}} \right) \right\rfloor \quad \forall i_{P1} \in [1, \dots, R_{P1}], j \in [1, \dots, V] \quad (16)$$

The goal of the model is to minimise the time spent to manufacture and assemble the PRV, i.e. total lead time⁸, $S_{R(P3)}^{P3}$.

The constants of the model are:

- P1 and P3 are the workstations, respectively a machine and an assembly station;
- R(x) represents the max number of sub-lots admitted on P1 and P3;
- V is the number of product (and process) variants;
- t_{xj} are the processing time of the product variant j on the workstation x;

⁸ <http://www.apics.org/industry-content-research/publications/apics-dictionary>

- $\text{set}(x)$ is the setup time on the workstation x ;
- q_{jT}^x is the total quantity of j that have to be processed on x ;
- q_{jA} is the quantity of j required to the station P3 to start a job.

The variables of the model are:

- $s_{i_x}^x$ and $e_{i_x}^x$ are the starting and completion time of a sub-lot i_x on P1 or P3;
- $q_{i_x j}^x$ is the quantity of variant j processed in the sub-lot i_x on P1 or P3;
- $Aq_{i_{P1} j}^{P1}$ is the available quantity of j at the end of the sub-lot i_{P1} on P3;
- $A_{i_{P3} j}^{P3}$ is the time availability of j before the sub-lot i_{P3} is processed on P3;
- M is a number much bigger than the possible end times on P3.

The constraints (1-2) are about the succession of the starting and completion time. The constraint (3) describes the job duration of the workstations. The constraints (4,5 and 11) avoid that two different sub-lots are processed on the same station at the same time. The constraints (7-8) determine if a setup is required or not, i.e. if the previous sub-lot was different or not from the next one. The constraints (9-10) fix the total quantities of the variants to be processed on the workstations. The constraint (12) constrains the jobs on P3 to start only when at least the quantity q_{jA} has been processed on P1 and so it is available for the assembly stage. The constraints (13-16) is about determining the availabilities of the products on P3 before a certain sub-lot i_{P3} .

Tab. 3.7 - result of the experience on the lot streaming model for PRV1.

	sub-lot 1	sub-lot 2	sub-lot 3	sub-lot 4
start time P1	-	75	160	370
end time P1	75	160	370	570
start time P3	75	225	378	611
end time P3	225	378	611	811

Tab. 3.8 - result of the experience on the lot streaming model for PRV2.

	sub-lot 1	sub-lot 2	sub-lot 3	sub-lot 4
start time P1	-	610	810	1010
end time P1	610	810	1010	1010
start time P3	610	864	907	1010
end time P3	713	907	1010	1050

The solutions found by the solver are global optima. The time distribution of the PRV 1 is in Tab. 3.7. The time distribution of the PRV 2 is in Tab. 3.8. The most noticeable result is that even if the PRV 1 has a degree of variety higher than the second PRV, it has a total production time lower (811 units of time) than the one related to the PRV with a lower variety (1050 units of time).

This case is in contradiction with the hypothesis that the minimisation of the degree of variety (inversely, the maximisation of the PV commonalities) can approximate the minimisation of the manufacturing costs (as in all the above discussed criteria). This means that something more has to be modelled to represent the variety/cost relation.

The aim of the next section is to analyse this result and to understand its nature. On the basis of this analysis, a new criterion that can model in a more comprehensive fashion the variety-costs relation is proposed.

3.2 - An approximation of the waiting time

As shown in the Tab. 3.7 and Tab. 3.8, the higher variety seems making easier the scheduling of the jobs process on the workstations P1 and P3. If we look at the job durations of the P3, something more about the reason of this gap between the two PRVs can be understood.

- Indeed, the first sub-lot on P3 for the PRV1 starts much later than the first one on the same workstation and for the PRV2.
- Moreover, the completion and starting times of the sub-lots on P3 for PRV1 are equal, i.e. the workstation P3 has never needed to stop between one sub-lot and another.

Since a sub-lot on P3 can start only if the necessary number of component manufactured on P1 is available, the reason about both these two points is due to the job duration on P1. Indeed, the workstation P3 has to wait much longer for the PRV2 than for the PRV1. Since the total processing time is the same between the two PRVs, the reason about the gap can be due only to the distribution of the times on the two workstations.

Indeed, in the Tab. 3.7 is shown how the times for PRV1 are much better balanced (500 for P1 and 730 for P3) than for PRV2 (1000 for P1 and 280 for P3). And this feature is strictly related to the nature of the process to manufacture the PRVs. Therefore this gap is directly related to the variety. In order to have a comprehensive metric to assess the variety-related costs, this point should be considered.

An exact model of this relation is not possible for two main reason: 1) it would be not treatable as can be read in the literature reviews (Chang * and Chiu 2005; Sarin and Jaiprakash 2007; Cheng, Mukherjee, and Sarin 2013) and moreover, 2) it is out of the scope of this work. Indeed, here the aim is to compare two varieties and not to give an absolute estimation of how much a variety costs to the enterprise. Therefore our proposition consists in an approximation of the attitude of a PV to be effectively scheduled. In order words, we propose a criterion to approximate how the time balancing of the processes related to a PV can impact the manufacturing costs related to the PV.

The main idea behind the proposed approximation is shown in Fig. 3.6. The aim is to approximate the computation of the *waiting* time of a couple of workstations that are on the different stages of a production system.

The **waiting** time of a generic workstation *MX* is calculated as the difference between the time to manufacture or assemble a part on *MX* and the time to manufacture or assemble on the workstation *MY* the minimum quantity of components that *MX* needs to start a job.

As shown in Fig. 3.6 on two machines M1 and M2, the *waiting* time on M1 is equal to the sum of the *waiting* time of each variant in the PV.

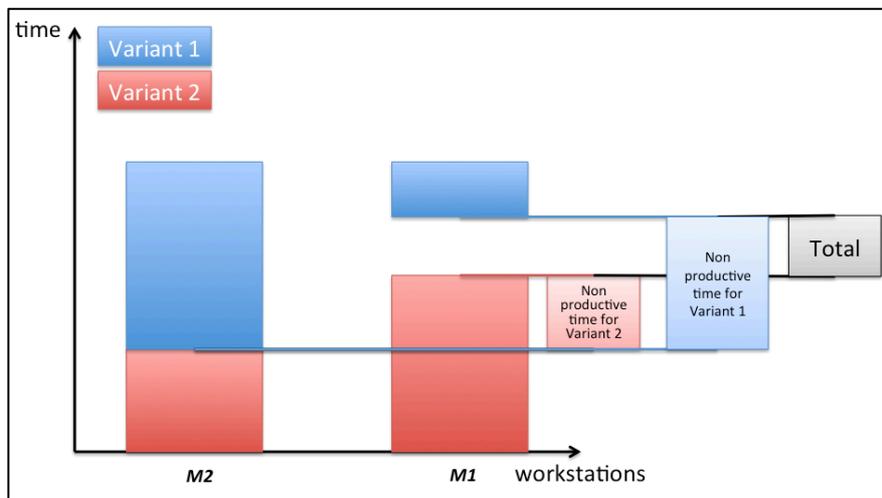


Fig. 3.6 - The graphical representation of the waiting time as in our approach.

The mathematical formalisation of this relation is the following.

$$C_i = \sum_{q=1}^2 C_{qD} \left| \sum_{m \in k_q} \left[\left(t_{mUM_{1q}} - t_{mUM_{2q}} Q_{mLM_{2q}} \right) l_m \right] \right| \quad (17)$$

where C_i is the total cost related to the *waiting* time; C_{qD} is the cost of a unit of *waiting* time on the manufacturing plant q ; m are the process variants in the plant q ; t_{mUM_1} and t_{mUM_2} are the times needed to process a unit of the job m on M1 and on M2 in the manufacturing plant q ; $Q_{mLM_{2q}}$ is the size of a lot for the job m on M2; l_m is the number of lots of m .

The content of the absolute value expresses how much time should be *waiting* in a certain manufacturing plant on the workstations M1-M2. Notice that, the lot size for each variant j is constant and equal to $Q_{mLM_{2q}}$.

The changes to the formulation of C_i to model production systems different from a two-stage assembly system are the following.

- If the machining time at the first stage are comparable, the t_{mUM_2} to be considered is the one that maximise $\left| t_{mUM_1} - t_{mUM_2} \right|$.
- If the *assembly* system has more than one stage, the C_i should be calculated for each couple of stages.
- To consider the presence of *parallel* workstations, the unitary processing time should be divided per the number of parallel machines.
- For *open* or *job* shops a C_i term should be considered for each couple of stages.

In the next section, the proposed criterion should complete the mathematical formulation of the model that is expected to answer to the question RQ2: the formulation is generalised to take into account more stages and machines.

3.3 - The implementation of the new criterion for the PV evaluation

In this section the implementation of the proposed criterion with the output of the method proposed in the second chapter is detailed.

At the end of the second chapter we proposed a method to assess the feasibility of the customer-process couples. These couples are normally associated to a customer demand. Now a mathematical model is needed to assess which is the best set of couples to cover all the customer profiles, i.e. all the customer profiles that are possible to link with at least one feasible manufacturing process.

Below there is the mathematical formulation of the general problem with:

- N_p production systems,
- each production system q (Fig. 3.7) has O_q stages,
- each stage o_q has M_{oq} assembly workstations,
- each assembly workstation μ_{oq} has one workstation μ_{oq}^* with processing times that maximise the sum, for each process variant, of $\left| t_{Um\mu_{oq}} - t_{Um\mu_{oq}^*} \right|$,
- each assembly workstation μ_{oq} has $E_{\mu_{oq}}$ workstation that manufactures its components or subassemblies.

$$\text{Min}(C_d + C_s + C_i) \tag{17}$$

$$C_d = \sum_{i=1}^{V_c} \sum_{j=1}^{V_p} C_{ij} Q_{ij} \tag{18}$$

$$C_s = \sum_{q=1}^{N_p} \sum_{m \in k_q} \sum_{o_q=2}^{O_q} \sum_{\mu_{o_q}=1}^{M_{o_q}} \sum_{\epsilon_{\mu_{o_q}}=1}^{E_{\mu_{o_q}}} \text{Set}_{m\mu_{o_q}} \left(\frac{t_{Um\mu_{o_q}} S_{m\mu_{o_q}}}{l_{m\mu_{o_q}} t_{Um\mu_{o_q}^*} S_{m\mu_{o_q}^*}} \right) \quad \text{card} \left(\bigcup_q k_q \right) = V_p, \forall x, y \in q, x \neq y, k_x \cap k_y \equiv \emptyset \tag{19}$$

$$C_i = \sum_{q=1}^{N_p} C_{qD} \sum_{m \in k_q} \sum_{o_q=1}^{O_q-1} \sum_{\mu_{o_q}=1}^{M_{o_q}} \left[\left(t_{Um\mu_{o_q}} - t_{Um\mu_{o_q}^*} \right) Q_{Lj\mu_{o_q}} \right] l_{m\mu_{o_q}} \quad \text{card} \left(\bigcup_q k_q \right) = V_p, \forall x, y \in q, x \neq y, k_x \cap k_y \equiv \emptyset \tag{20}$$

$$Q_{Lj\mu_{o_q}} l_{j\mu_{o_q}} = \left(\sum_{i=1}^{V_c} Q_{ij} \right) \prod_{s=2}^{o_q} S_{j\mu_s \mu_s} \quad \forall q \in [1, N_p], j \in k_q, o_q \in [1, O_q], \mu_1 \in [1, M_{o_q}], \exists! \mu_{o_q}^* \in [1, E_{\mu_{o_q}}], \epsilon_{\mu_{o_q}} \in [1, E_{\mu_{o_q}}], \epsilon_{\mu_{o_q}} = \mu_{o_q}^*, \mu_s = \epsilon_{\mu_{s-1}} \tag{21}$$

$$S_{j\mu_{o_q} \mu_{o_q}} \leq Q_{Lj\mu_{o_q}} \quad \forall q \in [1, N_p], j \in k_q, o_q \in [1, O_q], \mu_{o_q} \in [1, M_{o_q}], \exists! \mu_{o_q}^* \in [1, E_{\mu_{o_q}}] \tag{22}$$

$$l_{j\mu_{o_q}} \cdot Q_{ij} \cdot Q_{Lj\mu_{o_q}} \in N^+ \quad \forall i \in [1, V_c], q \in [1, N_p], j \in [1, V_p], o_q \in [1, O_q], \mu_{o_q} \in [1, M_{o_q}], \exists! \mu_{o_q}^* \in [1, E_{\mu_{o_q}}] \tag{23}$$

The constant in the model are the following:

- C_{ijT} is the cost of the process j to meet the requirements of the customer i ;
- $Set_{m\epsilon_{\mu_{oq}}}$ is the setup cost for the process m on the workstation $\epsilon_{\mu_{oq}}$ (a setup is needed even if the two consecutive lots are of the same variant);
- t_{UmX} is the unitary process duration for the variant m on the workstation X ;
- S_{jXY} is the number of components manufactured or assembled on the workstation Y that are needed to start one job on the workstation X ;
- V_c and V_p are the number of customers and processes;
- k_q is the set of process variants of the production system q .

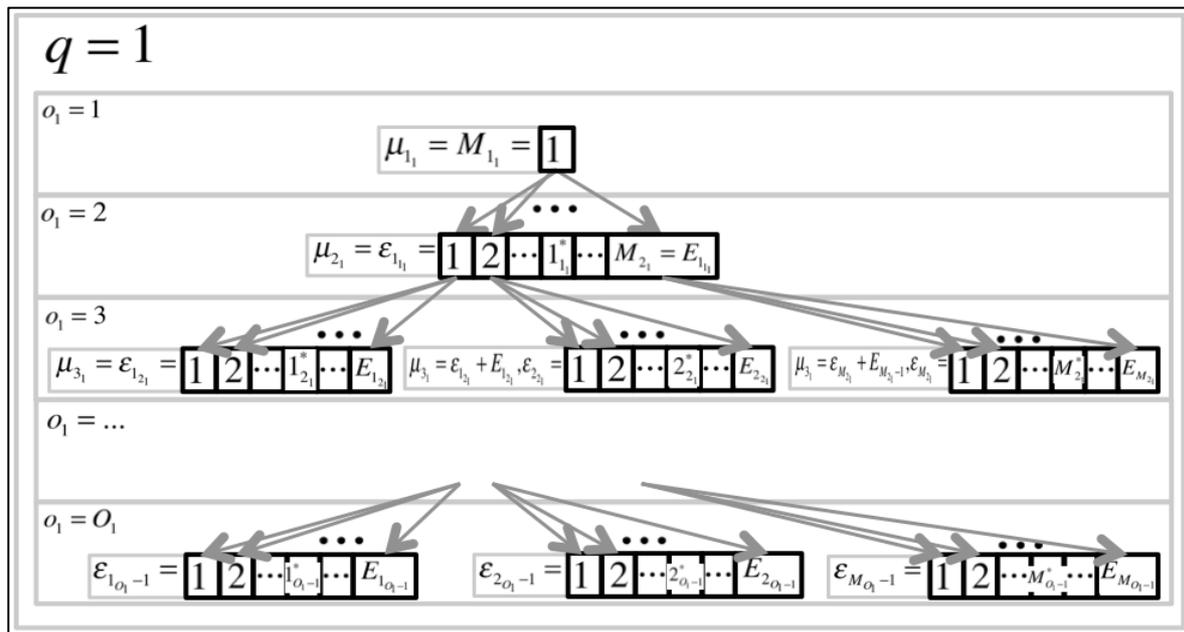


Fig. 3.7 - representation of the production system 1 ($q=1$); each black box represents a workstation; at the first stage (i.e. the last chronologically) there is only one assembly workstation that has to provide the final product.

The variables are:

- Q_{ij} is the volume for the couple of the customer i and the process j ;
- $l_{m\mu_{oq}^*}$ is the number of lot for the process m on μ_{oq}^* ;
- Q_{LmX} is the lot size for the process m on the workstation X , i.e. each process variant has a constant lot size.

The three more terms to minimise are in the constraints (24-26): C_d represents the process *direct* costs; C_s represents the costs of the setups (the costs of the setups on the last assembly stage can be included in the costs of the workstation at the previous stage by transforming the latter ones into averages). The constraint (27) defines the relation between the lot sizes and the customer demand met by the process variant j . The constraint (28) imposes the lot size on the workstation μ_{oq}^* bigger than the minimum size to make a job start on the assembly workstation μ_{oq} . The constraint (29) is about the domain of the variables.

This optimisation model allows to find the best PV among the alternatives produced by implementing the approach in the previous chapter. The solution of this optimisation problem allows to select the best set of *customer-process* couples that should define the best PV. The solution of this model should be deployed to answer each one of the industrial issues. Indeed,

- to solve the first one, a customer profile should be linked with the best available product variant;
- to solve the second one, a product variant should be connected with the best available process variant;
- to solve the third one, a new customer should be connected to the best process variant, also considering the already designed PV.

4 - Conclusions

In this chapter a proposal to answer the

RQ2 - How to compare different variety alternatives, i.e. process varieties that allow to manufacture PVs able to meet the same customer variety?

has been described (Fig. 3.8). The main point of the proposed criterion is the consideration of the *lot streaming* variables for the definition of the manufacturing costs of a PV. By using the *lot streaming* fundamentals, we formalised a mathematical model that showed how the degree of variety is not enough to describe the variety-costs relations. The analysis of the results of the model allowed to propose a new term for a more comprehensive criterion to describe the variety-costs relations. The application of this criterion on evaluation of the feasible *customer-process* couples (i.e. output of the method proposed in the chapter 2) has been presented.

In the next chapter the criterion is implemented on the selection of the best variety of water coils.

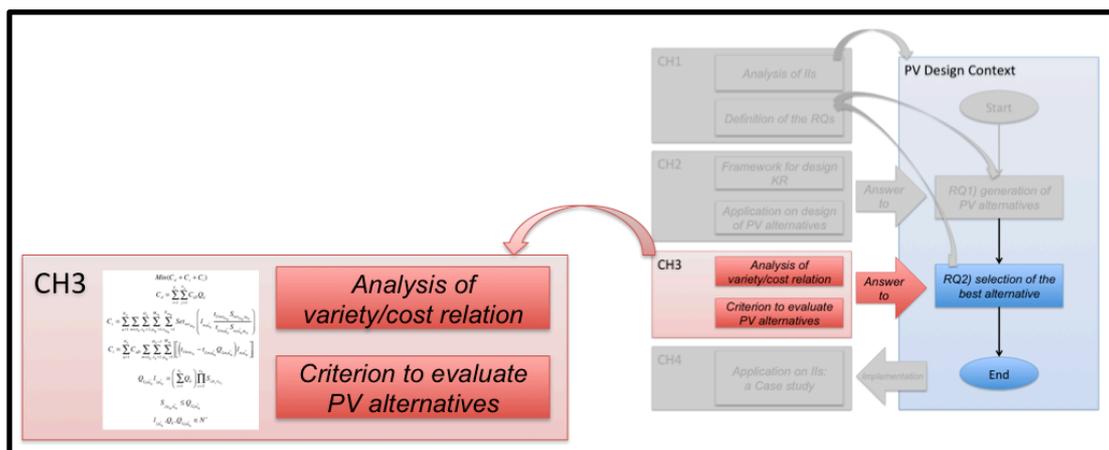


Fig. 3.8 – results of the third chapter.

Chapter 4 Case study: water coil assembly

The aim of this chapter is to discuss the applications of the proposed framework and criterion on a case study (Fig. 4.1). In the first section, an application of our proposals in *section CH2.3.1.1* (i.e. instantiation of the conceptual model of the framework) on a *water coil* assembly is presented. On the basis of this case study, a *top-down* PV design scenario is simulated (application of the proposal in *section CH2.3.1.3*, i.e. the connection of different pieces of represented knowledge – instances of *experience*). Finally, the instantiation of the algorithm proposed in *section CH2.3.1.4* (i.e. algorithm to map two instances of *experience*) is discussed.

In the second part of the chapter, the Java tool developed to convert the knowledge about the *water coil* into a set of feasibility constraints is presented (instantiation of proposals in *sections CH2.3.2.1* and *CH2.3.2.2*). The main features of the tool are discussed on the *water coil* case. The same tool allows the application of the proposed criterion for the PV evaluations (proposals in *sections CH3.3.2* and *CH3.3.3*).

In the third and final part, the results of the case study are discussed. The development choices and the impacts of the proposed solutions on the questions RQ1, RQ2 and the industrial issues are argued. Finally the limits concerning the application of the proposals are discussed.

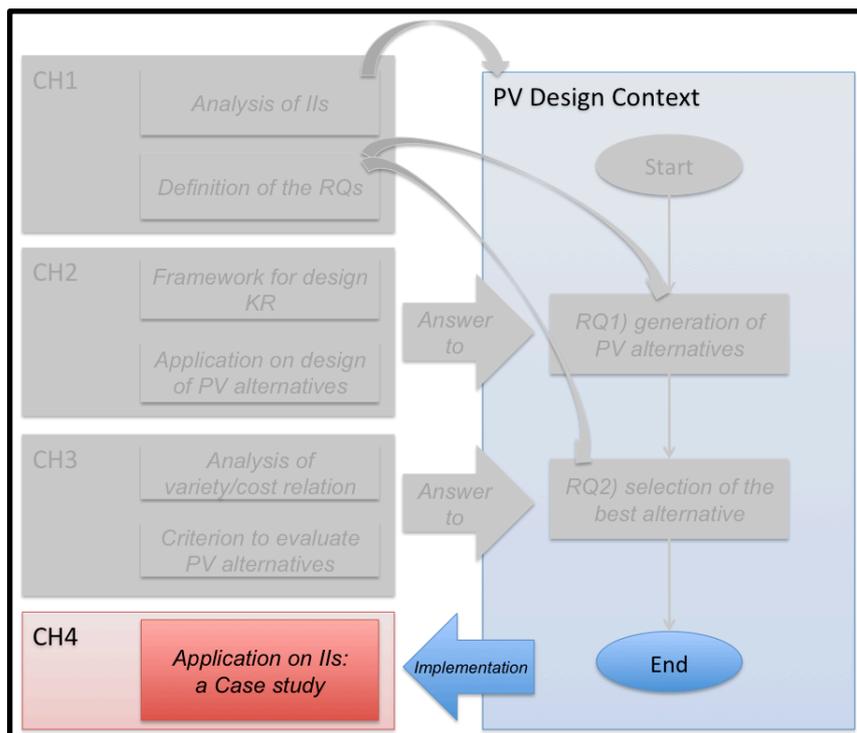


Fig. 4.1 - role of the chapter in the thesis structure.

1 - The multi-domain KR framework: a case study

The subject of the application is a *water coil*, i.e. a family of components of the *fan coil* PV (Fig. 4.2). A *water coil* is made of a set of punched aluminium fins. The fins are assembled on a coil of copper tubes that contains a flow of water. A motor activates a fan. The fan generates and orients an airstream towards the coil for the heat transfer. The usual *fan coil* implementation is for hotel rooms,

universities, offices and so on. The corresponding *water coil* customer variety has about 50 millions of variants.



Fig. 4.2 - Pictures of the Trane fan coil PV.

The formalised knowledge about the *water coil* behaviour is from the following standards for the HVAC domain.

- The chapter 1 in (ASHRAE 2009) about *psychrometrics* has been used to model the air temperature and humidity relations.
- The chapter 4 about *heat transfer* in (ASHRAE 2009) has been used to model the relations between heat transfer and fins geometry.
- The chapter 22 about *air-cooling and dehumidifying coils* in (ASHRAE 2012) has been used to model the *water coil* system behaviour.
- The AHRI standard on *Forced-Circulation Air-Cooling and Air-Heating Coils* (Air-conditioning, Heating & Refrigeration Institute 2001) has been used to model the relations between the coil geometry and the heat transfer.

The complete model adapted and expressed in Lingo⁹ language is shown in the Annex 3. In the next sections, 1) the usage of the CAD software and 2) the formalisation of the knowledge about the *water coil* as instance of the *experience* concept are shown.

1.1 - The water coil knowledge represented in CATIA V5

For the knowledge formalisation on the CAD software CATIA V5, the *Knowledgeware*¹⁰ module has been employed: this module has been used to express the constraints between the properties related to the volumes represented in the software. The instance of the conceptual model presented in the section CH2.3.1.1 is represented as follows on CATIA.

- An *experience* is a CAD part or assembly of parts. In the case of the *water coil* (Fig. 4.3), the *experience* is the coil assembly that includes the fins, the straight tubes and the *C-tubes* (i.e. short copper tubes that are bent and welded to close and orient the flow of water in the coil).
- The *shape*, the *time* and the *space* of a *measurement* are represented as CATIA part parameters (at the left of the Fig. 4.4). If the *measurement* is T_{A1} then T_{A1} represents the

⁹ http://www.lindo.com/index.php?option=com_content&view=article&id=2&Itemid=10

¹⁰ http://www.3ds.com/products-services/catia/portfolio/catia-v5/all-products/domain/Product_Synthesis/

shape, *vol_T_A1* represents the *space* and *time_T_A1* represents the *time*. All the *times* are related each others.

- The parameter *vol_T_A1* and the other parameters representing the *spaces* of the *measurements* are related, by means of a CATIA *formula* (i.e. $vol_T_A1 = air_front \setminus volume$), to the result of the *measure item* CATIA function (*air_front \setminus volume* in the Fig. 4.3). This type of relations allow to relate the *measurements* to the *spaces*.
- For each *property*, the *UOM*, the *tolerance* and the *range of values* can be associated to the CATIA parameters as shown in Fig. 4.5, Fig. 4.6 and Fig. 4.7.
- The transformation are formalised using the *check rules* in the *knowledgeware* module of CATIA. For instance let us consider the following mathematical relation (i.e. an instance of *transformation*) between coil areas,

$$A_0 = A_P + A_S .$$

The representation of this relation in CATIA is shown in Fig. 4.8. In the same figure, a red or green light can be seen beside each constraint represented by a *check*. The green light notifies that the values of the parameters respect the constraint. The red light notifies that the constraint is violated.

This usage of CATIA has allowed the representation of the *water coil* experience as in Fig. 4.3 and Fig. 4.4. In the next section, a simulation of how the *water coil* knowledge should be retrieved to perform a *top-down* PV design is shown.

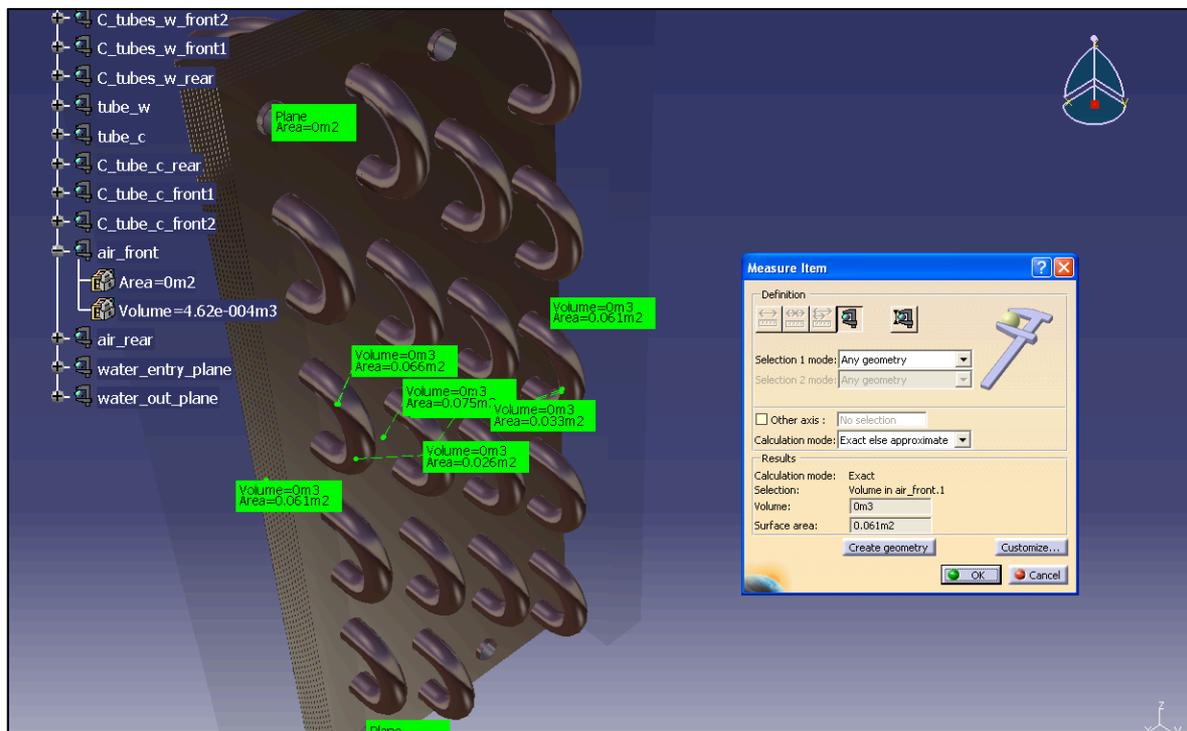


Fig. 4.3 - The representation of the space of the measurement *T_A1*.

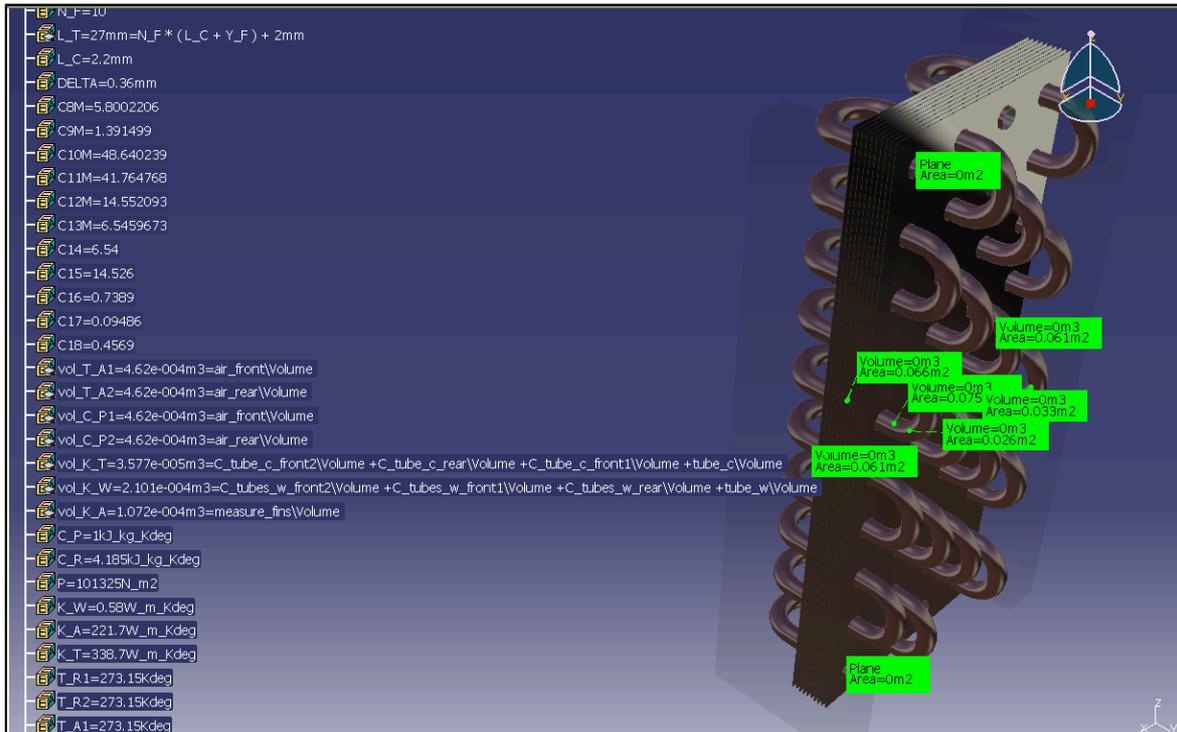


Fig. 4.4 - The representation of the water coil experience on CATIA V5.

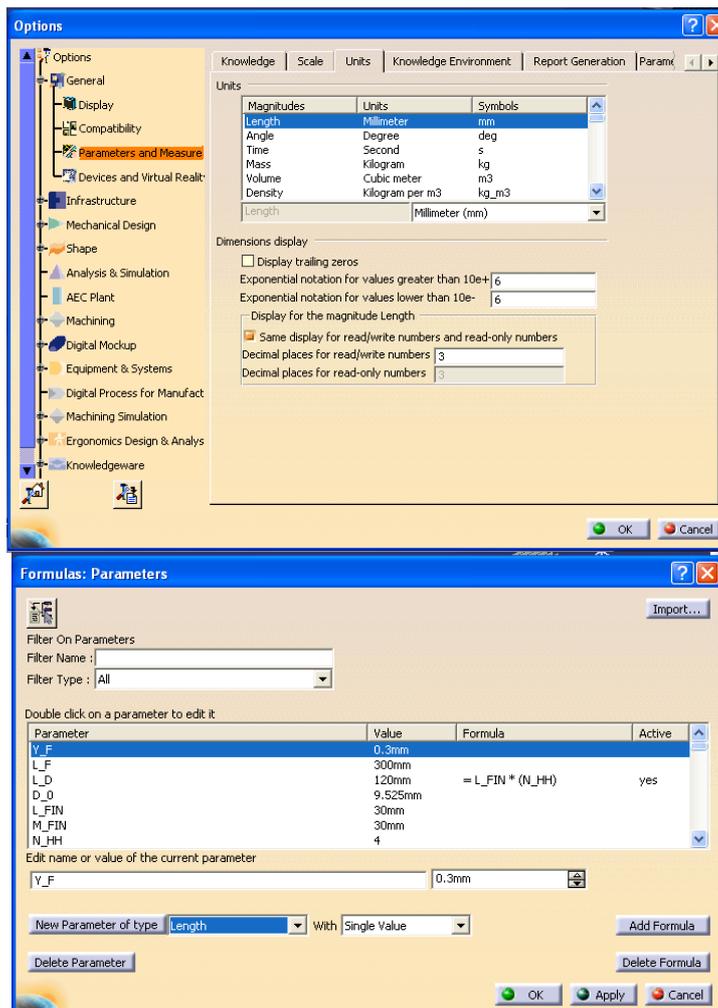


Fig. 4.5 - The representation of the UOM related to the parameter Y_F, i.e. fin thickness.

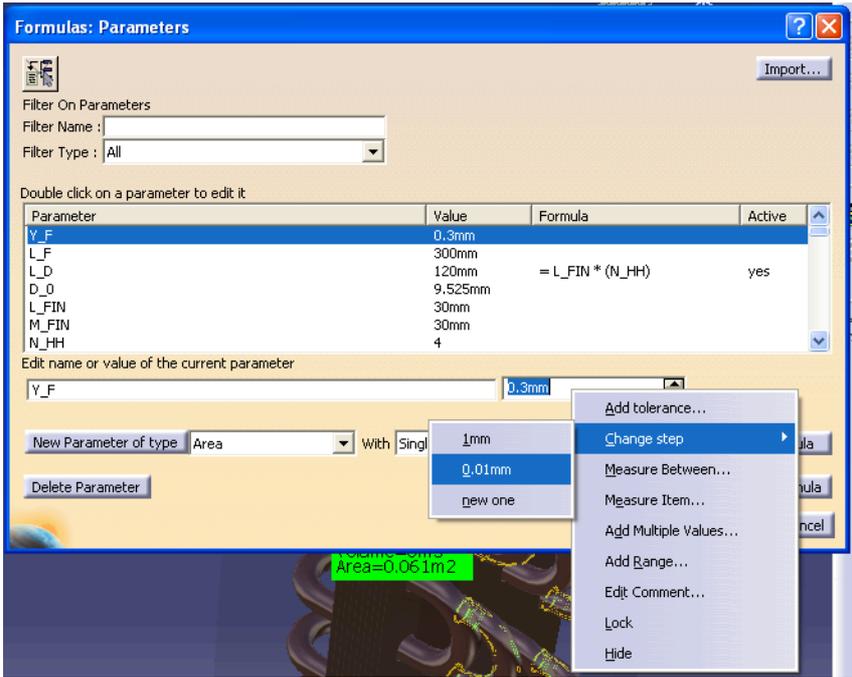


Fig. 4.6 - The representation of the tolerance related to the parameter Y_F.

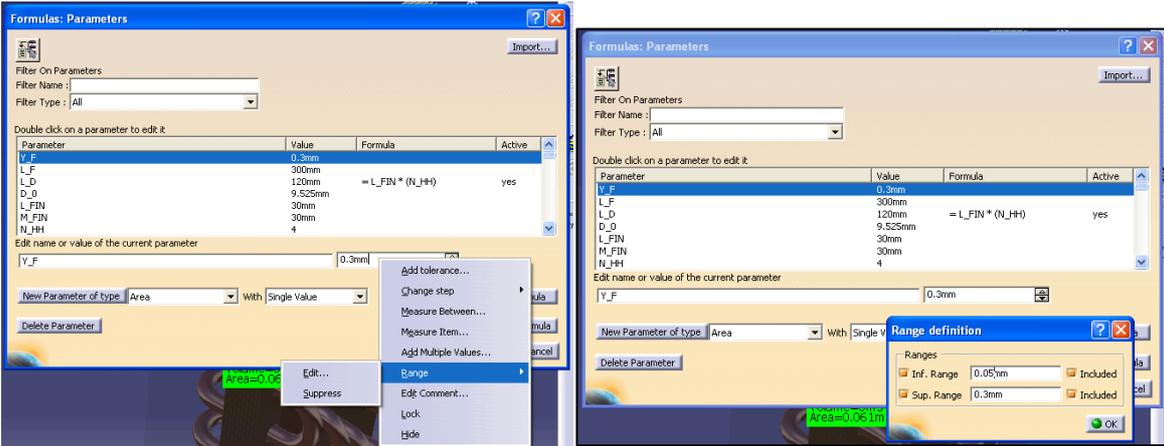


Fig. 4.7 - The representation of the range of values for the parameter Y_F.

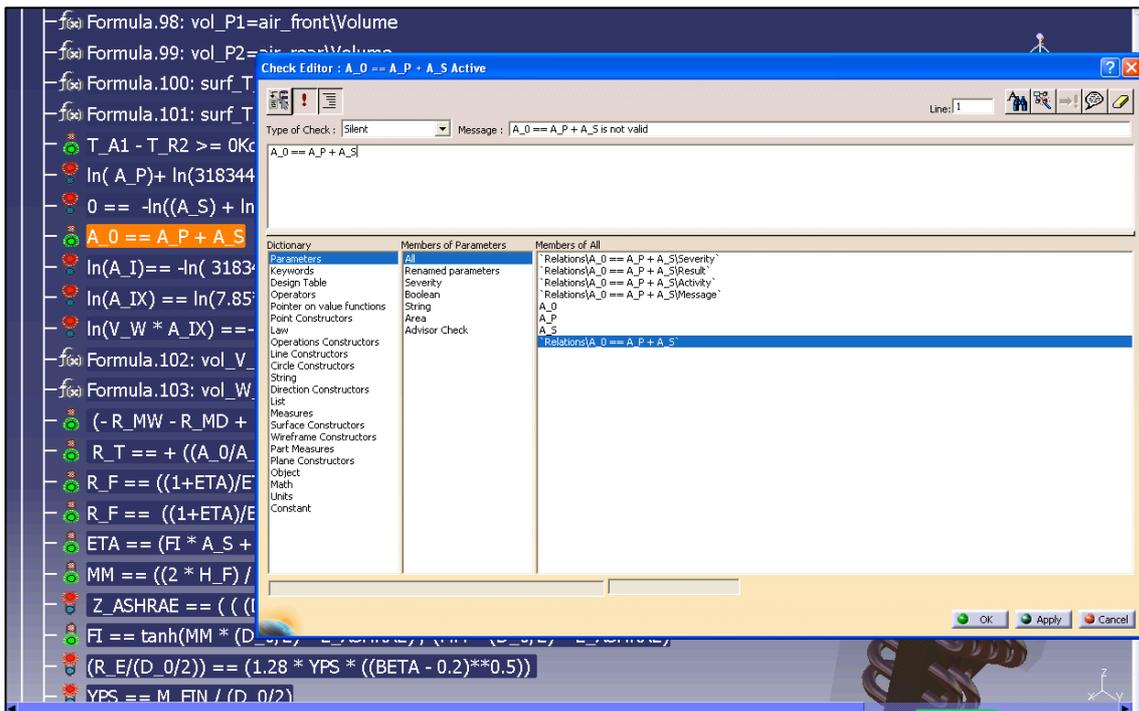


Fig. 4.8 - The representation on CATIA of the transformations as check rules. Beside each check a red or green light shows if the parameter values are compliant with the formalised constraint.

1.2 - The design of the water coil variety

In this section we consider a simulation of the *top-down* PV design approach. Starting from a customer variety definition, the retrieval of the *water coil* knowledge is shown. In order to start the design stage, the customer profiles should be formalised as *properties* of some *measurements*. In our case study the customer variety is represented by the following requirements.

- An air temperature and a humidity level have to be maintained in a certain volume.
 - The temperature and the relative humidity represent two shapes. In order to characterise a shape, as for each property, the UOM, the tolerance and the range of values have to be specified. For the temperature (X_3) let us consider Celsius degrees, ± 0.1 degrees and values of 25.7. For the relative humidity (X_4) consider $\pm 0.01\%$ and 50%. The tolerance of the volume (space) characterisation is of ± 0.1 mm. Since the values of the shapes have to be “maintained in a certain volume” there is no-dependence on time of the values of the spaces and the shapes. The characterisation of the air is due to the value of the specific heat in the same volume of the temperature and humidity spaces. The space of these measurements are not specified.
- The sensible power is constant.
 - The sensible power (X_5) characterises the required heat transfer. In the fan coil design this value characterises the room usage and features (e.g. presence of windows, computers, ovens). In our case the value is 6.9 kW with an error of ± 0.1 kW. The value is constant therefore its time is equal to the

times of the temperature and humidity above. The space is equal to the volume of the above temperature and humidity.

- The temperature of a water flow is fixed at two points of the flow.
 - The water is characterised by its density and specific heat. The temperatures are: 1) 11.4 to 13.3 C° with an error of ± 0.1 (X_1); 2) 6.0 to 6.4 C° with an error of ± 0.1 (X_2). The value of the first temperature is equal or higher the one of the second temperature. The space of these measurements are not specified.

All these *measurement properties* represent the customer variety requirements and they constrain the compatibility of an *experience*. Every property of the measurement that is not specified is considered as an engineer degree of freedom. In other words, the defined properties *ranges*, *UOMs* and *tolerances* are the only limits to determine the compatibility of a formalised experience. All other STSs defined in the experience are considered as not relevant for the customer satisfaction. Therefore their values can be fixed on the basis of the impacts on the manufacturing costs.

In the next subsections, the simulation of the knowledge retrieval to perform a PV design stage is shown. Starting from the above requirements, the *experiences* retrieval will allow the connection of the customer requirements to the degrees of freedom that the engineers can handle to meet the customer requirements.

Customer-to-Product

In the above requirements a volume with specific heat, temperatures, humidities and sensible powers has been specified. The *times* of all these measurements are equal to the same value, i.e. stationary conditions. In all the eventual available *experiences*, a volume in which all these measurements are observed is searched. Therefore the association of the customer-defined *shapes* with the ones in the *experiences* is based on the *shapes* that are measured in a certain volume in a certain time. In order to verify that a customer-defined *shape* matches with a *shape* in an *experience*, the *UOMs* (values of strings have to match), the *ranges of values* (customer-defined ranges have to be included in the ranges of the *experience*) and the *tolerances* (customer-defined values need to be equal or higher than values in the *experience*) of the *properties* of the *measurements* have to match.

In the volume in Fig. 4.9, the *water coil experience* represents three measurements:

- 15 to 30 ± 0.01 °C;
- 35 to 65 ± 0.01 %;
- 0.5 to 13 ± 0.1 kW.



Fig. 4.9 - the coil and the volume of air in the customer requirements.

Also for the water temperatures the procedure is the same. Two different temperature in two different surfaces are measured on a flow of water at the coil entry and at the coil exit. The density and the specific heat measured in the surfaces matches with the customer-defined ones. The temperature *shapes* observed in the two couples of surfaces (in orange, at the top and at the bottom of Fig. 4.10) in the experience are:

- 5 to 12 ± 0.01 °C;
- 10 to 20 ± 0.01 °C.

Since the five *measurements* (3 temperatures, humidity and power) respect the constraints about the *UOMs*, *ranges* and *tolerance* of the customer-defined STSs, the *water coil experience* formalises the knowledge to meet the requirements of all the defined customer variety.

Notice that in this case study, the customer defined *spaces* have not been specified. If a precise region of the space is constrained by the customer then also a volume comparison is required to verify the compatibility of an experience with the customer requirements. Also this verification has to be performed on the basis of the UOMs, ranges of values and tolerances of the customer defined *spaces* with the related region defined in the experience.

Since the *spaces* are surfaces or volumes in the CAD file, two regions of the space are equivalent when their intersection is equal to both the regions. In order to check the compatibility of the *spaces*, this verification should be performed for each value of each properties relative to each volume to be compared.



Fig. 4.10 - surfaces in which the entering and exiting water temperature can be observed.

Product-to-Process

Once the water coil experience can meet the customer requirements, the measurements that matches with the customer-defined properties are removed from the engineer's degrees of freedom. Indeed, all the non-constrained properties are automatically the properties that the engineer should control to meet the customer requirements.

In the water coil examples, the degrees of freedom include the thermal resistance of the tube and fin material, the water mass flow, the exiting air temperature and so on. But also the geometries of the measurement spaces are included. All these parameters become the requirement of a manufacturing or assembly process.

As for the customer-defined requirements, the compatibility is based on the *UOMs*, the *ranges of values* and the *tolerances*. As general rule, the starting point for the compatibility verification is the volume comparison between the requirements (depending on customers, on products or on other processes) and the volume evolutions in the *experiences* representing the processes. In other words, all the requirements have to match with one or more experiences at least at a certain time (to guarantee that the process is able to build what is required to meet the requirements).

When an assembly process is formalised, at the starting point (at *time=0*), the CATIA file should show the components that are still not connected. If we look at the transformations at that time, subset of *properties* are identifiable: each *property* belongs only to a subset and can be related only to *properties* of the same subset. Each subset can be considered as an independent component. Therefore the requirements of the assembly *experience* can be met by more than one other *experience* (e.g. one for each independent component). Since the identification of the subset should add another complexity to the knowledge retrieval, an equivalent solution is deployable, i.e. add an ID of the component in the name of the related properties (e.g. the parameter *Y* should become *Y_x*, where *x* is the ID of the component).

In the water coil case, the experiences that are retrieved to connect the customer requirements to the manufacturing process properties are the following.

- The installation process - since no-effects of this process on the heat transfer are modelled in the considered HVAC standards, the installation is considered as a movement of the water into the coil and of the coil in the airstream. In CATIA this time-dependences of the parameters have been modelled by means of the macros (Fig. 4.11). In Fig. 4.12 and Fig. 4.13 the relative position of the coil and the airstreams depend on the value of the parameter *time_1*.
 - Here the airstream speed is considered equal to the fan speed (Y_{11}), i.e. 3.56 ± 0.01 m/s.

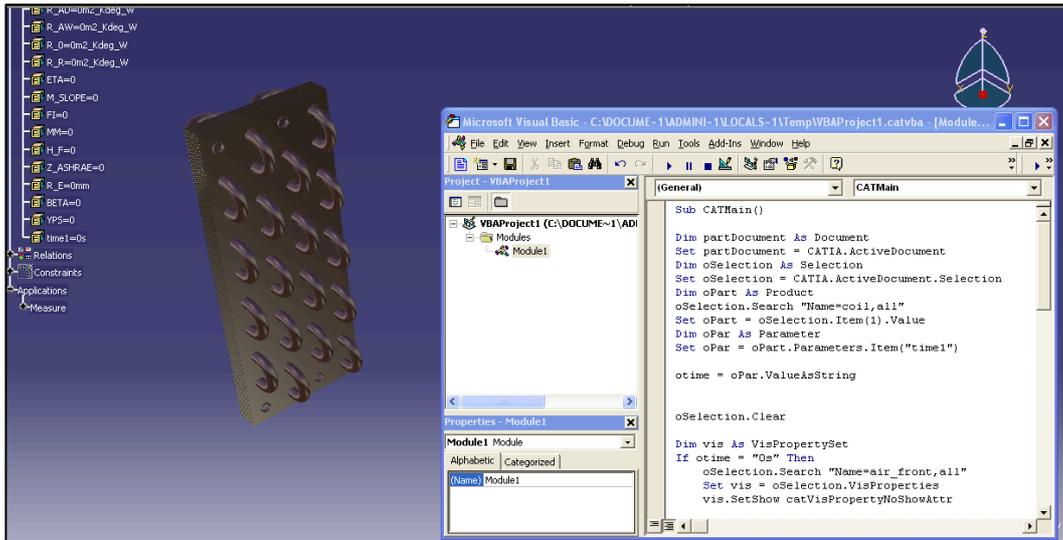


Fig. 4.11 - the macro for representing the time-dependences in the installation process.

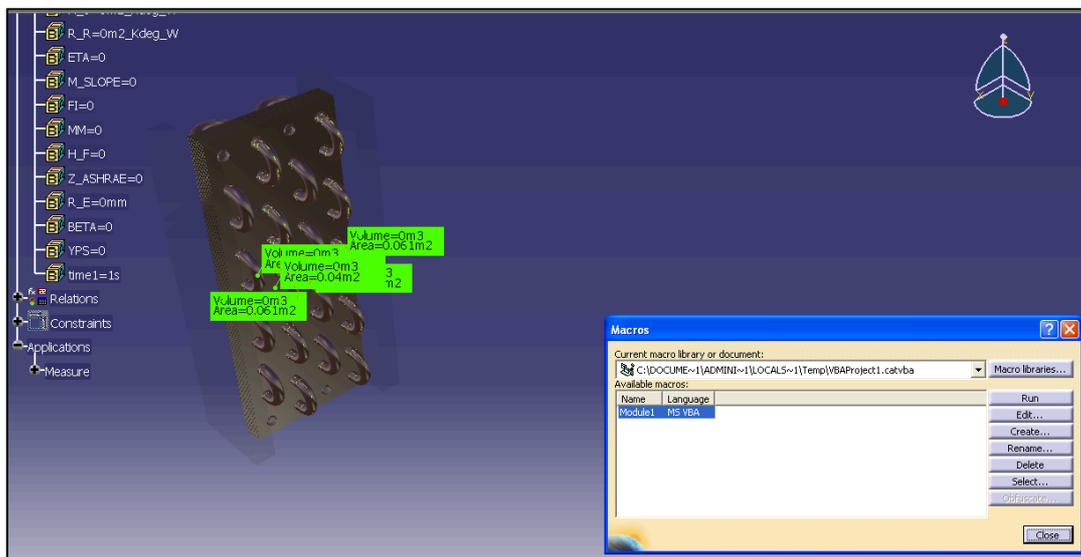


Fig. 4.12 - the geometries for the parameter $time1=1s$.

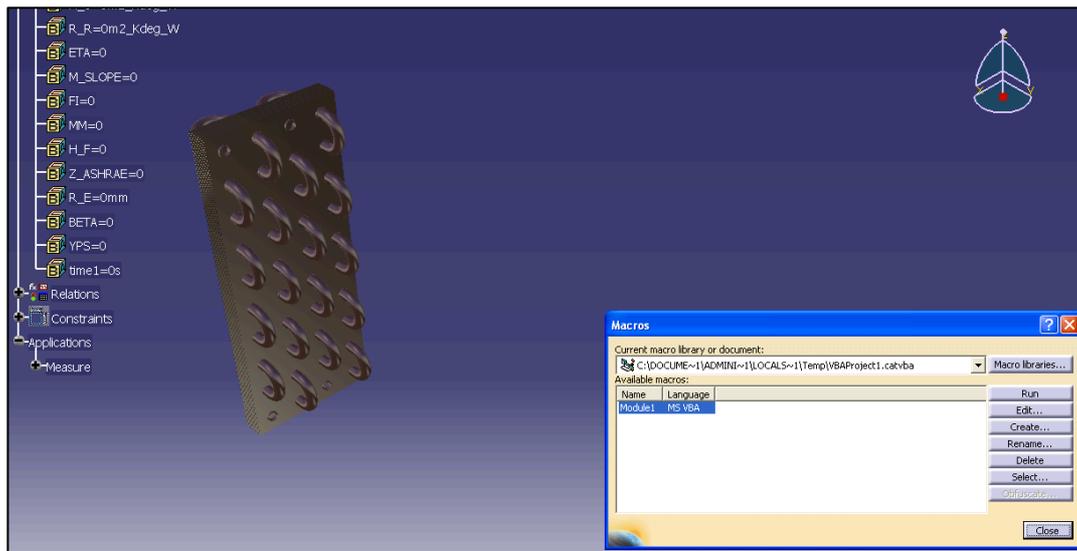


Fig. 4.13 - the geometries for the parameter $time1=0s$.

- The C-tubes welding process - the welding process has been considered as a simple movement of the C-tubes. The position of the C-tubes determines the number of coil circuits (Y_2) that is equal to 3 or 4 circuits, depending on the considered production system (in the next section the other difference between the two considered production systems are detailed).
- The tube expansion - this process is modelled as a change of the relative position of the tubes and the fin holes.
- The assembly stage - the assembly process is modelled as a change in the relative positions of the aluminium fins and the copper tubes.
- The fin punching process - the process is modelled as the cutting (i.e. change of the length of the fin) and punching (i.e. manufacturing of the fin holes) of an aluminium sheet. The degrees of freedom of this process are considered as characterising the process variety:
 - number of horizontal holes (Y_1), with value equal to 4;
 - the punching tool depth that impact the fin collar (Y_8) that determine the distance between two fins, equal to 2.2 ± 0.01 mm ;
 - the aluminium sheet height (Y_7), equal to 400 ± 0.01 mm;
 - the aluminium sheet thickness (Y_5), equal to 0.15 ± 0.01 mm;
 - the horizontal (Y_9 , 25.4 ± 0.01 mm) and vertical (Y_{10} , 22 ± 0.01 mm) distance between two punching tool features (i.e. the features that hole the aluminium sheet).
- The tube cutting process - the process is modelled as the change of length of a copper tube. The degrees of freedom of this process are considered as characterising the process variety:
 - the copper tube thickness (Y_3) goes from 0.81 to 0.82 ± 0.01 mm;
 - the copper tube diameter (Y_4) goes from 9.56 to 9.60 ± 0.01 mm;
 - the frequency of tube cuts that impacts the tube length (Y_6); the resulting length is equal to 330 ± 0.01 mm.

1.3 - *The unambiguity of the water coil knowledge*

In order to test the conclusions (1 to 6) of the algorithm proposed in *section CH2.3.1.4* (i.e. the algorithm about the knowledge mapping between two *experiences*), examples of experiences about the water coil are considered.

The algorithm proposed in chapter 2 is expected to estimate the relations between a couple of experiences. The algorithm can infer about:

- the *inconsistency* of the two *experiences*, i.e. the *experiences* formalise the same *measurements* but they have two different behaviour formalised by the *transformations* (conclusion 1);
- the complete or partial equivalence of the represented experiences (conclusions 2, 5 and 6);
- the absence of relations between the two experiences (conclusion 3, 4).

Two main conditions determine the difference of conclusions that the algorithm can provide: 1) The verifications of the compatibility of the measurements that are based on the STSs comparison, as for the knowledge retrieval in the PV design stage; if two measurement from the two experiences are compatible only for part of the ranges of values, the analysis of the algorithm is performed only for these values; 2) the verifications of the transformations that are realised by comparing the constraints in the two experiences that are violated by a set of values for the common *measurements*. The first kind of comparisons can be performed on the basis of the parameter values of the *experiences* formalised on CATIA. The second kind of comparison should be performed on a mathematical solver able to deal with MINLP models, e.g. Lingo.

In the following subsection, all the possible conclusions are discussed on the *water coil* example. The first subsection deals with two experiences that represents the same measurements (i.e. STSs are compatible). The second subsection deals with two experiences that have some measurements that are represented in both the experiences. The *conclusion 3* is not further explained because it is simply due to an incompatibility of the STSs. This means that the experiences are representing a completely different piece of knowledge, i.e. the non-ambiguity can be trivially verified.

1.3.1 - *The two experiences represent the same measurements*

The branch of the flowchart that deals with two experiences that represent the same measurements is in Fig. 4.14. In this case, the two experiences formalise an experiment done on the same measurements (i.e. STSs are compatible).

- Let us consider the *water coil* example. Suppose that the same measurements represented in the above shown experience are formalised in two different CATIA files. In order to test the constraint violation, set of values for the observed measurements have to be tested on the solver.
- Conclusion 1 - When the solver results are not compatible for the two experiences, they are inconsistent. Two are the possible explanations for this result. 1) The inconsistency is due to a factor that is not observed in both the experiences and that takes different values during the two experiences, e.g. the thermal resistance of the duct that orient the airstream. 2) The inconsistency is due to a high level of tolerance for some properties (e.g. rugosity if the inner

tube surface) that can hide effects of the STSs on the system behaviour. In both of these situations, the mathematical regression of the observed variables can lead to different results and so to different mathematical relations between the STSs, i.e. the transformations. Therefore the Lingo solver launched on two models that have the same values for STSs but different constraints (i.e. transformations in the mathematical model) should provide different results, e.g. one model is feasible and the other one is infeasible for the same values of the properties.

- Conclusion 2 - When the solver gives the same result (feasible or not) for the two experiences for each given set of values for the STSs, the two experiences are completely equivalent. In the case of the water coil, this means that two experiences on CATIA describe the same system (same STSs and equivalent transformations) therefore they can be used as alternative pieces of knowledge in a PV design stage.

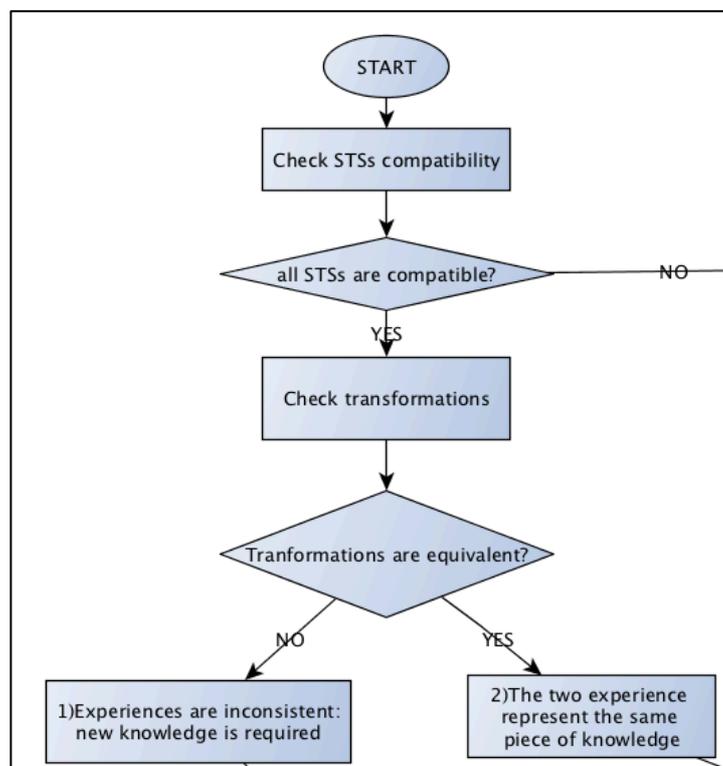


Fig. 4.14 - excerpt of algorithm proposed in chapter 2: conclusions if all STSs are compatible.

1.3.2 - The two experiences shares the representation of some measurements

The two *experiences* can eventually share only part of the observed measurements (branch of the flowchart in Fig. 4.15). In this case the verification of the constraint violation have to be done with a particular attention on the values considered for the non-common *measurements* (NCM).

- Conclusion 4 - If the experiences STSs are never compatible then either 1) they are describing different pieces of knowledge or 2) they are inconsistent. In order to verify which is the right conclusion, further experiments considering the NCM should be performed. This means that for no-values of the thermal resistance of the duct, the solver gives compatible results between the two experiences.

- *Conclusion 5* - If for all values of the NCM, the solver results are compatible for values of the common STSs, the NCM are not influential on the behaviour of the represented system. This means that the even if the thermal resistance of the duct is considered in only one experience, its values cannot impact the relations between the other *measurements*.
- *Conclusion 6* - When the solver results are compatible only for some values of the NCM that values are the values that would have been observed in the experience that does not represent them if these NCM were taken into account. This means that, when the solver gives compatible results, the related values of the thermal resistance of the duct would have been observed during the experiment run following the knowledge formalised in the experience that does not represent it.

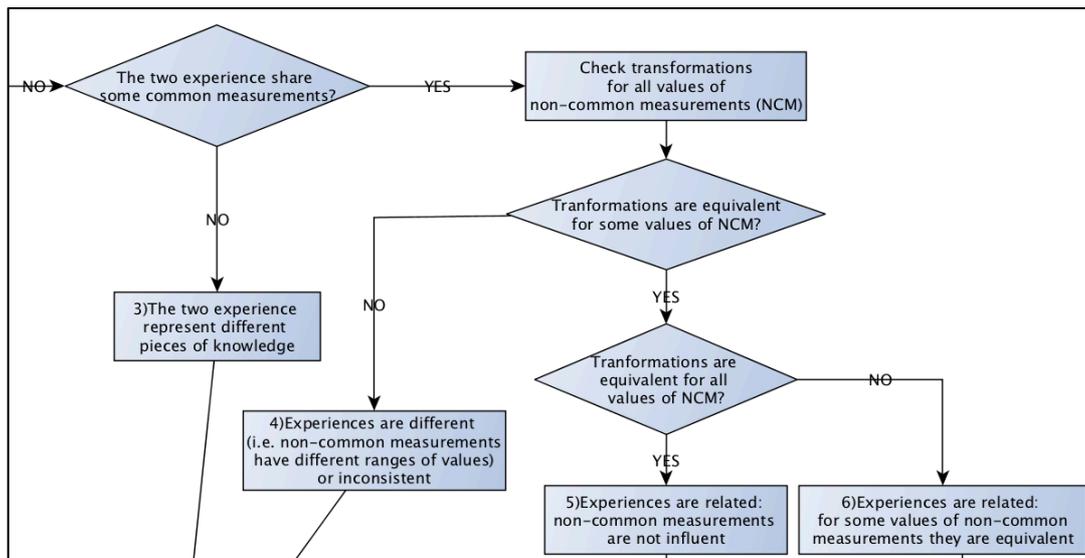


Fig. 4.15 - excerpt of algorithm proposed in chapter 2: conclusions if only some STSs are compatible.

2 - Water coil variety alternatives: generation and evaluation

As proposed in the *section CH2.3.2*, to use the retrieved *transformations* for the PV design process, the discretisation and the formulation of a MINLP model are required. To do so, we developed a tool in Java to interface the definitions of the *customer-process* varieties and of the *transformations* with the MINLP solver.

The developed tools (Fig. 4.16) allows (1) to collect the data about the customers and the processes from a structured text file. This data provides the input for the generation of the feasibility tests for the *customer-process* couples. (2) The tool allows the interaction with the solver Lingo for the solution of the MINLP problems. (3) Once all the *customer-process* couples have been tested a report file is generated. (4) On the basis of this report file, the tool formulates the optimisation problem and (5) send it to the solver that (6) finally provide a solution about the best set of couple that define the optimal PV variety.

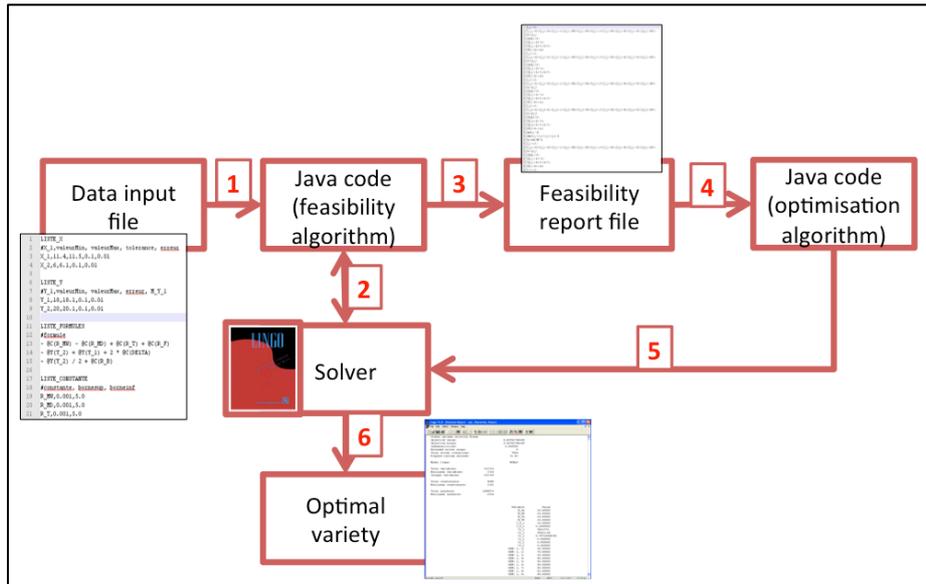


Fig. 4.16 - flows of the developed java tool.

2.1 - The step one: the structure of the data file and the constraints in Lingo14

In Fig. 4.17, an extract of the data input file for the *water coil* case is shown. The file represents separately the *properties* formalising the customer requirements (the *X* variables) and the *properties* characterising the process variety (the *Y* variables). The ranges of values for the *properties* and the *tolerances* allow to build the ranges for each customer or process in the MINLP model.

Each customer *property* name is followed by four values: 1) the minimum and 2) maximum values that the *property* can have; 3) the error that is tolerated by the customer; 4) the precision that the knowledge in the experience can guarantee.

```

LISTE_X
#X_1,valeurMin, valeurMax, tolerance, erreur
X_1,11.4,13.3,0.1,0.01
X_2,6,6.4,0.1,0.01
X_3,25.7,25.7,0.1,0.01
X_4,0.5,0.5,0.01,0.001
X_5,6.9,6.9,0.1,0.01
LISTE_Y
#Y_1,valeurMin, valeurMax, erreur, M_Y_1
INT:Y_1,4,4
INT:Y_2,4,4
Y_3,0.81,0.82,0.01,0.001
Y_4,9.56,9.60,0.01,0.001
Y_5,0.15,0.15,0.01,0.001
Y_6,400.0,400.0,1,0.1
Y_7,330.0,330.0,1,0.1
Y_8,2.2,2.2,0.01,0.001
Y_9,25.4,25.4,0.1,0.01
Y_10,22.0,22.0,0.1,0.01
Y_11,3.56,3.56,0.01,0.001
  
```

Fig. 4.17 - extract of the input file that represents the property values and the constraint to generate the customer-process feasibility tests.

Each process property name is followed by other four values: as for the customer properties, 1) the minimum and 2) maximum values that the *property* can have; 3) the precision that the process knowledge can guarantee for the property; 4) a small number to define the range in Lingo. Indeed, the solver does not accept “<” and “>” constraints. Therefore in order to formalise the value of a property for a feasibility test, a small number is needed.

For instance, let us consider the temperature of the water exiting the coil (i.e. the customer requirement X_1 in Fig. 4.17). Since the tolerance is ± 0.1 , only two customer profiles are formalised, one for 11.4 ± 0.1 and the other for 11.5 ± 0.1 . The constraint that has to be represented in Lingo for the customer $X_1 = 11.4 \pm 0.1$ (i.e. $11.3 < X_1 < 11.5$) is the following

$$11.31 \leq X_1 \leq 11.49,$$

where the lower bound is equal to

$$11.4 \text{ (nominal value)} - 0.1 \text{ (customer tolerance)} + 0.01 \text{ (knowledge precision)}$$

and the upper bound is equal to

$$11.4 \text{ (nominal value)} + 0.1 \text{ (customer tolerance)} - 0.01 \text{ (knowledge precision)}.$$

Since for the processes there is only the knowledge precision, the process properties ranges need the definition of another small number to build the lower and upper bounds. For instance, consider the external coil tube diameter (Y_4 in Fig. 4.17). The lower bound is calculated as follows

$$9.57 \text{ (nominal value)} - 0.01 \text{ (knowledge precision)} + 0.001 \text{ (number smaller than precision)},$$

instead the upper bound is equal to

$$9.57 \text{ (nominal value)} + 0.01 \text{ (knowledge precision)} - 0.001 \text{ (number smaller than precision)}.$$

The properties can be also represented as discrete intervals (e.g. Y_1 in the Fig. 4.17). In this case the constraint in Lingo is a simple equality, e.g. $Y_1 = 4$.

In the data files, also the formulas to compute the customer demand (for each profile) and the process costs (for each process) are formalised. This formulas allow to generate the optimisation model at the step 4. When the process variety includes processes related to more than one production system (i.e. set of machines usually corresponding to the shop floor of a manufacturing plant), more than one data files are necessary. In this case, the java tool harmonizes the data files information and the results of the related report files to generate the MINLP optimisation model.

For the *water coil* case the demand and the costs are below. Two production systems are considered, i.e. S1 and S2. The shop floor of S1 has a better equipment therefore the process duration (t_{xy} , where x is the workstation and y the production system) are shorter and setups (Set_{xy}) are cheaper. On the other hand the costs allocated per unit of time (C_{yD}) are 40% higher. The direct costs (first equation below) are equivalent: the better equipment is equalled by the cheaper labour cost. Setups on the assembly workstation (M1) are not considered. The workstation M2 is for the aluminium fin punching process. The M3 is for the copper tube cutting and bending. In S2, the copper tube cutting and bending process is outsourced, i.e. same price and $t_{M_3, S_2} = 0$. The provided copper tubes have a slightly different diameter (D_0), i.e. values from 10.06 to 10.10 ± 0.01 mm.

$$C_{iT S_1} = C_{iT S_2} = 10 * PUNCH_CUT * Y_F * L_T + L_T(L_F / L_FIN - 1)$$

$$Q_{iT} = 60$$

$$Set_{jM_2 S_1} = 1$$

$$Set_{jM_3 S_1} = 1$$

$$Set_{jM_2 S_2} = 1.5$$

$$Set_{jM_3 S_2} = 0$$

$$S_{jM_1 M_2} = \lfloor (L_T - 40) / (L_C + Y_F) \rfloor$$

$$S_{jM_1 M_3} = \lfloor ((L_F / L_FIN) - 1) * PUNCH_CUT \rfloor$$

$$t_{M_1 S_1} = \lfloor ((L_F / L_FIN) - 1) * PUNCH_CUT \rfloor / 2$$

$$t_{M_2 S_1} = PUNCH_CUT * 0.1$$

$$t_{M_3 S_1} = L_T * 0.1$$

$$t_{M_1 S_2} = \lfloor ((L_F / L_FIN) - 1) * PUNCH_CUT \rfloor / 1.5$$

$$t_{M_2 S_2} = PUNCH_CUT * 0.15$$

$$t_{M_3 S_2} = 0$$

$$C_{1D} = 70$$

$$C_{2D} = 50$$

2.2 - The step two: the interface with Lingo

Once the all the ranges for the properties are formalised in Lingo, a feasibility test can be performed to check if a solution that respect all the constraint exists. The eventual solution represents the values of the process properties to manufacture a product that is able to satisfy the customer requirements.

In Fig. 4.18 and Fig. 4.19 two screenshots of one MINLP model on Lingo are shown. At the top of the model (Fig. 4.18) there is the definitions of the ranges for the *customer-process* couple. At the bottom (Fig. 4.19) all the transformations in the retrieved experiences are formalised in Lingo language to represent the constraints of the model.

```

@BND (11.31, X_1_1_1, 11.49);
@BND (5.91, X_2_1_1, 6.09);
@BND (25.61, X_3_1_1, 25.79);
@BND (0.491, X_4_1_1, 0.509);
@BND (6.810000000000005, X_5_1_1, 6.99);
@BND (4.0, Y_1_1_1, 4.0);
@BND (4.0, Y_2_1_1, 4.0);
@BND (0.801, Y_3_1_1, 0.8190000000000001);
@BND (9.561, Y_4_1_1, 9.579);
@BND (0.141, Y_5_1_1, 0.159);
@BND (926.1, Y_6_1_1, 927.9);
@BND (481.1, Y_7_1_1, 482.9);
@BND (2.191000000000003, Y_8_1_1, 2.209);
@BND (25.31, Y_9_1_1, 25.49);
@BND (21.91, Y_10_1_1, 22.09);
@BND (3.551, Y_11_1_1, 3.569);

@BND (0, D_I, 10);
@BND (0, R_B, 50);
@BND (0.3, V_W, 2.4);
@BND (0.01, W_R, 5);
@BND (0.01, W_A, 5);
@BND (0, A_O, 1000);
@BND (0, A_P, 1000);
@BND (0, A_S, 1000);
@BND (0, A_I, 1000);
@BND (0, A_IX, 0.001);
@BND (0, T_A2, 50);
@BND (250, T_2, 350);
@BND (0, T_WB_A2, 50);
@BND (0, T_A2_DEW, 50);
@BND (0, T_WB_A1, 50);
@BND (0, T_A1_DEW, 50);
@BND (0, H_DEW_A1, 100);
@BND (250, T_1, 350);
@BND (250, TW1, 350);
@BND (250, T_AS2, 350);
@BND (0, T_AB, 50);
@BND (0, T_S2, 50);
@BND (0, T_SB, 50);

```

Fig. 4.18 - customer variety and process variety definitions in the MINLP model.

```

@BND (11.31, X_1_1_1, 11.49);
@BND (5.91, X_2_1_1, 6.09);
@BND (25.61, X_3_1_1, 25.79);
@BND (0.491, X_4_1_1, 0.509);
@BND (6.810000000000005, X_5_1_1, 6.99);
@BND (4.0, Y_1_1_1, 4.0);
@BND (4.0, Y_2_1_1, 4.0);
@BND (0.801, Y_3_1_1, 0.8190000000000001);
@BND (9.561, Y_4_1_1, 9.579);
@BND (0.141, Y_5_1_1, 0.159);
@BND (926.1, Y_6_1_1, 927.9);
@BND (481.1, Y_7_1_1, 482.9);
@BND (2.191000000000003, Y_8_1_1, 2.209);
@BND (25.31, Y_9_1_1, 25.49);
@BND (21.91, Y_10_1_1, 22.09);
@BND (3.551, Y_11_1_1, 3.569);

@BND (0, D_I, 10);
@BND (0, R_B, 50);
@BND (0.3, V_W, 2.4);
@BND (0.01, W_R, 5);
@BND (0.01, W_A, 5);
@BND (0, A_O, 1000);
@BND (0, A_P, 1000);
@BND (0, A_S, 1000);
@BND (0, A_I, 1000);
@BND (0, A_IX, 0.001);
@BND (0, T_A2, 50);
@BND (250, T_2, 350);
@BND (0, T_WB_A2, 50);
@BND (0, T_A2_DEW, 50);
@BND (0, T_WB_A1, 50);
@BND (0, T_A1_DEW, 50);
@BND (0, H_DEW_A1, 100);
@BND (250, T_1, 350);
@BND (250, TW1, 350);
@BND (250, T_AS2, 350);
@BND (0, T_AB, 50);
@BND (0, T_S2, 50);
@BND (0, T_SB, 50);

```

Fig. 4.19 - transformations in the MINLP model.

The execution time of the solver is limited to 10s in our case. Since the infeasibility result can require too much time to the solver, the couples for which the solver do not find a feasible solution in less than 10s are considered infeasible. In Fig. 4.20 and Fig. 4.21 the report of a feasible and an infeasible couple is shown.

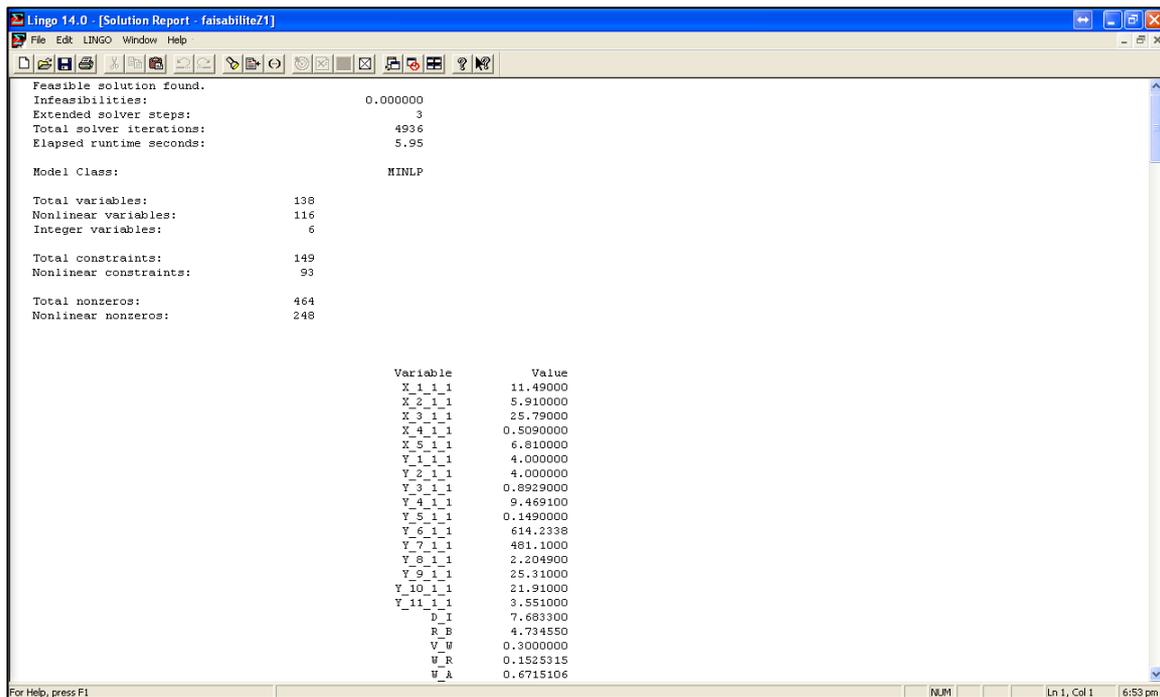


Fig. 4.20 - the Lingo report for a feasible customer-process couple.

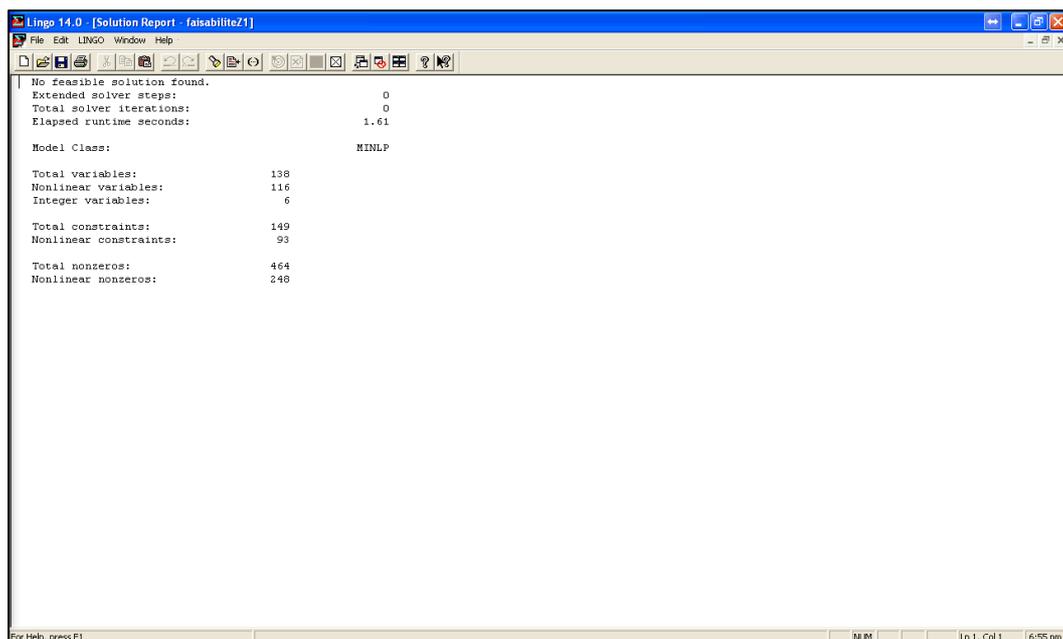


Fig. 4.21 - the Lingo report for an infeasible customer-process couple.

2.3 - The step three: the generation of the feasibility report file

Reading the Lingo reports allows to build a feasibility report file (Fig. 4.22). A feasible couple is represented as Z_x_y , where x is the index for the customer and y is the index for the process. For a feasible couple, 1) the report file contains all the nominal values of the *properties* describing the *customer-process* couple, 2) all the unitary durations and costs of the related manufacturing and assembly stages and 3) the demand for each customer profile. The (2) and (3) are computed on the basis of the nominal values of the couple properties and by using the formulas in the data files.

```

Z_1_1 = 1.0 ;
X_1_1_1 = 11.4 ; X_2_1_1 = 6.0 ; X_3_1_1 = 25.7 ; X_4_1_1 = 0.5 ; X_5_1_1 = 6.9 ; Y_1_1_1 = 4.0 ; Y_2_1_1 = 4.0 ; Y_3_1_1 = 0.81 ; Y_4_1_1 = 9.57 ; Y_5_1_1 = 0.15 ; Y_6_1_1 = 400.0 ;
Lot : 1 ; Z_1_1
S_M1_M2_1 = @FLOOR(( 400.0 - 40) / ( 2.2 + 0.15 )) ;
S_M1_M3_1 = @FLOOR(( 330.0/25.4 - 1 ) * 4.0) ;
T_M1_1_1 = @FLOOR(( 330.0/25.4 - 1 ) * 4.0) / 2 ;
T_M2_1_1 = 4.0 * 0.1 ;
T_M3_1_1 = 400.0 * 0.1 ;
COST_1 = 330.0 * 0.15 * 10 * 4.0 + 400.0 * ( 330.0 / 25.4 - 1) ;

```

Fig. 4.22 - extract from the report file collecting the feasible customer-process couples.

For the *water coil* case, with the above data, the 59.7% of the couples are feasible on the production system 1 and 25.5% on the production system 2. Indeed, none of the requirements of the customer profiles can be meet with the highest values of the copper tube external diameter.

2.4 - The step four: the optimisation model generation

On the basis of the customer demand, the manufacturing costs and the feasibility reports for the two production systems, another piece of code developed in Java allows to generate the optimisation problem. This model is developed in Lingo language and implements the criterion proposed in the chapter 3 for the PV evaluation (Fig. 4.23). The mathematical formulation of the problem in chapter 3 (equations (23-29)) for the water coil case is the following:

$$\text{Min}(C_d + C_s + C_i) \quad (24)$$

$$C_d = \sum_{i=1}^{V_C} \sum_{j=1}^{V_P} C_{ijT} Q_{ij} \quad (25)$$

$$C_s = \sum_{q=1}^2 \sum_{m \in k_q} \left(\text{Set}_{mM_{2q}} l_m + \text{Set}_{mM_{2q}} \left(l_j \frac{t_{mUM_{2q}} S_{mM_{1q}M_{2q}}}{t_{mUM_{2q}} S_{mM_{1q}M_{2q}}} \right) \right) \quad (26)$$

$$C_i = \sum_{q=1}^2 C_{qD} \left| \sum_{m \in k_q} \left[(t_{mUM_{1q}} - t_{mUM_{2q}} Q_{mLM_{2q}}) l_m \right] \right| \quad \text{card} \left(\bigcup_q k_q \right) = V_P, \forall x, y \in q, x \neq y, k_x \cap k_y \equiv \emptyset \quad (27)$$

$$Q_{jLM_{2q}} l_j = \left(\sum_{i=1}^{V_C} Q_{ij} \right) S_{jM_{1q}M_{2q}} \quad \forall j \in [1, \dots, V_P], q \in \{1, 2\} \quad (28)$$

$$S_{jM_{1q}M_{2q}} \leq Q_{jLM_{2q}} \quad \forall j \in [1, \dots, V_P], q \in \{1, 2\} \quad (29)$$

$$l_j, Q_{ij}, Q_{jLM_{2q}} \in N^+ \quad \forall i \in [1, \dots, V_C], j \in [1, \dots, V_P], q \in \{1, 2\} \quad (30)$$

The model is representing: two production systems ($N_P=2$); both of two stages ($O_1=2, O_2=2$); 1 assembly workstation (M1) and two manufacturing workstations (M2-M3) for each production system; each j represents a process variant in a production system. M2 is the workstation that maximise

$$\left| \sum_{m \in k_q} (t_{mUM_{1q}} - t_{mUX_q}) \right|$$

where X represents either M2 or M3

```

Lingo 14.0 - [Lingo Model - final]
File Edit LINGO Window Help
@GIN(LOT_S1_6);@GIN(Q_S1_LOT_T_6);@GIN(Q_S1_TOT_6);
Q_S1_TOT_6=Z_4_8+Z_6_8+Z_7_8+Z_10_8+Z_11_8+Z_12_8+Z_13_8+Z_14_8+Z_15_8+Z_23_8+Z_24_8+Z_25_8+Z_27_8+Z_28_8+Z_30_8+Z_32_8+Z_33_8+Z_34_8+Z_37;
Q_S1_LOT_T_6<=LOT_S1_6*S_S1_M1_M2_6*400000;
Q_S1_LOT_T_6=Q_S1_TOT_6*S_S1_M1_M2_6;
LOT_S1_6=@IF(Q_S1_TOT_6#LT# 1, 0, LOT_S1_6);
LOT_S1_6<=Q_S1_TOT_6;
@GIN(LOT_S1_7);@GIN(Q_S1_LOT_T_7);@GIN(Q_S1_TOT_7);
Q_S1_TOT_7=Z_1_7+Z_3_7+Z_4_7+Z_5_7+Z_6_7+Z_8_7+Z_9_7+Z_10_7+Z_11_7+Z_12_7+Z_17_7+Z_18_7+Z_19_7+Z_20_7+Z_21_7+Z_22_7+Z_23_7+Z_25_7+Z_26_7+Z_29_7+Z_31_7+Z_35;
Q_S1_LOT_T_7<=LOT_S1_7*S_S1_M1_M2_7*400000;
Q_S1_LOT_T_7=Q_S1_TOT_7*S_S1_M1_M2_7;
LOT_S1_7=@IF(Q_S1_TOT_7#LT# 1, 0, LOT_S1_7);
LOT_S1_7<=Q_S1_TOT_7;
@GIN(LOT_S1_8);@GIN(Q_S1_LOT_T_8);@GIN(Q_S1_TOT_8);
Q_S1_TOT_8=Z_2_8+Z_3_8+Z_4_8+Z_5_8+Z_6_8+Z_7_8+Z_9_8+Z_10_8+Z_13_8+Z_14_8+Z_15_8+Z_23_8+Z_24_8+Z_25_8+Z_27_8+Z_28_8+Z_30_8+Z_32_8+Z_33_8+Z_34_8+Z_37;
Q_S1_LOT_T_8<=LOT_S1_8*S_S1_M1_M2_8*400000;
Q_S1_LOT_T_8=Q_S1_TOT_8*S_S1_M1_M2_8;
LOT_S1_8=@IF(Q_S1_TOT_8#LT# 1, 0, LOT_S1_8);
LOT_S1_8<=Q_S1_TOT_8;
@GIN(LOT_S1_9);@GIN(Q_S1_LOT_T_9);@GIN(Q_S1_TOT_9);
Q_S1_TOT_9=Z_2_9+Z_3_9+Z_4_9+Z_5_9+Z_6_9+Z_7_9+Z_8_9+Z_10_9+Z_15_9+Z_16_9+Z_19_9+Z_20_9+Z_21_9+Z_22_9+Z_23_9+Z_25_9+Z_27_9+Z_30_9+Z_31_9+Z_35;
Q_S1_LOT_T_9<=LOT_S1_9*S_S1_M1_M2_9*400000;
Q_S1_LOT_T_9=Q_S1_TOT_9*S_S1_M1_M2_9;
LOT_S1_9=@IF(Q_S1_TOT_9#LT# 1, 0, LOT_S1_9);
LOT_S1_9<=Q_S1_TOT_9;
@GIN(LOT_S1_10);@GIN(Q_S1_LOT_T_10);@GIN(Q_S1_TOT_10);
Q_S1_TOT_10=Z_4_10+Z_5_10+Z_6_10+Z_9_10+Z_10_10+Z_13_10+Z_16_10+Z_18_10+Z_19_10+Z_20_10+Z_21_10+Z_27_10+Z_28_10+Z_29_10+Z_30_10+Z_34_10+Z_35;
Q_S1_LOT_T_10<=LOT_S1_10*S_S1_M1_M2_10*400000;
Q_S1_LOT_T_10=Q_S1_TOT_10*S_S1_M1_M2_10;
LOT_S1_10=@IF(Q_S1_TOT_10#LT# 1, 0, LOT_S1_10);
LOT_S1_10<=Q_S1_TOT_10;
@GIN(LOT_S1_10);@GIN(Q_S1_LOT_T_10);@GIN(Q_S1_TOT_10);
C1_1 = (Z_1_1+Z_2_1+Z_4_1+Z_5_1+Z_6_1+Z_8_1+Z_10_1+Z_12_1+Z_13_1+Z_14_1+Z_15_1+Z_17_1+Z_18_1+Z_19_1+Z_20_1+Z_21_1+Z_22_1+Z_23_1+Z_24_1+Z_26_1+Z_27_1+Z_28_1+Z_29_1+Z_31_1+Z_32_1+Z_33_1+Z_34_1+Z_37) - (S_S1_M1_M3_1 * T_S1_M3_1_1) / (S_S1_M1_M2_1 * T_S1_M2_1_1) * SET_S1_3 + LOT_S1_2 * SET_S1_2 + (LOT_S1_2 * ((S_S1_C3_1 - C_D_1) * -1 * (LOT_S1_1 * T_S1_M1_1_1 - (T_S1_M2_1_1 * Q_S1_LOT_T_1)) + (LOT_S1_2 * T_S1_M1_1_2 - (T_S1_M2_1_2 * Q_S1_LOT_T_2)) + (LOT_S1_3 * T_S1_M1_S_S2_M1_M3_1 = @FLOOR((400.0 - 40) / (2.2 + 0.15)) );
T_S2_M1_1_1 = @FLOOR((330.0/25.4 - 1) * 4.0) ;
T_S2_M2_1_1 = 4.0 * 0.15 ;
T_S2_M3_1_1 = 0 ;
COST_S2_1 = 330.0 * 0.15 * 8 * 4.0 + 400.0 * (330.0 / 25.4 - 1) ;
S_S2_M1_M2_2 = @FLOOR((400.0 - 40) / (2.2 + 0.15)) );
S_S2_M1_M3_2 = @FLOOR((330.0/25.4 - 1) * 4.0) ;
T_S2_M1_1_2 = @FLOOR((330.0/25.4 - 1) * 4.0) / 1.5 ;
T_S2_M2_1_2 = 4.0 * 0.15 ;
T_S2_M3_1_2 = 0 ;
COST_S2_2 = 330.0 * 0.15 * 8 * 4.0 + 400.0 * (330.0 / 25.4 - 1) ;

```

Fig. 4.23 - an example of the optimisation model on Lingo.

2.5 - The two last steps: the solution of the optimisation model

The optimisation model is solved by using Lingo. In the Fig. 23 a screenshot of the Lingo solution report is shown. The solution contains the quantities of each *customer-process* couple that define the best PV. For each customer profile, these quantities allow to determine which are the best process variants and how many times the process should be run.

For instance, for the *water coil* case, the customer profile 1 is defined as follows:

- $T_{R2} = 11.4 \pm 0.1$ °C,
- $T_{R1} = 6.0 \pm 0.1$ °C,
- $T_{A1} = 25.7 \pm 0.1$ °C
- $RH = 0.5 \pm 0.01$
- $Q_S = 6.9 \pm 0.1$ kW.

The forecasted customer-related demand was of 60 products. The best PV associates to this customer profile the following process to cover all the customer demand:

- PUNCH_CUT = 4,
- $N_C = 4$,
- $\Delta = 0.81 \pm 0.01$ mm,
- $D_0 = 10.07 \pm 0.01$ mm,
- $Y_F = 0.15 \pm 0.01$ mm,
- $L_F = 400 \pm 0.1$ mm,
- $L_T = 330 \pm 0.1$ mm,
- $L_C = 2.20 \pm 0.01$ mm,
- $L_{FIN} = 25.40 \pm 0.01$ mm,
- $M_{FIN} = 22.00 \pm 0.01$ mm,

- $V_A = 3.56 \pm 0.01$ m/s.

This process is from S2. Notice that, a customer profile can be associated to more than one process variant, even if they are from different production systems. This means that two different manufacturing processes can satisfy the same customer requirements and they are used together to cover the customer-related demand.

In our case, S2 is used to satisfy great part of the customer demand. S1 should manufacture only the variants that cannot be manufactured in S2. This result is due to the presence of the proposed term C_3 : indeed, if only the direct costs were considered, all the demand would have been satisfied by S1 (the test has been performed on Lingo removing the additional term C_3 from the objective function).

Another effect of the presence of the term C_3 is the number of lots. Indeed, since the duration of the assembly process on M1 (for both sites) was shorter than the time to manufacture the aluminium fins required to start a job, the optimal solution has small lot sizes to reduce the waiting time.

Finally, each customer profile is related to only one process. In our vision, this is due to the high similarity of the process variants. Indeed, even if we considered 20 process variant (10 variants for S1 and 10 for S2), they had almost the same impact on production times and waiting times. Therefore it was not possible to balance the cost difference with a production time reduction obtained with an appropriate lot strategy.

2.6 - Modification of the models to reduce the execution time of the solver

In order to have a result of the design in a reasonable amount of time, two main modifications of the MINLP models have been done. Notice that, the used algorithm is not a heuristic therefore it provides

Parallelisation of the MINLP problems

Actually, the feasibility verifications and the optimisation stages can be merged in one unique phase. In this case the MINLP model should contain the constraints for each customer-process couples, the costs and the demands.

A test of this configuration of the model has been tested for a really small case with unsatisfactory results. Even the aggregation of more than one feasibility test in a single model make the execution time of the solver increase rapidly.

Since the result of the feasibility tests are completely independent, the tests have been performed individually and so they have been performed independently from the optimisation stage. Doing so, the complexity of the single problem and so the performances of the solver have been improved.

Linearization of the lot-relative constraints

For the same reason, a further modification has been done on the mathematical formulation of the PV design problem. Actually this modification is an approximation. Indeed, the mathematical model formalised in chapter 3 cannot be solved in a reasonable time for real-size instance.

In order to cope with this point, a linearization of the lot-relative constraints has been performed to reduce the complexity of the optimisation model. Indeed, the product at the left of this constraint

$$Q_{jLM_{2q}} l_j = \left(\sum_{i=1}^{V_c} Q_{ij} \right) S_{jM_{1q}M_{2q}}$$

has been replaced by $Q_{jLM_{2q}}$ (total quantity of the processes j to be run on M2). The constraint has been linearized and the term C_i has been modified as follows:

$$Q_{jTM_{2q}} \leq l_j S_{jM_{1q}M_{2q}} D$$

$$Q_{jTM_{2q}} = S_{jM_{1q}M_{2q}} Q_{jTM_{1q}}$$

$$Q_{jTM_{1q}} = \left(\sum_j Q_{ij} \right)$$

$$C_i = \sum_{q=1}^2 C_{qD} \left| \sum_{m \in k_q} \left[\left(l_m t_{mUM_{1q}} - t_{mUM_{2q}} Q_{mTM_{1q}} \right) \right] \right|,$$

where D is a big number and $Q_{jTM_{1q}}$ and $Q_{jTM_{2q}}$ is the total number of processes to be run on M1 and M2.

Notice that, this is an approximation because the lot size on the M2 (i.e. Q_{jLM_2} in the previous formulation of the model) is not constrained to be an integer. This approximation is applicable when the lot sizes have reasonably large values.

3 - Conclusions

In this fourth chapter, the *water coil* case study (Fig. 4.24) has been discussed. The formalisation on CATIA of the *water coil* knowledge has been showed. On the basis of this knowledge, a *top-down* PV design has been simulated: this stage has provided the definition of the customer variety and of the process variety to be used for the feasibility tests. Then this KR has been used to illustrate how the algorithm can map the relations between *experiences*.

In the second part of the chapter, a java tool to implement the feasibility tests for the customer-process couples has been shown. The tool allows to convert the information about the variety (collected in a text file) into a set of MINLP models. These models are then executed by a MINLP solver (Lingo) and the results are collected in a report file. Finally this file is used to formulate the optimisation problem that is solved by using Lingo.

In the final part of the chapter, the impacts of the showed implementation of the research questions and industrial issues have been discussed. Some limits regarding the approximations and the treatability have been highlighted in the discussion. Finally, how the proposals can answer to the industrial issue has been argued.

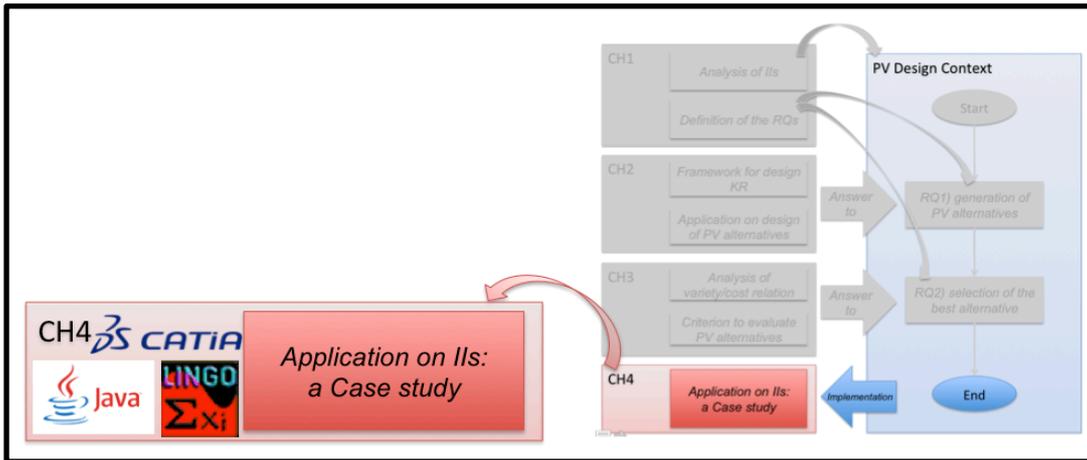


Fig. 4.24 - results of the fourth chapter.

Conclusions: limits and perspectives of the proposals

1 - General conclusions

In this dissertation we presented our approaches about the variety modeling. Starting from three industrial issues about the PV design, we focused on the definition of a KR framework that should support the *design of the variants and the variety at the same stage*. We applied the framework for the PV alternatives generation. Finally we propose a criterion for the evaluations of the alternatives. (Fig. 0.1) The presented proposals represent the results of our works on knowledge formalization (Giovannini et al. 2012a; Giovannini et al. 2012b) and on how to interface the formalised design knowledge with the customers (Giovannini et al. 2013; Giovannini et al. 2014).

The main points of our contribution are the following.

- The concept of *measurement* has been developed to provide a framework for the unambiguous KR. The *measurement* concept, from the transposition of *perception* in the *anti-logicist* works, allows to formalise the knowledge avoiding the instantiation of concepts and the abstract reasoning. In this way, it is possible to guarantee a unique possible interpretation of the represented knowledge.
- The deployment of this framework allows the development of a multi-domain design knowledge model that currently represents the only support to perform a *top-down* PV design. For this reason, our approach is also relevant in the variety management domain.
- The alternatives of PV have been compared considering the impact of the *lot streaming* variables on the manufacturing costs of the PVs. We have shown that the common hypothesis of variety/costs proportionality can be reversed. Starting from a lot-streaming modelling, we have studied the nature of the variety/costs relations for proposing a new comprehensive criterion for the PV alternative evaluation.

In order to support our positions we have developed:

- an algorithm to demonstrate the unambiguity of the proposed KR; this algorithm has been applied on the *water coil* case; our aim was to show how the mapping process between two pieces of knowledge is based only on the syntax: i.e. by avoiding the instantiation of abstract concept, the framework guarantees the unique interpretation of each represented knowledge;
- a *lot-streaming* model formalised with the MINLP has been presented; this model has been solved with the exact solver Lingo to study the variety/cost relations; the result of the analysis has shown the relevance of the sub-lot size and the operations overlapping when a PV evaluation is performed.

The deployed tools for the implementation of the proposed approaches are the following.

- A method to instantiate our framework and enrich with knowledge the CAD files has been proposed. This method has been applied on CATIA but it is theoretically applicable on every

parametric CAD software. This method can support the design for variety approaches by providing:

- an unambiguous way to connect pieces of knowledge from different domains;
- a sound approach to build PV models robust to customer requirement changes.
- A Java tool to transform the customer variety and the process variety into a set of MINLP problem has been developed. The tool also provides an interface with a MINLP solver (Lingo) for the solution of the *customer-process* couple feasibility tests and for the formulation of the PV optimisation problem.

In the next sections, the limits and the perspectives of our proposals vis-à-vis the research questions and the industrial issues are discussed.

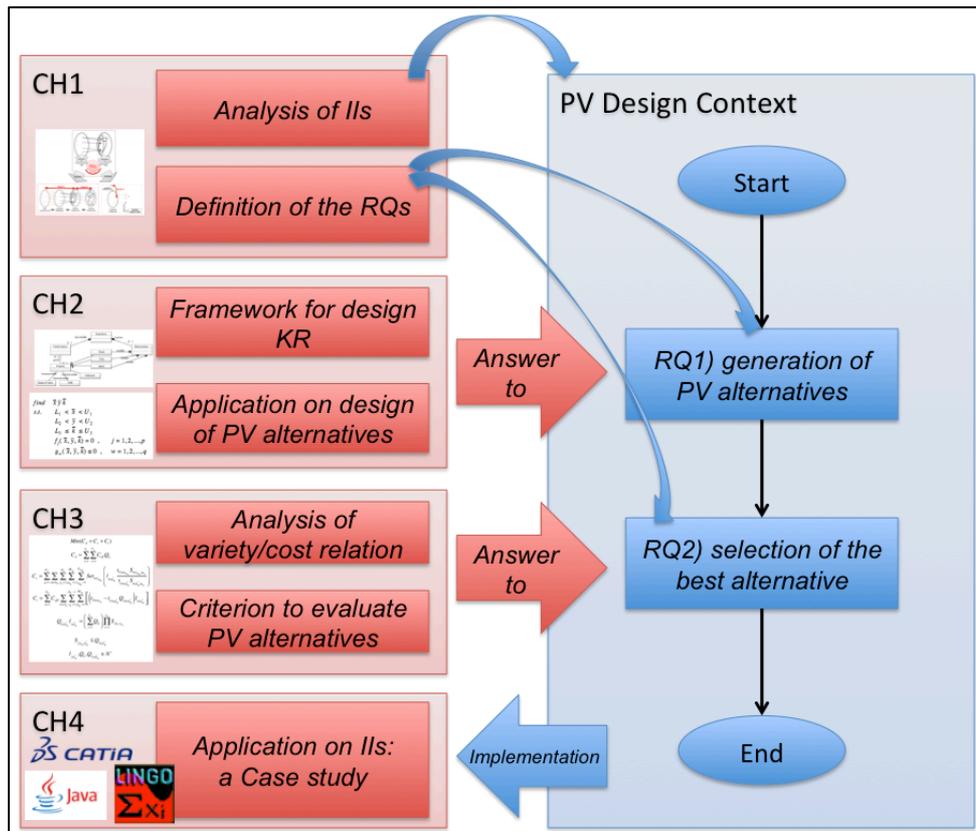


Fig. 0.1 - overview of the structure of the thesis.

2 - Research conclusions

In this section, the aim is to discuss the impacts of the implementation of the proposed approaches on the research questions.

2.1 - First research question (RQ1)

The first research question defined in this thesis is the following.

RQ1 - How to model the variety in order to support a PV design process that at the same time takes into account knowledge that relates the customer profiles, the product variants and the manufacturing process variants?

The proposed implementation uses CATIA V5 to formalise the knowledge from the product and the process domains. The knowledge is formalised following the framework in the chapter 2. This allows to retrieve and connect knowledge coming from different domains without ambiguities.

In the case study, we showed how a PV design can be performed for a *water coil* PV. Formalising the customer profiles, the product and the process features as *measurements* (e.g. temperatures, diameters, features of the punching tools) allows to build a PV model that consider simultaneously customers, products and processes. For the retrieval and the connection of the knowledge needed to satisfy the customer requirements, no-human interpretation was necessary. Even the different experiences can be mapped without ambiguity, as shown during the implementation of the algorithm in *section 1.4*.

The knowledge model represent design knowledge and therefore provide a support for a *top-down* PV design approach. In the section 2, the implementation of this knowledge to identify the feasible *customer-process* couples has been shown.

2.1.1 - Limits of the proposals

The limits of the proposed solution, concern mainly the complexity of the PV knowledge and the number of variants.

- The proposed framework provides a non-ambiguous KR working on the syntax for the knowledge retrieval and mapping. But the syntax is based on set of values for the *properties*. And the more the complexity grows the more the knowledge becomes hard to manage, for the knowledge retrieval, the mapping and for the solution of the MINLP problems. Indeed, even for the water coil, that is only a component, the time to test a couple takes about 10 seconds. For more complex components and systems, coping with the feasibility tests with an exact algorithm should be not treatable. Indeed, notice that, all the algorithms developed and used are exact. Neither approximations nor heuristics have been developed to reduce the problem complexity.
- If the above point concerns the complexity of one single *customer-process* couple, the same treatability issue can be highlighted for the number of the couples to be tested. Indeed, in the case study only 2000 couples are tested on the two production sites (1000 couples/site). Also here, neither approximations nor heuristics have been developed. Therefore all the possibilities (even the less credible) are considered for the feasibility tests.

2.2 - Second research question (RQ2)

The second research question identified in this thesis is the following.

RQ2 - How to compare different process variety alternatives, i.e. process varieties that allow to manufacture PVs able to meet the same customer variety?

In the previous chapter, the implementation of the criterion proposed in the chapter 3 has been showed. The software Lingo was used to model the PV optimisation problem and to solve it. In order to obtain a solution in a reasonable time, some adaptations of the original optimisation model (section CH3.3.3) has been proposed. The water coil case has shown the feasibility of the proposed approach to compare different PVs, even if manufactured on different production systems. Indeed, the unambiguity provided by the measurement-based KR allows this kind of comparisons. In the case study we compared manufacturing process parameters and make-or-buy decisions (i.e. outsourcing of the copper tubes) concerning two different production systems.

2.2.1 - Limits of the proposals

Concerning this proposal, two limits can be identified.

- the approximation of the lot-streaming problem with our criterion represent a limit per-se. In fact, this approximation can lead to suboptimal PVs when the costs allocated on the waiting time are much more relevant than other costs, i.e. when the effect of the approximation become more evident.
- The linearization represents another limit. Indeed, suppressing the integrity constraint for the lot sizes can lead to difficult rounding issues for small lot sizes.

2.3 - Perspectives and possible improvements

From the discussion on the impacts and the limits of the proposed approaches on the research questions, let us derive the following research perspectives.

About the knowledge representation: toward automatic mapping and knowledge reuse

As shown with the algorithm to demonstrate the unambiguity of our KR, a knowledge mapping and retrieval is possible only on the basis of the syntax of the representation. The main perspective related to our knowledge representation framework is to **deeply understand the implication of an unambiguous knowledge representation on problems like the automatic mapping of knowledge**. Indeed, the mapping of knowledge that is at most semi-automatic at the moment (Kalfoglou and Schorlemmer 2003; Rahm and Bernstein 2001). Moreover this same conclusion can lead to some relevant improvement to the **automation of the knowledge reuse**.

A connected perspective concern **the automation and improvement of the knowledge retrieval** as formalised in chapter 4, i.e. on a CAD software. Since this processes are based also on the volume comparison, also here, an appropriate algorithm can provide a more effective support to an automatic mapping or knowledge retrieval (Shah et al. 2001).

About the size and complexity of the design problem

- In order to cope with the limits regarding the number and the complexity of the customer-process feasibility tests, the focus can be on the development of an algorithm that can **enlarge the size of the treatable PV design problem instances**. For instance, this algorithm

could exploit the similarities between the customer-process feasibility tests. Indeed, even if it is not possible to identify a pattern between the constraints of the MINLP problems (i.e. they depend only on the design knowledge), the feasibility tests involve always a verification of intervals. If a **set inversion** using the **interval analysis** (Jaulin and Walter 1993) was possible (i.e. the projection of the process variety on the customer domain estimated in one unique run in place of the customer-process unitary verifications), then the variety would be verifiable in a unique stage, i.e. without the requirement of the discretisation of the domains.

- If the previous perspective concerned the limits of the feasibility tests, the study of an appropriate algorithm can also **overcome the approximation of the lot constraints** for the PV optimisation problem. Indeed, a deeper study of the *lot-streaming* literature can provide the base for the development of an effective algorithm to take into account the integrity of the lots by preserving a reasonable execution time.

3 - Industrial conclusions

Starting from the discussion of the effects of our proposals on the industrial issues, it is possible to highlight the following limits and perspectives.

3.1 - First industrial issue (II1)

The first industrial issues, as formalised in the first chapter is the following.

II1 - *If a customer describes its requirements to more than one seller then more than one different set of resulting products can be proposed to the customer (for the same needs). Are the proposed product variants equally good for the customer? Is there a best alternative for the customer? Is there a best alternative for Trane? And so, given a set of customers, how to design a PV that represents the best way to cover the customers' needs, i.e. the best subset C in Fig. 1.6?*

In the case study, we showed how a customer profile could be formalised as measurements. This allows the non-ambiguity of the customer requirements definition and therefore two sellers should not associate the same customer requirements to different profiles.

Indeed, the result of the PV optimisation on Lingo provides the optimal list of customer-process couples. This list should be the core of the selection tool for the PV. Then, when a set of customer requirements is defined as *measurements*, the requirements are connected in an unambiguous way to the optimal process(es) that can meet them. When more than one process is connected to a customer profile, the decision is based on the quantity of customers that the process should satisfy (i.e. manufacture the product to meet the customer-related requirements) and the quantity of customer that the process has already satisfied.

3.2 - Second industrial issue (II2)

The second industrial issue formalised in chapter 1 is the following.

II2 - *Is it possible to conjugate design effort and local manufacturing cost reduction? In other words, how to link the product variety C and process variety D (Fig. 1.10) in order to impose to D, only the constraints that guarantee the satisfaction of the customer requirements? And consequently, how to find the best process variety, i.e., the best subset of D coming from different manufacturing plants?*

As shown for the water coil case, the proposed knowledge formalisation allows to conjugate product and process knowledge in the same model. Doing so, the product engineers define only the knowledge that is essential to answer to the customer requests. The remainder of the *measurements* is included in the degrees of freedom of the process engineers. In this way, it is also possible to formalise in the same model, the product knowledge and a manufacturing process knowledge relative to more than one manufacturing plant. In the previous chapter, we have shown how two production systems can be modelled in the same PV optimisation problem.

3.3 - Third industrial issue (II3)

The third industrial issue defined in chapter 1 is the following.

II3 - *At Trane, the special design stages can represent the 20% of the order of certain product families. What is the right way to link the main design stage (i.e. related to the PV) with the secondary ones (i.e. for special orders)? Can a PV model adapt the product offer to the new customer requirements? If yes, how to model it?*

In this chapter, the application of our framework for the unambiguous representation of the design knowledge has been discussed. As shown during the knowledge retrieval on CATIA (*section 1.2*) and in the knowledge mapping algorithm implementation (*section 1.3*), the represented *experiences* can be retrieved and reused without human interpretation and therefore without possible ambiguities. Hence, this framework allows to build PV models that are completely reusable.

Let us consider a possible special requirement X.

- The requirement X can concern a new *measurement* to be constrained.
 - If the measurement is already present in the PV model then, in order to reuse the model, we need only to constraint the measurement and automatically remove a degree of freedom for the engineers.
 - If the measurement is not present in the current PV model and there is no-experiences that matches with X, then new knowledge is required to meet X.
- The requirement X can concern a new value for a certain measurement *property* (i.e. *UOM* and *tolerance* are compatible).
 - If the value is included in the *range of value* of the *property* in the current PV model, then no-changes are required to reuse the PV model.

- If the value is not included in the *range of value* of the *property* in the current PV model, then the retrieval of a new *experience* is required. If it does not exist, then new knowledge is necessary to meet the requirement X.

3.4 - Limits and possible improvements

The proposed approach to represent the design knowledge can be used to develop a **knowledge base** including the most relevant pieces of product and process knowledge of the enterprise. For instance, the representation of pieces of knowledge describing the behaviour of components that are shared between many Trane PVs (e.g. *compressors, heat exchangers, sheet metal panels, fans* and others) can speed up the new product development stages. Moreover, using this kind of representation can provide a great improvement in the reactivity when a new variant is necessary or a PV has to be renewed or a PV has to be manufactured by another plant and so forth. For instance, a plug-in can extend the selection tool in order to perform automatically small design modifications to meet some *special* customer requirement.

When the **size of the variety** becomes too large, the proposed approach can be hard to be deployed (see the chapter 4 for a more detailed discussion on this point). In this cases, a degree of process variety can be fixed (to contain the costs, e.g. limiting the production to certain manufacturing plants, machines, tools, etc.) and the projection on the customer profiles that is possible to satisfy can be studied. For instance, an experiment can be designed to understand how the processes cover the customer domain by testing only the more relevant *customer-process* couples. In this way the extensive solution of complex MINLP problems can be avoided. The implementation of this perspective should not require large modifications to the prototype in the chapter 4. Another perspective related to the size constraint is the development of an appropriate heuristic.

At the same way, a projection on the customer domain can be performed to understand how an **existing PV cover the customer requirements**. This can be useful when, at the design stage, the definition of the customer set was fuzzy. Using the proposed framework, the definition of the customer variety can be represented unambiguously. This definition and the design knowledge can be used to understand if the current PV includes variants that are linked with no-customers, customers that are not satisfied and/or a degree of variety highest than the one required from the customer domain. Also this implementation of this perspective should not require large modifications to the prototype in the chapter 4.

List of acronyms

CAD	Computer-Aided Design
II	Industrial Issue
KR	Knowledge Representation
MINLP	Mixed-Integer Non-Linear Programming
PRV	Process Variety
PV	Product Variety
QFD	Quality Function Deployment
RQ	Research Question
HVAC	Heating, Ventilation and Air Conditioning
STS	Space, Time, Shape characterisation

List of figures

FIG. 1.1 - STRUCTURE OF THE THESIS.	11
FIG. 1.2 - ROLE OF THE CHAPTER IN THE THESIS STRUCTURE.	12
FIG. 1.3 - TRANE MANUFACTURING PLANTS OVER THE WORLD.	13
FIG. 1.4 - TRANE PRODUCTS AND SERVICES.	14
FIG. 1.5 - TRANE PRODUCT SELECTION PROCESS ON BPMN.	15
FIG. 1.6 - TRANE PRODUCT SELECTION: BUILDING LINKS FROM THE REAL CUSTOMERS TO THE PRODUCT VARIANTS.	16
FIG. 1.7 - TRANE PRODUCT SELECTION: SELECTION TOOL SCREENSHOT.	17
FIG. 1.8 - TRANE DESIGN AND DEVELOPMENT PROCESS ACCORDING GLOBAL STRATEGY 1.	18
FIG. 1.9 - TRANE DESIGN AND DEVELOPMENT PROCESS ACCORDING STRATEGY 2.	19
FIG. 1.10 - TRANE DESIGN PROCESS: BUILDING LINKS FROM THE PRODUCT VARIANTS TO THE PROCESS VARIANTS.	19
FIG. 1.11 - TRANE ORDER ENTRY FOR SPECIAL ORDERS.	20
FIG. 1.12 - TRANE SPECIAL ORDER ISSUE: STRESS ON THE KNOWLEDGE REUSE.	21
FIG. 1.13 - LINKS BETWEEN DOMAINS INVOLVED IN THE PV DESIGN STAGE.	22
FIG. 1.14 - REPRESENTATION OF THE INDUSTRIAL ISSUES AS MAPPING PROBLEM BETWEEN VARIETY REPRESENTATIONS: FROM DIFFERENT DOMAINS (1ST AND 2ND ISSUES) OR THE SAME DOMAIN (3RD ISSUE).	23
FIG. 1.15 - RESULTS OF THE FIRST CHAPTER.	25
FIG. 1.16 - ILLUSTRATION OF THE RESEARCH QUESTIONS.	25
FIG. 2.1 - ROLE OF THE CHAPTER IN THE THESIS STRUCTURE.	28
FIG. 2.2 - DOMAIN CONNECTIONS TO BE REVIEWED AND SECTION NUMBERS OF THE REVIEWS.	30
FIG. 2.3 - THE DIFFERENT REPRESENTATIONS OF THE VARIETY PERFORMED BY THE CUSTOMER.	31
FIG. 2.4 - THE ACTUAL RESULTS OF THE PAPERS REVIEWED IN THE SECTION 1.2.	35
FIG. 2.5 - PART OF THE REACTIVE SYSTEM ARCHITECTURE IN (KAELBLING 1987).	41
FIG. 2.6 - REPRESENTATION OF HOW THE KNOWLEDGE IS USED IN A LOGICIST AND ANTI-LOGICIST-BASED SYSTEMS.	42
FIG. 2.7 - CONCEPTUAL MODEL OF THE FRAMEWORK.	45
FIG. 2.8 - REPRESENTATION OF THE BARE TUBE COIL EXPERIENCE ON CATIA V5.	46
FIG. 2.9 - REPRESENTATION OF THE BARE TUBE COIL: THE VOLUME HIGHLIGHTED IS RELATED TO THE C_P, RO_A AND T_A1 MEASUREMENTS.	47
FIG. 2.10 - REPRESENTATION OF THE BARE TUBE COIL: THE TRANSFORMATIONS IN CATIA V5.	48
FIG. 2.11 - FLOWCHART REPRESENTING THE ALGORITHM PRESENTED IN THE CURRENT SECTION.	50
FIG. 2.12 - THE CONNECTIONS BETWEEN VARIETY MODELS COMING FROM DIFFERENT DOMAINS PROVIDED BY THE PROPOSED FRAMEWORK.	51
FIG. 2.13 - RESULTS OF THE SECOND CHAPTER.	55
FIG. 3.1 - ROLE OF THE CHAPTER IN THE THESIS STRUCTURE.	56
FIG. 3.2 - EXAMPLES OF QFD MATRIX FOR THE GVI COMPUTATION, FROM (MARTIN,2000).	58
FIG. 3.3 - EXAMPLES OF MATRIX FOR THE CPI COMPUTATION, FROM (MARTIN,2000).	61
FIG. 3.4 - SCENARIO IN WHICH AN ASSEMBLY SYSTEM CAN BE MODELLED AS A FLOW SHOP.	68
FIG. 3.5 - MODELLED ASSEMBLY SYSTEM.	69
FIG. 3.6 - THE GRAPHICAL REPRESENTATION OF THE WAITING TIME AS IN OUR APPROACH.	73

FIG. 3.7 - REPRESENTATION OF THE PRODUCTION SYSTEM 1 ($Q=1$); EACH BLACK BOX REPRESENTS A WORKSTATION; AT THE FIRST STAGE (I.E. THE LAST CHRONOLOGICALLY) THERE IS ONLY ONE ASSEMBLY WORKSTATION THAT HAS TO PROVIDE THE FINAL PRODUCT.....	75
FIG. 3.8 - RESULTS OF THE THIRD CHAPTER.....	76
FIG. 4.1 - ROLE OF THE CHAPTER IN THE THESIS STRUCTURE.....	77
FIG. 4.2 - PICTURES OF THE TRANE FAN COIL PV.....	78
FIG. 4.3 - THE REPRESENTATION OF THE SPACE OF THE MEASUREMENT T_{A1}	79
FIG. 4.4 - THE REPRESENTATION OF THE WATER COIL EXPERIENCE ON CATIA V5.....	80
FIG. 4.5 - THE REPRESENTATION OF THE UOM RELATED TO THE PARAMETER Y_F , I.E. FIN THICKNESS.....	80
FIG. 4.6 - THE REPRESENTATION OF THE TOLERANCE RELATED TO THE PARAMETER Y_F	81
FIG. 4.7 - THE REPRESENTATION OF THE RANGE OF VALUES FOR THE PARAMETER Y_F	81
FIG. 4.8 - THE REPRESENTATION ON CATIA OF THE TRANSFORMATIONS AS CHECK RULES. BESIDE EACH CHECK A RED OR GREEN LIGHT SHOWS IF THE PARAMETER VALUES ARE COMPLIANT WITH THE FORMALISED CONSTRAINT.....	82
FIG. 4.9 - THE COIL AND THE VOLUME OF AIR IN THE CUSTOMER REQUIREMENTS.....	84
FIG. 4.10 - SURFACES IN WHICH THE ENTERING AND EXITING WATER TEMPERATURE CAN BE OBSERVED.....	85
FIG. 4.11 - THE MACRO FOR REPRESENTING THE TIME-DEPENDENCES IN THE INSTALLATION PROCESS.....	86
FIG. 4.12 - THE GEOMETRIES FOR THE PARAMETER $TIME1=1s$	86
FIG. 4.13 - THE GEOMETRIES FOR THE PARAMETER $TIME1=0s$	87
FIG. 4.14 - EXCERPT OF ALGORITHM PROPOSED IN CHAPTER 2: CONCLUSIONS IF ALL STSs ARE COMPATIBLE.....	89
FIG. 4.15 - EXCERPT OF ALGORITHM PROPOSED IN CHAPTER 2: CONCLUSIONS IF ONLY SOME STSs ARE COMPATIBLE.....	90
FIG. 4.16 - FLOWS OF THE DEVELOPED JAVA TOOL.....	91
FIG. 4.17 - EXTRACT OF THE INPUT FILE THAT REPRESENTS THE PROPERTY VALUES AND THE CONSTRAINT TO GENERATE THE CUSTOMER-PROCESS FEASIBILITY TESTS.....	91
FIG. 4.18 - CUSTOMER VARIETY AND PROCESS VARIETY DEFINITIONS IN THE MINLP MODEL.....	94
FIG. 4.19 - TRANSFORMATIONS IN THE MINLP MODEL.....	94
FIG. 4.20 - THE LINGO REPORT FOR A FEASIBLE CUSTOMER-PROCESS COUPLE.....	95
FIG. 4.21 - THE LINGO REPORT FOR AN INFEASIBLE CUSTOMER-PROCESS COUPLE.....	95
FIG. 4.22 - EXTRACT FROM THE REPORT FILE COLLECTING THE FEASIBLE CUSTOMER-PROCESS COUPLES.....	96
FIG. 4.23 - AN EXAMPLE OF THE OPTIMISATION MODEL ON LINGO.....	97
FIG. 4.24 - RESULTS OF THE FOURTH CHAPTER.....	100
FIG. 0.1 - OVERVIEW OF THE STRUCTURE OF THE THESIS.....	102

List of Tables

TAB. 2.1 - MAIN CHARACTERISTICS OF THE PAPERS ABOUT THE MANUFACTURING DOMAIN INTEGRATION IN THE PV DESIGN.	35
TAB. 2.2 - REVIEWED REFERENCES AND DOMAIN CONNECTIONS.....	37
TAB. 2.3 - HOW THE VARIETY IS REPRESENTED IN THE WORKS ABOUT CONFIGURATION SYSTEM DESIGN (FROM (GIOVANNINI ET AL. 2014)).....	38
TAB. 2.4 - TRANSPOSITION OF CONCEPTS.	43
TAB. 3.1A - INDEX FOR PV EVALUATION BASED ON PRODUCT FEATURES.....	59
TAB. 3.2 - INDEX FOR PV EVALUATION BASED ON PROCESS FEATURES.	62
TAB. 3.3 - CRITERIA FOR VARIETY ASSESSMENT.	65
TAB. 3.4 - DATA ABOUT THE WORKING STATIONS.	69
TAB. 3.5 - DATA ABOUT THE FIRST PRV.....	69
TAB. 3.6 - DATA ABOUT THE SECOND PRV.....	70
TAB. 3.7 - RESULT OF THE EXPERIENCE ON THE LOT STREAMING MODEL FOR PRV1.....	71
TAB. 3.8 - RESULT OF THE EXPERIENCE ON THE LOT STREAMING MODEL FOR PRV2.....	71

Personal references

International journals

Giovannini, A., Aubry, A., Panetto, H., Dassisti, M., El Haouzi, H., 2012b. Ontology-Based System for supporting Manufacturing Sustainability. *Annual Reviews in Control* 36, 309–317.

Giovannini, A., Aubry, A., Panetto, H., El Haouzi, H., Pierrel, L., Dassisti, M., SUBMITTED. Anti-logicist framework for design-knowledge representation. *Annual Reviews in Control*.

International conferences

Giovannini, A., Aubry, A., Panetto, H., Dassisti, M., El Haouzi, H., 2012a. Knowledge-Based System for Manufacturing Sustainability. Presented at the 14th IFAC Symposium on Information Control Problems in Manufacturing - INCOM, Bucharest.

Giovannini, A., Aubry, A., Panetto, H., El Haouzi, H., 2013. Mass customisation in sustainable networked enterprises, in: *Proceedings of the 14th IFIP Working Conference on VIRTUAL ENTERPRISE*, Sept. 30th-Oct. 2nd, Dresden, Germany, Collaborative Systems for Reindustrialization. Springer, pp. 657–666.

Giovannini, A., Aubry, A., Panetto, H., El Haouzi, H., Pierrel, L., Dassisti, M., 2014. Approach for the rationalisation of product lines variety. Presented at the 19th IFAC World Congress, Cape Town, South Africa, pp. 3280–3291.

Other publications

Giovannini, A., Aubry, A., Panetto, H., El Haouzi, H., 2013. A systemic perspective for mass customization: an approach for product lines definition. *Insight Journal of INCOSE* 16, 10–12.

Giovannini, A., Aubry, A., Panetto, H., El Haouzi, H., Pierrel, L., 2014. Knowledge-based system for customer variety representation in product lines definition, in: *IFIP (Ed.), 6th Doctoral Workshop on "Product and Asset Lifecycle Management", PALM 2014*. Les Diablerets, Switzerland.

References

- Agard, Bruno, and Bernard Penz. 2009. "A Simulated Annealing Method Based on a Clustering Approach to Determine Bills of Materials for a Large Product Family." *International Journal of Production Economics* 117 (2): 389–401.
- Agre, Philip E., and David Chapman. 1987. "Pengi: An Implementation of a Theory of Activity." In *AAAI*, 87:286–272. <http://www.aaai.org/Library/AAAI/1987/aaai87-048.php>.
- Air-conditioning, Heating & Refrigeration Institute. 2001. "AHRI Standard 410-2001 with Addenda 1, 2 and 3: Forced-Circulation Air-Cooling and Air-Heating Coils."
- Akao, Yoji. 2004. *Quality Function Deployment: Integrating Customer Requirements Into Product Design*. Productivity Press.
- Aldanondo, M., and E. Vareilles. 2008. "Configuration for Mass Customization: How to Extend Product Configuration towards Requirements and Process Configuration." *Journal of Intelligent Manufacturing* 19 (5): 521–35.
- Alizon, Fabrice, Steven B. Shooter, and Timothy W. Simpson. 2009. "Assessing and Improving Commonality and Diversity within a Product Family." *Research in Engineering Design* 20 (4): 241–53.
- ASHRAE. 2009. *2009 ASHRAE Handbook: Fundamentals*. 2009 Ashrae Handbook - Fundamentals. American Society of Heating, Refrigeration and Air-Conditioning Engineers. <http://books.google.fr/books?id=D65XPgAACAAJ>.
- . 2012. *2012 ASHRAE Handbook: Heating, Ventilating, and Air-Conditioning Systems and Equipment*. Atlanta, Ga.: ASHRAE.
- Barajas, Marco, and Bruno Agard. 2009. *The Use of Fuzzy Logic in Product Family Development: Literature Review and Opportunities*. CIRRELT. http://www.researchgate.net/publication/225417015_The_use_of_fuzzy_logic_in_product_family_development_literature_review_and_opportunities/file/9c9605298be9b89a57.pdf.
- Belloni, Alexandre, Robert Freund, Matthew Selove, and Duncan Simester. 2008. "Optimizing Product Line Designs: Efficient Methods and Comparisons." *Management Science* 54 (9): 1544–52.
- Blecker, T., N. Abdelkafi, B. Kaluza, and G. Kreutler. 2004. "Mass Customization vs. Complexity: A Gordian Knot?" In , 890–903. <http://mpr.a.uni-muenchen.de/5290/>.
- Brooks, Rodney A. 1991. "Intelligence without Representation." *Artificial Intelligence* 47 (1): 139–59.
- Chan, F. T. S., T. C. Wong, and L. Y. Chan. 2008. "Lot Streaming for Product Assembly in Job Shop Environment." *Robotics and Computer-Integrated Manufacturing* 24 (3): 321–31. doi:10.1016/j.rcim.2007.01.001.
- Chang *, Jen Huei, and Huan Neng Chiu. 2005. "A Comprehensive Review of Lot Streaming." *International Journal of Production Research* 43 (8): 1515–36. doi:10.1080/00207540412331325396.
- Chauvet, G. A. 1993. "Hierarchical Functional Organization of Formal Biological Systems: A Dynamical Approach. I. The Increase of Complexity by Self-Association Increases the Domain of Stability of a Biological System." *Philosophical Transactions of the Royal Society of London. Series B: Biological Sciences* 339 (1290): 425–44.
- Cheng, M., N.J. Mukherjee, and S.C. Sarin. 2013. "A Review of Lot Streaming." *International Journal of Production Research* 51 (23-24): 7023–46. doi:10.1080/00207543.2013.774506.
- Chowdhury, Souma, Achille Messac, and Ritesh A. Khire. 2011. "Comprehensive Product Platform Planning (cp3) Framework." *Journal of Mechanical Design* 133 (10): 101004.
- Collier, David A. 1981. "The Measurement and Operating Benefits of Component Part Commonality." *Decision Sciences* 12 (1): 85–96.
- Davis, Ernest. 1998. "Naive Physics Perplex." *AI Magazine* 19 (4): 51.

- Davis, Stanley M. 1987. *Future Perfect*. Addison-Wesley.
- Du, X., J. Jiao, and M. M. Tseng. 2001. "Product Platform Representation: A Graph Grammar Approach." *Manufacturing Systems* 31 (2): 1–7.
- EIMaraghy, H., G. Schuh, W. EIMaraghy, F. Piller, P. Schönsleben, M. Tseng, and A. Bernard. 2013. "Product Variety Management." *CIRP Annals - Manufacturing Technology* 62 (2): 629–52. doi:10.1016/j.cirp.2013.05.007.
- Fayyad, Usama, Gregory Piatetsky-Shapiro, and Padhraic Smyth. 1996. "From Data Mining to Knowledge Discovery in Databases." *AI Magazine* 17 (3): 37.
- Fellini, Ryan, Michael Kokkolaras, Nestor Michelena, Panos Papalambros, A. Perez-Duarte, Kazuhiro Saitou, and Peter Fenyes. 2004. "A Sensitivity-Based Commonality Strategy for Family Products of Mild Variation, with Application to Automotive Body Structures." *Structural and Multidisciplinary Optimization* 27 (1-2): 89–96.
- Fellini, Ryan, Michael Kokkolaras, Panos Papalambros, and Alexis Perez-Duarte. 2005. "Platform Selection under Performance Bounds in Optimal Design of Product Families." *Journal of Mechanical Design* 127 (4): 524–35.
- Ferguson, Scott, Andrew Olewnik, and Phil Cormier. 2011. "Exploring Marketing to Engineering Information Mapping in Mass Customization: A Presentation of Ideas, Challenges and Resulting Questions." In *ASME 2011 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, 609–23. American Society of Mechanical Engineers. <http://proceedings.asmedigitalcollection.asme.org/proceeding.aspx?articleid=1640658>.
- Fixson, Sebastian K. 2007. "Modularity and Commonality Research: Past Developments and Future Opportunities." *Concurrent Engineering* 15 (2): 85–111.
- Fujita, Kikuo, Hirofumi Amaya, and Ryota Akai. 2013. "Mathematical Model for Simultaneous Design of Module Commonalization and Supply Chain Configuration toward Global Product Family." *Journal of Intelligent Manufacturing* 24 (5): 991–1004. doi:10.1007/s10845-012-0641-x.
- Fujita, Kikuo, and Hiroko Yoshida. 2004. "Product Variety Optimization Simultaneously Designing Module Combination and Module Attributes." *Concurrent Engineering* 12 (2): 105–18.
- Gao, Fei, Yuanming Zhang, and Gang Xiao. 2013. "A Methodology to Support Platform Parameters Determination through Considering Main Factors in Product Life Cycle." *International Journal of Computer Integrated Manufacturing* 27 (5): 470–78. doi:10.1080/0951192X.2013.814166.
- Gao, P., L. F. Lin, M. Cai, and J. X. Dong. 2003. "Automatic Acquisition of Product Configuration Model Based on Ontology Mapping." *Jisuanji Jicheng Zhizao Xitong/Computer Integrated Manufacturing System(China)* 9 (9): 810–16.
- Giovannini, A., A. Aubry, H. Panetto, M. Dassisti, and H. El Haouzi. 2012a. "Knowledge-Based System for Manufacturing Sustainability." In . Bucharest. <http://hal.archives-ouvertes.fr/hal-00672215/>.
- . 2012b. "Ontology-Based System for Supporting Manufacturing Sustainability." *Annual Reviews in Control* 36 (2): 309–17.
- Giovannini, A., A. Aubry, H. Panetto, and H. El Haouzi. 2013. "Mass Customisation in Sustainable Networked Enterprises." In *Proceedings of the 14th IFIP Working Conference on VIRTUAL ENTERPRISE*, Sept. 30th-Oct. 2nd, Dresden, Germany, Collaborative Systems for Reindustrialization, 408:657–66. Springer. <http://hal.archives-ouvertes.fr/hal-00850144/>.
- Giovannini, A., A. Aubry, H. Panetto, H. El Haouzi, L. Pierrel, and M. Dassisti. 2014. "Approach for the Rationalisation of Product Lines Variety." <http://www.nt.ntnu.no/users/skoge/prost/proceedings/ifac2014/media/files/2226.pdf>.
- Green, Paul E., and Venkatachary Srinivasan. 1978. "Conjoint Analysis in Consumer Research: Issues and Outlook." *Journal of Consumer Research*, 103–23.
- Helo, P. T., Q. L. Xu, S. J. Kyllönen, and R. J. Jiao. 2010. "Integrated Vehicle Configuration System—Connecting the Domains of Mass Customization." *Computers in Industry* 61 (1): 44–52.

- Hölttä-Otto, Katja, and Olivier de Weck. 2007. "Degree of Modularity in Engineering Systems and Products with Technical and Business Constraints." *Concurrent Engineering* 15 (2): 113–26.
- Hong, G., L. Hu, D. Xue, Y. L. Tu, and Y. L. Xiong. 2008. "Identification of the Optimal Product Configuration and Parameters Based on Individual Customer Requirements on Performance and Costs in One-of-a-Kind Production." *International Journal of Production Research* 46 (12): 3297–3326.
- Inakoshi, H., S. Okamoto, Y. Ohta, and N. Yugami. 2001. "Effective Decision Support for Product Configuration by Using Cbr." In *Proceedings of the Fourth International Conference on Case-Based Reasoning (ICCBR), Workshop Casebased Reasoning in Electronic Commerce*, Vancouver, Canada. <http://www.aic.nrl.navy.mil/papers/2001/AIC-01-003/ws3/ws3toc4.pdf>.
- Jacobs, F. Robert, and Daniel J. Bragg. 1988. "Repetitive Lots: Flow-Time Reductions Through Sequencing and Dynamic Batch Sizing." *Decision Sciences* 19 (2): 281–94.
- Janitza, Dennis, Martin Lacher, Maik Maurer, Udo Pulm, and Henning Rudolf. 2003. "A Product Model for Mass – Customisation Products." In *Knowledge-Based Intelligent Information and Engineering Systems*, edited by Vasile Palade, Robert Howlett, and Lakhmi Jain, 2774:1023–29. *Lecture Notes in Computer Science*. Springer Berlin / Heidelberg. <http://www.springerlink.com/content/2vhvy11bq93r1b2p/abstract/>.
- Jariri, Fereydoon, and Seyed Hesameddin Zegordi. 2008. "Quality Function Deployment Planning for Platform Design." *The International Journal of Advanced Manufacturing Technology* 36 (5-6): 419–30.
- Jaulin, Luc, and Eric Walter. 1993. "Set Inversion via Interval Analysis for Nonlinear Bounded-Error Estimation." *Automatica* 29 (4): 1053–64.
- Jiao, Jianxin, and Mitchell M. Tseng. 2000. "Understanding Product Family for Mass Customization by Developing Commonality Indices." *Journal of Engineering Design* 11 (3): 225–43.
- Kaelbling, Leslie Pack. 1987. "An Architecture for Intelligent Reactive Systems." *Reasoning about Actions and Plans*, 395–410.
- Kalfoglou, Yannis, and Marco Schorlemmer. 2003. "Ontology Mapping: The State of the Art." *The Knowledge Engineering Review* 18 (01): 1–31.
- Kazemzadeh, R. B., Majid Behzadian, Mohammad Aghdasi, and Amir Albadvi. 2009. "Integration of Marketing Research Techniques into House of Quality and Product Family Design." *The International Journal of Advanced Manufacturing Technology* 41 (9-10): 1019–33. doi:10.1007/s00170-008-1533-2.
- Khajavirad, Aida, and Jeremy J. Michalek. 2008. "A Decomposed Gradient-Based Approach for Generalized Platform Selection and Variant Design in Product Family Optimization." *Journal of Mechanical Design* 130 (7): 071101.
- . 2009. "A Deterministic Lagrangian-Based Global Optimization Approach for Quasiseparable Nonconvex Mixed-Integer Nonlinear Programs." *Journal of Mechanical Design* 131 (5): 051009.
- Khajavirad, Aida, Jeremy J. Michalek, and Timothy W. Simpson. 2009. "An Efficient Decomposed Multiobjective Genetic Algorithm for Solving the Joint Product Platform Selection and Product Family Design Problem with Generalized Commonality." *Structural and Multidisciplinary Optimization* 39 (2): 187–201.
- Khalaf, Radwan El Hadj, Bruno Agard, and Bernard Penz. 2010a. "An Experimental Study for the Selection of Modules and Facilities in a Mass Customization Context." *Journal of Intelligent Manufacturing* 21 (6): 703–16.
- . 2010b. "Simultaneous Design of a Product Family and Its Related Supply Chain Using a Tabu Search Algorithm." *International Journal of Production Research* 49 (19): 5637–56. doi:10.1080/00207543.2010.519737.
- Kota, Sridhar, Kannan Sethuraman, and Raymond Miller. 2000. "A Metric for Evaluating Design Commonality in Product Families." *Journal of Mechanical Design* 122 (4): 403–10.
- Lamothe, Jacques, Khaled Hadj-Hamou, and Michel Aldanondo. 2006. "An Optimization Model for Selecting a Product Family and Designing Its Supply Chain." *European Journal of Operational Research* 169 (3): 1030–47. doi:10.1016/j.ejor.2005.02.007.

- Li, Bin, Liping Chen, Zhengdong Huang, and Yifang Zhong. 2006. "Product Configuration Optimization Using a Multiobjective Genetic Algorithm." *The International Journal of Advanced Manufacturing Technology* 30 (1): 20–29. doi:10.1007/s00170-005-0035-8.
- Lieber, Romain, Jean-Marc Dupont, Fabien Bouffaron, and Gérard Morel. 2013. "Improving physical-physiological interaction requirements for maintenance enabling systems specification." In *12th IFAC/IFIP/IFORS/IEA Symposium on Analysis, Design, and Evaluation of Human-Machine Systems*. Las Vegas, États-Unis. <http://hal.archives-ouvertes.fr/hal-00859454>.
- Li, L., and G. Q. Huang. 2009. "Multiobjective Evolutionary Optimisation for Adaptive Product Family Design." *International Journal of Computer Integrated Manufacturing* 22 (4): 299–314. doi:10.1080/09511920802014920.
- Liu, X. B., C. F. Yuan, T. Y. Gao, W. Sun, and W. L. Wang. 2003. "Layered Product Configuration Model Based on Feature and Faced to Customer." *Jisuanji Jicheng Zhizao Xitong/Computer Integrated Manufacturing System(China)* 9 (7): 527–31.
- Liu, Zhuo, Yoke San Wong, and Kim Seng Lee. 2011. "A Manufacturing-Oriented Approach for Multi-Platforming Product Family Design with Modified Genetic Algorithm." *Journal of Intelligent Manufacturing* 22 (6): 891–907.
- Martin, Mark V., and Kosuke Ishii. 1997. "Design for Variety: Development of Complexity Indices and Design Charts." In *Proceedings of 1997 ASME Design Engineering Technical Conferences*, 14–17. http://me317.stanford.edu/twiki/pub/ME317AdminWeb/ME317CourseReaderDFVProductStructureGraph/aa19.rdr317a.dfv_.pdf.
- . 2000. "Design for Variety: A Methodology for Developing Product Platform Architectures." *Proceedings of DETC2000*, 10–13.
- Mazo, R., C. Salinesi, D. Diaz, O. Djebbi, and A. Lora-Michiels. 2012. "Constraints: The Heart of Domain and Application Engineering in the Product Lines Engineering Strategy." *International Journal of Information System Modeling and Design (IJISMD)* 3 (2): 33–68.
- Michalek, Jeremy J., Oben Ceryan, Panos Y. Papalambros, and Yoram Koren. 2006. "Balancing Marketing and Manufacturing Objectives in Product Line Design." *Journal of Mechanical Design* 128 (6): 1196–1204.
- Moily, Jaya P. 1986. "Optimal and Heuristic Procedures for Component Lot-Splitting in Multi-Stage Manufacturing Systems." *Management Science* 32 (1): 113–25.
- Ostrosi, Egon, A.-J. Fougères, Michel Ferney, and Didier Klein. 2012. "A Fuzzy Configuration Multi-Agent Approach for Product Family Modelling in Conceptual Design." *Journal of Intelligent Manufacturing* 23 (6): 2565–86.
- Park, Jaeil, and Timothy W. Simpson. 2007. "Toward an Activity-Based Costing System for Product Families and Product Platforms in the Early Stages of Development." *International Journal of Production Research* 46 (1): 99–130. doi:10.1080/00207540600825240.
- Pohl, K., G. Bockle, and F. Van Der Linden. 2005. *Software Product Line Engineering*. Vol. 10. Springer. http://cs5292.userapi.com/u11728334/docs/7ca820593ad0/Klaus_Pohl_Software_Product_Line_Engineering.pdf.
- Qin, Y. H., and G. X. Wei. 2010. "Product Configuration Based on Modular Product Family Modelling." *Journal of Computational Information Systems* 6 (7): 2321–31.
- Qu, T., S. Bin, George Q. Huang, and H.D. Yang. 2010. "Two-Stage Product Platform Development for Mass Customisation." *International Journal of Production Research* 49 (8): 2197–2219. doi:10.1080/00207541003738881.
- Rahm, Erhard, and Philip A. Bernstein. 2001. "A Survey of Approaches to Automatic Schema Matching." *The VLDB Journal* 10 (4): 334–50.
- Rojas Arciniegas, Alvaro J., and Harrison M. Kim. 2010. "Optimal Component Sharing in a Product Family by Simultaneous Consideration of Minimum Description Length and Impact Metric." *Engineering Optimization* 43 (2): 175–92. doi:10.1080/0305215X.2010.486032.

- Ruben, Robert A., and Farzad Mahmoodi. 1998. "Lot Splitting in Unbalanced Production Systems*." *Decision Sciences* 29 (4): 921–49.
- Russell, Bertrand, and Alfred North Whitehead. 1925. "Principia Mathematica Vol." <http://philpapers.org/rec/RUSPMV>.
- Sandborn, Peter, Varun Prabhakar, and Bo Eriksson. 2008. "The Application of Product Platform Design to the Reuse of Electronic Components Subject to Long-Term Supply Chain Disruptions." In *ASME 2008 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, 989–98. American Society of Mechanical Engineers. <http://proceedings.asmedigitalcollection.asme.org/proceeding.aspx?articleid=1627276>.
- Sarin, Subhash C., and Puneet Jaiprakash. 2007. *Flow Shop Lot Streaming*. Springer. http://books.google.fr/books?hl=en&lr=&id=vSmRkPo2zQEC&oi=fnd&pg=PR7&dq=Sarin+and+Jaiprakash+lot+streaming+review&ots=U5NfMTKMcX&sig=GqEA1u_XHOT3_4dasPB4Ccf9dJU.
- Sarin, Subhash C., Liming Yao, and Dan Trietsch. 2011. "Single-batch Lot Streaming in a Two-stage Assembly System." *International Journal of Planning and Scheduling* 1 (1): 90–108. doi:10.1504/IJPS.2011.044604.
- Shah, Jami J., David Anderson, Yong Se Kim, and Sanjay Joshi. 2001. "A Discourse on Geometric Feature Recognition from CAD Models." *Journal of Computing and Information Science in Engineering* 1 (1): 41–51.
- Shahzad, Kashif M., and Khaled Hadj-Hamou. 2013. "Integrated Supply Chain and Product Family Architecture under Highly Customized Demand." *Journal of Intelligent Manufacturing* 24 (5): 1005–18. doi:10.1007/s10845-012-0630-0.
- Siddique, Zahed, David W. Rosen, and Nanxin Wang. 1998. "On the Applicability of Product Variety Design Concepts to Automotive Platform Commonality." In *ASME Design Engineering Technical Conferences-Design Theory and Methodology*. http://www.researchgate.net/publication/243771347_On_the_applicability_of_product_variety_design_concepts_to_automotive_platform_commonality/file/60b7d529fb9cc72466.pdf.
- Siddique, Zahed, and Lindsay Wilmes. 2007. "An Application of Design Space for Assembly Process Reasoning to Utilize Current Assembly Plant Resources for New Product Family Members." *Journal of Intelligent Manufacturing* 18 (1): 171–84.
- Simpson, Timothy W., Carolyn Conner Seepersad, and Farrokh Mistree. 2001. "Balancing Commonality and Performance within the Concurrent Design of Multiple Products in a Product Family." *Concurrent Engineering* 9 (3): 177–90.
- Simpson, Timothy W., Zahed Siddique, and Jianxin Roger Jiao. 2006. "Platform-Based Product Family Development." In *Product Platform and Product Family Design*, 1–15. Springer. http://link.springer.com/chapter/10.1007/0-387-29197-0_1.
- Slowinski, Roman. 1998. *Fuzzy Sets in Decision Analysis, Operations Research and Statistics*. Kluwer Academic Publishers USA. <http://www.kybernetika.cz/content/1999/5/665/paper.pdf>.
- Sullivan, Eric, Scott Ferguson, and Joseph Donndelinger. 2011. "Exploring Differences in Preference Heterogeneity Representation and Their Influence in Product Family Design." In *ASME 2011 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, 81–92. American Society of Mechanical Engineers. <http://proceedings.asmedigitalcollection.asme.org/proceeding.aspx?articleid=1640530>.
- Thevenot, Henri J., Fabrice Alizon, Timothy W. Simpson, and Steven B. Shooter. 2007. "An Index-Based Method to Manage the Tradeoff between Diversity and Commonality during Product Family Design." *Concurrent Engineering* 15 (2): 127–39.
- Thevenot, Henri J., and Timothy W. Simpson. 2006. "Commonality Indices for Product Family Design: A Detailed Comparison." *Journal of Engineering Design* 17 (2): 99–119. doi:10.1080/09544820500275693.
- . 2007. "A Comprehensive Metric for Evaluating Component Commonality in a Product Family." *Journal of Engineering Design* 18 (6): 577–98. doi:10.1080/09544820601020014.

- Tseng, M. M., and J. Jiao. 2001. "Mass Customization." *Handbook of Industrial Engineering*, 3rd Edition, New York: Wiley, 684–709.
- Tucker, Conrad S., and Harrison M. Kim. 2008. "Optimal Product Portfolio Formulation by Merging Predictive Data Mining with Multilevel Optimization." *Journal of Mechanical Design* 130 (4): 041103.
- Van Harmelen, Frank, Vladimir Lifschitz, and Bruce Porter. 2008. *Handbook of Knowledge Representation*. Vol. 1. Elsevier.
http://books.google.com/books?hl=en&lr=&id=xwBDyIHhJhYC&oi=fnd&pg=PP2&dq=Handbook+of+Knowledge+Representation&ots=WRMJpiwwTZ&sig=3343orjQ8_c2UvhrqBU6hxjabBo.
- Von Bertalanffy, Ludwig. 1956. *General Systems Theory*. Vol. 1.
<http://www.theinnerprocess.com/pdf/general-systems-theory.pdf>.
- Wacker, John G., and Mark Treleven. 1986. "Component Part Standardization: An Analysis of Commonality Sources and Indices." *Journal of Operations Management* 6 (2): 219–44.
doi:10.1016/0272-6963(86)90026-4.
- Wang, YAPING, GUIHUA Han, JIANGHUA Ge, JINGRUI Qi, and JIANYUAN Xu. 2011. "Research on Design Method of Demand-Driven Product Configuration for Mass Customization." *Journal of Advanced Manufacturing Systems* 10 (1): 117–25.
- Whitman, Lawrence E., and Hervé Panetto. 2006. "The Missing Link: Culture and Language Barriers to Interoperability." *Annual Reviews in Control* 30 (2): 233–41.
- Williams, Christopher B., Janet K. Allen, David W. Rosen, and Farrokh Mistree. 2007. "Designing Platforms for Customizable Products and Processes in Markets of Non-Uniform Demand." *Concurrent Engineering* 15 (2): 201–16.
- Wong, T. C., Felix T. S. Chan, and L. Y. Chan. 2009. "A Resource-Constrained Assembly Job Shop Scheduling Problem with Lot Streaming Technique." *Computers & Industrial Engineering* 57 (3): 983–95. doi:10.1016/j.cie.2009.04.002.
- Xie, H., P. Henderson, and M. Kernahan. 2005. "Modelling and Solving Engineering Product Configuration Problems by Constraint Satisfaction." *International Journal of Production Research* 43 (20): 4455–69. doi:10.1080/00207540500142381.
- Yang, Dong, and Ming Dong. 2012. "A Constraint Satisfaction Approach to Resolving Product Configuration Conflicts." *Advanced Engineering Informatics* 26 (3): 592–602.
doi:10.1016/j.aei.2012.03.008.
- Yao, Liming, and Subhash C. Sarin. 2014. "Multiple-Lot Lot Streaming in a Two-Stage Assembly System." In *Essays in Production, Project Planning and Scheduling*, edited by P. Simin Pulat, Subhash C. Sarin, and Reha Uzsoy, 357–88. *International Series in Operations Research & Management Science* 200. Springer US. http://link.springer.com/chapter/10.1007/978-1-4614-9056-2_15.
- Ye, Xiaoli, and John K. Gershenson. 2008. "Attribute-Based Clustering Methodology for Product Family Design." *Journal of Engineering Design* 19 (6): 571–86.
doi:10.1080/09544820802471123.
- Ye, Xiaoli, Henri J. Thevenot, Fabrice Alizon, John K. Gershenson, Kiran Khadke, Timothy W. Simpson, and Steven B. Shooter. 2009. "Using Product Family Evaluation Graphs in Product Family Design." *International Journal of Production Research* 47 (13): 3559–85.
doi:10.1080/00207540802183646.
- Zacharias, Nicolas A., and Ali A. Yassine. 2008. "Optimal Platform Investment for Product Family Design." *Journal of Intelligent Manufacturing* 19 (2): 131–48.
- Zdravković, Milan, and Miroslav Trajanović. 2009. "Integrated Product Ontologies for Inter-Organizational Networks." *Computer Science and Information Systems* 6 (2): 29–46.
- Zhou, Chunjing, Zhihang Lin, and Chuntao Liu. 2008. "Customer-Driven Product Configuration Optimization for Assemble-to-Order Manufacturing Enterprises." *The International Journal of Advanced Manufacturing Technology* 38 (1): 185–94. doi:10.1007/s00170-007-1089-6.

-
- Zhu, Bin, Zhan Wang, Haicheng Yang, Rong Mo, and Yanwei Zhao. 2008. "Applying Fuzzy Multiple Attributes Decision Making for Product Configuration." *Journal of Intelligent Manufacturing* 19 (5): 591–98. doi:10.1007/s10845-008-0132-2.

Annex 1: Lingo syntax

Lingo syntax is not case sensitive and a “;” separates commands.

Arithmetic Operator	Interpretation
^	Exponentiation
*	Multiplication
/	Division
+	Addition
-	Subtraction

Logical Operator	Interpretation
#NOT#	True if operand is false
#NE#	True if operands are not equal
#EQ#	True if operands are equal
#GT#	True if the left operand is strictly greater than the right operand
#GE#	True if the left operand is greater than or equal the right operand
#LT#	True if the left operand is strictly less than the right operand
#LE#	True if the left operand is less than or equal the right operand
#AND#	True if both the arguments are true
#OR#	True if at least one of the argument is true

Relational Operator	Interpretation
=	Equal
<=	Less than or equal
>=	Greater than or equal

Mathematical Function	Interpretation
@ABS(X)	Absolute value of X
@ACOS(X)	Inverse cosine of X
@ACOSH(X)	Inverse hyperbolic cosine of X
@ASIN(X)	Inverse sine of X
@ASINH(X)	Inverse hyperbolic sine of X
@ATAN(X)	Inverse tangent of X
@ATANH(X)	Inverse hyperbolic tangent of X
@COS(X)	Cosine of X
@COSH(X)	Hyperbolic cosine of X
@EXP(X)	Euler's number raised to the power X
@FLOOR(X)	Returns the integer part of X. If X is negative, it returns the most negative integer, I, such that I>=X
@INT(X)	Returns the integer part of X. If X is negative, it returns the largest negative integer, I, such that X>=I
@LOG(X)	Natural logarithm
@LOG10(X)	Base-10 logarithm

@PI()	Returns the value of PI
@POW(X,Y)	X raised to the Y power
@SIN(X)	Sine of X
@SINH(X)	Hyperbolic sine of X
@SMAX(X1,X2,...,Xn)	Returns the maximum value of X1,X2,...,Xn
@SMIN(X1,X2,...,Xn)	Returns the minimum value of X1,X2,...,Xn
@TAN(X)	Tangent of X
@TANH(X)	Hyperbolic tangent of X

Variable Domain Function	Interpretation
@BIN(X)	Constrains X values to be binary
@GIN(X)	Constrains X values to be integer
@FREE(X)	Lets X have negative values (by default, variables values are constrained to be positive)
@BND(lower_bound, X, upper_bound)	Define the lower and upper bounds for the X values

Set Definition	Interpretation
setname / member1..memberN / [: attribute_list];	Allows to define a set with N members. Each member can have one or more attributes. Each member has the same attributes.

Set Looping Function	Interpretation
@FOR(setname[(set_index_list)[cond_qualifier]]:exp_list)	This generates the expressions contained in <i>exp_list</i> for all members of the set <i>setname</i> that respect the condition <i>cond_qualifier</i> .
@MAX(setname[(set_index_list)[cond_qualifier]]:expression)	This returns the maximum value of <i>expression</i> taken over the member of <i>setname</i> that respect the condition <i>cond_qualifier</i> .
@MIN(setname[(set_index_list)[cond_qualifier]]:expression)	This returns the minimum value of <i>expression</i> taken over the member of <i>setname</i> that respect the condition <i>cond_qualifier</i> .
@SUM(setname[(set_index_list)[cond_qualifier]]:expression)	This returns the sum of an <i>expression</i> over the member of <i>setname</i> that respect the condition <i>cond_qualifier</i> .
@PROD(setname[(set_index_list)[cond_qualifier]]:expression)	This returns the product of an <i>expression</i> over the member of <i>setname</i> that respect the condition <i>cond_qualifier</i> .

Annex 2: assembly system for the analysis of the variety/cost relation

Variables in chapter 3	Variables in Lingo code
$e_{i_x}^x$	END
$s_{i_x}^x$	START
$q_{i_xj}^x$	QTY_P1_RUN, QTY_P3_RUN,
t_{xj}	TIME_P1, TIME_P3
$z_{i_xj}^x$	Z_P1_RUN, Z_P3_RUN,
$set(x)$	SETUP1, SETUP2
q_{jT}^x	QTY_P1, QTY_P3
q_{jA}	QTY_P1_P3
$u_{i_{p3}i_{p1}j}$	U_P1
$A_{i_{p3}j}^{P3}$	AV_TIME_P3
$Aq_{i_{p1}j}^{P1}$	AV_P1
R_{P1}	RUNS_P1
R_{P3}	RUNS_P3
V	VARIANTS
x	P1, P3

Syntax of the Model of PRV1	Interpretation
MODEL:	Start of the model
DATA: RUNS_P1 = 4; VARIANTS_P1 = 2; RUNS_P3 = 4; VARIANTS_P3 = 2; ENDDATA	Set-relative data values: <ul style="list-style-type: none"> $runs_X$ is the number of permitted sub-lots on X; $variants_X$ is the number of process variants on X.
SETS: RUN_P1 /1..RUNS_P1/ : START_P1, END_P1, Z1_P1_RUN, s1; VARIANT_P1 /1..VARIANTS_P1/ : TIME_P1, QTY_P1; QUANTITY_P1 (VARIANT_P1,	Definition of sets: <ul style="list-style-type: none"> run_X defines the jobs for the sub-lot manufacturing on X; each member of the set has a start time ($start_X$), an end time (end_X), a setup time ($s1$ on P1 and $s3$

```

RUN_P1):QTY_P1_RUN, AV_P1;

RUN_P3 /1..RUNS_P3/ : START_P3,
END_P3,s3;

VARIANT_P3 /1..VARIANTS_P3/ :
TIME_P3, QTY_P3, QTY_P1_P3;

QUANTITY_P3 (VARIANT_P3,
RUN_P3):QTY_P3_RUN, AV_TIME_P3,
Z_P3_RUN;

U_VALUE_P1 (RUN_P3, RUN_P1,
VARIANT_P1): U_P1;

```

ENDSETS

on P3); on P1, a binary variable (z_{P1_run}) that is equal to 1 when a setup is required;

- *variant_X* defines the attributes relative to the variants on X; each member of the set has a unitary processing time ($time_X$) and a demanded quantity (qty_X); on P3, qty_{p1_p3} defines the number of component manufactured on P1 are required to start a job on P3;
- *quantity_P1* defines the jobs for the sublot manufacturing per variant on P1; each member of the set has a quantity (qty_{P1_run}) and the availability of each variant at the end of each job (av_{p1});
- *quantity_P3* defines the jobs for the sublot manufacturing per variant on P3; each member of the set has a quantity (qty_{P3_run}), the available time of each variant at each job starting time and a binary variable (z_{P3_run}) that is equal to 1 when a setup is required;
- *u_value_p1* defines the availability between P1 and P3; each member of the set has a binary property ($u_{p1}(i,j,k)$) that is equal to 1 when during the job i on P3, the variant k is used and k has been manufactured during the job j on P1.

```

TIME_P1(1) = 2;
TIME_P1(2) = 4;
QTY_P1(1) = 200;
QTY_P1(2) = 150;
SETUP1 = 10;
START_P1(1) = 0;
TIME_P3(1) = 40;
TIME_P3(2) = 100;
QTY_P3(1) = 2;
QTY_P3(2) = 2;
QTY_P1_P3(1) = 100;
QTY_P1_P3(2) = 75;
SETUP3 = 3;

```

Input data.

```

@FOR(RUN_P1(I) | I #EQ#1:
    END_P1(I) >= START_P1(I);
    END_P1(I) >= 1;
);
@FOR(RUN_P1(I) | I #GT# 1:

```

Starting and completion time constraints.

```

        END_P1(I-1) = START_P1(I);
        END_P1(I) >= START_P1(I);
    );
    @FOR(RUN_P3(I) | I #EQ# 1:
        END_P3(I) >= START_P3(I);
        END_P3(I) >= 1;
    );
    @FOR(RUN_P3(I) | I #GT# 1:
        END_P3(I-1) <= START_P3(I);
        END_P3(I) >= START_P3(I);
    );

```

@FOR(RUN_P1(I) | I #EQ# RUNS_P1: Constraints about the durations of the jobs.

```

END_P1(I) - START_P1(I) =
@SUM(QUANTITY_P1(J,I):
QTY_P1_RUN(J,I) * TIME_P1(J));
);

```

```

@FOR(RUN_P1(I) | I #LT# RUNS_P1:

END_P1(I) - START_P1(I) =
@SUM(QUANTITY_P1(J,I):
QTY_P1_RUN(J,I) * TIME_P1(J)) + SETUP1
* @IF( (Z1_P1_RUN(I+1) ) #NE#
(Z1_P1_RUN(I) ) , 1,0);

s1 = SETUP1 * @IF( (Z1_P1_RUN(I+1) )
#NE# (Z1_P1_RUN(I) ) , 1,0);

);

```

```

@FOR(RUN_P3(I) | I #EQ# RUNS_P3:

END_P3(I) - START_P3(I) =
@SUM(QUANTITY_P3(J,I):
QTY_P3_RUN(J,I) * TIME_P3(J));

);

```

```

@FOR(RUN_P3(I) | I #LT# RUNS_P3:

END_P3(I) - START_P3(I) =
@SUM(QUANTITY_P3(J,I):
QTY_P3_RUN(J,I) * TIME_P3(J)) + SETUP3
* @IF(Z_P3_RUN(1,I+1) #ne#
Z_P3_RUN(1,I) #or# Z_P3_RUN(2,I+1)
#ne# Z_P3_RUN(2,I) , 1,0);

s3 = SETUP3 * @IF(Z_P3_RUN(1,I+1)
#ne# Z_P3_RUN(1,I) #or#
Z_P3_RUN(2,I+1) #ne# Z_P3_RUN(2,I) ,
1,0);

);

```

@FOR(VARIANT_P1(J) : For each variant, the produced quantity has to be equal to the demanded quantity.
 @SUM(QUANTITY_P1(J,I):
 QTY_P1_RUN(J,I)) = QTY_P1(J);

<pre>); @FOR(VARIANT_P3(J) : @SUM(QUANTITY_P3(J,I) : QTY_P3_RUN(J,I) = QTY_P3(J) ;);</pre>	
<pre>@FOR(QUANTITY_P1(J,I) : @GIN(QTY_P1_RUN(J,I)) ;);</pre>	Quantities produced on P1 are constrained to be integer.
<pre>@FOR(RUN_P1(I) : @BIN(Z1_P1_RUN(I)); -10 * QTY_P1(2) * (Z1_P1_RUN(I)) + QTY_P1_RUN(2,I) <= 0; +10 * QTY_P1(1) * (Z1_P1_RUN(I)) - 10 * (QTY_P1(1) - QTY_P1_RUN(1,I)) <= 0;);</pre>	<p>The variable $z_{p1_run(i)}$ has to be:</p> <ul style="list-style-type: none"> • equal to 0, when during the job i, a sub-lot of the variant 1 is manufactured; • equal to 1, when during the job i, a sub-lot of the variant 2 is manufactured.
<pre>@FOR(QUANTITY_P3(J,I) : @GIN(QTY_P3_RUN(J,I)); @BIN(Z_P3_RUN); -1000 * Z_P3_RUN + QTY_P3_RUN <= 0; -Z_P3_RUN + QTY_P3_RUN >= 0;);</pre>	<p>The variable $z_{p3_run(i,j)}$ has to be:</p> <ul style="list-style-type: none"> • equal to 0, when during the job i, a sub-lot of the variant j is not manufactured; • equal to 1, when during the job i, a sub-lot of the variant j is manufactured.
<pre>@FOR(RUN_P3(I) : @SUM(QUANTITY_P3(J,I) : QTY_P3_RUN(J,I)) <= 1;);</pre>	The sub-lot size on the assembly workstation P3 is constrained to be less than or equal to 1.
<pre>START_P3(1) = @SUM(QUANTITY_P3(J,I) I #EQ# 1 : Z_P3_RUN * AV_TIME_P3); START_P3(1) = @MIN(QUANTITY_P3(J,I) I #EQ# 1 : AV_TIME_P3) ; @FOR(RUN_P3(I) I #GT# 1: START_P3(I) >= @SUM(QUANTITY_P3(J,I) : Z_P3_RUN * AV_TIME_P3););</pre>	These constraints force a job on P3 to start only if the required quantity of the components has been manufactured on P1.
<pre>@FOR(QUANTITY_P3(I,J) J#GT#1: av_time_p3 = @MIN(U_VALUE_P1(J,K,I) : 10000 * (U_P1) + (END_P1(K))) ;); @FOR(QUANTITY_P3(I,J) J#EQ#1: av_time_p3 = @MIN(U_VALUE_P1(J,K,I) : 10000 * (U_P1) + (END_P1(K))) ;</pre>	These constraints define the availability time of a variant j to be manufactured on P3 during the job i .

);	
@FOR(U_VALUE_P1(I,J,K) : @BIN(U_P1) ; -10000 * (1 - U_P1) + (AV_P1(K,J) - @SUM(QUANTITY_P3(K,N) N #LT# I : QTY_P3_RUN(K,N))) <= 0 ; -10000 * (U_P1) - (- 0.9 + (AV_P1(K,J) - @SUM(QUANTITY_P3(K,N) N #LT# I: QTY_P3_RUN(K,N)))) <= 0 ;);	These constraints define the value of <i>u_p1</i> .
@FOR(QUANTITY_P1(J,I) : @GIN(AV_P1) ; AV_P1 = (@SUM(QUANTITY_P1(J,K) K #LE# I:QTY_P1_RUN(J,K))/QTY_P1_P3(J)) ;);	These constraints define the value of <i>av_p1</i> .
MIN= END_P3(4) ;	The objective function.
END	End of the model.

Syntax of the Model of PRV2	Interpretation
MODEL:	As for PRV1.
DATA: RUNS_P1 = 4 ; VARIANTS_P1 = 3 ; RUNS_P3 = 4 ; VARIANTS_P3 = 3 ; ENDDATA	As for PRV1.
SETS: RUN_P1 /1..RUNS_P1/ : START_P1, END_P1, Z1_P1_RUN, Z2_P1_RUN, Z3_P1_RUN, s1 ; VARIANT_P1 /1..VARIANTS_P1/ : TIME_P1, QTY_P1 ; QUANTITY_P1 (VARIANT_P1, RUN_P1) :QTY_P1_RUN, AV_P1 ; RUN_P3 /1..RUNS_P3/ : START_P3, END_P3, s3 ; VARIANT_P3 /1..VARIANTS_P3/ : TIME_P3, QTY_P3, QTY_P1_P3 ; QUANTITY_P3 (VARIANT_P3, RUN_P3) :QTY_P3_RUN, AV_TIME_P3, Z_P3_RUN ;	As for PRV1. For the member of the set <i>run_p1</i> more attributes are defined. Indeed, the binary values of the variables <i>zX_p1_run</i> , where X is the process variant, are used to determine the presence of a setup between two jobs on P1.

```
U_VALUE_P1 (RUN_P3, RUN_P1,
VARIANT_P1): U_P1;
```

ENDSETS

```
TIME_P1(1) = 2;                               As for PRV1.
TIME_P1(2) = 2;
TIME_P1(3) = 1;
QTY_P1(1) = 100;
QTY_P1(2) = 100;
QTY_P1(3) = 150;
SETUP1 = 10;
START_P1(1) = 0;
TIME_P3(1) = 230;
TIME_P3(2) = 200;
TIME_P3(3) = 150;
QTY_P3(1) = 1;
QTY_P3(2) = 1;
QTY_P3(3) = 2;
QTY_P1_P3(1) = 100;
QTY_P1_P3(2) = 100;
QTY_P1_P3(3) = 75;
SETUP3 = 3;
```

```
@FOR(RUN_P1(I) | I #EQ# 1:                   As for PRV1.
    END_P1(I) >= START_P1(I);
    END_P1(I) >= 1;
);
@FOR(RUN_P1(I) | I #GT# 1:
    END_P1(I-1) = START_P1(I);
    END_P1(I) >= START_P1(I);
);

@FOR(RUN_P3(I) | I #EQ# 1:
    END_P3(I) >= START_P3(I);
    END_P3(I) >= 1;
);
@FOR(RUN_P3(I) | I #GT# 1:
    END_P3(I-1) <= START_P3(I);
    END_P3(I) >= START_P3(I);
);
```

```
@FOR(RUN_P1(I) | I #EQ# RUNS_P1:             As for PRV1.

END_P1(I) - START_P1(I) =
@SUM(QUANTITY_P1(J,I):
QTY_P1_RUN(J,I) * TIME_P1(J));
);

@FOR(RUN_P1(I) | I #LT# RUNS_P1:

END_P1(I) - START_P1(I) =
@SUM(QUANTITY_P1(J,I):
QTY_P1_RUN(J,I) * TIME_P1(J)) +
SETUP1 * @IF( Z1_P1_RUN(I+1) #NE#
Z1_P1_RUN(I) #OR# Z2_P1_RUN(I+1)
#NE# Z2_P1_RUN(I) #OR# Z3_P1_RUN(I)
#NE# Z3_P1_RUN(I+1) , 1,0);

s1 = SETUP1 * @IF( Z1_P1_RUN(I+1)
```

```

#NE# Z1_P1_RUN(I) #OR#
Z2_P1_RUN(I+1) #NE# Z2_P1_RUN(I)
#OR# Z3_P1_RUN(I) #NE#
Z3_P1_RUN(I+1) , 1,0);

);

@FOR(RUN_P3(I) | I #EQ# RUNS_P3:

END_P3(I) - START_P3(I) =
@SUM(QUANTITY_P3(J,I):
QTY_P3_RUN(J,I) * TIME_P3(J));
);

@FOR(RUN_P3(I) | I #LT# RUNS_P3:

END_P3(I) - START_P3(I) =
@SUM(QUANTITY_P3(J,I):
QTY_P3_RUN(J,I) * TIME_P3(J) +
SETUP3 * @IF(Z_P3_RUN(1,I+1) #EQ#
Z_P3_RUN(1,I) #AND# Z_P3_RUN(2,I+1)
#EQ# Z_P3_RUN(2,I) , 0,1);

s3 = SETUP3 * @IF(Z_P3_RUN(1,I+1)
#EQ# Z_P3_RUN(1,I) #AND#
Z_P3_RUN(2,I+1) #EQ# Z_P3_RUN(2,I) ,
0,1);

);

```

```

@FOR(VARIANT_P1(J):
As for PRV1.

@SUM(QUANTITY_P1(J,I):
QTY_P1_RUN(J,I) = QTY_P1(J);

);

```

```

@FOR(VARIANT_P3(J):

@SUM(QUANTITY_P3(J,I):
QTY_P3_RUN(J,I) = QTY_P3(J);

);

```

```

@FOR(QUANTITY_P1(J,I):
As for PRV1.
@GIN(QTY_P1_RUN(J,I));
);

```

```

@FOR(RUN_P1(I):
The values of the 3 binary variables z1_p1_run,
z2_p1_run and z3_p1_run are computed as
z_p1_run in PRV1.

@BIN( Z1_P1_RUN(I) );

-10 * QTY_P1(2) * (Z1_P1_RUN(I) ) +
QTY_P1_RUN(2,I) <= 0;

+10 * QTY_P1(1) * (Z1_P1_RUN(I) ) -
10 * (QTY_P1(1) - QTY_P1_RUN(1,I)) <=
0;

```

```

@BIN( Z2_P1_RUN(I) );

-10 * QTY_P1(2) * (Z2_P1_RUN(I) ) +
QTY_P1_RUN(2,I) <= 0;

+10 * QTY_P1(3) * (Z2_P1_RUN(I) ) -
10 * (QTY_P1(3) - QTY_P1_RUN(3,I)) <=
0;

@BIN( Z3_P1_RUN(I) );

-10 * QTY_P1(3) * (Z3_P1_RUN(I) ) +
QTY_P1_RUN(3,I) <= 0;

+10 * QTY_P1(1) * (Z3_P1_RUN(I) ) -
10 * (QTY_P1(1) - QTY_P1_RUN(1,I)) <=
0;

);

@FOR(QUANTITY_P3(J,I) :                               As for PRV1.

@GIN(QTY_P3_RUN(J,I) );
@BIN( Z_P3_RUN );

-1000 * Z_P3_RUN + QTY_P3_RUN <= 0;

-Z_P3_RUN + QTY_P3_RUN >= 0;

);

@FOR(RUN_P3(I) :                                       As for PRV1.

@SUM(QUANTITY_P3(J,I) :
QTY_P3_RUN(J,I) ) <= 1;

);

START_P3(1) = @SUM(QUANTITY_P3(J,I) | I #EQ# 1 : Z_P3_RUN * AV_TIME_P3
);

START_P3(1) = @MIN(QUANTITY_P3(J,I)
| I #EQ# 1 : AV_TIME_P3 ) ;

@FOR(RUN_P3(I) | I #GT# 1:

START_P3(I) >=
@SUM(QUANTITY_P3(J,I) : Z_P3_RUN *
AV_TIME_P3 );

);

@FOR(QUANTITY_P3(I,J) | J#GT#1:                       As for PRV1.

av_time_p3 = @MIN(U_VALUE_P1(J,K,I) :
10000 * (U_P1 ) + (END_P1(K))) ;

);

@FOR(QUANTITY_P3(I,J) | J#EQ#1:

```

<pre> av_time_p3 = @MIN(U_VALUE_P1(J,K,I): 10000 * (U_P1) + (END_P1(K))) ;); </pre>	
<pre> @FOR(U_VALUE_P1(I,J,K): @BIN(U_P1); -10000 * (1 - U_P1) + (AV_P1(K,J) - @SUM(QUANTITY_P3(K,N) N #LT# I: QTY_P3_RUN(K,N))) <= 0; -10000 * (U_P1) - (- 0.9 + (AV_P1(K,J) - @SUM(QUANTITY_P3(K,N) N #LT# I: QTY_P3_RUN(K,N)))) <= 0;); </pre>	As for PRV1.
<pre> @FOR(QUANTITY_P1(J,I): @GIN(AV_P1); AV_P1 = (@SUM(QUANTITY_P1(J,K) K #LE# I:QTY_P1_RUN(J,K))/QTY_P1_P3(J));); </pre>	As for PRV1.
<pre> MIN= END_P3(4) ; </pre>	As for PRV1.
<pre> END </pre>	As for PRV1.

Annex 3: water coil variety generation model

Water coil model for a <i>customer-process</i> couple	Interpretation
<pre> @BND (13.11,X_1_1_1,13.29) ; @BND (5.91,X_2_1_1,6.09) ; @BND (25.61,X_3_1_1,25.79) ; @BND (0.491,X_4_1_1,0.509) ; @BND (6.8100000000000005,X_5_1_1,6.99) ; @BND (4.0,Y_1_1_1,4.0) ;Y_1_1_1 =4; @BND (4.0,Y_2_1_1,4.0) ;Y_2_1_1 =4; @BND (0.801,Y_3_1_1,0.8190000000000001) ; @BND (10.051,Y_4_1_1,10.069) ; @BND (0.141,Y_5_1_1,0.159) ; @BND (399.1,Y_6_1_1,400.9) ; @BND (329.1,Y_7_1_1,330.9) ; @BND (2.1910000000000003,Y_8_1_1,2.209) ; @BND (25.31,Y_9_1_1,25.49) ; @BND (21.91,Y_10_1_1,22.09) ; @BND (3.551,Y_11_1_1,3.569) ; </pre>	<p>Variables from the simulation of PV design performed in section 3.1.3 - .</p>
<pre> @BND (0, D_I ,10) ; @BND (0, R_B ,50) ; @BND (0.3, V_W ,2.4) ; @BND (0.01, W_R ,5) ; @BND (0.01, W_A ,5) ; @BND (0, A_0 ,1000) ; @BND (0, A_P ,1000) ; @BND (0, A_S ,1000) ; @BND (0, A_I ,1000) ; @BND (0, A_IX ,0.001) ; @BND (0, T_A2 ,50) ; @BND (250, T_2 ,350) ; @BND (0, T_WB_A2 ,50) ; @BND (0, T_A2_DEW ,50) ; @BND (0, T_WB_A1 ,50) ; @BND (0, T_A1_DEW ,50) ; @BND (0, H_DEW_A1 ,100) ; @BND (250, T_1 ,350) ; @BND (250, TW1 ,350) ; @BND (250, T_AS2 ,350) ; @BND (0, T_AB ,50) ; @BND (0, T_S2 ,50) ; @BND (0, T_SB ,50) ; @BND (0, T_RB ,50) ; @BND (0, T_S_MEAN ,50) ; @BND (0, T_W_MEAN ,50) ; @BND (0, H_A1 ,100) ; @BND (0, H_A2_DEW ,100) ; @BND (0.250, T_2M ,0.350) ; @BND (0.250, T_1M ,0.350) ; @BND (0.250, T_AS2M ,0.350) ; @BND (0.250, TW1M ,0.350) ; @BND (0, H_A2 ,100) ; @BND (0, H_AB ,100) ; @BND (0, H_S2 ,100) ; @BND (0, H_SB ,100) ; @BND (0, YM ,1) ; @BND (0, C ,1) ; </pre>	<p>Variables in standards.</p>

```

@BND(0, Q_T ,100);
@BND(0, LOAD_T ,100);
@BND(0, Q_TW ,100);
@BND(0, Q_TD ,100);
@BND(0, Q_E ,50);
@BND(0, Q_W ,50);
@BND(1, P_W1 ,20);
@BND(1, P_W2 ,20);
@BND(1, P_WS_WET1 ,20);
@BND(1, P_WS_WET2 ,20);
@BND(1, P_WS1 ,20);
@BND(1, P_WS_S2 ,20);
@BND(0, ALFA1 ,3);
@BND(0, ALFA2 ,3);
@BND(0.001, W_1 ,0.090);
@BND(0.001, W_2 ,0.090);
@BND(0.001, W_S_WET1 ,0.090);
@BND(0.001, W_S_WET2 ,0.090);
@BND(0.001, W_S1 ,0.090);
@BND(0.001, W_S2 ,0.090);
@BND(0, H_W3 ,200);
@BND(0, H_W4 ,200);
@BND(2500, H_G4 ,2600);
@BND(2300, H_FG4 ,2600);
@BND(100, H_G4 ,5000);
@BND(100, H_FG4 ,5000);
@BND(0, H_FW ,10);
@BND(0, D_T_M ,50);
@BND(0, D_T_MS ,50);
@BND(0, D_H_M ,100);
@BND(0, A_D ,1000);
@BND(0, A_W ,1000);
@BND(0, CM ,10);
@BND(0.01, M_SLOPE ,100);
@BND(0.001, R_AD ,5);
@BND(0.001, R_AW ,5);
@BND(0.001, R_MD ,5);
@BND(0.001, R_MW ,5);
@BND(0.001, R_T ,5);
@BND(0.001, R_F ,5);
@BND(0.001, R_0 ,5);
@BND(0.001, R_R ,5);
@BND(25, H_F ,250);
@BND(0.001, Z_ASHRAE ,100);
@BND(0.001, MM ,100);
@BND(0.001, FI ,1);
@BND(0.001, ETA ,1);
@BND(0.001, R_E ,5000);
@BND(1, BETA ,100);
@BND(1, MU_TW ,2);
@BND(1, MU_W ,2);
@BND(100, G_W ,2000);
@BND(1, RE ,200000);
@BND(7, PR_W ,14.5);
@BND(-6, A_1 ,24);
@BND(-70, A_2 ,7);
@BND(-0.2, A_3 ,4);
@BND(-12, A_4 ,0);
@BND(0, A_5 ,2.5);
@BND(-0.2, A_6 ,0);
@BND(0, ST_W ,10);

```

```

@BND(0.001, J_W ,0.1);
@BND(1, F_W ,10000);
C_P = 1;
C_R = 4.185;
P = 101.325;
c8M = 5.8002206;
c9M = 1.391499;
c10M = 48.640239;
c11M = 41.764768;
c12M = 14.552093;
c13M = 6.5459673;
C14 = 6.54;
C15 = 14.526;
C16 = 0.7389;
C17 = 0.09486;
C18 = 0.4569;
K_W = 0.58;
K_A = 221.7 ;
K_T = 338.7 ;

```

```

N_T = PUNCH_HOLES * Y_1_1_1;
N_H = N_T;
N_P = N_T/N_C;
@GIN(N_P);
@GIN(N_F);
N_F1 = ((Y_6_1_1 - 20) / (Y_8_1_1 +
Y_5_1_1));
N_F = (@FLOOR(N_F1));
Y_9_1_1 * (PUNCH_HOLES + 1) = Y_7_1_1;
Y_10_1_1 * Y_1_1_1 = L_D;
L_S = Y_6_1_1;
0 = (- R_MW - R_MD + R_T + R_F);
0 = (- Y_4_1_1 + D_I + 2 * Y_3_1_1);
0 = (- Y_4_1_1 / 2 + R_B);
0 = (- R_T + ((B * D_I) / (2 *
338.7)) * @LOG(Y_4_1_1 / D_I));
0 = (reta - (1 + ETA) / ETA);
0 = (- R_F + reta * R_AD);
0 = (- R_F + reta * (R_AW * (1/
M_SLOPE)));

```

These are the constraints that characterise the water coil behaviour. These formulas are from the following standards.

- The chapter 1 in (ASHRAE 2009) about *psychrometrics* has been used to model the air temperature and humidity relations.
- The chapter 4 about *heat transfer* in (ASHRAE 2009) has been used to model the relations between heat transfer and fins geometry.
- The chapter 22 about *air-cooling and dehumidifying coils* in (ASHRAE 2012) has been used to model the *water coil* system behaviour.
- The AHRI standard on *Forced-Circulation Air-Cooling and Air-Heating Coils* (Air-conditioning, Heating & Refrigeration Institute 2001) has been used to model the relations between the coil geometry and the heat transfer.

```

0 = (- ETA + (FI * A_S + A_P) / A_0);

0 = (- MM + ((2 * H_F) / (221.7 *
Y_5_1_1))^0.5);

0 = (- Z_ASHRAE + (( rapp ) - 1) * (1
+ 0.35 * @LOG( rapp )));

0 = ( rapp - R_E/R_B);

0 = (-HYP + (MM * R_B * Z_ASHRAE));

0 = (-HYP1 + @COSH(HYP));

0 = (-HYP2 + @SINH(HYP));

0 = (-HYP3 * HYP1 + HYP2);

0 = (- FI + HYP3 / HYP );

0 = (- (R_E/R_B) + (1.28 * YPS *
(BETA - 0.2)^0.5) );

0 = (-YPS + Y_10_1_1 / R_B);

0 = (- BETA + Y_9_1_1 / Y_10_1_1);

0 <= ( Y_9_1_1 - Y_10_1_1);

0 = (- H_F + (10.45 - Y_11_1_1 + 10 *
Y_11_1_1^0.5));

0 = (-nf_yf + N_F * Y_5_1_1);

0 = (-N_P * Y_2_1_1 + N_T);

0 <= (- (N_F * Y_5_1_1) + Y_6_1_1);

0 = (- @log( A_P) - @log(318344) +
@log(N_T * Y_4_1_1 * Y_6_1_1 - N_T *
nf_yf * ( Y_4_1_1 - 2 * Y_8_1_1 )));

0 = (-@log(A_S) + @log(1*10^(-5)) +
@log(N_F * ( ( Y_7_1_1 * L_D / 5 ) -
(N_H * ( Y_4_1_1 + 2 * Y_5_1_1 )^2) /
6.36688 + ( ( Y_4_1_1 + 2 * Y_5_1_1 ) *
( N_H - N_T) * Y_8_1_1 ) / 3.18344 )));

0 = (-A_0 + A_S + A_P);

0 = (- @log(A_I) -@log( 318344) +
@log(D_I * Y_6_1_1 * N_T));

0 = (-A_I * B + A_0);

0 = (-@log(A_IX) + @log(7.85*10^-7) +
@log( D_I ^ 2 * Y_2_1_1));

0 = (-N_T + N_P * Y_2_1_1);

```

```

0 = (-@log(V_W * A_IX) - @log( 1000 *
0.998) + @log(W_R));

0 = ( @log( MU_W /1000) - @log(2.414 *
10^(-5)) - @log(10^(247.8/(T_W_MEAN +
273.25 - 140)))));

0 = (T_W_MEAN - (X_1_1_1 + X_2_1_1) /
2);

0 = ( @log( MU_TW /1000) - @log( 2.414
* 10^(-5)) - @log( 10^(247.8/(T_S_MEAN
+ 273.15 - 140)))));

0 = (G_W - 998.927 * V_W);

0 = (MU_W * RE - (D_I * G_W));

0 = ( PR_W * 0.58 - 4.185 * (MU_W)) ;

0 = ((4.185 * G_W * 1000) * ST_W -
F_W);

0 = ( -J_W + (ST_W * (PR_W^(2/3))) *
(MU_TW / MU_W)^0.14);

0 = ( -@LOG(J_W) + A_1 + A_2 * logre
+ A_3 * logre ^2 + (A_4 + A_5 * logre
+ A_6* logre ^2) * @LOG(L_S / D_I));

0 = (-logre + @log(re));

0 = ( - A_1 + RE_L * 0.620576 + RE_TR
* 23.9147 + RE_TU * -5.2036);

0 = (- A_2 + RE_L * (0.66666 ) + RE_TR
* (-6.2352) + RE_TU * (0.073562));

0 = (-A_3 + RE_L * 0.0 + RE_TR *
0.329875+ RE_TU * -0.01184);

0 = (-A_4 + RE_L * -0.33333 + RE_TR *
-11.359 + RE_TU * 0.0);

0 = (-A_5 +RE_L * 0.0 + RE_TR *
2.46100 + RE_TU * 0.0);

0 = ( -A_6 + RE_L * 0.0 + RE_TR * -
0.13330+ RE_TU * 0.0);

RE_L + RE_TR + RE_TU = 1;

RE_L = @IF(RE #LT# 2100, 1, 0);

RE_TR = @IF(RE #GE# 2100 #AND# RE #LT#
10000, 1, 0);

RE_TU = @IF(RE #GE# 10000, 1, 0);

```

```

@BIN(RE_L);

@BIN(RE_TR);

@BIN(RE_TU);

0 = (- M_SLOPE + (H_SB - H_S2) / (T_SB
- T_S2));

( X_3_1_1 - T_AB )>= 0;

( T_AB - T_A2 )>= 0;

( H_A1 - H_AB )>= 0;

( H_AB - H_A2 )>= 0;

( T_SB - T_S2 )>= 0;

( H_SB - H_S2 )>= 0;

( X_1_1_1 - T_RB )>= 0;

( T_RB - X_2_1_1 )>= 0;

( T_A2 - T_S2 )>= 0;

( T_S2 - X_2_1_1 )>= 0;

( T_AB - T_SB )>= 0;

( T_SB - T_RB )>= 0;

( X_3_1_1 - X_1_1_1 )>= 0;

( H_AB - H_SB )>= 0;

( H_A2 - H_S2 )>= 0;

0 = ( X_3_1_1_DEW - T_SB);

0 = ( H_DEW_A1 - H_SB);

0 = ( -T_2 + T_WB_A2 + 273.15);

0 = (@LOG(P_WS_WET2 * 1000) - (-
5.8002206/ (T_2M) + 1.391499 -
48.640239 * (T_2M) + 41.764768*((T_2M)
^2) - 14.552093*((T_2M) ^3) +
6.5459673*@log((T_2M * 1000))));

0 = (T_2M * 1000 - T_2);

0 = ((101.325- P_WS_wet2) * w_s_wet2 -
(0.621945 * P_WS_wet2)) ;

0 = (-(2501 + 1.86*T_A2 -
4.186*T_WB_A2) * W_2 + ((2501 -
2.326*T_WB_A2) * w_s_wet2 - 1.006 *

```

```

(T_A2 - T_WB_A2))) ;

0 = (-H_A2 + 1.006 * T_A2 + W_2 * (2501
+ 1.86 * T_A2));

0 = ((0.621945 + W_2) * p_w2 -
(101.325* W_2)) ;

0 = (-T_A2_dew + 6.54 + 14.526*alfa2 +
0.7389*alfa2^2 + 0.09486*alfa2^3 +
0.4569 * (p_w2)^0.1984);

0 = ((alfa2) - @log( p_w2 ));

0 = (-H_A2_dew + 1.006 * T_A2_dew + W_2
* (2501 + 1.86 * T_A2_dew));

0 = (-T_1 + X_3_1_1 + 273.15);

0 = (@LOG(P_WS1 * 1000) - (-5.8002206/
(T_1M) + 1.391499 - 48.640239 * (T_1M)
+ 41.764768*((T_1M) ^2) -
14.552093*((T_1M) ^3) +
6.5459673*@log((T_1M * 1000))));

0 = (T_1M * 1000 - T_1);

0 = (P_W1 - X_4_1_1 * P_WS1);

0 = (w_1-(0.621945 * p_w1) / (101.325-
p_w1));

0 = (-H_A1 + 1.006 * X_3_1_1 + W_1 *
(2501 + 1.86 * X_3_1_1));

0 = (w_s1 - (0.621945 * P_WS1) /
(101.325- P_WS1));

0 = (p_w1 - (101.325* w_1) / (0.621945 +
w_1));

0 = (alfa1 - @log(p_w1));

0 = (-X_3_1_1_dew + 6.54 + 14.526*alfa1
+ 0.7389*alfa1^2 + 0.09486*alfa1^3 +
0.4569 * (p_w1)^0.1984);

0 = (-H_dew_A1 + 1.006 * X_3_1_1_dew +
W_1 * (2501 + 1.86 * X_3_1_1_dew));

0 = (-(2501 + 1.86*X_3_1_1 -
4.186*T_WB_A1) * w_1 + ((2501 -
2.326*T_WB_A1) * w_s_wet1 - 1.006 *
(X_3_1_1 - T_WB_A1))) ;

0 = (-TW1 + T_WB_A1 + 273.15);

0 = (@LOG(P_WS_WET1 * 1000) - (-
5.8002206/ (TW1M) + 1.391499 -
48.640239 * (TW1M) + 41.764768*((TW1M)

```

```

^2) - 14.552093*((TW1M) ^3) +
6.5459673*@log((TW1M * 1000)));

0 = (TW1M * 1000 - TW1);

0 = (w_s_wet1 - (0.621945 * P_WS_wet1)
/ (101.325- P_WS_wet1));

0 = ( T_as2 - T_s2 - 273.15);

0 = (@LOG(P_WS_S2 * 1000) - (-
5.8002206/ (T_AS2M) + 1.391499 -
48.640239 * (T_AS2M) +
41.764768*((T_AS2M) ^2) -
14.552093*((T_AS2M) ^3) +
6.5459673*@log((T_AS2M* 1000))));

0 = (T_AS2M * 1000 - T_AS2);

0 = (-w_s2 +(0.621945 * p_WS_s2) /
(101.325- p_WS_s2));

0 = (-H_s2 + 1.006 * T_s2 + W_s2 *
(2501 +1.86 * T_s2));

0 = ( -YM * ( H_A1 - H_A2 - H_FW)+
X_1_1_1 - X_2_1_1);

0 = ( X_3_1_1 - T_AB ) * 1- ( H_A1 -
H_AB ) ;

0 = (R_R * F_W - B);

0 = (LOAD_T * W_A - Q_T);

0 = (LOAD_T - (Q_E + Q_W + X_5_1_1));

0 = (-LOAD_T + (H_A1 - H_A2) - ( W_1
- W_2) * H_W3);

0 = (H_W3 - X_1_1_1 * 4.186);

0 = (-H_W4 + X_3_1_1_DEW * 4.186);

0 = (-H_G4 + 2501 + (1.805 *
X_3_1_1_DEW));

0 = (-H_FG4 + H_G4 - H_W4);

0 = (Q_E - H_FG4 * (W_1 - W_2));

0 = (-Q_W - X_5_1_1 + (H_A1 - H_A2) -
( W_1 - W_2) * (H_FG4 + H_W3));

0 = (Q_W - ( W_1 - W_2) * (H_W4 -
H_W3));

0 = (-X_5_1_1 + (H_A1 - H_A2) - ( W_1
- W_2) * H_G4);

```

```

0 = (Q_T - W_A * ( H_A1 - H_A2 - H_FW
));

0 = (Q_T - W_R * 4.185 * ( X_1_1_1 -
X_2_1_1 ));

0 = (H_FW - ( W_1 - W_2 ) * 4.186 *
(X_1_1_1));

0 = (Q_TD - W_A * 1* ( X_3_1_1 - T_AB
));

0 = (Q_TD - W_R * 4.185 * (X_1_1_1 -
T_RB ));

0 = (-Q_T + Q_TD + Q_TW);

0 = (Q_TW - W_R * 4.185 * ( T_RB -
X_2_1_1 ));

0 = (-R_0 + R_R + R_AD + R_MD);

0 = (C * 1* R_AW - (R_MW + R_R));

0 = (-C * ( H_AB - H_SB ) + T_SB -
T_RB);

0 = (-C * ( H_A2 - H_S2 ) + T_S2 -
X_2_1_1);

0 = (YM * W_R * 4.185 - W_A);

0 = ( -H_AB * ( C + YM ) + X_3_1_1_DEW
- X_1_1_1 + YM * H_A1 + C * H_DEW_A1);

0 = ( -T_RB + X_1_1_1 - YM * 1* (
X_3_1_1 - T_AB ));

0 = (-D_T_M * @LOG( ( X_3_1_1 -
X_1_1_1 ) / ( T_AB - T_RB ) ) + (
X_3_1_1 - X_1_1_1 ) - ( T_AB - T_RB ));

0 = (A_D * D_T_M - 1000 * Q_TD * R_0);

0 = (-D_H_M * @LOG(( H_AB - H_SB ) / (
H_A2 - H_S2 )) + ( H_AB - H_SB ) - (
H_A2 - H_S2 )) ;

0 = (-D_T_MS * @LOG(( T_SB - T_RB ) /
( T_S2 - X_2_1_1 )) + ( T_SB - T_RB ) -
( T_S2 - X_2_1_1 )) ;

0 = (A_W * D_H_M - Q_TW * 1000 * R_AW
* 1);

0 = (A_W * D_T_MS - Q_TW * 1000* (
R_MW + R_R ));

0 = (A_0 - A_W - A_D);

```

```
0 = (CM * 1000 * W_A * 1* R_AD - A_0);  
  
0 = (T_A2 - T_S_MEAN + @EXP(-CM) * (  
X_3_1_1 - T_S_MEAN));  
  
0 = (Q_A - Y_11_1_1 * A_A);  
  
0 = 1000000 * A_A - Y_6_1_1 * Y_7_1_1;  
  
0 = W_A - 1.2 * Q_A;  
  
@BIN(RE_L);  
  
@BIN(RE_TR);  
  
@BIN(RE_TU);  
  
@GIN(PUNCH_HOLES);
```

Annex 4: water coil PV optimisation model

Variables in chapter 4	Variables in Lingo code
C_d	C1
C_s	C2
C_i	C3
C_{ijT}	COST
V_C	NUMBER OF CUSTOMERS
V_P	NUMBER OF PROCESSES
$Set_{mM_{2q}}$	SET_S1_2, SET_S2_2
l_m	LOT_S1, LOT_S2
$Set_{mM_{3q}}$	SET_S1_3, SET_S2_3
$t_{mUM_{3q}}$	T_S1_M3, T_S2_M3
$S_{mM_{1q}M_{3q}}$	S_S1_M1_M3, S_S2_M1_M3
$t_{mUM_{2q}}$	T_S1_M2, T_S2_M2
$S_{mM_{1q}M_{2q}}$	S_S1_M1_M2, S_S2_M1_M2
C_{qD}	C_D_1, C_D_2
$t_{mUM_{1q}}$	T_S1_M1, T_S2_M1
Q_{ij}	Z, K
Q_{mTM_1}	Q_S1_TOT, Q_S2_TOT
Q_{jTM_2}	Q_S1_LOT_T, Q_S2_LOT_T
D	40000
PRODUCTION SYSTEM 1	S1
PRODUCTION SYSTEM 2	S2

Optimisation model	Interpretation
$S_{S1_M1_M2_1} = @FLOOR((400.0 - 40) / (2.2 + 0.15)) ;$ $S_{S1_M1_M3_1} = @FLOOR((330.0/25.4 - 1) * 4.0) ;$	Constraints about the process durations, setup costs and direct costs on the production system 1.

```

T_S1_M1_1_1 = @FLOOR(( 330.0/25.4 - 1 ) * 4.0) /
2 ;

T_S1_M2_1_1 = 4.0 * 0.1 ;

T_S1_M3_1_1 = 400.0 * 0.1 ;

COST_S1_1 = 330.0 * 0.15 * 10 * 4.0 + 400.0 * (
330.0 / 25.4 - 1) ;

S_S1_M1_M2_2 = @FLOOR(( 400.0 - 40) / ( 2.2 +
0.15 )) ;

S_S1_M1_M3_2 = @FLOOR(( 330.0/25.4 - 1 ) * 4.0)
;

T_S1_M1_1_2 = @FLOOR(( 330.0/25.4 - 1 ) * 4.0) /
2 ;

T_S1_M2_1_2 = 4.0 * 0.1 ;

T_S1_M3_1_2 = 400.0 * 0.1 ;

COST_S1_2 = 330.0 * 0.15 * 10 * 4.0 + 400.0 * (
330.0 / 25.4 - 1) ;

S_S1_M1_M2_3 = @FLOOR(( 400.0 - 40) / ( 2.2 +
0.15 )) ;

S_S1_M1_M3_3 = @FLOOR(( 330.0/25.4 - 1 ) * 4.0)
;

T_S1_M1_1_3 = @FLOOR(( 330.0/25.4 - 1 ) * 4.0) /
2 ;

T_S1_M2_1_3 = 4.0 * 0.1 ;

T_S1_M3_1_3 = 400.0 * 0.1 ;

COST_S1_3 = 330.0 * 0.15 * 10 * 4.0 + 400.0 * (
330.0 / 25.4 - 1) ;

S_S1_M1_M2_4 = @FLOOR(( 400.0 - 40) / ( 2.2 +
0.15 )) ;

S_S1_M1_M3_4 = @FLOOR(( 330.0/25.4 - 1 ) * 4.0)
;

T_S1_M1_1_4 = @FLOOR(( 330.0/25.4 - 1 ) * 4.0) /
2 ;

T_S1_M2_1_4 = 4.0 * 0.1 ;

T_S1_M3_1_4 = 400.0 * 0.1 ;

COST_S1_4 = 330.0 * 0.15 * 10 * 4.0 + 400.0 * (
330.0 / 25.4 - 1) ;

S_S1_M1_M2_5 = @FLOOR(( 400.0 - 40) / ( 2.2 +

```

```

0.15 )) ;

S_S1_M1_M3_5 = @FLOOR(( 330.0/25.4 - 1 ) * 4.0)
;

T_S1_M1_1_5 = @FLOOR(( 330.0/25.4 - 1 ) * 4.0) /
2 ;

T_S1_M2_1_5 = 4.0 * 0.1 ;

T_S1_M3_1_5 = 400.0 * 0.1 ;

COST_S1_5 = 330.0 * 0.15 * 10 * 4.0 + 400.0 * (
330.0 / 25.4 - 1) ;

S_S1_M1_M2_6 = @FLOOR(( 400.0 - 40) / ( 2.2 +
0.15 )) ;

S_S1_M1_M3_6 = @FLOOR(( 330.0/25.4 - 1 ) * 4.0)
;

T_S1_M1_1_6 = @FLOOR(( 330.0/25.4 - 1 ) * 4.0) /
2 ;

T_S1_M2_1_6 = 4.0 * 0.1 ;

T_S1_M3_1_6 = 400.0 * 0.1 ;

COST_S1_6 = 330.0 * 0.15 * 10 * 4.0 + 400.0 * (
330.0 / 25.4 - 1) ;

S_S1_M1_M2_7 = @FLOOR(( 400.0 - 40) / ( 2.2 +
0.15 )) ;

S_S1_M1_M3_7 = @FLOOR(( 330.0/25.4 - 1 ) * 4.0)
;

T_S1_M1_1_7 = @FLOOR(( 330.0/25.4 - 1 ) * 4.0) /
2 ;

T_S1_M2_1_7 = 4.0 * 0.1 ;

T_S1_M3_1_7 = 400.0 * 0.1 ;

COST_S1_7 = 330.0 * 0.15 * 10 * 4.0 + 400.0 * (
330.0 / 25.4 - 1) ;

S_S1_M1_M2_8 = @FLOOR(( 400.0 - 40) / ( 2.2 +
0.15 )) ;

S_S1_M1_M3_8 = @FLOOR(( 330.0/25.4 - 1 ) * 4.0)
;

T_S1_M1_1_8 = @FLOOR(( 330.0/25.4 - 1 ) * 4.0) /
2 ;

T_S1_M2_1_8 = 4.0 * 0.1 ;

T_S1_M3_1_8 = 400.0 * 0.1 ;

```

```

COST_S1_8 = 330.0 * 0.15 * 10 * 4.0 + 400.0 * (
330.0 / 25.4 - 1) ;

S_S1_M1_M2_9 = @FLOOR(( 400.0 - 40) / ( 2.2 +
0.15 )) ;

S_S1_M1_M3_9 = @FLOOR(( 330.0/25.4 - 1 ) * 4.0)
;

T_S1_M1_1_9 = @FLOOR(( 330.0/25.4 - 1 ) * 4.0) /
2 ;

T_S1_M2_1_9 = 4.0 * 0.1 ;

T_S1_M3_1_9 = 400.0 * 0.1 ;

COST_S1_9 = 330.0 * 0.15 * 10 * 4.0 + 400.0 * (
330.0 / 25.4 - 1) ;

S_S1_M1_M2_10 = @FLOOR(( 400.0 - 40) / ( 2.2 +
0.15 )) ;

S_S1_M1_M3_10 = @FLOOR(( 330.0/25.4 - 1 ) * 4.0)
;

T_S1_M1_1_10 = @FLOOR(( 330.0/25.4 - 1 ) * 4.0)
/ 2 ;

T_S1_M2_1_10 = 4.0 * 0.1 ;

T_S1_M3_1_10 = 400.0 * 0.1 ;

COST_S1_10 = 330.0 * 0.15 * 10 * 4.0 + 400.0 *
( 330.0 / 25.4 - 1) ;

```

```
C_D_1=70.0;
```

```
SET_S1_2 = 100;
```

```
SET_S1_3 = 100;
```

```

Q_S1_TOT_1=Z_1_1 +Z_2_1 +Z_4_1 +Z_5_1 +Z_6_1
+Z_8_1 +Z_10_1 +Z_12_1 +Z_13_1 +Z_14_1 +Z_15_1
+Z_17_1 +Z_18_1 +Z_19_1 +Z_20_1 +Z_21_1 +Z_22_1
+Z_23_1 +Z_24_1 +Z_27_1 +Z_29_1 +Z_31_1 +Z_32_1
+Z_33_1 +Z_34_1 +Z_39_1 +Z_41_1 +Z_42_1 +Z_43_1
+Z_44_1 +Z_45_1 +Z_46_1 +Z_47_1 +Z_51_1 +Z_52_1
+Z_53_1 +Z_55_1 +Z_56_1 +Z_59_1 +Z_61_1 +Z_62_1
+Z_64_1 +Z_65_1 +Z_66_1 +Z_68_1 +Z_69_1 +Z_71_1
+Z_72_1 +Z_73_1 +Z_74_1 +Z_76_1 +Z_77_1 +Z_78_1
+Z_82_1 +Z_83_1 +Z_84_1 +Z_85_1 +Z_86_1 +Z_87_1
+Z_89_1 +Z_90_1 +Z_91_1 +Z_92_1 +Z_93_1 +Z_95_1
+Z_97_1 +Z_98_1 +Z_100_1 ;

```

```
Q_S1_LOT_T_1<=LOT_S1_1* S_S1_M1_M2_1*40000 ;
```

```
Q_S1_LOT_T_1=Q_S1_TOT_1* S_S1_M1_M2_1 ;
```

Unitary costs and constraints about the lot sizes and numbers for the production system 1.

```

LOT_S1_1= @IF(Q_S1_TOT_1#LT# 1, 0,LOT_S1_1) ;

LOT_S1_1<=Q_S1_TOT_1 ;

@GIN(LOT_S1_1);@GIN(Q_S1_LOT_T_1);@GIN(Q_S1_TOT_
1);

Q_S1_TOT_2=Z_1_2 +Z_2_2 +Z_3_2 +Z_4_2 +Z_7_2
+Z_9_2 +Z_10_2 +Z_11_2 +Z_13_2 +Z_14_2 +Z_15_2
+Z_20_2 +Z_21_2 +Z_22_2 +Z_23_2 +Z_24_2 +Z_26_2
+Z_29_2 +Z_30_2 +Z_31_2 +Z_32_2 +Z_34_2 +Z_35_2
+Z_36_2 +Z_37_2 +Z_38_2 +Z_39_2 +Z_40_2 +Z_41_2
+Z_42_2 +Z_43_2 +Z_46_2 +Z_47_2 +Z_48_2 +Z_49_2
+Z_51_2 +Z_53_2 +Z_54_2 +Z_55_2 +Z_57_2 +Z_58_2
+Z_59_2 +Z_63_2 +Z_64_2 +Z_65_2 +Z_67_2 +Z_68_2
+Z_69_2 +Z_71_2 +Z_72_2 +Z_74_2 +Z_76_2 +Z_77_2
+Z_78_2 +Z_79_2 +Z_82_2 +Z_83_2 +Z_88_2 +Z_90_2
+Z_94_2 +Z_95_2 +Z_98_2 +Z_99_2 +Z_100_2 ;

Q_S1_LOT_T_2<=LOT_S1_2* S_S1_M1_M2_2*40000 ;

Q_S1_LOT_T_2=Q_S1_TOT_2* S_S1_M1_M2_2 ;

LOT_S1_2= @IF(Q_S1_TOT_2#LT# 1, 0,LOT_S1_2) ;

LOT_S1_2<=Q_S1_TOT_2 ;

@GIN(LOT_S1_2);@GIN(Q_S1_LOT_T_2);@GIN(Q_S1_TOT_
2);

Q_S1_TOT_3=Z_2_3 +Z_4_3 +Z_5_3 +Z_7_3 +Z_8_3
+Z_9_3 +Z_16_3 +Z_17_3 +Z_19_3 +Z_20_3 +Z_21_3
+Z_22_3 +Z_23_3 +Z_24_3 +Z_25_3 +Z_26_3 +Z_28_3
+Z_30_3 +Z_32_3 +Z_37_3 +Z_38_3 +Z_40_3 +Z_41_3
+Z_42_3 +Z_43_3 +Z_47_3 +Z_48_3 +Z_49_3 +Z_50_3
+Z_51_3 +Z_53_3 +Z_56_3 +Z_65_3 +Z_66_3 +Z_68_3
+Z_75_3 +Z_77_3 +Z_78_3 +Z_80_3 +Z_82_3 +Z_83_3
+Z_87_3 +Z_91_3 +Z_94_3 +Z_95_3 +Z_96_3 +Z_100_3
;

Q_S1_LOT_T_3<=LOT_S1_3* S_S1_M1_M2_3*40000 ;

Q_S1_LOT_T_3=Q_S1_TOT_3* S_S1_M1_M2_3 ;

LOT_S1_3= @IF(Q_S1_TOT_3#LT# 1, 0,LOT_S1_3) ;

LOT_S1_3<=Q_S1_TOT_3 ;

@GIN(LOT_S1_3);@GIN(Q_S1_LOT_T_3);@GIN(Q_S1_TOT_
3);

Q_S1_TOT_4=Z_1_4 +Z_2_4 +Z_3_4 +Z_4_4 +Z_5_4
+Z_6_4 +Z_8_4 +Z_9_4 +Z_12_4 +Z_13_4 +Z_14_4
+Z_15_4 +Z_18_4 +Z_20_4 +Z_22_4 +Z_23_4 +Z_24_4
+Z_25_4 +Z_26_4 +Z_27_4 +Z_29_4 +Z_30_4 +Z_32_4
+Z_33_4 +Z_34_4 +Z_35_4 +Z_36_4 +Z_37_4 +Z_38_4
+Z_39_4 +Z_40_4 +Z_41_4 +Z_42_4 +Z_44_4 +Z_49_4
+Z_50_4 +Z_51_4 +Z_52_4 +Z_53_4 +Z_55_4 +Z_56_4
+Z_57_4 +Z_58_4 +Z_59_4 +Z_60_4 +Z_61_4 +Z_65_4
+Z_66_4 +Z_68_4 +Z_70_4 +Z_72_4 +Z_73_4 +Z_74_4

```

```

+Z_75_4 +Z_76_4 +Z_83_4 +Z_85_4 +Z_88_4 +Z_90_4
+Z_92_4 +Z_93_4 +Z_94_4 +Z_95_4 +Z_98_4 +Z_99_4
+Z_100_4 ;

Q_S1_LOT_T_4<=LOT_S1_4* S_S1_M1_M2_4*40000 ;

Q_S1_LOT_T_4=Q_S1_TOT_4* S_S1_M1_M2_4 ;

LOT_S1_4= @IF(Q_S1_TOT_4#LT# 1, 0,LOT_S1_4) ;

LOT_S1_4<=Q_S1_TOT_4 ;

@GIN(LOT_S1_4);@GIN(Q_S1_LOT_T_4);@GIN(Q_S1_TOT_
4);

Q_S1_TOT_5=Z_4_5 +Z_5_5 +Z_8_5 +Z_10_5 +Z_13_5
+Z_14_5 +Z_18_5 +Z_19_5 +Z_20_5 +Z_21_5 +Z_22_5
+Z_23_5 +Z_25_5 +Z_27_5 +Z_28_5 +Z_29_5 +Z_30_5
+Z_31_5 +Z_33_5 +Z_38_5 +Z_40_5 +Z_41_5 +Z_44_5
+Z_45_5 +Z_46_5 +Z_47_5 +Z_48_5 +Z_50_5 +Z_51_5
+Z_53_5 +Z_54_5 +Z_56_5 +Z_58_5 +Z_61_5 +Z_64_5
+Z_67_5 +Z_68_5 +Z_69_5 +Z_72_5 +Z_73_5 +Z_76_5
+Z_78_5 +Z_82_5 +Z_83_5 +Z_84_5 +Z_88_5 +Z_89_5
+Z_91_5+Z_92_5 +Z_95_5 +Z_99_5 +Z_100_5 ;

Q_S1_LOT_T_5<=LOT_S1_5* S_S1_M1_M2_5*40000 ;

Q_S1_LOT_T_5=Q_S1_TOT_5* S_S1_M1_M2_5 ;

LOT_S1_5= @IF(Q_S1_TOT_5#LT# 1, 0,LOT_S1_5) ;

LOT_S1_5<=Q_S1_TOT_5 ;

@GIN(LOT_S1_5);@GIN(Q_S1_LOT_T_5);@GIN(Q_S1_TOT_
5);

Q_S1_TOT_6=Z_2_6 +Z_3_6 +Z_5_6 +Z_6_6 +Z_7_6
+Z_10_6 +Z_11_6 +Z_12_6 +Z_13_6 +Z_14_6 +Z_17_6
+Z_18_6 +Z_19_6 +Z_21_6 +Z_22_6 +Z_24_6 +Z_25_6
+Z_26_6 +Z_27_6 +Z_29_6 +Z_30_6 +Z_31_6 +Z_32_6
+Z_33_6 +Z_34_6 +Z_35_6 +Z_36_6 +Z_37_6 +Z_38_6
+Z_39_6 +Z_41_6 +Z_42_6 +Z_45_6 +Z_46_6 +Z_47_6
+Z_48_6 +Z_49_6 +Z_50_6 +Z_52_6 +Z_53_6 +Z_54_6
+Z_56_6 +Z_57_6 +Z_59_6 +Z_60_6 +Z_61_6 +Z_62_6
+Z_64_6 +Z_66_6 +Z_67_6 +Z_68_6 +Z_69_6 +Z_70_6
+Z_72_6 +Z_74_6 +Z_75_6 +Z_76_6 +Z_78_6 +Z_79_6
+Z_80_6 +Z_81_6 +Z_82_6 +Z_83_6 +Z_84_6 +Z_85_6
+Z_87_6 +Z_90_6 +Z_91_6 +Z_92_6 +Z_93_6 +Z_95_6
+Z_96_6 +Z_97_6 +Z_98_6 +Z_99_6 ;

Q_S1_LOT_T_6<=LOT_S1_6* S_S1_M1_M2_6*40000 ;

Q_S1_LOT_T_6=Q_S1_TOT_6* S_S1_M1_M2_6 ;

LOT_S1_6= @IF(Q_S1_TOT_6#LT# 1, 0,LOT_S1_6) ;

LOT_S1_6<=Q_S1_TOT_6 ;

@GIN(LOT_S1_6);@GIN(Q_S1_LOT_T_6);@GIN(Q_S1_TOT_

```

6) ;

```

Q_S1_TOT_7=Z_1_7 +Z_3_7 +Z_4_7 +Z_5_7 +Z_6_7
+Z_8_7 +Z_9_7 +Z_10_7 +Z_11_7 +Z_12_7 +Z_17_7
+Z_18_7 +Z_19_7 +Z_20_7 +Z_21_7 +Z_22_7 +Z_23_7
+Z_25_7 +Z_26_7 +Z_27_7 +Z_29_7 +Z_31_7 +Z_32_7
+Z_33_7 +Z_34_7 +Z_36_7 +Z_37_7 +Z_38_7 +Z_39_7
+Z_40_7 +Z_41_7 +Z_42_7 +Z_46_7 +Z_51_7 +Z_52_7
+Z_54_7 +Z_55_7 +Z_56_7 +Z_57_7 +Z_60_7 +Z_61_7
+Z_62_7 +Z_65_7 +Z_66_7 +Z_68_7 +Z_70_7 +Z_72_7
+Z_73_7 +Z_74_7 +Z_76_7 +Z_77_7 +Z_78_7 +Z_80_7
+Z_82_7 +Z_84_7 +Z_85_7 +Z_86_7 +Z_87_7 +Z_88_7
+Z_89_7 +Z_90_7 +Z_92_7 +Z_94_7 +Z_96_7 +Z_97_7
+Z_99_7 +Z_100_7 ;

```

```
Q_S1_LOT_T_7<=LOT_S1_7* S_S1_M1_M2_7*40000 ;
```

```
Q_S1_LOT_T_7=Q_S1_TOT_7* S_S1_M1_M2_7 ;
```

```
LOT_S1_7= @IF(Q_S1_TOT_7#LT# 1, 0,LOT_S1_7) ;
```

```
LOT_S1_7<=Q_S1_TOT_7 ;
```

```
@GIN(LOT_S1_7);@GIN(Q_S1_LOT_T_7);@GIN(Q_S1_TOT_7);
```

```

Q_S1_TOT_8=Z_4_8 +Z_6_8 +Z_7_8 +Z_10_8 +Z_11_8
+Z_12_8 +Z_13_8 +Z_14_8 +Z_15_8 +Z_23_8 +Z_24_8
+Z_25_8 +Z_27_8 +Z_28_8 +Z_30_8 +Z_32_8 +Z_33_8
+Z_34_8 +Z_37_8 +Z_40_8 +Z_41_8 +Z_42_8 +Z_44_8
+Z_46_8 +Z_47_8 +Z_48_8 +Z_50_8 +Z_51_8 +Z_52_8
+Z_53_8 +Z_54_8 +Z_55_8 +Z_56_8 +Z_60_8 +Z_61_8
+Z_63_8 +Z_66_8 +Z_68_8 +Z_71_8 +Z_73_8 +Z_74_8
+Z_75_8 +Z_76_8 +Z_77_8 +Z_78_8 +Z_79_8 +Z_80_8
+Z_81_8 +Z_82_8 +Z_83_8 +Z_84_8 +Z_86_8 +Z_87_8
+Z_89_8 +Z_90_8 +Z_91_8 +Z_94_8 +Z_97_8 +Z_99_8
+Z_100_8 ;

```

```
Q_S1_LOT_T_8<=LOT_S1_8* S_S1_M1_M2_8*40000 ;
```

```
Q_S1_LOT_T_8=Q_S1_TOT_8* S_S1_M1_M2_8 ;
```

```
LOT_S1_8= @IF(Q_S1_TOT_8#LT# 1, 0,LOT_S1_8) ;
```

```
LOT_S1_8<=Q_S1_TOT_8 ;
```

```
@GIN(LOT_S1_8);@GIN(Q_S1_LOT_T_8);@GIN(Q_S1_TOT_8);
```

```

Q_S1_TOT_9=Z_2_9 +Z_3_9 +Z_4_9 +Z_5_9 +Z_6_9
+Z_7_9 +Z_8_9 +Z_10_9 +Z_15_9 +Z_16_9 +Z_19_9
+Z_20_9 +Z_21_9 +Z_22_9 +Z_23_9 +Z_25_9 +Z_27_9
+Z_30_9 +Z_31_9 +Z_34_9 +Z_35_9 +Z_38_9 +Z_39_9
+Z_40_9 +Z_42_9 +Z_43_9 +Z_44_9 +Z_47_9 +Z_49_9
+Z_51_9 +Z_54_9 +Z_55_9 +Z_58_9 +Z_60_9 +Z_66_9
+Z_70_9 +Z_71_9 +Z_72_9 +Z_73_9 +Z_75_9 +Z_76_9
+Z_81_9 +Z_82_9 +Z_83_9 +Z_86_9 +Z_88_9 +Z_91_9
+Z_92_9 +Z_93_9 +Z_94_9 +Z_96_9 +Z_97_9 +Z_99_9
+Z_100_9 ;

```

```

Q_S1_LOT_T_9<=LOT_S1_9* S_S1_M1_M2_9*40000 ;

Q_S1_LOT_T_9=Q_S1_TOT_9* S_S1_M1_M2_9 ;

LOT_S1_9= @IF(Q_S1_TOT_9#LT# 1, 0,LOT_S1_9) ;

LOT_S1_9<=Q_S1_TOT_9 ;

@GIN(LOT_S1_9);@GIN(Q_S1_LOT_T_9);@GIN(Q_S1_TOT_
9);

Q_S1_TOT_10=Z_4_10 +Z_5_10 +Z_6_10 +Z_9_10
+Z_10_10 +Z_13_10 +Z_16_10 +Z_18_10 +Z_19_10
+Z_20_10 +Z_21_10 +Z_27_10 +Z_28_10 +Z_29_10
+Z_30_10 +Z_34_10 +Z_35_10 +Z_39_10 +Z_41_10
+Z_44_10 +Z_46_10 +Z_51_10 +Z_52_10 +Z_54_10
+Z_55_10 +Z_56_10 +Z_57_10 +Z_59_10 +Z_60_10
+Z_62_10 +Z_65_10 +Z_66_10 +Z_69_10 +Z_71_10
+Z_72_10 +Z_80_10 +Z_82_10 +Z_84_10 +Z_86_10
+Z_88_10 +Z_92_10 +Z_94_10 +Z_96_10 +Z_100_10 ;

Q_S1_LOT_T_10<=LOT_S1_10* S_S1_M1_M2_10*40000 ;

Q_S1_LOT_T_10=Q_S1_TOT_10* S_S1_M1_M2_10 ;

LOT_S1_10= @IF(Q_S1_TOT_10#LT# 1, 0,LOT_S1_10) ;

LOT_S1_10<=Q_S1_TOT_10 ;

@GIN(LOT_S1_10);@GIN(Q_S1_LOT_T_10);@GIN(Q_S1_TO
T_10);

```

```

C1_1 = (Z_1_1 +Z_2_1 +Z_4_1 +Z_5_1 +Z_6_1 +Z_8_1
+Z_10_1 +Z_12_1 +Z_13_1 +Z_14_1 +Z_15_1 +Z_17_1
+Z_18_1 +Z_19_1 +Z_20_1 +Z_21_1 +Z_22_1 +Z_23_1
+Z_24_1 +Z_27_1 +Z_29_1 +Z_31_1 +Z_32_1 +Z_33_1
+Z_34_1 +Z_39_1 +Z_41_1 +Z_42_1 +Z_43_1 +Z_44_1
+Z_45_1 +Z_46_1 +Z_47_1 +Z_51_1 +Z_52_1 +Z_53_1
+Z_55_1 +Z_56_1 +Z_59_1 +Z_61_1 +Z_62_1 +Z_64_1
+Z_65_1 +Z_66_1 +Z_68_1 +Z_69_1 +Z_71_1 +Z_72_1
+Z_73_1 +Z_74_1 +Z_76_1 +Z_77_1 +Z_78_1 +Z_82_1
+Z_83_1 +Z_84_1 +Z_85_1 +Z_86_1 +Z_87_1 +Z_89_1
+Z_90_1 +Z_91_1 +Z_92_1 +Z_93_1 +Z_95_1 +Z_97_1
+Z_98_1 +Z_100_1 ) * COST_S1_1 + (Z_1_2 +Z_2_2
+Z_3_2 +Z_4_2 +Z_7_2 +Z_9_2 +Z_10_2 +Z_11_2
+Z_13_2 +Z_14_2 +Z_15_2 +Z_20_2 +Z_21_2 +Z_22_2
+Z_23_2 +Z_24_2 +Z_26_2 +Z_29_2 +Z_30_2 +Z_31_2
+Z_32_2 +Z_34_2 +Z_35_2 +Z_36_2 +Z_37_2 +Z_38_2
+Z_39_2 +Z_40_2 +Z_41_2 +Z_42_2 +Z_43_2 +Z_46_2
+Z_47_2 +Z_48_2 +Z_49_2 +Z_51_2 +Z_53_2 +Z_54_2
+Z_55_2 +Z_57_2 +Z_58_2 +Z_59_2 +Z_63_2 +Z_64_2
+Z_65_2 +Z_67_2 +Z_68_2 +Z_69_2 +Z_71_2 +Z_72_2
+Z_74_2 +Z_76_2 +Z_77_2 +Z_78_2 +Z_79_2 +Z_82_2
+Z_83_2 +Z_88_2 +Z_90_2 +Z_94_2 +Z_95_2 +Z_98_2
+Z_99_2 +Z_100_2 ) * COST_S1_2 + (Z_2_3 +Z_4_3
+Z_5_3 +Z_7_3 +Z_8_3 +Z_9_3 +Z_16_3 +Z_17_3
+Z_19_3 +Z_20_3 +Z_21_3 +Z_22_3 +Z_23_3 +Z_24_3

```

Optimisation terms for the production system 1.

```

+z_25_3 +z_26_3 +z_28_3 +z_30_3 +z_32_3 +z_37_3
+z_38_3 +z_40_3 +z_41_3 +z_42_3 +z_43_3 +z_47_3
+z_48_3 +z_49_3 +z_50_3 +z_51_3 +z_53_3 +z_56_3
+z_65_3 +z_66_3 +z_68_3 +z_75_3 +z_77_3 +z_78_3
+z_80_3 +z_82_3 +z_83_3 +z_87_3 +z_91_3 +z_94_3
+z_95_3 +z_96_3 +z_100_3 ) * COST_S1_3 + (z_1_4
+z_2_4 +z_3_4 +z_4_4 +z_5_4 +z_6_4 +z_8_4 +z_9_4
+z_12_4 +z_13_4 +z_14_4 +z_15_4 +z_18_4 +z_20_4
+z_22_4 +z_23_4 +z_24_4 +z_25_4 +z_26_4 +z_27_4
+z_29_4 +z_30_4 +z_32_4 +z_33_4 +z_34_4 +z_35_4
+z_36_4 +z_37_4 +z_38_4 +z_39_4 +z_40_4 +z_41_4
+z_42_4 +z_44_4 +z_49_4 +z_50_4 +z_51_4 +z_52_4
+z_53_4 +z_55_4 +z_56_4 +z_57_4 +z_58_4 +z_59_4
+z_60_4 +z_61_4 +z_65_4 +z_66_4 +z_68_4 +z_70_4
+z_72_4 +z_73_4 +z_74_4 +z_75_4 +z_76_4 +z_83_4
+z_85_4 +z_88_4 +z_90_4 +z_92_4 +z_93_4 +z_94_4
+z_95_4 +z_98_4 +z_99_4 +z_100_4 ) * COST_S1_4
+ (z_4_5 +z_5_5 +z_8_5 +z_10_5 +z_13_5 +z_14_5
+z_18_5 +z_19_5 +z_20_5 +z_21_5 +z_22_5 +z_23_5
+z_25_5 +z_27_5 +z_28_5 +z_29_5 +z_30_5 +z_31_5
+z_33_5 +z_38_5 +z_40_5 +z_41_5 +z_44_5 +z_45_5
+z_46_5 +z_47_5 +z_48_5 +z_50_5 +z_51_5 +z_53_5
+z_54_5 +z_56_5 +z_58_5 +z_61_5 +z_64_5 +z_67_5
+z_68_5 +z_69_5 +z_72_5 +z_73_5 +z_76_5 +z_78_5
+z_82_5 +z_83_5 +z_84_5 +z_88_5 +z_89_5 +z_91_5
+z_92_5 +z_95_5 +z_99_5 +z_100_5 ) * COST_S1_5
+ (z_2_6 +z_3_6 +z_5_6 +z_6_6 +z_7_6 +z_10_6
+z_11_6 +z_12_6 +z_13_6 +z_14_6 +z_17_6 +z_18_6
+z_19_6 +z_21_6 +z_22_6 +z_24_6 +z_25_6 +z_26_6
+z_27_6 +z_29_6 +z_30_6 +z_31_6 +z_32_6 +z_33_6
+z_34_6 +z_35_6 +z_36_6 +z_37_6 +z_38_6 +z_39_6
+z_41_6 +z_42_6 +z_45_6 +z_46_6 +z_47_6 +z_48_6
+z_49_6 +z_50_6 +z_52_6 +z_53_6 +z_54_6 +z_56_6
+z_57_6 +z_59_6 +z_60_6 +z_61_6 +z_62_6 +z_64_6
+z_66_6 +z_67_6 +z_68_6 +z_69_6 +z_70_6 +z_72_6
+z_74_6 +z_75_6 +z_76_6 +z_78_6 +z_79_6 +z_80_6
+z_81_6 +z_82_6 +z_83_6 +z_84_6 +z_85_6 +z_87_6
+z_90_6 +z_91_6 +z_92_6 +z_93_6 +z_95_6 +z_96_6
+z_97_6 +z_98_6 +z_99_6 ) * COST_S1_6 + (z_1_7
+z_3_7 +z_4_7 +z_5_7 +z_6_7 +z_8_7 +z_9_7
+z_10_7 +z_11_7 +z_12_7 +z_17_7 +z_18_7 +z_19_7
+z_20_7 +z_21_7 +z_22_7 +z_23_7 +z_25_7 +z_26_7
+z_27_7 +z_29_7 +z_31_7 +z_32_7 +z_33_7 +z_34_7
+z_36_7 +z_37_7 +z_38_7 +z_39_7 +z_40_7 +z_41_7
+z_42_7 +z_46_7 +z_51_7 +z_52_7 +z_54_7 +z_55_7
+z_56_7 +z_57_7 +z_60_7 +z_61_7 +z_62_7 +z_65_7
+z_66_7 +z_68_7 +z_70_7 +z_72_7 +z_73_7 +z_74_7
+z_76_7 +z_77_7 +z_78_7 +z_80_7 +z_82_7 +z_84_7
+z_85_7 +z_86_7 +z_87_7 +z_88_7 +z_89_7 +z_90_7
+z_92_7 +z_94_7 +z_96_7 +z_97_7 +z_99_7 +z_100_7
) * COST_S1_7 + (z_4_8 +z_6_8 +z_7_8 +z_10_8
+z_11_8 +z_12_8 +z_13_8 +z_14_8 +z_15_8 +z_23_8
+z_24_8 +z_25_8 +z_27_8 +z_28_8 +z_30_8 +z_32_8
+z_33_8 +z_34_8 +z_37_8 +z_40_8 +z_41_8 +z_42_8
+z_44_8 +z_46_8 +z_47_8 +z_48_8 +z_50_8 +z_51_8
+z_52_8 +z_53_8 +z_54_8 +z_55_8 +z_56_8 +z_60_8
+z_61_8 +z_63_8 +z_66_8 +z_68_8 +z_71_8 +z_73_8
+z_74_8 +z_75_8 +z_76_8 +z_77_8 +z_78_8 +z_79_8
+z_80_8 +z_81_8 +z_82_8 +z_83_8 +z_84_8 +z_86_8
+z_87_8 +z_89_8 +z_90_8 +z_91_8 +z_94_8 +z_97_8
+z_99_8 +z_100_8 ) * COST_S1_8 + (z_2_9 +z_3_9

```

```

+Z_4_9 +Z_5_9 +Z_6_9 +Z_7_9 +Z_8_9 +Z_10_9
+Z_15_9 +Z_16_9 +Z_19_9 +Z_20_9 +Z_21_9 +Z_22_9
+Z_23_9 +Z_25_9 +Z_27_9 +Z_30_9 +Z_31_9 +Z_34_9
+Z_35_9 +Z_38_9 +Z_39_9 +Z_40_9 +Z_42_9 +Z_43_9
+Z_44_9 +Z_47_9 +Z_49_9 +Z_51_9 +Z_54_9 +Z_55_9
+Z_58_9 +Z_60_9 +Z_66_9 +Z_70_9 +Z_71_9 +Z_72_9
+Z_73_9 +Z_75_9 +Z_76_9 +Z_81_9 +Z_82_9 +Z_83_9
+Z_86_9 +Z_88_9 +Z_91_9 +Z_92_9 +Z_93_9 +Z_94_9
+Z_96_9 +Z_97_9 +Z_99_9 +Z_100_9 ) * COST_S1_9
+ (Z_4_10 +Z_5_10 +Z_6_10 +Z_9_10 +Z_10_10
+Z_13_10 +Z_16_10 +Z_18_10 +Z_19_10 +Z_20_10
+Z_21_10 +Z_27_10 +Z_28_10 +Z_29_10 +Z_30_10
+Z_34_10 +Z_35_10 +Z_39_10 +Z_41_10 +Z_44_10
+Z_46_10 +Z_51_10 +Z_52_10 +Z_54_10 +Z_55_10
+Z_56_10 +Z_57_10 +Z_59_10 +Z_60_10 +Z_62_10
+Z_65_10 +Z_66_10 +Z_69_10 +Z_71_10 +Z_72_10
+Z_80_10 +Z_82_10 +Z_84_10 +Z_86_10 +Z_88_10
+Z_92_10 +Z_94_10 +Z_96_10 +Z_100_10 ) *
COST_S1_10 ;

```

```

C2_1 = LOT_S1_1 * SET_S1_2 + (LOT_S1_1 *
((S_S1_M1_M3_1 * T_S1_M3_1_1) / (S_S1_M1_M2_1 *
T_S1_M2_1_1))) * SET_S1_3 + LOT_S1_2 * SET_S1_2
+ (LOT_S1_2 * ((S_S1_M1_M3_2 * T_S1_M3_1_2)
/ (S_S1_M1_M2_2 * T_S1_M2_1_2))) * SET_S1_3 +
LOT_S1_3 * SET_S1_2 + (LOT_S1_3 * ((S_S1_M1_M3_3
* T_S1_M3_1_3) / (S_S1_M1_M2_3 * T_S1_M2_1_3))) *
SET_S1_3 + LOT_S1_4 * SET_S1_2 + (LOT_S1_4 *
((S_S1_M1_M3_4 * T_S1_M3_1_4) / (S_S1_M1_M2_4 *
T_S1_M2_1_4))) * SET_S1_3 + LOT_S1_5 * SET_S1_2
+ (LOT_S1_5 * ((S_S1_M1_M3_5 * T_S1_M3_1_5)
/ (S_S1_M1_M2_5 * T_S1_M2_1_5))) * SET_S1_3 +
LOT_S1_6 * SET_S1_2 + (LOT_S1_6 * ((S_S1_M1_M3_6
* T_S1_M3_1_6) / (S_S1_M1_M2_6 * T_S1_M2_1_6))) *
SET_S1_3 + LOT_S1_7 * SET_S1_2 + (LOT_S1_7 *
((S_S1_M1_M3_7 * T_S1_M3_1_7) / (S_S1_M1_M2_7 *
T_S1_M2_1_7))) * SET_S1_3 + LOT_S1_8 * SET_S1_2
+ (LOT_S1_8 * ((S_S1_M1_M3_8 * T_S1_M3_1_8)
/ (S_S1_M1_M2_8 * T_S1_M2_1_8))) * SET_S1_3 +
LOT_S1_9 * SET_S1_2 + (LOT_S1_9 * ((S_S1_M1_M3_9
* T_S1_M3_1_9) / (S_S1_M1_M2_9 * T_S1_M2_1_9))) *
SET_S1_3 + LOT_S1_10 * SET_S1_2 + (LOT_S1_10 *
((S_S1_M1_M3_10 * T_S1_M3_1_10) / (S_S1_M1_M2_10
* T_S1_M2_1_10))) * SET_S1_3;

```

```

C3_1 = C_D_1 * @ABS((LOT_S1_1 * T_S1_M1_1_1 - (
T_S1_M2_1_1 * Q_S1_LOT_T_1)) + (LOT_S1_2 *
T_S1_M1_1_2 - (T_S1_M2_1_2 * Q_S1_LOT_T_2)) +
(LOT_S1_3 * T_S1_M1_1_3 - (T_S1_M2_1_3 *
Q_S1_LOT_T_3)) + (LOT_S1_4 * T_S1_M1_1_4 - (
T_S1_M2_1_4 * Q_S1_LOT_T_4)) + (LOT_S1_5 *
T_S1_M1_1_5 - (T_S1_M2_1_5 * Q_S1_LOT_T_5)) +
(LOT_S1_6 * T_S1_M1_1_6 - (T_S1_M2_1_6 *
Q_S1_LOT_T_6)) + (LOT_S1_7 * T_S1_M1_1_7 - (
T_S1_M2_1_7 * Q_S1_LOT_T_7)) + (LOT_S1_8 *
T_S1_M1_1_8 - (T_S1_M2_1_8 * Q_S1_LOT_T_8)) +
(LOT_S1_9 * T_S1_M1_1_9 - (T_S1_M2_1_9 *
Q_S1_LOT_T_9)) + (LOT_S1_10 * T_S1_M1_1_10 - (
T_S1_M2_1_10 * Q_S1_LOT_T_10)));

```

```

S_S2_M1_M2_1 = @FLOOR(( 400.0 - 40) / ( 2.2 +
0.15 )) ;

S_S2_M1_M3_1 = @FLOOR(( 330.0/25.4 - 1 ) * 4.0)
;

T_S2_M1_1_1 = @FLOOR(( 330.0/25.4 - 1 ) * 4.0) /
1.5 ;

T_S2_M2_1_1 = 4.0 * 0.15 ;

T_S2_M3_1_1 = 0 ;

COST_S2_1 = 330.0 * 0.15 * 8 * 4.0 + 400.0 * (
330.0 / 25.4 - 1) ;

S_S2_M1_M2_2 = @FLOOR(( 400.0 - 40) / ( 2.2 +
0.15 )) ;

S_S2_M1_M3_2 = @FLOOR(( 330.0/25.4 - 1 ) * 4.0)
;

T_S2_M1_1_2 = @FLOOR(( 330.0/25.4 - 1 ) * 4.0) /
1.5 ;

T_S2_M2_1_2 = 4.0 * 0.15 ;

T_S2_M3_1_2 = 0 ;

COST_S2_2 = 330.0 * 0.15 * 8 * 4.0 + 400.0 * (
330.0 / 25.4 - 1) ;

S_S2_M1_M2_3 = @FLOOR(( 400.0 - 40) / ( 2.2 +
0.15 )) ;

S_S2_M1_M3_3 = @FLOOR(( 330.0/25.4 - 1 ) * 4.0)
;

T_S2_M1_1_3 = @FLOOR(( 330.0/25.4 - 1 ) * 4.0) /
1.5 ;

T_S2_M2_1_3 = 4.0 * 0.15 ;

T_S2_M3_1_3 = 0 ;

COST_S2_3 = 330.0 * 0.15 * 8 * 4.0 + 400.0 * (
330.0 / 25.4 - 1) ;

S_S2_M1_M2_4 = @FLOOR(( 400.0 - 40) / ( 2.2 +
0.15 )) ;

S_S2_M1_M3_4 = @FLOOR(( 330.0/25.4 - 1 ) * 4.0)
;

T_S2_M1_1_4 = @FLOOR(( 330.0/25.4 - 1 ) * 4.0) /

```

Constraints about the process durations, setup costs and direct costs on the production system 2.

```

1.5 ;

T_S2_M2_1_4 = 4.0 * 0.15 ;

T_S2_M3_1_4 = 0 ;

COST_S2_4 = 330.0 * 0.15 * 8 * 4.0 + 400.0 * (
330.0 / 25.4 - 1) ;

S_S2_M1_M2_5 = @FLOOR(( 400.0 - 40) / ( 2.2 +
0.15 )) ;

S_S2_M1_M3_5 = @FLOOR(( 330.0/25.4 - 1 ) * 4.0)
;

T_S2_M1_1_5 = @FLOOR(( 330.0/25.4 - 1 ) * 4.0) /
1.5 ;

T_S2_M2_1_5 = 4.0 * 0.15 ;

T_S2_M3_1_5 = 0 ;

COST_S2_5 = 330.0 * 0.15 * 8 * 4.0 + 400.0 * (
330.0 / 25.4 - 1) ;

```

```
C_D_2=50.0;
```

```
SET_S2_2 = 150;
```

```
SET_S2_3 = 0;
```

```

Q_S2_TOT_1=K_1_1 +K_4_1 +K_5_1 +K_6_1 +K_9_1
+K_13_1 +K_14_1 +K_15_1 +K_19_1 +K_21_1 +K_22_1
+K_23_1 +K_25_1 +K_27_1 +K_29_1 +K_31_1 +K_32_1
+K_33_1 +K_36_1 +K_37_1 +K_38_1 +K_39_1 +K_40_1
+K_42_1 +K_43_1 +K_44_1 +K_46_1 +K_48_1 +K_50_1
+K_52_1 +K_57_1 +K_58_1 +K_59_1 +K_60_1 +K_61_1
+K_62_1 +K_64_1 +K_65_1 +K_71_1 +K_72_1 +K_74_1
+K_78_1 +K_79_1 +K_80_1 +K_82_1 +K_83_1 +K_84_1
+K_85_1 +K_86_1 +K_87_1 +K_89_1 +K_91_1 +K_92_1
+K_94_1 +K_95_1 +K_96_1 +K_97_1 +K_99_1 ;

```

```
Q_S2_LOT_T_1<=LOT_S2_1* S_S2_M1_M2_1*40000 ;
```

```
Q_S2_LOT_T_1=Q_S2_TOT_1* S_S2_M1_M2_1 ;
```

```
LOT_S2_1= @IF(Q_S2_TOT_1#LT# 1, 0,LOT_S2_1) ;
```

```
LOT_S2_1<=Q_S2_TOT_1 ;
```

```
@GIN(LOT_S2_1);@GIN(Q_S2_LOT_T_1);@GIN(Q_S2_TOT_1);
```

```

Q_S2_TOT_2=K_1_2 +K_3_2 +K_4_2 +K_5_2 +K_7_2
+K_8_2 +K_9_2 +K_11_2 +K_12_2 +K_13_2 +K_14_2
+K_15_2 +K_21_2 +K_22_2 +K_23_2 +K_24_2 +K_27_2

```

Unitary costs and constraints about the lot sizes and numbers for the production system 2.

```

+K_29_2 +K_32_2 +K_36_2 +K_38_2 +K_39_2 +K_40_2
+K_42_2 +K_44_2 +K_45_2 +K_48_2 +K_49_2 +K_55_2
+K_56_2 +K_58_2 +K_60_2 +K_62_2 +K_63_2 +K_64_2
+K_65_2 +K_66_2 +K_67_2 +K_68_2 +K_70_2 +K_71_2
+K_74_2 +K_79_2 +K_81_2 +K_83_2 +K_84_2 +K_86_2
+K_87_2 +K_89_2 +K_90_2 +K_91_2 +K_92_2 +K_93_2
+K_96_2 +K_100_2 ;

Q_S2_LOT_T_2<=LOT_S2_2* S_S2_M1_M2_2*40000 ;

Q_S2_LOT_T_2=Q_S2_TOT_2* S_S2_M1_M2_2 ;

LOT_S2_2= @IF(Q_S2_TOT_2#LT# 1, 0,LOT_S2_2) ;

LOT_S2_2<=Q_S2_TOT_2 ;

@GIN(LOT_S2_2);@GIN(Q_S2_LOT_T_2);@GIN(Q_S2_TOT_
2);

Q_S2_TOT_3=K_3_3 +K_5_3 +K_6_3 +K_7_3 +K_10_3
+K_16_3 +K_17_3 +K_22_3 +K_23_3 +K_26_3 +K_29_3
+K_31_3 +K_35_3 +K_36_3 +K_37_3 +K_39_3 +K_41_3
+K_44_3 +K_46_3 +K_49_3 +K_51_3 +K_52_3 +K_53_3
+K_54_3 +K_56_3 +K_58_3 +K_61_3 +K_63_3 +K_64_3
+K_65_3 +K_66_3 +K_67_3 +K_68_3 +K_71_3 +K_72_3
+K_74_3 +K_78_3 +K_79_3 +K_81_3 +K_82_3 +K_83_3
+K_84_3 +K_85_3 +K_86_3 +K_89_3 +K_90_3 +K_91_3
+K_92_3 +K_94_3 +K_97_3 ;

Q_S2_LOT_T_3<=LOT_S2_3* S_S2_M1_M2_3*40000 ;

Q_S2_LOT_T_3=Q_S2_TOT_3* S_S2_M1_M2_3 ;

LOT_S2_3= @IF(Q_S2_TOT_3#LT# 1, 0,LOT_S2_3) ;

LOT_S2_3<=Q_S2_TOT_3 ;

@GIN(LOT_S2_3);@GIN(Q_S2_LOT_T_3);@GIN(Q_S2_TOT_
3);

Q_S2_TOT_4=K_1_4 +K_2_4 +K_5_4 +K_9_4 +K_10_4
+K_12_4 +K_15_4 +K_16_4 +K_17_4 +K_18_4 +K_19_4
+K_20_4 +K_24_4 +K_25_4 +K_26_4 +K_27_4 +K_28_4
+K_29_4 +K_33_4 +K_34_4 +K_35_4 +K_38_4 +K_41_4
+K_42_4 +K_45_4 +K_49_4 +K_52_4 +K_53_4 +K_55_4
+K_57_4 +K_58_4 +K_60_4 +K_63_4 +K_65_4 +K_68_4
+K_73_4 +K_77_4 +K_78_4 +K_80_4 +K_83_4 +K_85_4
+K_87_4 +K_88_4 +K_90_4 +K_93_4 +K_96_4 +K_97_4
+K_99_4 ;

Q_S2_LOT_T_4<=LOT_S2_4* S_S2_M1_M2_4*40000 ;

Q_S2_LOT_T_4=Q_S2_TOT_4* S_S2_M1_M2_4 ;

LOT_S2_4= @IF(Q_S2_TOT_4#LT# 1, 0,LOT_S2_4) ;

LOT_S2_4<=Q_S2_TOT_4 ;

@GIN(LOT_S2_4);@GIN(Q_S2_LOT_T_4);@GIN(Q_S2_TOT_

```

```

4) ;

Q_S2_TOT_5=K_4_5 +K_5_5 +K_6_5 +K_7_5 +K_8_5
+K_9_5 +K_10_5 +K_11_5 +K_16_5 +K_18_5 +K_19_5
+K_20_5 +K_24_5 +K_28_5 +K_29_5 +K_38_5 +K_40_5
+K_41_5 +K_44_5 +K_46_5 +K_48_5 +K_49_5 +K_54_5
+K_56_5 +K_57_5 +K_58_5 +K_60_5 +K_62_5 +K_65_5
+K_66_5 +K_68_5 +K_70_5 +K_72_5 +K_73_5 +K_74_5
+K_77_5 +K_80_5 +K_82_5 +K_83_5 +K_84_5 +K_93_5
+K_96_5 +K_98_5 +K_99_5 ;

Q_S2_LOT_T_5<=LOT_S2_5* S_S2_M1_M2_5*40000 ;

Q_S2_LOT_T_5=Q_S2_TOT_5* S_S2_M1_M2_5 ;

LOT_S2_5= @IF(Q_S2_TOT_5#LT# 1, 0,LOT_S2_5) ;

LOT_S2_5<=Q_S2_TOT_5 ;

@GIN(LOT_S2_5);@GIN(Q_S2_LOT_T_5);@GIN(Q_S2_TOT_
5);
    
```

```

C1_2 = (K_1_1 +K_4_1 +K_5_1 +K_6_1 +K_9_1
+K_13_1 +K_14_1 +K_15_1 +K_19_1 +K_21_1 +K_22_1
+K_23_1 +K_25_1 +K_27_1 +K_29_1 +K_31_1 +K_32_1
+K_33_1 +K_36_1 +K_37_1 +K_38_1 +K_39_1 +K_40_1
+K_42_1 +K_43_1 +K_44_1 +K_46_1 +K_48_1 +K_50_1
+K_52_1 +K_57_1 +K_58_1 +K_59_1 +K_60_1 +K_61_1
+K_62_1 +K_64_1 +K_65_1 +K_71_1 +K_72_1 +K_74_1
+K_78_1 +K_79_1 +K_80_1 +K_82_1 +K_83_1 +K_84_1
+K_85_1 +K_86_1 +K_87_1 +K_89_1 +K_91_1 +K_92_1
+K_94_1 +K_95_1 +K_96_1 +K_97_1 +K_99_1 ) *
COST_S2_1 + (K_1_2 +K_3_2 +K_4_2 +K_5_2 +K_7_2
+K_8_2 +K_9_2 +K_11_2 +K_12_2 +K_13_2 +K_14_2
+K_15_2 +K_21_2 +K_22_2 +K_23_2 +K_24_2 +K_27_2
+K_29_2 +K_32_2 +K_36_2 +K_38_2 +K_39_2 +K_40_2
+K_42_2 +K_44_2 +K_45_2 +K_48_2 +K_49_2 +K_55_2
+K_56_2 +K_58_2 +K_60_2 +K_62_2 +K_63_2 +K_64_2
+K_65_2 +K_66_2 +K_67_2 +K_68_2 +K_70_2 +K_71_2
+K_74_2 +K_79_2 +K_81_2 +K_83_2 +K_84_2 +K_86_2
+K_87_2 +K_89_2 +K_90_2 +K_91_2 +K_92_2 +K_93_2
+K_96_2 +K_100_2 ) * COST_S2_2 + (K_3_3 +K_5_3
+K_6_3 +K_7_3 +K_10_3 +K_16_3 +K_17_3 +K_22_3
+K_23_3 +K_26_3 +K_29_3 +K_31_3 +K_35_3 +K_36_3
+K_37_3 +K_39_3 +K_41_3 +K_44_3 +K_46_3 +K_49_3
+K_51_3 +K_52_3 +K_53_3 +K_54_3 +K_56_3 +K_58_3
+K_61_3 +K_63_3 +K_64_3 +K_65_3 +K_66_3 +K_67_3
+K_68_3 +K_71_3 +K_72_3 +K_74_3 +K_78_3 +K_79_3
+K_81_3 +K_82_3 +K_83_3 +K_84_3 +K_85_3 +K_86_3
+K_89_3 +K_90_3 +K_91_3 +K_92_3 +K_94_3 +K_97_3
) * COST_S2_3 + (K_1_4 +K_2_4 +K_5_4 +K_9_4
+K_10_4 +K_12_4 +K_15_4 +K_16_4 +K_17_4 +K_18_4
+K_19_4 +K_20_4 +K_24_4 +K_25_4 +K_26_4 +K_27_4
+K_28_4 +K_29_4 +K_33_4 +K_34_4 +K_35_4 +K_38_4
+K_41_4 +K_42_4 +K_45_4 +K_49_4 +K_52_4 +K_53_4
+K_55_4 +K_57_4 +K_58_4 +K_60_4 +K_63_4 +K_65_4
+K_68_4 +K_73_4 +K_77_4 +K_78_4 +K_80_4 +K_83_4
    
```

Optimisation terms for the production system 2.

```

+K_85_4 +K_87_4 +K_88_4 +K_90_4 +K_93_4 +K_96_4
+K_97_4 +K_99_4 ) * COST_S2_4 + (K_4_5 +K_5_5
+K_6_5 +K_7_5 +K_8_5 +K_9_5 +K_10_5 +K_11_5
+K_16_5 +K_18_5 +K_19_5 +K_20_5 +K_24_5 +K_28_5
+K_29_5 +K_38_5 +K_40_5 +K_41_5 +K_44_5 +K_46_5
+K_48_5 +K_49_5 +K_54_5 +K_56_5 +K_57_5 +K_58_5
+K_60_5 +K_62_5 +K_65_5 +K_66_5 +K_68_5 +K_70_5
+K_72_5 +K_73_5 +K_74_5 +K_77_5 +K_80_5 +K_82_5
+K_83_5 +K_84_5 +K_93_5 +K_96_5 +K_98_5 +K_99_5
) * COST_S2_5 ;

```

```

C2_2 = LOT_S2_1 * SET_S2_2 + (LOT_S2_1 *
((S_S2_M1_M3_1 * T_S2_M3_1_1) / (S_S2_M1_M2_1 *
T_S2_M2_1_1))) * SET_S2_3 + LOT_S2_2 * SET_S2_2
+ (LOT_S2_2 * ((S_S2_M1_M3_2 * T_S2_M3_1_2)
/ (S_S2_M1_M2_2 * T_S2_M2_1_2))) * SET_S2_3 +
LOT_S2_3 * SET_S2_2 + (LOT_S2_3 * ((S_S2_M1_M3_3
* T_S2_M3_1_3) / (S_S2_M1_M2_3 * T_S2_M2_1_3))) *
SET_S2_3 + LOT_S2_4 * SET_S2_2 + (LOT_S2_4 *
((S_S2_M1_M3_4 * T_S2_M3_1_4) / (S_S2_M1_M2_4 *
T_S2_M2_1_4))) * SET_S2_3 + LOT_S2_5 * SET_S2_2
+ (LOT_S2_5 * ((S_S2_M1_M3_5 * T_S2_M3_1_5)
/ (S_S2_M1_M2_5 * T_S2_M2_1_5))) * SET_S2_3;

```

```

C3_2 = C_D_2 * @ABS((LOT_S2_1 * T_S2_M1_1_1 - (
T_S2_M2_1_1 * Q_S2_LOT_T_1)) + (LOT_S2_2 *
T_S2_M1_1_2 - (T_S2_M2_1_2 * Q_S2_LOT_T_2)) +
(LOT_S2_3 * T_S2_M1_1_3 - (T_S2_M2_1_3 *
Q_S2_LOT_T_3)) + (LOT_S2_4 * T_S2_M1_1_4 - (
T_S2_M2_1_4 * Q_S2_LOT_T_4)) + (LOT_S2_5 *
T_S2_M1_1_5 - (T_S2_M2_1_5 * Q_S2_LOT_T_5)));

```

```

Z_1_1 + Z_1_2 + Z_1_4 + Z_1_7 + K_1_1 + K_1_2 +
K_1_4 = 100* 9.6 - 40;

```

Constraints about the customer demand.

```

Z_2_1 + Z_2_2 + Z_2_3 + Z_2_4 + Z_2_6 + Z_2_9 +
K_2_4 = 100* 9.6 - 40;

```

```

Z_3_2 + Z_3_4 + Z_3_6 + Z_3_7 + Z_3_9 + K_3_2 +
K_3_3 = 100* 9.6 - 40;

```

```

Z_4_1 + Z_4_2 + Z_4_3 + Z_4_4 + Z_4_5 + Z_4_7 +
Z_4_8 + Z_4_9 + Z_4_10 + K_4_1 + K_4_2 + K_4_5
= 100* 9.6 - 40;

```

```

Z_5_1 + Z_5_3 + Z_5_4 + Z_5_5 + Z_5_6 + Z_5_7 +
Z_5_9 + Z_5_10 + K_5_1 + K_5_2 + K_5_3 + K_5_4
+ K_5_5 = 100* 9.6 - 40;

```

```

Z_6_1 + Z_6_4 + Z_6_6 + Z_6_7 + Z_6_8 + Z_6_9 +
Z_6_10 + K_6_1 + K_6_3 + K_6_5 = 100* 9.6 -
40;

```

```

Z_7_2 + Z_7_3 + Z_7_6 + Z_7_8 + Z_7_9 + K_7_2 +
K_7_3 + K_7_5 = 100* 9.6 - 40;

```

```

Z_8_1 + Z_8_3 + Z_8_4 + Z_8_5 + Z_8_7 + Z_8_9 +

```

$$K_{8_2} + K_{8_5} = 100 * 9.6 - 40;$$

$$Z_{9_2} + Z_{9_3} + Z_{9_4} + Z_{9_7} + Z_{9_10} + K_{9_1} + K_{9_2} + K_{9_4} + K_{9_5} = 100 * 9.6 - 40;$$

$$Z_{10_1} + Z_{10_2} + Z_{10_5} + Z_{10_6} + Z_{10_7} + Z_{10_8} + Z_{10_9} + Z_{10_10} + K_{10_3} + K_{10_4} + K_{10_5} = 100 * 9.6 - 40;$$

$$Z_{11_2} + Z_{11_6} + Z_{11_7} + Z_{11_8} + K_{11_2} + K_{11_5} = 100 * 9.6 - 40;$$

$$Z_{12_1} + Z_{12_4} + Z_{12_6} + Z_{12_7} + Z_{12_8} + K_{12_2} + K_{12_4} = 100 * 9.6 - 40;$$

$$Z_{13_1} + Z_{13_2} + Z_{13_4} + Z_{13_5} + Z_{13_6} + Z_{13_8} + Z_{13_10} + K_{13_1} + K_{13_2} = 100 * 9.6 - 40;$$

$$Z_{14_1} + Z_{14_2} + Z_{14_4} + Z_{14_5} + Z_{14_6} + Z_{14_8} + K_{14_1} + K_{14_2} = 100 * 9.6 - 40;$$

$$Z_{15_1} + Z_{15_2} + Z_{15_4} + Z_{15_8} + Z_{15_9} + K_{15_1} + K_{15_2} + K_{15_4} = 100 * 9.6 - 40;$$

$$Z_{16_3} + Z_{16_9} + Z_{16_10} + K_{16_3} + K_{16_4} + K_{16_5} = 100 * 9.6 - 40;$$

$$Z_{17_1} + Z_{17_3} + Z_{17_6} + Z_{17_7} + K_{17_3} + K_{17_4} = 100 * 9.6 - 40;$$

$$Z_{18_1} + Z_{18_4} + Z_{18_5} + Z_{18_6} + Z_{18_7} + Z_{18_10} + K_{18_4} + K_{18_5} = 100 * 9.6 - 40;$$

$$Z_{19_1} + Z_{19_3} + Z_{19_5} + Z_{19_6} + Z_{19_7} + Z_{19_9} + Z_{19_10} + K_{19_1} + K_{19_4} + K_{19_5} = 100 * 9.6 - 40;$$

$$Z_{20_1} + Z_{20_2} + Z_{20_3} + Z_{20_4} + Z_{20_5} + Z_{20_7} + Z_{20_9} + Z_{20_10} + K_{20_4} + K_{20_5} = 100 * 9.6 - 40;$$

$$Z_{21_1} + Z_{21_2} + Z_{21_3} + Z_{21_5} + Z_{21_6} + Z_{21_7} + Z_{21_9} + Z_{21_10} + K_{21_1} + K_{21_2} = 100 * 9.6 - 40;$$

$$Z_{22_1} + Z_{22_2} + Z_{22_3} + Z_{22_4} + Z_{22_5} + Z_{22_6} + Z_{22_7} + Z_{22_9} + K_{22_1} + K_{22_2} + K_{22_3} = 100 * 9.6 - 40;$$

$$Z_{23_1} + Z_{23_2} + Z_{23_3} + Z_{23_4} + Z_{23_5} + Z_{23_7} + Z_{23_8} + Z_{23_9} + K_{23_1} + K_{23_2} + K_{23_3} = 100 * 9.6 - 40;$$

$$Z_{24_1} + Z_{24_2} + Z_{24_3} + Z_{24_4} + Z_{24_6} + Z_{24_8} + K_{24_2} + K_{24_4} + K_{24_5} = 100 * 9.6 - 40;$$

$$Z_{25_3} + Z_{25_4} + Z_{25_5} + Z_{25_6} + Z_{25_7} + Z_{25_8} + Z_{25_9} + K_{25_1} + K_{25_4} = 100 * 9.6 - 40;$$

40;

$$Z_{26_2} + Z_{26_3} + Z_{26_4} + Z_{26_6} + Z_{26_7} + \\ K_{26_3} + K_{26_4} = 100 * 9.6 - 40;$$

$$Z_{27_1} + Z_{27_4} + Z_{27_5} + Z_{27_6} + Z_{27_7} + \\ Z_{27_8} + Z_{27_9} + Z_{27_10} + K_{27_1} + K_{27_2} + \\ K_{27_4} = 100 * 9.6 - 40;$$

$$Z_{28_3} + Z_{28_5} + Z_{28_8} + Z_{28_10} + K_{28_4} + \\ K_{28_5} = 100 * 9.6 - 40;$$

$$Z_{29_1} + Z_{29_2} + Z_{29_4} + Z_{29_5} + Z_{29_6} + \\ Z_{29_7} + Z_{29_10} + K_{29_1} + K_{29_2} + K_{29_3} + \\ K_{29_4} + K_{29_5} = 100 * 9.6 - 40;$$

$$Z_{30_2} + Z_{30_3} + Z_{30_4} + Z_{30_5} + Z_{30_6} + \\ Z_{30_8} + Z_{30_9} + Z_{30_10} = 100 * 9.6 - 40;$$

$$Z_{31_1} + Z_{31_2} + Z_{31_5} + Z_{31_6} + Z_{31_7} + \\ Z_{31_9} + K_{31_1} + K_{31_3} = 100 * 9.6 - 40;$$

$$Z_{32_1} + Z_{32_2} + Z_{32_3} + Z_{32_4} + Z_{32_6} + \\ Z_{32_7} + Z_{32_8} + K_{32_1} + K_{32_2} = 100 * 9.6 - \\ 40;$$

$$Z_{33_1} + Z_{33_4} + Z_{33_5} + Z_{33_6} + Z_{33_7} + \\ Z_{33_8} + K_{33_1} + K_{33_4} = 100 * 9.6 - 40;$$

$$Z_{34_1} + Z_{34_2} + Z_{34_4} + Z_{34_6} + Z_{34_7} + \\ Z_{34_8} + Z_{34_9} + Z_{34_10} + K_{34_4} = 100 * 9.6 \\ - 40;$$

$$Z_{35_2} + Z_{35_4} + Z_{35_6} + Z_{35_9} + Z_{35_10} + \\ K_{35_3} + K_{35_4} = 100 * 9.6 - 40;$$

$$Z_{36_2} + Z_{36_4} + Z_{36_6} + Z_{36_7} + K_{36_1} + \\ K_{36_2} + K_{36_3} = 100 * 9.6 - 40;$$

$$Z_{37_2} + Z_{37_3} + Z_{37_4} + Z_{37_6} + Z_{37_7} + \\ Z_{37_8} + K_{37_1} + K_{37_3} = 100 * 9.6 - 40;$$

$$Z_{38_2} + Z_{38_3} + Z_{38_4} + Z_{38_5} + Z_{38_6} + \\ Z_{38_7} + Z_{38_9} + K_{38_1} + K_{38_2} + K_{38_4} + \\ K_{38_5} = 100 * 9.6 - 40;$$

$$Z_{39_1} + Z_{39_2} + Z_{39_4} + Z_{39_6} + Z_{39_7} + \\ Z_{39_9} + Z_{39_10} + K_{39_1} + K_{39_2} + K_{39_3} = \\ 100 * 9.6 - 40;$$

$$Z_{40_2} + Z_{40_3} + Z_{40_4} + Z_{40_5} + Z_{40_7} + \\ Z_{40_8} + Z_{40_9} + K_{40_1} + K_{40_2} + K_{40_5} = \\ 100 * 9.6 - 40;$$

$$Z_{41_1} + Z_{41_2} + Z_{41_3} + Z_{41_4} + Z_{41_5} + \\ Z_{41_6} + Z_{41_7} + Z_{41_8} + Z_{41_10} + K_{41_3} + \\ K_{41_4} + K_{41_5} = 100 * 9.6 - 40;$$

$$Z_{42_1} + Z_{42_2} + Z_{42_3} + Z_{42_4} + Z_{42_6} + \\ Z_{42_7} + Z_{42_8} + Z_{42_9} + K_{42_1} + K_{42_2} +$$

$$K_{42_4} = 100 * 9.6 - 40;$$

$$Z_{43_1} + Z_{43_2} + Z_{43_3} + Z_{43_9} + K_{43_1} = 100 * 9.6 - 40;$$

$$Z_{44_1} + Z_{44_4} + Z_{44_5} + Z_{44_8} + Z_{44_9} + Z_{44_10} + K_{44_1} + K_{44_2} + K_{44_3} + K_{44_5} = 100 * 9.6 - 40;$$

$$Z_{45_1} + Z_{45_5} + Z_{45_6} + K_{45_2} + K_{45_4} = 100 * 9.6 - 40;$$

$$Z_{46_1} + Z_{46_2} + Z_{46_5} + Z_{46_6} + Z_{46_7} + Z_{46_8} + Z_{46_10} + K_{46_1} + K_{46_3} + K_{46_5} = 100 * 9.6 - 40;$$

$$Z_{47_1} + Z_{47_2} + Z_{47_3} + Z_{47_5} + Z_{47_6} + Z_{47_8} + Z_{47_9} = 100 * 9.6 - 40;$$

$$Z_{48_2} + Z_{48_3} + Z_{48_5} + Z_{48_6} + Z_{48_8} + K_{48_1} + K_{48_2} + K_{48_5} = 100 * 9.6 - 40;$$

$$Z_{49_2} + Z_{49_3} + Z_{49_4} + Z_{49_6} + Z_{49_9} + K_{49_2} + K_{49_3} + K_{49_4} + K_{49_5} = 100 * 9.6 - 40;$$

$$Z_{50_3} + Z_{50_4} + Z_{50_5} + Z_{50_6} + Z_{50_8} + K_{50_1} = 100 * 9.6 - 40;$$

$$Z_{51_1} + Z_{51_2} + Z_{51_3} + Z_{51_4} + Z_{51_5} + Z_{51_7} + Z_{51_8} + Z_{51_9} + Z_{51_10} + K_{51_3} = 100 * 9.6 - 40;$$

$$Z_{52_1} + Z_{52_4} + Z_{52_6} + Z_{52_7} + Z_{52_8} + Z_{52_10} + K_{52_1} + K_{52_3} + K_{52_4} = 100 * 9.6 - 40;$$

$$Z_{53_1} + Z_{53_2} + Z_{53_3} + Z_{53_4} + Z_{53_5} + Z_{53_6} + Z_{53_8} + K_{53_3} + K_{53_4} = 100 * 9.6 - 40;$$

$$Z_{54_2} + Z_{54_5} + Z_{54_6} + Z_{54_7} + Z_{54_8} + Z_{54_9} + Z_{54_10} + K_{54_3} + K_{54_5} = 100 * 9.6 - 40;$$

$$Z_{55_1} + Z_{55_2} + Z_{55_4} + Z_{55_7} + Z_{55_8} + Z_{55_9} + Z_{55_10} + K_{55_2} + K_{55_4} = 100 * 9.6 - 40;$$

$$Z_{56_1} + Z_{56_3} + Z_{56_4} + Z_{56_5} + Z_{56_6} + Z_{56_7} + Z_{56_8} + Z_{56_10} + K_{56_2} + K_{56_3} + K_{56_5} = 100 * 9.6 - 40;$$

$$Z_{57_2} + Z_{57_4} + Z_{57_6} + Z_{57_7} + Z_{57_10} + K_{57_1} + K_{57_4} + K_{57_5} = 100 * 9.6 - 40;$$

$$Z_{58_2} + Z_{58_4} + Z_{58_5} + Z_{58_9} + K_{58_1} + K_{58_2} + K_{58_3} + K_{58_4} + K_{58_5} = 100 * 9.6 - 40;$$

$$Z_{59_1} + Z_{59_2} + Z_{59_4} + Z_{59_6} + Z_{59_10} + K_{59_1} = 100 * 9.6 - 40;$$

$$Z_{60_4} + Z_{60_6} + Z_{60_7} + Z_{60_8} + Z_{60_9} + Z_{60_10} + K_{60_1} + K_{60_2} + K_{60_4} + K_{60_5} = 100 * 9.6 - 40;$$

$$Z_{61_1} + Z_{61_4} + Z_{61_5} + Z_{61_6} + Z_{61_7} + Z_{61_8} + K_{61_1} + K_{61_3} = 100 * 9.6 - 40;$$

$$Z_{62_1} + Z_{62_6} + Z_{62_7} + Z_{62_10} + K_{62_1} + K_{62_2} + K_{62_5} = 100 * 9.6 - 40;$$

$$Z_{63_2} + Z_{63_8} + K_{63_2} + K_{63_3} + K_{63_4} = 100 * 9.6 - 40;$$

$$Z_{64_1} + Z_{64_2} + Z_{64_5} + Z_{64_6} + K_{64_1} + K_{64_2} + K_{64_3} = 100 * 9.6 - 40;$$

$$Z_{65_1} + Z_{65_2} + Z_{65_3} + Z_{65_4} + Z_{65_7} + Z_{65_10} + K_{65_1} + K_{65_2} + K_{65_3} + K_{65_4} + K_{65_5} = 100 * 9.6 - 40;$$

$$Z_{66_1} + Z_{66_3} + Z_{66_4} + Z_{66_6} + Z_{66_7} + Z_{66_8} + Z_{66_9} + Z_{66_10} + K_{66_2} + K_{66_3} + K_{66_5} = 100 * 9.6 - 40;$$

$$Z_{67_2} + Z_{67_5} + Z_{67_6} + K_{67_2} + K_{67_3} = 100 * 9.6 - 40;$$

$$Z_{68_1} + Z_{68_2} + Z_{68_3} + Z_{68_4} + Z_{68_5} + Z_{68_6} + Z_{68_7} + Z_{68_8} + K_{68_2} + K_{68_3} + K_{68_4} + K_{68_5} = 100 * 9.6 - 40;$$

$$Z_{69_1} + Z_{69_2} + Z_{69_5} + Z_{69_6} + Z_{69_10} = 100 * 9.6 - 40;$$

$$Z_{70_4} + Z_{70_6} + Z_{70_7} + Z_{70_9} + K_{70_2} + K_{70_5} = 100 * 9.6 - 40;$$

$$Z_{71_1} + Z_{71_2} + Z_{71_8} + Z_{71_9} + Z_{71_10} + K_{71_1} + K_{71_2} + K_{71_3} = 100 * 9.6 - 40;$$

$$Z_{72_1} + Z_{72_2} + Z_{72_4} + Z_{72_5} + Z_{72_6} + Z_{72_7} + Z_{72_9} + Z_{72_10} + K_{72_1} + K_{72_3} + K_{72_5} = 100 * 9.6 - 40;$$

$$Z_{73_1} + Z_{73_4} + Z_{73_5} + Z_{73_7} + Z_{73_8} + Z_{73_9} + K_{73_4} + K_{73_5} = 100 * 9.6 - 40;$$

$$Z_{74_1} + Z_{74_2} + Z_{74_4} + Z_{74_6} + Z_{74_7} + Z_{74_8} + K_{74_1} + K_{74_2} + K_{74_3} + K_{74_5} = 100 * 9.6 - 40;$$

$$Z_{75_3} + Z_{75_4} + Z_{75_6} + Z_{75_8} + Z_{75_9} = 100 * 9.6 - 40;$$

$$Z_{76_1} + Z_{76_2} + Z_{76_4} + Z_{76_5} + Z_{76_6} + Z_{76_7} + Z_{76_8} + Z_{76_9} = 100 * 9.6 - 40;$$

$$Z_{77_1} + Z_{77_2} + Z_{77_3} + Z_{77_7} + Z_{77_8} + K_{77_4} + K_{77_5} = 100 * 9.6 - 40;$$

$$Z_{78_1} + Z_{78_2} + Z_{78_3} + Z_{78_5} + Z_{78_6} + Z_{78_7} + Z_{78_8} + K_{78_1} + K_{78_3} + K_{78_4} = 100 * 9.6 - 40;$$

$$Z_{79_2} + Z_{79_6} + Z_{79_8} + K_{79_1} + K_{79_2} + K_{79_3} = 100 * 9.6 - 40;$$

$$Z_{80_3} + Z_{80_6} + Z_{80_7} + Z_{80_8} + Z_{80_10} + K_{80_1} + K_{80_4} + K_{80_5} = 100 * 9.6 - 40;$$

$$Z_{81_6} + Z_{81_8} + Z_{81_9} + K_{81_2} + K_{81_3} = 100 * 9.6 - 40;$$

$$Z_{82_1} + Z_{82_2} + Z_{82_3} + Z_{82_5} + Z_{82_6} + Z_{82_7} + Z_{82_8} + Z_{82_9} + Z_{82_10} + K_{82_1} + K_{82_3} + K_{82_5} = 100 * 9.6 - 40;$$

$$Z_{83_1} + Z_{83_2} + Z_{83_3} + Z_{83_4} + Z_{83_5} + Z_{83_6} + Z_{83_8} + Z_{83_9} + K_{83_1} + K_{83_2} + K_{83_3} + K_{83_4} + K_{83_5} = 100 * 9.6 - 40;$$

$$Z_{84_1} + Z_{84_5} + Z_{84_6} + Z_{84_7} + Z_{84_8} + Z_{84_10} + K_{84_1} + K_{84_2} + K_{84_3} + K_{84_5} = 100 * 9.6 - 40;$$

$$Z_{85_1} + Z_{85_4} + Z_{85_6} + Z_{85_7} + K_{85_1} + K_{85_3} + K_{85_4} = 100 * 9.6 - 40;$$

$$Z_{86_1} + Z_{86_7} + Z_{86_8} + Z_{86_9} + Z_{86_10} + K_{86_1} + K_{86_2} + K_{86_3} = 100 * 9.6 - 40;$$

$$Z_{87_1} + Z_{87_3} + Z_{87_6} + Z_{87_7} + Z_{87_8} + K_{87_1} + K_{87_2} + K_{87_4} = 100 * 9.6 - 40;$$

$$Z_{88_2} + Z_{88_4} + Z_{88_5} + Z_{88_7} + Z_{88_9} + Z_{88_10} + K_{88_4} = 100 * 9.6 - 40;$$

$$Z_{89_1} + Z_{89_5} + Z_{89_7} + Z_{89_8} + K_{89_1} + K_{89_2} + K_{89_3} = 100 * 9.6 - 40;$$

$$Z_{90_1} + Z_{90_2} + Z_{90_4} + Z_{90_6} + Z_{90_7} + Z_{90_8} + K_{90_2} + K_{90_3} + K_{90_4} = 100 * 9.6 - 40;$$

$$Z_{91_1} + Z_{91_3} + Z_{91_5} + Z_{91_6} + Z_{91_8} + Z_{91_9} + K_{91_1} + K_{91_2} + K_{91_3} = 100 * 9.6 - 40;$$

$$Z_{92_1} + Z_{92_4} + Z_{92_5} + Z_{92_6} + Z_{92_7} + Z_{92_9} + Z_{92_10} + K_{92_1} + K_{92_2} + K_{92_3} = 100 * 9.6 - 40;$$

$$Z_{93_1} + Z_{93_4} + Z_{93_6} + Z_{93_9} + K_{93_2} + K_{93_4} + K_{93_5} = 100 * 9.6 - 40;$$

$$Z_{94_2} + Z_{94_3} + Z_{94_4} + Z_{94_7} + Z_{94_8} + Z_{94_9} + Z_{94_10} + K_{94_1} + K_{94_3} = 100 * 9.6 - 40;$$

- 40;

$$Z_{95_1} + Z_{95_2} + Z_{95_3} + Z_{95_4} + Z_{95_5} + Z_{95_6} + K_{95_1} = 100 * 9.6 - 40;$$

$$Z_{96_3} + Z_{96_6} + Z_{96_7} + Z_{96_9} + Z_{96_10} + K_{96_1} + K_{96_2} + K_{96_4} + K_{96_5} = 100 * 9.6 - 40;$$

$$Z_{97_1} + Z_{97_6} + Z_{97_7} + Z_{97_8} + Z_{97_9} + K_{97_1} + K_{97_3} + K_{97_4} = 100 * 9.6 - 40;$$

$$Z_{98_1} + Z_{98_2} + Z_{98_4} + Z_{98_6} + K_{98_5} = 100 * 9.6 - 40;$$

$$Z_{99_2} + Z_{99_4} + Z_{99_5} + Z_{99_6} + Z_{99_7} + Z_{99_8} + Z_{99_9} + K_{99_1} + K_{99_4} + K_{99_5} = 100 * 9.6 - 40;$$

$$Z_{100_1} + Z_{100_2} + Z_{100_3} + Z_{100_4} + Z_{100_5} + Z_{100_7} + Z_{100_8} + Z_{100_9} + Z_{100_10} + K_{100_2} = 100 * 9.6 - 40;$$

Objective function.

$$\text{MIN} = C1_1 + C2_1 + C3_1 + C1_2 + C2_2 + C3_2 ;$$

Abstract - The product variety design is an essential process in order to deal with the flexibility requested by the mass-customisation. During the product variety stage, customers and expert are involved in the definition of the best variety. Therefore a deep understanding of the links between knowledge coming from the customer domain, product domain and process domain is needed. In this thesis the research focus is on the formalisation of this knowledge. Indeed, even if many efforts are present in the knowledge representation literature, *logics* are always used to build these links. But appropriate reflections about the use of *logics* can lead to recognise the risk of ambiguity of the representations, i.e. more than one interpretation of the same represented object are possible. This ambiguity would make the represented knowledge not appropriate for the product variety design. In this work, we propose a framework for the knowledge representation based on the *anti-logicism*. Since the samples of *anti-logicist* systems (e.g. multi-agents, robots) have shown an intelligent behaviour without a representation based on logics, we use the principles the *anti-logicism* to propose our knowledge representation framework. A knowledge representation framework that allows to connect the customer requirements to the manufacturing process parameters is proposed. The core feature of the models based on this framework is the non-ambiguity. Indeed, each piece of knowledge that composes the model can be interpreted in one unique way. This feature allows the perfect collaboration between customer, product engineers and process engineering during the variety design stage. Once the pieces of knowledge coming from different domains are integrated in one model, the framework explains how to generate alternatives of product-process variety by starting from a given customer set. Finally a criterion to compare the different generated alternatives of product-process variety is proposed. A method to instantiate the framework on a 3D CAD has been developed. Moreover, a prototype that uses the knowledge model along with a mathematical solver to propose the best variety has been developed. The impact of the framework on the selection process and on the design process of a customisable product (i.e. water coil) is tested. The test of the instantiation and the prototype allows to show the advantages and the limit of the proposals.

Keywords – knowledge representation, design knowledge, product variety design, product family, product line

Résumé - La conception de variété (ou diversité) de produit est un processus essentiel pour atteindre le niveau de flexibilité requis par la personnalisation de masse. Pendant le processus de conception de la variété, les clients et les experts sont impliqués dans la définition de la meilleure solution. Par conséquent, la compréhension des liens entre les connaissances provenant de ces différents domaines, i.e. client, produit, processus est devenue nécessaire. Dans cette thèse, nous nous intéressons en particulier à la formalisation de ces connaissances. En effet, même si plusieurs efforts ont été accomplis dans le domaine de la représentation de la connaissance, la pensée *logiciste* (i.e. utilisation de méthode à base de logiques formelles) reste la base de la majeure partie des travaux sur la formalisation de la connaissance. Des réflexions appropriées sur l'utilisation des logiques peuvent montrer les risques d'ambiguïté de la représentation: l'utilisation de la logique conduit souvent à une représentation sujette à plusieurs interprétations, i.e. une représentation ambiguë. Une représentation avec cette caractéristique ne répond pas à l'exigence de bien comprendre les liens entre les différentes connaissances impliquées dans la conception de la variété. Notre travail s'intéresse, donc, au développement d'un cadre de modélisation de la connaissance de conception basé sur l'*anti-logicisme*. Les travaux sur les systèmes développés à partir des principes de cette école de représentation de la connaissance montrent à travers des applications concrètes dans les domaines de la robotique ou des systèmes multi-agents que les comportements intelligents peuvent être obtenus sans une représentation de la connaissance basée sur les logiques. Ce cadre permet de développer une variété de produit-processus à partir d'une clientèle définie au départ. Finalement, un critère pour comparer les différentes alternatives de variété générées est aussi proposé. Une méthode pour instancier le cadre de modélisation sur un logiciel de CAO 3D a été développée. De plus, un prototype pour utiliser les modèles de connaissance avec un solveur mathématique a été conçu et développé. Les propositions ont été testées sur un cas d'étude industriel, i.e. batterie froide d'un appareil de réfrigération. Ce test a permis de discuter les avantages et les limites de nos propositions.

Mots-clefs – représentation de la connaissance, connaissances de conception, conception de la variété de produit, famille de produits, ligne de produits