A dependencies satisfaction tool to support the cost oriented configuration of industrial products in the conceptual design stage

Margherita Peruzzini [0000-0003-2260-0392][1], Mauro Pistacchi [0000-0001-5265-7117][2], Andrea Savoretti [0000-0001-5098-8641][3], Marco Mandolini [0000-0003-0962-5982][4] and Roberto Raffaeli [0000-0003-0301-454X][5]

[1] University of Modena and Reggio Emilia, margherita.peruzzini@unimore.it
[2] Polytechnic University of Marche, mauro.pistacchi@gmail.com
[3] Polytechnic University of Marche, a.savoretti@pm.univpm.it
[4] Polytechnic University of Marche, m.mandolini@univpm.it
[5] eCampus University, roberto.raffaeli@uniecampus.it

ABSTRACT

Sustainability and cost optimization are actually the main drivers of product and system design in modern companies. However, cost assessments are usually carried out at the end of the design process to check the validity of the decisions already taken. Therefore, when targets are not achieved, numerous time-consuming iteration loops are necessary to optimize the initial solution. The paper situates in a research aimed at merging functional-based and design-to-cost approaches to propose a CAD-based platform able to assess new product variants from the earliest stages by configuring and assessing feasible design solutions. In particular, an approach is proposed for dealing with dependencies among design parameters in order to support the designer in a rapid definition of valid solutions and optimise them. The approach is described and then applied on an industrial test case, a bridge-crane design process.

Keywords: design-to-cost, knowledge representation, object-oriented design.

1 INTRODUCTION

Modern product design is based on the contemporary satisfaction of market needs, time to market, and global costs. The definition of product variants is a critical activity that involves expertise, product knowledge and the correct estimation of design and production efforts. This trend needs supporting tools integrated with 3D CAD environments that are able to properly structure the design knowledge and support the designers in configuring the best design solution with respect to desired product performance, required manufacturing processes and production quantities. Such an approach considers costs, performances until lifecycle assessments (LCA) to promote sustainability-oriented design strategies.

Indeed, in the modern industrial practice, product cost is estimated at the beginning of the design process by qualitative methods, but is deeply analyzed only at the end of the project by quantitative estimating techniques, when design changes are difficult to implement and very expensive, so that only minor corrective actions can be taken, generating a long iteration loops [15]. Over the last 20

years, some methodologies have been defined to assess cost before product realization in order to optimize product and process design, such as Design for Manufacturing and Assembly (DFMA) approach [2] to Feature-Based Costing (FBC) [28], to Design to Cost (DTC) [9], until knowledge-based engineering (KBE) systems [8, 25]. On the other hand, several approaches like knowledge-based engineering (KBE) [16], artificial intelligence (AI) algorithms [20], object-oriented (OO) design and functional programming [14] have been conceived to easily capture, structure and reuse the design knowledge in a fast and easy way, by automating repetitive tasks and optimization stages. However, their application in industry is difficult due to long data analysis, complex knowledge structuring as well as numerous resources to be involved.

In this context, the research proposes a knowledge-based tool to be used from the conceptual design stages. It supports the modelling phase from functional modules, the identification of the functional features of the product, the representation of the product related knowledge according to design-to-cost vision, and the generation of reliable design solution according to the constraints of the specific application.

It is well known that in the early design phase some activities require a priori choices, which leads to different alternatives depending to the chosen path. Moreover, several constraints, like weight or product overall dimensions will be known only after some activities of the design process have been performed. Consequently, it is useful to explore as many design variants as possible in order to be sure that the best decisions have been made to focus on the best solution.

Then, the proposed method focuses on the design activities that are affected by mutual dependencies in order to explore and optimize different product variants. It makes use of the design structure matrix (DSM) to manage the dependencies between the design parameters. Partition tools allow finding the parameters that are affected by cyclic dependencies and clustering algorithms to reduce the whole design problems in smaller and affordable steps. Solvers of constraint satisfaction problem (CSP) determines the domain of the possible solutions, thus different product alternatives can be evaluated taking advantage of multiple configurations.

The paper presents the approach, the prototypal configuration tool, and some preliminary results on an industrial case study. In particular, the approach has been experimented on a family of cranes for industrial plants in collaboration with an industrial partner. Examples of the definition of the architecture of some crane modules are then described to show the benefits of the approach in finding valid technical solutions and optimizing them regarding cost or weight.

2   BACKGROUND

In their revision of methods to acquire and formalize design knowledge, Chandrasegaran et al. group the representation approaches in five categories [6]: pictorial, symbolic, linguistic, virtual, and algorithmic. The scientific literature provides a broad overview of approaches and methodologies to obtain structured functional representation of a product, with the main goal to offer a wider range of variants and models to meet the consumers' needs [21, 26]. However, as it emerges from representation patterns such as the Function-Behavior-Structure (FBS), additional point of views on the product must be considered.

2.1   Modelling and exploring complex design problems

When the full set of product components, as well as inputs and outputs are available, it is possible to structuring the product function modules from exploiting Design Structure Matrix (DSM). The ability of DSM to manage complex systems is well known [11] and its use has been experimented in design process [1]. In fact, DSM allows representing precedence relationships between activities or parameters of the design process. Moreover, by means of partition tools, it is possible to reorder the DSM and eventually find the coupled clusters of parameters that are affected by mutual dependencies. Aggregation, decomposition and tearing are the possible alternatives to solve the coupled clusters [3]. Since the first two algorithms are not always feasible, the tearing consists in making assumptions that would allow the design process to proceed. A priori choices are properly characteristics of the design process i.e. the material selection of a part. Usually assumptions must be verified after an iteration.

For example, a constraint on the weight of a part can be verified only after several other choices have been made.

On the other hand, the Design Of Experiment (DOE) theory [17] is a traditional way to explore possible solutions. DOE analysis uses a multi-variation approach. Investigators define the number of level for each variable and vary them simultaneously during the tests. Maximum information is thus obtained by means of a full factorial plan, in which all the possible level combinations are performed. Using a regression model is then possible to obtain the system response at different operating conditions. The full factorial plan requires a number of tests increasing with the number of parameters, and easily outgrows the resources of most investigations. In general, fractional factorial plans are adopted in order to reduce the number of tests. In fact, evaluation of all interaction terms is usually unnecessary as the higher order interactions are of gradually decreasing size. These fractional factorial designs are among the most widely used types of designs for product and process design and for process improvement [7].

2.2   Design synthesis as a constraint satisfaction problem

Chakrabarti et al. in [5] affirm that engineering design is the process of satisfying requirements by developing and synthesizing building blocks into meaningful designs that meet the requirements to fulfil needs and desires. The paper shows an overview in design synthesis research, dividing the three major branches of function-based synthesis, grammar-based synthesis, and analogy-based design. The first one includes several methodologies coalesced around the functional decomposition of solutions, taking designers through steps that help decompose a design problem and build conceptual solutions based on the product functionality. All these models try to find connections between functional aspects of product and behavioral, structural and environmental ones.

About the second category, generative graph and spatial grammars are used for computationally encoding knowledge about creating designs: defining a vocabulary and a rule-set that operates over it, the generated design language can be used to rapidly generating standard or novel design alternatives. For instance, Wyatt et al. [29] represent product architecture as a network in which nodes symbolize components and links symbolize connections or other relations. The space of possible architectures is the complete set of possible arrangements of elements that satisfy the constraints. The challenge for the future in grammar-based synthesis is the automatic learning of grammar rules. In fact, engineering grammars would be more readily applied if rules could be learned and adapted on their own by the software.

Finally, analogy is defined as the cognitive process of transferring information from a particular subject to another one. In particular, the case base reasoning (CBR) is used to adapt old situations to meet new demands, using old cases to critique and interpret new solutions [13].

A constraint satisfaction problem (CSP) can be defined on a finite set of variables whose values belong to the definition of finite domains and a set of constraints. A constraint on a set of variables is a restriction on the values that those variables can take simultaneously. Conceptually, a constraint is seen as a set of all the values that the variables can take on at the same time, and that can be represented by means of matrices, equations, inequalities or relations. In a similar manner, engineering design is the process of satisfying requirements by developing and synthesizing building blocks into meaningful designs that meet the requirements to fulfill needs and desires, but always respecting design constraints, which are fundamental in determining the resulting solutions [4]. Numerical constraints and restrictions on the possible values of a design parameter have been used in parametric design [18] or, alternatively, constraints may represent high-level qualitative information being part of the problem definition.

Design process can be considered as a CSP under a set of requirements, assumptions and design limits [19]. A *design* is consistent if it does not violate any of the constraints and is complete if it includes all variables. CSP on finite domains are typically solved using a form of search. The approach for the resolution is based on ad-hoc algorithms that compare all found solutions at the same depth (with usually the same cost) to avoid running in infinite paths. The most used techniques are variants of backtracking, constraint propagation, and local search.

The branching factor at each level of the tree is equal to the number of variables not yet assigned, multiplied by the number of values that each variable can take. Several CSP languages and solvers have

been introduced to reduce the branching factor. *Gecode* is a toolkit for the development of "constraint-based", systems and applications; *MiniZinc* is a modeling language to mid-level constraints, which exploits the advantages of a library of pre-defined constraints; *Cream* is a Class Library for Constraint Programming; *Jacop* is a Java-based constraint solver; *Choco* is a free Java library dedicated to Constraint Programming. Optimization tools can work in adjunction to CSP solver and allow finding out a valid solution that minimizes or maximizes an objective function.

In conclusion, a major goal of design synthesis is to support creation of a large number of alternatives of high value. However, the tools available to the designers in the early stages of product conceptualization are relatively new, and the majority of knowledge to be encapsulated at this stage is tacit, so that its formalization is more difficult with respect to in embodiment and detailed design stages. Ultimately, the research presented in the paper mainly addresses these issues:

1) Design process knowledge formalization;
2) Exploration of solution spaces enhancing solution optimization;
3) Definition of tool functionalities to support design synthesis in an usable manner.

3   THE APPROACH

The main issue of this work is to support synthesis in the early design phases. Symbolic descriptions are combined with geometric descriptions, which provide overall dimensions, locations and spatial relationships between parts. In order to configure a certain design solution according to the design-to-cost principles and the requirements of the specific application, the proposed system is based on the functional modeling of the product to be designed, the representation of the company design knowledge, and the selection of the best design solution thanks to AI algorithms.

3.1   The research approach

The research activity can be synthetized in the following steps:

1. Identification of the product functional structure;
2. Formalization of the design knowledge, including explicit and tacit knowledge related to the product based on the company best practices as well as the human expertise;
3. Definition of the product function modules and input-output parameters, where dependencies are defined by DSM approach, in particular design activity-based DSM and parameter-based DSM;
4. Definition of a number of CSPs, to capture and process the company knowledge, to suggest effective solutions considering the possible product alternatives with their associated costs;
5. Dimensioning of the product modules and instantiation of the optimized solution.

While the first step is well covered in the literature [21], in [23] the authors proposed an approach to formalize the knowledge of a company in order to be reused to generate offers. The method has been described as a sequence of steps, which moves from the data acquisition of the past projects to the definition of a cost function based on dimensioning parameters. Design process can be represented, e.g. by means an IDEF, as a network of dimensioning activities connected each other through input-output parameters. Historic data as well as the human experience matured during the years allow cost functions to be defined for the main product modules based on design parameters. The conversion of the product requirements into a functional structure is also required. It is advisable to concentrate on product linked by developing a requirements-based functional model. In this way, the requirements are connected to the physical blocks of the product.

The focus of this paper is on the following steps. The approach poses the cost estimation problem as the design of a product structure in response to the formulation of the design problem as a constraint satisfaction problem (CSP) to be solved by AI algorithms. The identified functional groups of the product are firstly configured. The resolution of the constraints is managed by a DSM generated by the dependencies in the structure under development. By means of partitioning and clustering algorithms, the configuration of the whole product is subdivided in a sequence of smaller problems

where the system can support the designer with effective solutions and rapid assessment of their associated costs.

3.2   Configuring the optimal solution

In order to give a correct functional mapping, a database of the product parts, features and constraints has to be set in advance. The product configuration framework is based on a representation of the company knowledge through a DB of products building blocks. The platform includes three different interfaces, which respectively represent the product according to its functions, its modules, and physical components. In this way, it allows designing a new product from a functional viewpoint by defining its functions into modules, and subsequently defining its architecture by connecting the defined functions to physical solutions [22].

The main elements of the approach are:
- The "*Block*", i.e. an instance of a technical solution for a sub-set of the functional structure. The block may be a novel solution or recover already implemented solutions, possibly in parametric and standardized form.
- The "*Blocks Structure*", i.e. a hierarchical tree of blocks capturing the needed product components. Such structure is driven by the functional decomposition and populated by the user in a semi-automatic manner.
- The "*Block Attributes*", i.e. the set of parameters, properties, characteristics of the building units of the designed product.
- The "*Dependencies*" among the blocks of the structure and the attributes. As detailed in [29], dependencies are in the form of structural (such as "attached to" or "contains"), behavioral (such as flows of energy, materials and signals), assignment (such as "comprises") and geometrical (such as positioning). Dependencies also include constraints on the structure such as the required number of a certain component, the compatibility of different parts, the need of a block given the presence of others. When data are available, cost models are defined according to a combined analytic-parametric approach [8].

In order to reuse the company knowledge for the new product design the following strategies are employed:
- A DB of already-implemented technical solutions, characterized in terms of a sub-structure of blocks, attributes and inner dependencies;
- A search function for similar product specifications, filter according to product characteristics, and explore past products with different features and compare;
- A search of past solutions to be compared with the new one by a similarity measure based on the Minkowski formula [24];
- To generate a quotation, parametric cost estimation formulas are included in the blocks definition. After having selected past products, it is possible to adapt them for the new product updating attributes, parameters and geometry. The embodied formulas will provide new cost estimations based on the new modules attributes.

The design process is based on the activities depicted in the Fig. 1. The desired product structure is composed in a hierarchical manner taking blocks from the database. Each block definition includes attributes and rules to determine costs and performances based on the values assumed by the attributes.

The designer adds dependencies among parts and their attributes. Solving tools include:
- Mathematical and logical constraints in the form of formulas among the attributes of the blocks
- Geometrical constraints among the geometric representations linked to the blocks
- Links to external design tools such as spreadsheets and CAE models (FEM, CFD, …) which needs to be run to solve specific design dimensioning activities

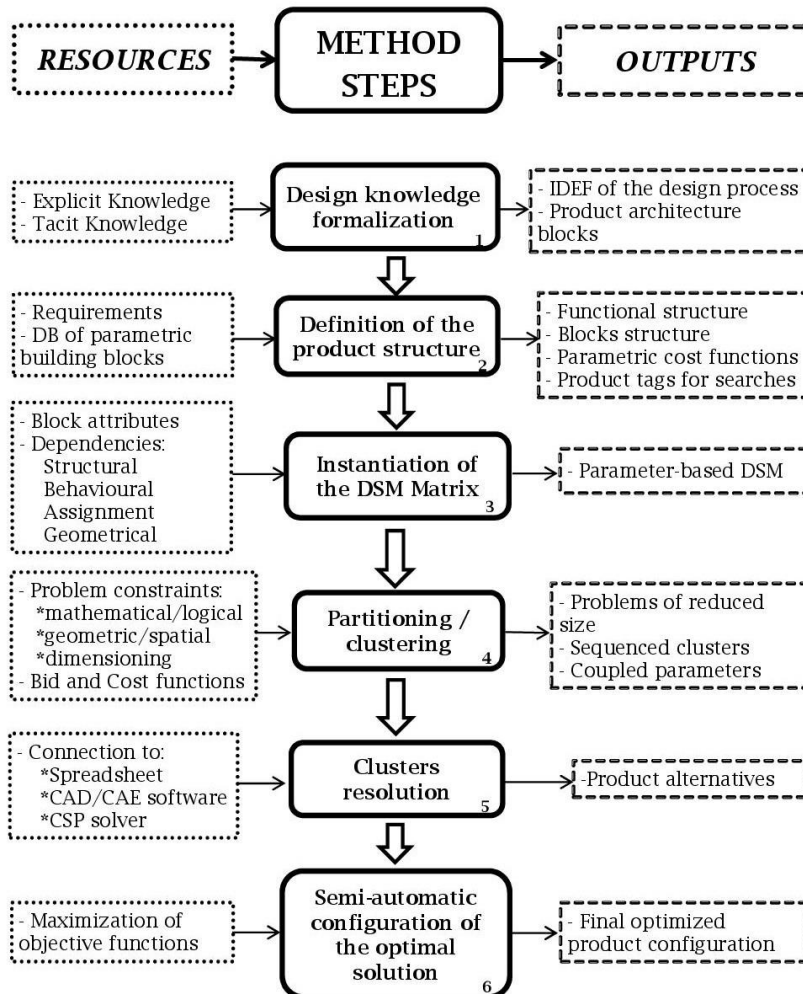- Quantities to be maximised/minimised such as costs, weight, overall dimensions and performances in general.

RESOURCES → **METHOD STEPS** → **OUTPUTS**

RESOURCES	METHOD STEPS	OUTPUTS
- Explicit Knowledge - Tacit Knowledge	**Design knowledge formalization** 1	- IDEF of the design process - Product architecture blocks
- Requirements - DB of parametric building blocks	**Definition of the product structure** 2	- Functional structure - Blocks structure - Parametric cost functions - Product tags for searches

- Block attributes
- Dependencies:
  Structural
  Behavioural
  Assignment
  Geometrical
	Instantiation of the DSM Matrix 3	- Parameter-based DSM

- Problem constraints:
  *mathematical/logical
  *geometric/spatial
  *dimensioning
- Bid and Cost functions
	Partitioning / clustering 4	- Problems of reduced size
		- Sequenced clusters
		- Coupled parameters

- Connection to:
  *Spreadsheet
  *CAD/CAE software
  *CSP solver
	Clusters resolution 5	-Product alternatives

- Maximization of objective functions	**Semi-automatic configuration of the optimal solution** 6	- Final optimized product configuration

Fig. 1: Flowchart of the proposed design procedure.

The introduced dependencies generally constitute an intricate and complex network with an elevate degree of coupling between elements. Such complexity can be identified as the main difficulty a designer is facing when trying to figure out a preliminary solution in the early phases. Since many parameters are mutually dependent, they must be solved together in an iterative manner, which is not feasible in the short time normally available in the conceptual phases.

DSM is then employed to organize design parameters dependencies for a number of reasons:

- Firstly, it allows simple, compact and visually effective representations of complex graph of networks;
- The partitioned DSM shows all the product parameters sequenced according to their dependencies, thus it is possible to know when a parameter must be defined in order to proceed with the design process. The partitioning algorithm groups around the diagonal those attributes that need to be defined in the same design step, as highly coupled. Thus, the

obtained parameters sequence allows minimizing the iterations during the phase of the parameters determination;

- DSM allows dividing the problem into simpler sub-problems of reduced size, which are expected to be much more manageable. This approach is close to the designers' perspective, which is used to dealing with problems of limited size. Relationships between the sub-systems as well as input and output data flow are highlighted and monitored.

Clusterization is based on a simulated annealing optimization procedure derived from the one proposed in [27]. The advance of such approach lies on the capability of controlling the number, size and composition of the clusters based on custom algorithm parameters and functions. In particular, "bid" and "cost" functions are defined in order to formulate respectively an offer for a parameter to be in a cluster and to assess the optimality of a certain configuration of clusters. Such functions are formulated in order to foster the presence in the same cluster of:

- parameters linked by pure mathematical rules, which can be effectively resolved by existing CSM solver;
- geometrical and spatial relationships, which can be effectively solved in parametric CAD environments;
- parameters which are linked by engineering tools introduced in the blocks structure such as spreadsheets or connections to external CAE software.

The sequenced clusters maintain couplings between them. While in each single cluster possible solutions are searched mostly using external tools (e.g., CAD, CSP, Spreadsheets, CAE) the overall process is controlled and managed by the proposed tool in a semi-automatic manner. The values for the attributes connected by the dependencies among different clusters are evaluated following one of the following strategies:

- asking to the user input values coming from his/her knowledge and experience;
- automatically exploring predefined domains, enhancing a backtracking strategy and some objective functions;
- programmed on selected values (DOE-like approach) to restrain the explosion of propagation paths.

In conclusion, the method moves from the identification of sub-problems characterized by a smaller amount of parameters affected by mutual dependencies. Then, it pursue the identification of the minimum number of parameters to be a priori defined in order to support the designer in wizard-like design process.

4   THE INDUSTRIAL CASE STUDY

The proposed approach has been preliminary tested on the "bridge cranes" product family in collaboration with an Italian company that manufactures lifting equipment. For this specific product family, a set of product information was retrieved, organized and classified, i.e. customer specifications, BOM, design documents, 3D models, 2D drawings, and cost data [23].

Bridge cranes are complex products since they are made of numerous parts performing several functions. Although the complexity, a certain amount of modules can be identified by functional analysis. In such a way, the company knowledge was ordered and functionally structured, so that it became easier and more efficient to visualize the product architecture. The Fig. 2 shows an exemplary portion of the crane design process (i.e., the drum and block dimensioning. For instance, the block weight is an output of the block dimensioning, which depends on rope dimensioning, but at the same time provides an input for the rope dimensioning, realizing coupled dependencies to be solved.
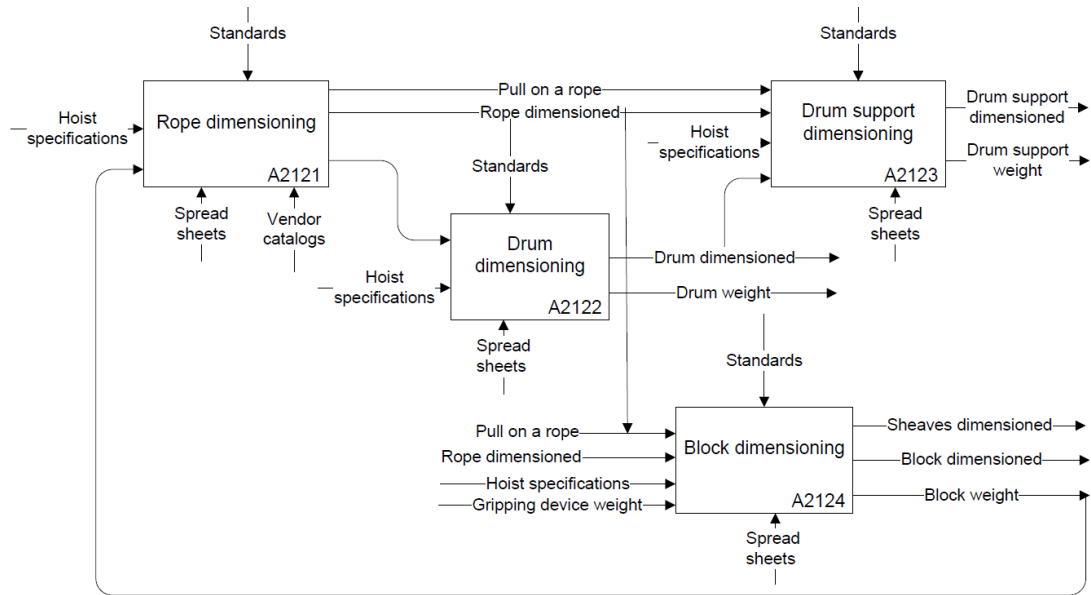
Fig. 2: Portion of IDEF-0 representing the crane design process.

The several specifications and customer requirements make the design always different in some aspects. In order to consider all possible combination of relations and customer requirements, a general functional model for the bridge crane was defined. In particular, a set of building blocks with their relative dimensioning parameters and specific cost functions was created. Fig. 3 shows the result of a preliminary set up obtained with the tool under development.

Fig. 3: The configuration tool interface for parameters-based configuration.

Once the crane structure has been defined, dependencies between modules and parameters have been analyzed and collected in a parameter-based DSM (Fig. 4). The convention used to represent dependencies in this paper is, according to [10], the IC/FBD convention, i.e. DSM feedback marks with inputs shown in columns and outputs in rows. The DSM has been initially reordered by means of partitioning algorithms. Because of the highly coupled problem nature, the partitioned DSM leads to just one big cluster. In order to subdivide the big problem into sub-problems, the clustering algorithm is applied. The resulting clustered DSM shows several blocks of highly coupled parameters, but the dependencies among clusters are much lower.

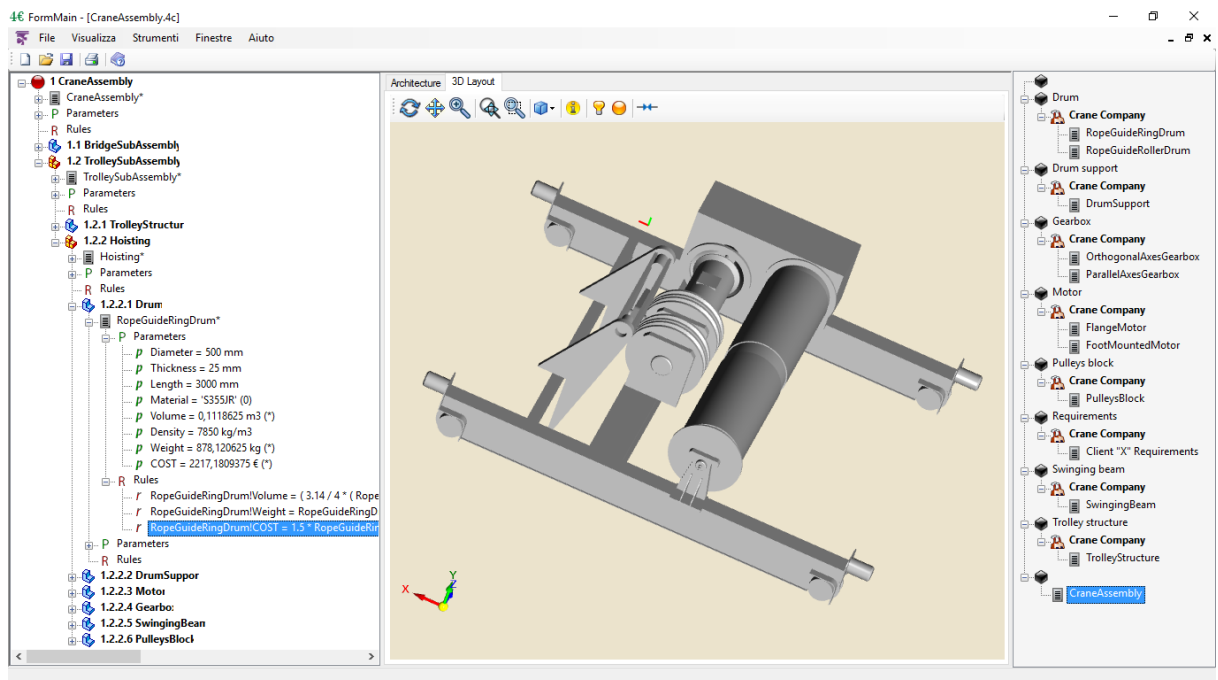| Parameter Name		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Pull on a rope	1	1																														
Rope diameter	2	1	2																													
Hook lift	3			3																												1
Allowable stress	4				4																											
Fleet angle	5			1		5	1																							1		
Drum length selected	6		1	1			6	1																								
Drum diameter selected	7		1					7	1	1																			1			
Allowable angle checking	8					1			8																							
Allowable stress checking	9				1					9	1																					
Von Mises stress	10	1	1				1	1			10	1									1											
Drum thickness selected	11							1	1			11									1											
Drum weight	12						1	1				1	12																			
Hoist speed	13			1										13																		
Braking time	14			1											14																	
Total inertia	15															15	1															
Disc inertia	16																16	1														
Brake code selected	17																	17	1									1				
Minimum torque checking	18																		18	1	1											
Minimum braking torque	19												1	1	1	1				19												
Torque of selected brake	20												1					1			20											
Trolley walking surface	21						1	1														21	1									
Trolley walkways	22																					1	22	1								
Trolley painted surface	23																							23								
Trolley walking weight	24												1									1			24							
Trolley structure weight	25												1												1	25						
Trolley total weight	26																								1	1	26					
Drum axis position	27																				1							27		1		1
Pulleys axis position	28																				1							1	28	1		1
Swinging beam position	29																													29		1
Pulleys diameter	30			1																											30	
Hook approach	31			1			1	1																				1	1	1		31

Fig.: 4: Portion of the clustered DSM.

Then, each matrix cluster is treated as an individual problem, solved independently form the others. Attempt values are assumed for the attributes connected to other clusters. Two exemplary clusters of coupled parameters identified in the DSM are here reported with more details.

4.1 Drum dimensioning

A cluster of 12 coupled parameters resulting from the clustered DSM is related to the dimensioning of the drum, a part of the crane hoist system which is responsible of rope winding.

The parameters of the block include dimensions, weight and stress of the drum due to the pull on the rope. Moreover, they also comprise relations with other clusters, such as the crane requirements: i.e. the rope diameter must ensure the support of the maximum load, and the drum length must be enough to ensure the hook lift. Checking parameters (True/False values) includes the verification of the following inequality constraint:

- *equivalent drum stress ≤ allowable stress*
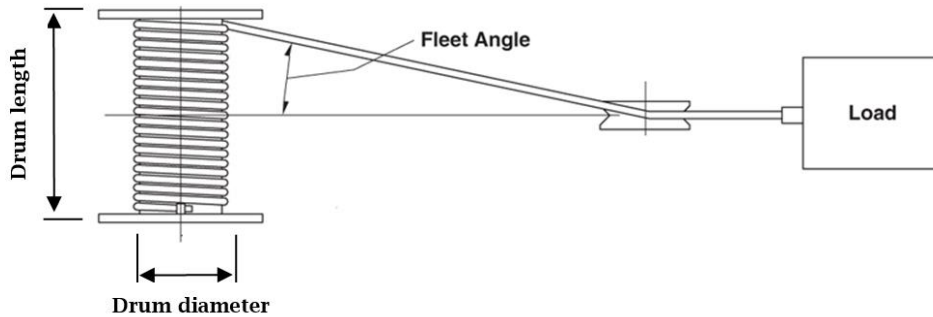- *fleet angle ≤ allowable fleet angle*

Fig. 5: The hoisting drum.

Stresses are calculated according to the applicable standards and fleet angle results from a simple trigonometric formula (Fig. 5). The partitioning of the coupled block leads to the DSM in Fig. 6(a). Although the parameters assignment can be made by a CSP integrated-optimization tool, the need of two a priori choices can be demonstrated. In fact, by removing the dependencies only for one parameter at a time, the condition of triangularization is not yet satisfied. Instead, searching for pairs of parameters, several alternatives have been found. Discarding the pairs containing verification parameters, the solving iteration gives as a possible solution to the *Drum Diameter* and *Drum Thickness*. The Fig. 6(b) shows the DSM partitioned after removing dependencies by this specific pair of parameters.

PARTITIONED DSM (a)

	Pull on a rope	Rope diameter	Hook lift	Allowable stress	Fleet angle	Drum length selected	Drum diameter selected	Allowable angle checking	Allowable stress checking	Von Mises stress	Drum thikness selected	Drum weight
Pull on a rope	1											
Rope diameter	1	2										
Hook lift			3									
Allowable stress				9								
Fleet angle			1		10	1						
Drum length selected		1		1		6	1					
Drum diameter selected		1					4	1	1			
Allowable angle checking					1			11				
Allowable stress checking				1					12	1		
Von Mises stress	1	1				1	1			8	1	
Drum thikness selected							1	1			5	
Drum weight						1	1			1		7

PARTITIONED DSM (b)

	Pull on a rope	Rope diameter	Hook lift	Drum diameter selected	Drum thikness selected	Drum length selected	Drum weight	Von Mises stress	Allowable stress	Fleet angle	Allowable angle checking	Allowable stress checking
Pull on a rope	1											
Rope diameter	1	2										
Hook lift			3									
Drum diameter selected				4								
Drum thikness selected					5							
Drum length selected		1		1	1	6						
Drum weight				1	1	1	7					
Von Mises stress	1	1		1	1	1		8				
Allowable stress									9			
Fleet angle		1				1				10		
Allowable angle checking										1	11	
Allowable stress checking									1	1		12

Fig.: 6: Partitioned DSM of the hoist drum before (a) and after (b) tearing.

The objective function to minimize the drum weight can be represented as follows:

f obj = Min (Weight)

This condition allows selecting among the space of solutions the best alternative in terms of cost, which is approximately dependent from the weight.

4.2 Trolley layout selection to comply with the overall dimensions

As stated before, many design problems cannot be formulated in the form of simple mathematical formulas and resolved by a CSP solver as in the previous example. Sometimes the problem is largely dependent on geometrical constraints and simplified parametric layouts are necessary to represent the problem. For instance, the trolley layout design (Fig. 7) includes the arrangement of components, such as the hoisting units, traversing units and trolley structure, depending on constraints given by customer specifications regarding the overall dimensions and the working area reachable by the hook.
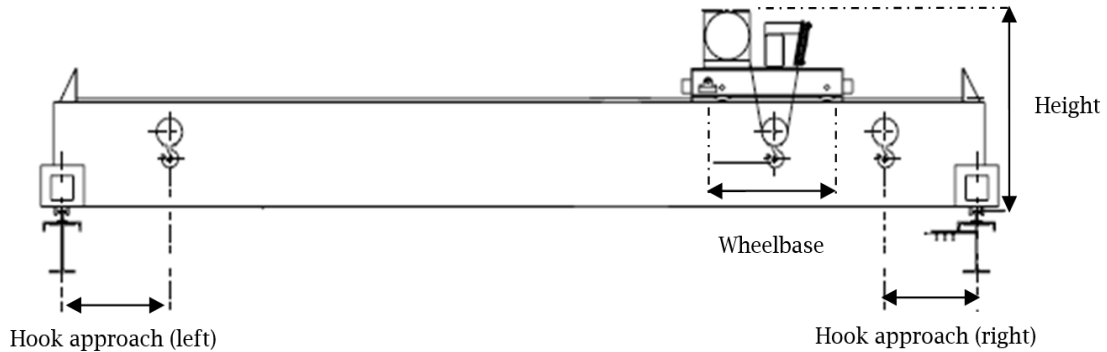
Fig.: 3: Typical crane layout.

A cluster of 13 coupled parameters has been identified within the trolley layout dependencies, but the partitioned DSM in Fig. 8(a) shows that only ten parameters are affected by mutual dependencies.

Applying the method, it has been found that 4 parameters need assumptions: 3 of them come from the resolution of relative clusters, i.e. the above cited *hoisting unit, traversing unit* and *trolley structure*; *layout selection* is the last attribute to be necessarily chosen to solve the loop. Fig. 8(b) shows the partitioned DSM after removing these backward dependencies, in order to eliminate coupled parameters. In fact, if the overall dimensions arising from the modules arrangement do not satisfy the constraints, probably the modules dimensioning needs a review: this iteration reflects what actually happens during the design process.

Fig. 4: Partitioned DSM of the trolley arrangement before (a) and after (b) tearing.

Since the designer expertise plays a determinant role in the layout assessment, the problem cannot be solved by an optimization tool but requires the human intervention. Thus, the user shall define a DOE table containing the solutions to explore and evaluate the response for each experiment. A regression

model would allow finding the optimal solution in the explored space, e.g. minimizing the hook approach.

It is worth pointing that most of the parameters belonging to this cluster depend on parameters in other clusters. The supporting tool will check for inconsistencies between parameters of different clusters. If the solver detects some inconsistencies, the values of the parameters must be modified. For example, the presented case of study has required a new dimensioning of the drum to meet the required hook approach. In particular, drum diameter has been decreased to reduce the trolley wheelbase, providing the trolley to get closer to the limits of the working area.

4.3   Results

The crane modules (i.e., motor, gearbox, hook, beams) have added in the software system along with a 3D geometry, identified parameters, relations and dimensioning rules. The system has been tested by the collaboration with designers from a partner company, after a short training about the main system functionalities. In particular, the requirements of a new crane have been assigned to few designers of different expertise. They have input the specifications in the systems and started the configuration process retrieving from the system DB the required functional modules. Parameters have been varied according to the rules in the sequence suggested by the system. Thus, designers have been guided to complete a preliminary design (Fig. 9). Surprisingly, the final solutions were very similar and cost-effective, confirming the capability of the approach of supporting also novices.

Finally, the users have highlighted some difficulties in the use of the system due to minor malfunctions or complex commands, which needs to be improved in the future.
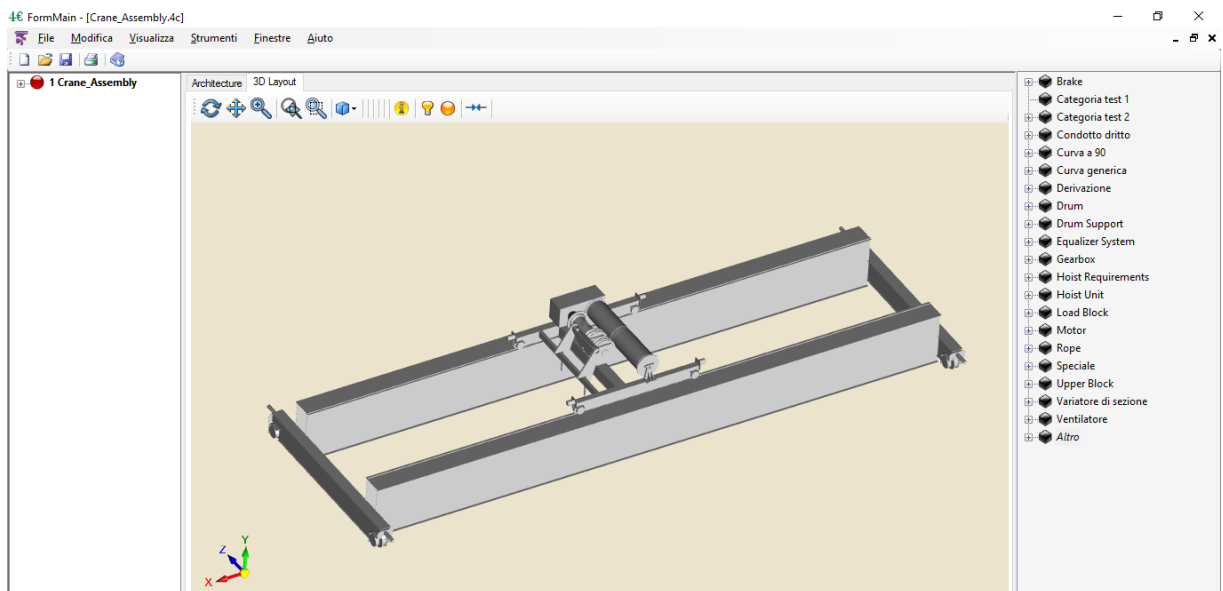

Fig. 9: Simplified 3D result of the configuration process of a double beam crane

5   CONCLUSIONS

The presented research focuses on the definition of a design tool to support the configuration of industrial products according to the design constraints and costs. Such a tool can be used from the conceptual stages to identify the best design solutions according to the customer requirements contained in a request for proposal.

In particular, this work presents an approach to face with highly coupled problems, which emerge as soon as preliminary architectures of configurable blocks are proposed. It makes a combined use of

clustering and partitioning algorithms applied to a parameter-based DSM in order to identify clusters dividing the whole problem into an ordered sequence of sub-problems easier to be solved, also making use of AI tools.

The main contribution of the paper is a method to tackle the rapid definition of an optimal design solution in terms of preliminary layout and dimensioning accompanied by automatic cost and performance evaluations. From one side, the approach is prone to be implemented on a software tool, activity that is currently under development. On the other side, it resembles the way of operating of an experienced design enhancing the speed of automatic evaluation. It also provides a guided wizard procedure, which is beneficial especially for novice designers, as confirmed by the preliminary experimentation in the crane design field. More extended experimentation is planned for the next future. Some possible test cases from different fields have already been collected and will be described in future works along with an assessment of the approach advantages and weaknesses.

Margherita Peruzzini, http://orcid.org/0000-0003-2260-0392
Mauro Pistacchi, http://orcid.org/0000-0001-5265-7117
Andrea Savoretti, http://orcid.org/0000-0001-5098-8641
Marco Mandolini, http://orcid.org/0000-0003-0962-5982
Roberto Raffaeli, http://orcid.org/0000-0003-0301-454X

REFERENCES

[1]   Black, T. A.; Fine, C. H.; Sachs, E. M.: A method for systems design using precedence relationships: An application to automotive brake systems, Sloan School of Management, Massachusetts Institute of Technology, 1990.
[2]   Boothroyd, G., Dewhurst, P., Knight, W.: Product Design for Manufacture and Assembly, Marcel Dekker, New York, 1994.
[3]   Browning, T. R.: Applying the Design Structure Matrix to System Decomposition and Integration Problems: A Review and New Directions, IEEE Transactions on Engineering Management, 48 (3), 2001, 292-306.
[4]   Chakrabarti, A.; Bligh, T. P.: An approach to functional synthesis of mechanical design concepts: theory, applications, and emerging research issues, Artificial Intelligence for Engineering, Design, Analysis and Manufacturing, 10 (04), 1996, 313-331.
[5]   Chakrabarti, A.; Shea, K.; Stone, R.; Cagan, J.; Campbell, M.; Hernandez, N. V.; Wood, K. L.: Computer-Based Design Synthesis Research: An Overview, Journal of Computing and Information Science in Engineering, 11(2), 2011, 021003. https://doi.org/10.1115/1.3593409
[6]   Chandrasegaran, S. K.; Ramani, K.; Sriram, R. D.; Horváth, I.; Bernard, A.; Harik, R. F; Gao, W.: The evolution, challenges, and future of knowledge representation in product design systems, Computer-Aided Design, 45(2), 2013, 204-228. http://dx.doi.org/10.1016/j.cad.2012.08.006
[7]   Cicconi, P.; Germani, M.; Bondi, S.; Zuliani, A.; Cagnacci, E.: A Design Methodology to Support the Optimization of Steel Structures. Procedia CIRP, 50, 2016, 58-64.
[8]   Cicconi, P.; Germani, M.; Mandolini, M.: How to Support Mechanical Product Cost Estimation in the Embodiment Design Phase, in Advanced Concurrent Engineering, Proc. 17th ISPE International Conference on Concurrent Engineering, CE 2010, Cracow, Poland; 6-10 September 2010, 419-430. http://dx.doi.org/10.1007/978-0-85729-024-3_45
[9]   Ehrlenspiel, K.; Kiewert, A.; Lindemann, U.: Cost-Efficient Design, ASME Press, 2007.
[10]   Eppinger, S. D.; Browning, T. R.: Design structure matrix methods and applications, MIT press, 2012.
[11]   Eppinger, S. D.; Whitney, D. E.; Smith, R. P.; Gebala, D. A.: Organizing the tasks in complex design projects. In: Sriram D., Logcher R., Fukuda S. (eds) Computer-Aided Cooperative Product Development. WCACPD 1989. Lecture Notes in Computer Science, vol 492. Springer, Berlin, Heidelberg, 1991, 229-252
[12]   Gorti, S. R.; Sriram, R. D.: From symbol to form: a framework for conceptual design, Computer-Aided Design, 28(11), 1996, 853-870. http://dx.doi.org/10.1016/0010-4485(95)00088-7
[13]   Kolodner, J. L.: Case-Based Reasoning, Morgan Kaufmann, California, 1993

[14]   La Rocca, G.: Knowledge based engineering: Between AI and CAD. Review of a language based technology to support engineering design, Advanced Engineering Informatics, 26, 2002, 159-179. http://dx.doi.org/10.1016/j.aei.2012.02.002

[15]   Layer, A.; Brinke, E.; Houten, F.; Kals, H.: Recent and future trends in cost estimation, International Journal of Computer Integrated Manufacturing, 15(6), 2002, 499-510. http://dx.doi.org/10.1080/09511920210143372

[16]   McMahon, C.; Lowe, A.; Culley, S.: Knowledge management in engineering design: personalization and codification, Journal of Engineering Design, 15, 2004, 307–325. http://dx.doi.org/10.1080/09544820410001697154

[17]   Montgomery, D. C.: Design and analysis of experiments, John Wiley & Sons, 2008.

[18]   Mullineux, G.; Hicks, B.; Medland, T.: Constraint-aided product design, Acta Polytechnica, 45 (3), 2005, 31-36.

[19]   Münzer, C. H. W.: Constraint-Based Methods for Automated Computational Design Synthesis of Solution Spaces, Ph.D. Thesis, Technische Universität München, Zurich, 2014.

[20]   Negnevitsky, M.: Artificial Intelligence – A Guide to Intelligent System, Addison-Wesley, Harlow, England, 2005.

[21]   Otto, K.; Hölttä-Otto, K.; Simpson, T.W., Krauser, D., Ripperda, S., Moon, S. K.: Global Views on Modular Design Research: Linking Alternative Methods to Support Modular Product Family Concept Development, Journal of Mechanical Design, 138(7), 2016, 071101. http://dx.doi.org/10.1115/1.4033654

[22]   Peruzzini, M., Raffaeli, R., Mandolini, M.: Towards a design-to-sustainability platform based on functional representations and simplified geometric layouts, Computer-Aided Design and Applications, 14(3), 2017, 301-312. http://dx.doi.org/10.1080/16864360.2016.1240451

[23]   Raffaeli, R; Savoretti, A.; Germani, M.: Design knowledge formalization to shorten the time to generate offers for Engineer To Order products, Advances on Mechanics, Design Engineering and Manufacturing: Proceedings of the International Joint Conference on Mechanics, Design Engineering & Advanced Manufacturing (JCM 2016), Springer, Catania, 2016, 1107-1114.

[24]   Ramesh, M.; Yip-Hoi, D; Dutta, D.: Feature-based Shape Similarity Measurement for Retrieval of Mechanical Parts, Journal of Computing and Information Science in Engineering, 1(3), 2001, 245-256. http://dx.doi.org/10.1115/1.1412456

[25]   Shehab, A.; Abdalla, E.: A Cost Effective Knowledge-Based Reasoning System for Design for Automation, Journal of Engineering Manufacture, 220(5), 2006, 729-743. http://dx.doi.org/10.1243/095440554JEM298

[26]   Stone, R. B.; Wood, K. L.; Crawford, R. H.: A heuristic method for identifying modules for product architectures, Design Studies 21, 2000, 5–31. https://doi.org/10.1016/s0142-694x(99)00003-4

[27]   Thebeau, R. E.: Knowledge management of system interfaces and interactions for product development processes, PhD Thesis, Massachusetts Institute of Technology, 2001.

[28]   Wierda, L. S.: Linking Design, Process Planning and Cost Information by Feature-based Modelling, Journal of Engineering Design, 2(1), 1991, 3-19, http://dx.doi.org/10.1080/09544829108901667

[29]   Wyatt, D. F.; Wynn, D. C.; Jarrett, J. P.; Clarkson, P. J.: Supporting product architecture design using computational design synthesis with network structure constraints, Research in Engineering Design, 23(1), 2012, 17-52.