



A modelling framework to support integrated design of production systems at early design stages

Shiva Abdoli¹

Received: 17 February 2022 / Accepted: 7 July 2022 / Published online: 8 August 2022
© The Author(s) 2022

Abstract

Multidisciplinary, large scale, and dynamic essence of production-logistic systems make their design knowledge complex. As a result, designers from different disciplines mostly design these systems with sequential approaches. This does not address the impact of single design decisions on overall system performance, which may lead to inconsistencies between different disciplines or failures. This paper aims to realise the integrated design of such systems by introducing a framework that incorporates Systems Engineering and Object-Oriented methods to develop a model that holistically embodies design knowledge of such systems. This model is constructed in Finite-State-Machine formalism to achieve an executable architecture and integrated with optimization models to allow simulation of alternatives and to observe the impact of design decisions on system behaviour. Supportive algorithms are introduced for refinements of design alternatives according to the simulation results. A fuzzy assessment approach is introduced to also assess the alternatives against qualitative criteria. The framework integrates simulation and fuzzy assessment results and performs a multi-criteria assessment to select an alternative for the detailed design. Therefore, the framework can stand as a decision support framework at early design stages, giving insights to designers about the impact of single design decisions on system overall performance and satisfaction of various objectives.

Keywords Design · Production systems · Modelling · Decision support systems

1 Introduction

Performance of assembly, production, and logistic systems are strongly dependent on the decisions made at their early design stages [1–3]. These systems operate based on dynamically interacting processes using soft/physical resources. Such systems are considered complex due to having large-scale and multidisciplinary design knowledge (design knowledge refers to the available information after requirement analysis) [4]. In this paper, the described systems are called ‘complex engineering process-based system’ (CEPS).

Some problems in the design of CEPSs are stated as aimed to address them by the introduced approach in this paper.

- 1- Inconsistency between different design disciplines
- 2- Incomprehensive design assessment/validation due to not having a holistic-integrated design approach.

- 2-1 Assessment without considering the impact of dynamic interactions on a CEPS’s performance.
- 2-2 Assessment against either qualitative or quantitative criteria and not both concurrently
- 3- Domain specific (not generic) approaches

Four overall phases can be defined in the design of CEPSs, requirements definition, conceptual design, configuration design, and detail design. It is not very easy to draw a line between design phases. The early-stage design, which is the scope of this paper, refers to the conceptual design stage.

The complex structure of CEPSs usually makes the designers to design different aspects of a CEPS sequentially [5]. As a result, the design at a higher-level is mostly finalized before proceeding to the next level [6]. In this context, operational policies are mostly designed after the physical design is finalized. For example, in designing a warehouse, designers decide on narrow aisles to reduce the space cost. This does not allow storing the new and picking the existing items in the same aisle concurrently (requires wide aisles). Concurrent storing-picking is an operational policy that is called double-command [7]. The operation time in the double-command

✉ Shiva Abdoli
s.abdoli@unsw.edu.au

¹ University of New South Wales, Sydney, NSW, Australia

process is mostly shorter than single-command, so fewer operators may deliver the same throughput. Therefore, a wide-aisle warehouse with the double-command-mode can lower the total cost. Generally, sequential approaches (not integrated) do not completely reflect the impact of single decisions on overall performance due to not having a holistic approach in addressing all types of design requirements simultaneously.

The performance of a CEPS is a function of the dynamic interactions between its resources, which can hardly be formulated with analytical methods (e.g., optimization). Therefore, it is necessary to use methods, such as simulation, that can capture the dynamic aspect of a CEPS to assess the impact of design decisions on its performance. However, simulation is often used at later design stages when important decisions are already made [8]. Integrated design in this paper means assessing the impact of design decisions on a CEPS's performance as a function of the dynamic interactions between its resources.

Design of a CEPS is mostly subjected to multiple objectives satisfaction; quantitative (e.g., cost) and qualitative (e.g., safety). Optimization is the most applied approach for design assessment against quantitative objectives. However, the latter does not fit well with optimization methods [9]. The different nature of conflicting objectives further complicates the assessment/selection of better design alternatives. These problems are magnified due to the gap between academic approaches and what practitioners exercise [10]. Companies may develop specific approaches, which are not generic enough to be applicable in other domains.

A set of objectives are stated and their satisfaction by application of the introduced approach manifests the contributions of this work to this research context.

I. Enabling integrated design by.

- a. holistically addressing all types of design requirements
- b. observing the design decisions' impact on the system's dynamic behaviour.

II. Enabling multidisciplinary design to achieve design consistency.

III. Enabling multi-aspect design assessment by supporting the decision-making regarding the selection of better alternatives by considering different types of performance criteria.

IV. Having a generic design approach such that the framework is applicable in different domains.

2 Literature review

Designers devise models as their artefacts to analyze a specific design knowledge and set up a solution [11]. In CEPS

design, modelling can serve three purposes; developing system architecture, design assessment, and validation [12]. System architecture embodies the system structure and function. Validation establishes evidence to assure an alternative accomplishes objectives [13]. This section reviews those research works that developed a modelling approach for CEPS design and did not focus on one modelling aspect of CEPSs, as shown in Table 1.

2.1 System architecting and validation

Different studies [14, 15, 17, 24, 26] used various methods for system architecting such as Object-Oriented (OO) or Function-Modelling (FM), but provided limited architecting guidelines.

The following works [14, 15, 24] used simulation-based validation while developing the architecture with languages, such as SysML/UML, which have simulation limitations [32]. Object-Process-Methodology (OPM) formalism offers simulation capabilities but has shortcomings in capturing system's dynamic state at a specific time. Although, it is tried to bridge this gap by Model-to-Model transformation (MtM), working with transformation engines requires high-level programming skills [17]. Thiers [7] introduced a design methodology to support the analysis of logistics systems by using model-based systems engineering. An abstract model of a token flow network was developed to realize the MtM transformation between SysML and Petri Net (PN). However, the full automation of MtM transformation was not achieved. Moreover, no explicit approach for system architecting was suggested. The methodology could not address different control and planning policies in one analysis model and suggested modelling the system behaviour in separated behavioural diagrams.

Koo developed a modelling language for system architecting, named Object-Process Network (OPN) [27]. OPN was based on the core concepts of PN and intended to support system architecting by automating communication and computational tasks in architectural reasoning. However, OPN had shortcomings in providing explicit mechanisms for automatically generating alternatives and modelling the constraints that were related to the integration of a system's entities. Moreover, OPN was not an intuitive language and had shortcomings in representing the static relationships between a system's entities.

Meng [14–16] introduced an approach to model a Reconfigurable Manufacturing System (RMS) by application of coloured timed object-oriented Petri Nets. The approach used the OO method to identify the elements in an RMS and suggested four classes as basic elements: material processing equipment, storage equipment, robot, and Automatic Guided Vehicle (AGV). From the OO and domain modelling perspective, the defined classes were not comprehensive enough

Table 1 Classification of modelling approaches

Approach	System architecting				Quantitative assessment		Qualitative assessment		Validation		MtM
	OO	Not defined	FM	Axiomatic design	Optimization	Fuzzy logic	Fuzzy logic	Formalism	Simulation		
UML, SysMI	[14–16]		[17]		[16]	[18]	[19–23]	Petri Net	[14–16, 24]	[14, 16, 24]	
OPM	[24]	[25]			[24]					[25]	
Not defined	[26, 27]			[28, 29]			[30]	FSM	[31]		

to illustrate a generic RMS. Generally, all RMSs do not use AGV and robots. Moreover, operational procedures were not addressed in that modelling approach. Also, it was not properly indicated how to formulate the problem to achieve a rigorous system configuration, such as equipment number. The presented work basically used OO at the solution level after the system architecture was finalized.

Silva and Alves [33] introduced a methodology for manufacturing systems design that produce a family of similar products. The methodology consisted of three phases: generic design, conceptual design, and detail design. This methodology was descriptive and did not suggest application of any method for system architecting, formulating solutions, and analysing the results.

Existing works mostly modelled operational policies as a fixed part of models, so simulation did not clarify the impact of different policies on performance. Abdoli and Kara [31] introduced architecting guidelines by incorporating the OO and Finite-State-Machine (FSM) methods to develop an executable system architecture (explained shortly).

2.2 Design assessment

Some works addressed quantitative and some qualitative assessment. Some studies [28, 29] used axiomatic design (AD) theory for architecting and assessment. However, AD gives less attention to design validation [34]. Cochran et al. [28, 29] introduced a methodology for designing manufacturing systems by application of AD, called: Manufacturing System Design Decomposition (MSDD). Return on investment was considered as the primary Functional Requirement (FR-1). Design Parameter-1(DP-1) was defined as ‘Manufacturing system design’ to fulfil FR-1. The next levels’ FRs were defined as measures of performance to maximize revenue. DPs were associated with FRs at each level. MSDD mainly provided the means to assess the design alternatives and did not address the design validation.

Dauby and Dagli [18] proposed an assessment approach by defining ‘canonical design primitives’ as possible genres for physical components and fuzzy logic was used to assess the impact of switching between primitives. That work only addressed the quantitative and not qualitative performance criteria. Moreover, the impact of interactions between design primitives on the system’s performance was not addressed. Kulak [30] used fuzzy logic for equipment selection, but the impact of equipment interactions with other system elements was not considered. Sasaki and Gen [35] used fuzzy logic in optimization formulation, in which the objectives and constraints were modelled as fuzzy variables. The research used GA as the search algorithm and encoded a chromosome that represented the generalized upper bounding for fuzzy objectives and variables. That approach only addressed the quantitative objectives. Moreover, that approach did not address the associated uncertainty with design choices.

Singh and Dagli [19] and Pape et al. [21] introduced similar approaches for evaluating system alternatives, called System of Systems (SoSs). The individual systems could decide whether to participate in an SoS or to make an interface with other individuals. Therefore, different SoS alternatives varied in the participation of individual systems or in developed interfaces. The performance characteristics of an SoS were defined by qualitative criteria, such as affordability. In the CEPS design context, it is needed to allocate a design option to each design requirement (such as machining equipment). Therefore, the decisions of participation or making interfaces were not equivalent to developing different alternatives in the CEPS design context. Other similar works assumed that individual systems might have priorities that affect their choice of participation [22, 23]. Accordingly, individual systems could negotiate with an SoS manager regarding deadlines and funding. The individual systems and SoS manager were modelled as agents and a fuzzy decision analysis was developed to perform the negotiations. Although, this approach resulted in a more comprehensive assessment, still had similar shortcomings to previously

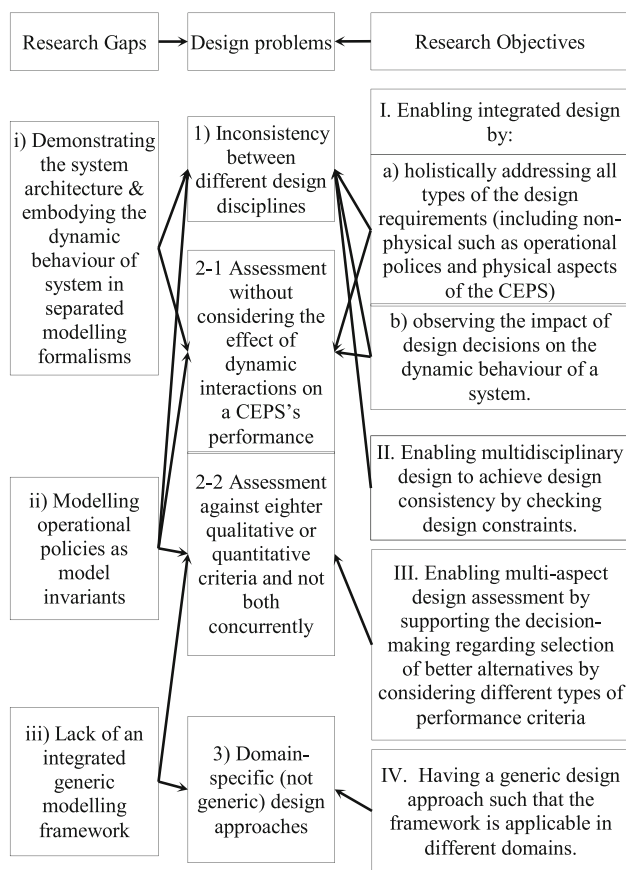


Fig. 1 Relation between problems, research objectives, and gaps

described work. Singh et al. [20] used fuzzy-Analytical Hierarchy Process (AHP) for decision-making on selection of manufacturing system types among; reconfigurable, dedicated, and flexible. Fuzzy logic was used to model the ratio scales because AHP has shortcomings in capturing the tied uncertainty into subjective judgments [36].

Some studies developed decision support frameworks for the selection of the most environmentally friendly supplier among existing ones by considering their logistic activities and processes performed to select the supplier that leads to a less environmental impact [36, 37]. These works compared different strategic options and did not address the system design in terms of configuring different options to develop and assess system alternatives.

2.3 Research gap

The following shortcomings are identified in reviewed works [12]. Figure 1 demonstrates how the research gaps led to a lack of practical design frameworks and consequently led to explained problems in design practices, shown on the left-hand side of Fig. 1. The right-hand side of Fig. 1 shows how the realization of objectives contributes to addressing the

practical design problems because this paper aims to present a framework that is practical and can be used in real design applications.

- i. Demonstrating the system architecture and embodying dynamic behaviour in separated modelling formalisms
- ii. Modelling operational policies as model invariants
- iii. Lack of an integrated generic modelling framework that: 1) addresses different aspects of modelling at early design stages, 2) provides a systematic interconnection between different models to interchange information.

Besides a recent work by Abdoli and Kara [31], the reviewed works realize some modelling aspects of CEPS design but not all in an integrated manner [18]. However, integrating different models enhances the consistency of the exchanged data/information [38]. Algorithmic approaches have not been used efficiently, while they can help in performing repetitive tasks [24]. The design process is knowledge propagation from one model to another. Thus, an integrated modelling approach can overcome the design methodologies' shortcomings that mainly guide the design process without generating solutions. Different modelling approaches answer narrowly defined design questions. Sequential design approaches also hardly integrate various design stages or capture the connection between low-level design decisions and high-level system objectives [39]. This highlights the importance of having an integrated design approach [40].

Model-based design is a way of formalization by defining the semantics of representing something before describing it. Model-based design relaxes the application of framework from the CEPS domain, assuring it is domain neutrality [16].

There is a lack of a generic modelling approach providing prescriptive guidelines for system architecting, multi-aspect assessment, and validation, which can assist in the integrated design of CEPSs [16, 41].

Modelling is both art and science, which conceptual modelling lying more at the artistic end [42]. Two designers may use one modelling approach and resultant models can be different with different capabilities. A prescriptive approach reduces the modelling effort and supports the design process more effectively [43]. The proposed framework utilizes different methods and supporting algorithms and systematically integrates them. The framework prescribes how to develop models (by introducing novel approaches for methods' application) and algorithms. Each model/algorithm serves a specific role in this framework. Systematic interconnection between models and algorithms allows a smooth inter-operability between them to serve framework objectives. Contributions of the proposed framework rely on how the methods are employed and the systematic models' integration. Each of the applied methods/formalisms may (may not) have been used before for other narrowly defined design

purposes, yet the prescribed approaches in this framework are novel and contribute to satisfying research objectives.

3 Framework structure and applied methods

An overview of the framework and justification of methods selection is provided in this section. This research used Systems Engineering (SE) for defining the framework structure, in terms of its tasks and their logical sequence, due to the multidisciplinary nature of CEPSs. Because SE encompasses interdisciplinary activities and concentrates on system properties rather than on single disciplines [44]. In SE process, after requirement analysis, the system architecture is developed [44], which is equivalent to design at early stages. In the trade study step, design alternatives are configured, their performance is studied, and one is nominated for detailed design. V-model of SE gives high attention to design validation from early stages, conforming to the importance of observing the dynamic behaviour at early stages by interconnecting the integration-validation phase with architecture design. Hence, the proposed framework addresses: architecture design, validation, and trade study, see Fig. 2.

The framework develops an executable architecture for observing the behaviour (dynamic performance) of alternatives by simulation. However, design is a process of design generation, evaluation, and redesign [45]. The proposed trade study mechanism serves a multitude of alternative generation, design constraints realization, and assessment. The OO modelling is a well-known approach for designing complex systems with multi-hierarchical structures [46]. Hence, the framework uses the OO method and architecting guidelines are developed for capturing the complexity of holistically modelling CEPSs.

In the framework, the validation happens at the system level. Discrete-Event-Simulation (DES) is a promising method for validation of the dynamic aspects of CEPSs [40]. PN is a widely used formalism in this context. However, a PN-model structure can look complicated even for small-scale systems [14, 47]. The FSM represents system structure by dynamic states of system elements. Hence, the framework maps the mentioned ‘architecting guidelines’ into FSM formalism to develop an executable architecture.

Design knowledge database in this framework stores design options with their properties, called design-objects. ‘Alternative generation’ algorithm configures the design alternatives by allocation of design-objects to design requirements while the ‘Feasibility checking’ algorithm crosses off the infeasible alternatives.

The framework uses optimization/quantification models to develop a numerical model for alternatives, which are later simulated for validation. The framework uses Linear-Programming (LP) and Automated Layout Design Program

(ALDeP). The latter is only used if layout design is needed. Other optimization methods can also be used because the analytical formulations of CEPSs belong to NP-hard problems and no optimization method can guarantee to find the global optimum [48]. The proposed framework used LP because of its broad application field including scheduling, and resource allocation. For layout development, CORE or CRAFT can also be used instead of ALDeP.

Many studies argue that qualitative properties are more critical for a system’s lifetime value [49]. Yet, qualitative aspects (e.g., safety) are not very easy to be assessed by optimization models due to the inherited uncertainty in qualitative assessment [50]. In the literature, combinations of qualitative assessment methods are used such as fuzzy-AHP or fuzzy-Quality Function Deployment to assess the qualitative aspects of a CEPS [51]. However, fuzzy logic played the key role in capturing the uncertainty in qualitative assessment. The framework includes a fuzzy assessment approach to assess the alternatives against qualitative objectives.

This framework uses simulation results as one and fuzzy assessment as another source of information in the trade study step. A feedback algorithm is introduced to modify an alternative according to the simulation results. TOPSIS (Technique for Order of Preference by Similarity to Ideal Solution) is widely used in Multi Criteria Decision Making problems because it allows different criteria have their own measuring units. TOPSIS also considers different importance for criteria. This framework uses TOPSIS to rank the alternatives and identify the final alternative that demonstrates better performance by considering simulation and fuzzy assessment results. The framework is implemented in MATLAB, however other platforms can be used too.

Examples of a warehouse design are given in the following sections to explain the methods’ applications and the connection between different artefacts in the framework. However, this does not impact the domain neutrality of the framework, as its application is presented in two case studies from different domains later in the paper.

4 Developing an executable system architecture

Abdoli and Kara [31] introduced architecting guidelines using the OO method for holistically modelling the architecture of a CEPS (decomposition hierarchy). Accordingly, a CEPS is decomposed into processes that deliver the main function of a CEPS, as the transformation of an item’s state from input to output. For instance, warehousing processes can be defined as; receiving, storing, picking, and shipping [52]. Each process can have any type of physical/operational enablers with their Design Requirements (DRs). The architecture is constructed in FSM formalism

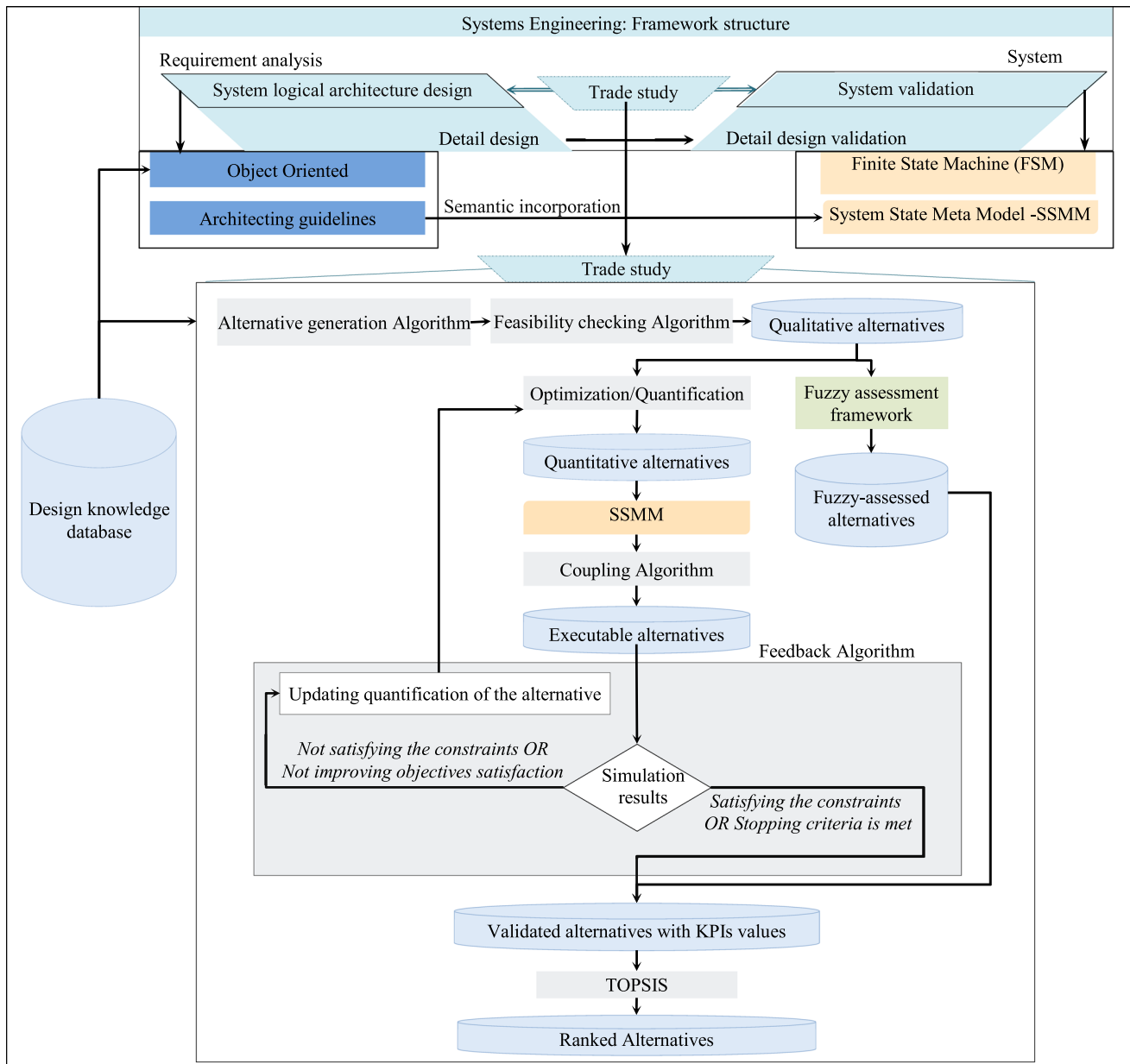


Fig. 2 The overall structure of the framework with applied methods and their data exchange flow

to achieve an executable architecture and is named System-State Meta-Model (SSMM). These architecting guidelines are briefly explained here and interested readers are referred to this reference [31] for details.

The OO-based hierarchal decomposition approach is mapped into FSM formalism by embodying nested states in SSMM, an example is shown in Fig. 3. In this approach, a CEPS is firstly modelled as a parent state in FSM and then is decomposed into sub-states to embody the CEPS's processes, called process-state. Each process-state is decomposed into further sub-states that represent its needed enablers, called enabler-state. Further, each enabler-state is decomposed into

sub-states, which address its specific DRs, called DR-state. The possible design options/genres for each DR-state are modelled as its sub-states, called object-state. The developed SSMM in FSM realizes a holistic design approach by embodying all types of DRs. For the warehouse case, enablers of storing process are defined as operational policy, equipment, and infrastructure. For instance, it is a DR to define an operational policy for allocating the Stock Keeping Units (SKUs) to storage modules. Hence, 'SKU-Allocation' is defined as a DR-state, which includes two sub-states showing two possible operational policy options: 'Class-Based Storage' (CBS) and 'Random'.

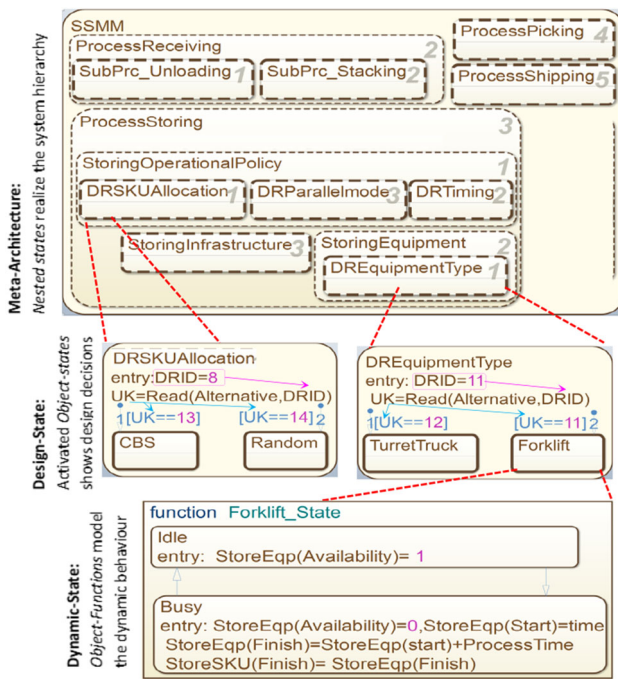


Fig. 3 SSMM for warehouse-case study

FSM allows defining functions for a state to model its behaviour. In this framework, a function is associated with each object-state to model its dynamic behaviour, called object-functions. An object-function returns the dynamic state of an object-state with certain outputs (e.g., equipment available capacity), called Dynamic Variables (DVs). Relation between object-functions (in exchanging DVs) must be defined to embody the interactions between enablers.

In SSMM, a design alternative is configured by activating one object-state for a DR-state. For instance, one alternative is configured by activating ‘CBS’ policy state while another is configured by activating ‘Random’ state. This model is named SSMM because it demonstrates system architecture by states while it is a meta-model that various system alternatives conform to it.

5 Trade study process

5.1 Alternative generation and feasibility checking algorithms

The alternative generation algorithm generates alternatives by exploring the stored design-objects in the design knowledge database [53]. The algorithm uses an array of variables, which each represents a DR with a unique ID, see Eq. 1. In this equation, Z and K are respectively the numbers of defined DRs and design-objects. The feasibility checking algorithm eliminates those alternatives that combination of employed

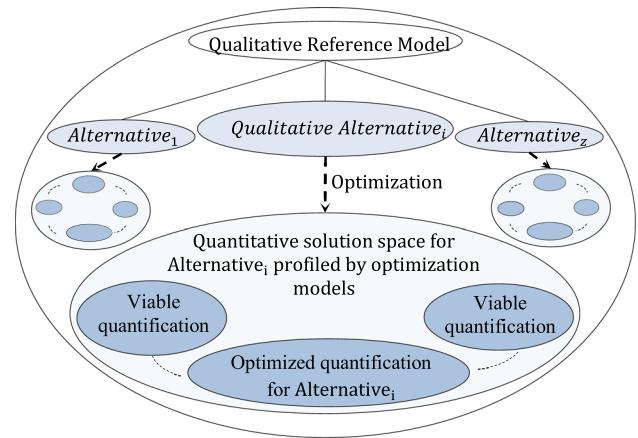


Fig. 4 Characterizing the solution space in the process of searching for a better design alternative

design-object results in an unfeasible system due to inconsistencies between their disciplines.

The applicability attribute of a design-object clarifies whether it can be employed for a DR_{ID}. The consistency attribute clarifies the consistency/compatibility of a design-object with others. Each design-object has an identifier, called Uniqueness-Key (UK). Generated alternatives at this stage are called ‘qualitative’ because they only show the employed genres for DRs.

$$CEPS_{Qualitative\ reference\ model} = [(-)_{DR_1}, \dots, (-)_{DR_{id}}, \dots, (-)_{DR_z}] \quad (1)$$

$$Applicability_{UK} = (Applicability_1, \dots, Applicability_z) \quad (2)$$

$$Applicability_{UK, id} = \begin{cases} 1; & Can\ be\ used\ in\ DR_{id} \quad \forall id \in Z \\ 0; & Otherwise \end{cases}$$

$$Consistency_{UK} = (Consistency_1, \dots, Consistency_k) \quad (3)$$

$$Consistency_{UK, j} = \begin{cases} 1; & Inconsistent\ with\ design - object_i \quad \forall i \in k \\ 0; & otherwise \end{cases}$$

5.2 Optimization/quantification

Optimization models are applied to qualitative alternatives to achieve a level of quantification, which with can be simulated. As explained, this framework uses LP and ALDeP for quantification of alternatives [31]. Quantification models return the required numbers of an employed design-object to satisfy objectives/constraints (e.g., three forklifts are required to meet throughput), as shown by ‘Number of Instances’ in Eq. 4. Figure 4 demonstrates how this framework characterizes the solution space. The quantified alternatives are stored in the design database.

$$\text{CEPS}_{\text{Quantitative-Alternative}_i} = [(-, -)_{\text{DR}_1}, \dots, (\text{UK}, \text{Number of Instances})_{\text{DR}_{1D}}, \dots, (-, -)_{\text{DR}_n}] \quad (4)$$

5.3 Fuzzy assessment framework

This framework utilizes a novel fuzzy framework for assessment of alternatives against qualitative objectives by modelling the domain expert's opinion [54]. The fuzzy framework has a hybrid-hierarchical structure and comprises fuzzy models in three levels: design-object, process, and system. The fuzzy models with respect to their hierarchical order are called 'object-fuzzy-model', 'Process-fuzzy-model', and 'System-fuzzy-model', see Fig. 5. The fuzzy framework has a hierarchical structure such that the output of a lower-level fuzzy-model is an input to the next-level model. The fuzzy framework has a hybrid structure, so the fuzzy models get extra inputs related to each assessment level. Hence, a fuzzy model at a higher level has at least two input sets: output of a lower-level fuzzy model and extra inputs from the design knowledge.

5.3.1 Fuzzy framework structure

The rule-based fuzzy structure is used in the fuzzy framework because it fits well to decision-making in the presence of linguistic information. Also, the Mamdani approach is used as the inference system because it fits better into rule-based models and returns a single output [55]. The novelty of the introduced fuzzy framework relies on its structure (hybrid-hierarchical), hence the technical details of fuzzy logic are not explained and interested readers are referred to literature [55].

In rule-based models, a set of if-then rules maps inputs to outputs [56]. Because the fuzzy framework aims to assess the qualitative aspects of a CEPS, so linguistic labels are defined for fuzzy inputs/outputs. The framework uses 'Efficiency' labels to demonstrate the input fuzzy sets and 'Goodness' labels for outputs, which Input labels are labelled as 'Not-Efficient', 'Marginally-Efficient', 'Efficient', and 'Very-Efficient', and output labels as: 'Not-Good', 'Marginally-Good', 'Good', and 'Very-Good'.

In the presence of linguistic information, experts define if-then rules and grade membership functions such that the model returns satisfactory results with a test dataset [56]. For instance, if the efficiency of a design-object against a 'very-important' objective is 'Very Efficient' and against a 'less-important' objective is 'Marginally Efficient', the goodness of design-object is assessed: 'Good'.

This paper uses the triangular function, which is extensively used in the application of fuzzy logic in different contexts. The framework uses the following widely used

procedure introduced by this reference [55] to grade the membership functions. It includes three steps that should be followed for each fuzzy variable:

- (1). Defining a range that indicates the discourse universe of a fuzzy variable; for example (0–1).
- (2). Experts mark an interval in favour of the meaning of each linguistic label used for the variable. This interval is defined (0.45, 0.8) for 'Efficient' label, see Fig. 5.
- (3). Experts mark a key-point within the defined interval, significantly representing the meaning of the linguistic label. The membership function gets its maximum truth degree in key-point. This is defined (0.75) for 'Efficient' label.

The membership function associated with each fuzzy set covers a sub-range in the discourse universe. For example, 'Not-Efficient' function covers (0–0.35) sub-range while 'Marginally-Efficient' covers (0.25–0.6). The same fuzzification is applied for variables that are exchanged between fuzzy models. This novel structure facilitates the data exchange between fuzzy models because both input and output variables embody the same fuzzy magnitude about the domain's expert opinion. For instance, fuzzification of 'Marginally-Efficient' is similar to 'Marginally-Good'.

All fuzzy models take the objectives' importance as a fuzzy input too. The fuzzy framework is implemented using MATLAB fuzzy toolbox as an "off-the-shelf" platform, used in different works [57]. However, other platforms can be used too.

5.3.2 Fuzzy models

'Object-fuzzy-model': A CEPS performance is directly related to the employed enablers for DRs. The 'Object-fuzzy-model' assesses the goodness of a design-object to be employed in an alternative. Thus, the domain expert's opinion about the efficiency of a design-object against objectives is modeled as a fuzzy input for 'Object-fuzzy-model'.

This framework uses a procedure to assist the domain experts to express their opinion regarding the efficiency of fuzzy inputs in all three fuzzy models:

- (1) The expert expresses their opinion about an input by selecting one efficiency label among: 'Not-efficient', 'Marginally-Efficient', 'Efficient', 'Very-Efficient'.
- (2) The expert picks a value from the sub-range covered by the selected efficiency label. For instance, if the expert selects 'Efficient', then they pick a value from the sub-range of (0.45–0.8).

In the warehouse case, 'Cargomatic' is a design-object that can be employed for Unloading – equipmentype₁ as

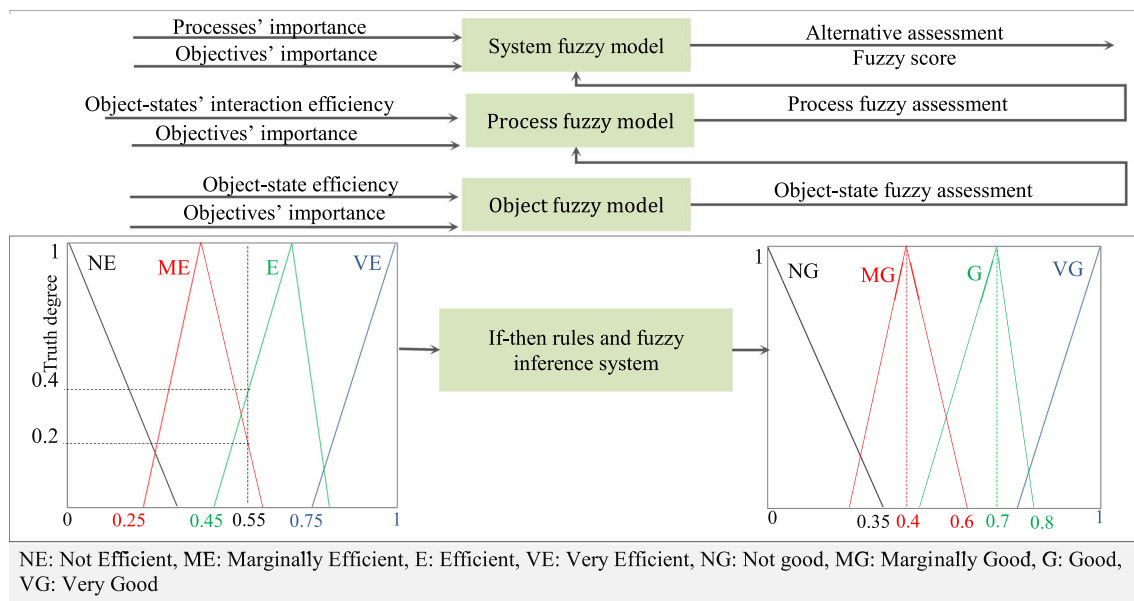


Fig. 5 Fuzzy Assessment framework structure

DR₁. Objectives are safety satisfaction and future expansion; the former is considered as very-important and the latter as less-important. The expert assesses 'Cargomatic' as 'Efficient' against safety satisfaction and picks 0.55 from the covered sub-range by 'Efficient' membership function. This value (0.55) belongs to two membership functions with different truth degrees: 'Efficient' and 'Marginally-Efficient'. This conforms to the main concept of fuzzy logic. Although the idea was to assess 'Cargomatic' as 'Efficient', multiple belongingness of 0.55 to 'Efficient' and 'Marginally-Efficient' embodies uncertainty in subjective assessment. It is worth mentioning that instead of using the above procedure, it is also possible to use the introduced approach by this reference [58] to calculate the assessment results in all three fuzzy models. Hence, the experts only pick an efficiency label (such as 'Very-Efficient'). Then the introduced FIS can be used to either return a single value or a linguistic term as fuzzy assessment output.

'Process-fuzzy-model': A process may need to have multiple interacting enablers to perform its function. 'Process-fuzzy-model' assesses the goodness of a process as the configuration of the employed design-object that interact with each other. Thus, the first input set to 'Process-fuzzy-model' is the goodness of design-objects (output of 'Object-fuzzy-model'). The expert's opinion about the interaction efficiency between every two design-objects is modelled as the second fuzzy input for the 'Process-fuzzy-model'. For example, the interaction efficiency of 'Cargomatic' and 'Sunken-door' against safety satisfaction is expressed as 'Very-Efficient', while interaction efficiency of 'Cargomatic' and 'Level-door' is expressed as 'Not-Efficient'.

Experts define if-then rules. For instance, if the goodness of all employed design-objects is 'Very-Good' and interaction efficiency of any employed design-object against very-importance objective is not 'Not-Efficient' then the process is assessed as 'Very-Good'.

'System-fuzzy-model': At the system level, certain processes can be more important due to their major impact on objectives' satisfaction. In the warehousing domain, the storing process is very critical and its design directly impacts the warehouse performance [53]. Thus, three input fuzzy sets are defined for 'system-fuzzy-model': 'Goodness' of processes (output of 'Process-fuzzy-model'), processes' importance, and objectives' importance. 'System-fuzzy-model' assesses an alternative as an integration of processes. For instance, if the goodness of very-important processes is 'Not-Good' and the goodness of less-important processes is not 'Very-Good' then the 'System-fuzzy-model' assesses the alternative as 'Not-Good'.

The fuzzy framework provides a single value as a design goodness score, which a higher score means a better alternative. The framework retrieves its needed data from the design knowledge database. For example, the design-objects have their interactions' efficiencies as their attributes.

5.4 System validation

The number of design alternatives and DRs can be massive. Accordingly, configuring an alternative in SSMM by activating the employed object-states in DR-states can take so much time. Hence, it is needed to facilitate such a tedious task to assure the practicality of the framework. On the other

hand, the simulation results can provide some insights into the dynamic behaviour of the alternatives, which can be used in a systematic manner to improve them. To address these issues, two supporting algorithms are introduced: ‘coupling algorithm’ and ‘feedback algorithm’.

5.5 Coupling algorithm

‘Coupling algorithm’ retrieves the quantified alternatives from the database and simulates them with SSMM automatically. This is called coupling algorithm because it configures the SSMM such that represents one specific alternative. Generally, alternatives vary in their employed design-objects for DRs. For each specific alternative, the coupling algorithm activates the object-states equivalent to its employed design-objects. For ease of referring, the alternative that is going to be simulated is called ‘current-alternative’.

In SSMM, each DR-state has a constant value called ‘DRID’, equivalent to the ID of DR_{ID} . For instance, Storing – Allocation₈ is a DR and its ID is 8. Accordingly, ‘SKU-Allocation’ state is equivalent to Storing – Allocation₈ and the DRID of ‘SKU-Allocation’ is also valued 8.

As explained, each design-object has an identifier called Uniqueness Key (UK). The equivalent object-state to a design-object carries a constant with the same value as UK. For instance, CBS is a design-object, and its UK is valued 13. Likewise, CBS-state carries 13 as its constant value.

The coupling algorithm retrieves the quantitative model of current-alternative from the database and obtains the UK of the employed design-object for each DR_{ID} . For the warehouse case, the current-alternative employed CBS as its design-object, which its UK is equal to 13, as shown below.

$$\text{Warehouse}_{\text{Quantitative model}} = [(-)_{DR_1}, \dots, (13)_{DR_8}, \dots, (-)_{DR_z}]$$

In SSMM, a transition condition is defined for each object-state to compare the retrieved UK by coupling algorithm with the carrying constant by an object-state. In the warehouse case, the algorithm finds that the employed design-object for Storing – Allocation₈ has UK of 13. There are two object-states in the ‘SKU-Allocation’ state: ‘CBS’ and ‘Random’. The transition condition in ‘CBS’ compares the retrieved UK (13) with the carrying constant by CBS, which is also 13. Therefore, the transition happens, and CBS-state is activated. While the ‘Random’ state carries 14 as its constant. When the transition is tested in the Random state, since the retrieved value for Storing – Allocation₈ is 13, not equal to 14, the transition does not happen.

The coupling algorithm checks each DR-state and activates the employed design-objects for each DR-state. At this stage, SSMM demonstrates a specific alternative that can be simulated.

Coupling algorithm:

Find quantitative model of an alternative for simulation.

For $id = 1: Z$ (number of DRs).

1. Find ‘DRID’ in DR_{id} – state equivalent to DR_{id}
2. In quantitative model, retrieve UK of employed design-object for DR_{id} .
3. Check transition conditions of object-states in DR_{id} – state
4. Activate the object-state that its constant is equal to retrieved UK

End.

5.5.1 Feedback algorithm

It is critical that a design alternative satisfies objectives and constraints (e.g., cost and throughput), so they are formulated in SSMM and are called Design-KPIs. The object-functions dynamically update these variables and return their value at the end of simulation.

Due to the dynamic infarctions between the enablers (more accurately the selected design-objects for DRs), the simulation may return different values for the design-KPIs compared to the calculated results by quantification models. For instance, the simulated throughput can be lower than what was formulated as an LP constraint. Likewise, the total cost can be higher than the LP result. Moreover, an alternative may have the potential for better performance (e.g., throughput can be satisfied with lower cost). This paper introduces the ‘feedback algorithm’ to study the simulation results and modify the quantification of alternatives to assure fulfilment of constraints or improve objectives’ satisfaction.

This paper recommends formulating the ‘queue’ between the processes and ‘utilization’ of physical enablers as ‘dynamic-KPIs’ in SSMM. The utilization is formulated for those enablers whose limited capacity may affect objectives satisfaction. Because optimization approaches mostly model a system such that its enablers use their nominal capacity. However, the different operation times of different enablers may lead to a desynchronization between processes. Consequently, enablers may run below their nominal capacity, impacting the throughput. The queue between two processes is formulated by considering the finish time of items in the first process and their start time in the subsequent process. The utilization of enablers is formulated by considering the accumulated values of their busy times during the available working time.

The feedback algorithm includes two procedures. The first addresses modifying the alternatives to meet the throughput if it was not satisfied according to simulation results. The second procedure aims to reduce the number of physical enablers

without compromising on throughput satisfaction to improve the alternatives from the cost reduction perspective.

If an alternative does not satisfy the throughput, the process that experiences a long queue (bottleneck) is modified by increasing the number of the physical enabler (e.g., machining equipment) that its limited available capacity caused the bottleneck situation. That is identified by studying the utilization of the physical enablers in the bottleneck process. Hence, the enabler with the highest utilization is the first candidate to increase its number of instances.

If the utilization of an enabler in one process is considerably low, the number of instances for that enabler is reduced, because the process might be able to meet throughput with a smaller number of enablers.

Feedback algorithm:

A) Throughput constraint satisfaction procedure

- I. Select Bottle-Neck process (BN).
- II. Define ‘Difference’ as: ‘required-throughput’ – ‘simulated-throughput’
- III. In BN, find enabler(s) that had highest simulated utilization; call it enabler_x
- IV. Find ‘Number of Instances’ for enabler_x in quantitative model of alternative
- V. ‘Updated-Instances’ = ‘Number of Instances’ + $\left(\frac{\text{Difference}}{\text{Capacity of enabler}_x} \times \text{operation time of enabler}_x \right)$
- VI. Call quantification to apply needed changes according to ‘Updated-Instances’

B) Objectives’ improvement procedure

- I. Do these steps for each enabler of a process
- II. Define: n = ‘Number of Instances’, ‘Minimum’ = ‘Number of Instances’
- III. Find Total-Busy-Time (TBT) of all instances of enabler
- IV. Perform this loop while the below equation holds
 - (a) For $k = 1: n-1$

$$\frac{(\text{Avaialbe time}) \times (n - k)}{(\text{Avaialbe time}) \times n} \leq \frac{\sum_{i=1}^n \text{TBT of instance } i}{(\text{Avaialbe time}) \times n} \leq \frac{(\text{Avaialbe time}) \times (n - k + 1)}{(\text{Avaialbe time}) \times n}$$

(b) If above equation holds then ‘Minimum’ = $n-k+1$

- V. ‘Updated-Instances’ = ‘Minimum’

The introduced feedback algorithm addresses simple modifications to alternatives in a systematic manner to accelerate the design process and reduce mistakes. A modified alternative is simulated again. The algorithm can be called several times until the alternative satisfies the constraints or

there is no place for further improving the objectives. A certain number of iterations can be defined as another stopping condition for the algorithm.

From the SE perspective, an alternative is validated at this stage. The SSMM, coupling algorithm, feedback algorithm, and quantification models work in tandem and exchange their results as shown in Fig. 2.

5.6 Technique for order of preference by similarity to ideal solution

The introduced framework provides two sources of information to analyse/assess the performance of alternatives: the simulation results for the quantitative objectives and the fuzzy assessment for the qualitative ones. This framework uses TOPSIS for ranking alternatives and final selection of a better alternative.

The TOPSIS procedure is explained briefly here and interested readers are referred to the literature for details. TOPSIS hypothesises two alternatives: positive and negative ideal alternatives. The former illustrates the best performance against all criteria while the latter demonstrates the worst. TOPSIS finds a real-alternative that is closest to the positive and is farthest from the negative ideal-alternative. The total number of feasible alternatives is defined as m while n is the total number of assessment criteria (e.g., simulated total cost and fuzzy assessment).

TOPSIS application:

I I- Do it for $i = 1: m$

1. Let x_{ij} be the given score of *alternative* _{i} against criterion j . Construct matrix X as: $X = [x_{ij}]_{m \times n}$
2. Let J be the set of benefit criterions, which their maximization is desired (e.g., throughput):
3. Let J' be the set of cost criterions, which their minimization is desired (e.g., cost):
4. Let z_m be each criterion weight:
5. Construct normalized weight matrix: $W = [w_j]_{1 \times n}$, $\sum_j w_j = 1$; $w_j = z_j / \sum_j z_j$ $m = 1: n$
6. Construct normalized decision matrix r : $r = [x_{ij}]_{m \times n}$; $r_{ij} = \frac{x_{ij}}{\sqrt{\sum_i x_{ij}^2}}$ for $i = 1: m$ and $j = 1: n$
7. Construct weighted normalized decision matrix: $v = [v_{ij}]_{m \times n}$, $v_{ij} = w_j \times r_{ij}$,
8. Positive Ideal-alternative: $A^* = \{v_1^*, \dots, v_n^*\}$; $v_j^* = \{ \max(v_{ij}) \text{ if } j \in J; \min(v_{ij}) \text{ if } j \in J' \}$
9. Negative Ideal-alternative: $A' = \{v_1', \dots, v_n'\}$; $v_j' = \{ \min(v_{ij}) \text{ if } j \in J; \max(v_{ij}) \text{ if } j \in J' \}$
10. Separation from Positive Ideal-alternative is: $S_j^* =$

$$\left[\sum_j (v_j^* - v_{ij})^2 \right]^{1/2} \quad i = 1 : m$$

11. Separation from Negative Ideal-alternative is: $S'_j = \left[\sum_j (v'_j - v_{ij})^2 \right]^{1/2} \quad i = 1 : m$
- 12 Calculate relative closeness to ideal solution, C_i^* as: $C_i^* = S'_j / (S'_j + S_j^*) \quad i = 1 : m \quad 0 < C_i^* < 1$

II. II- Select the alternative with C_i^* closest to 1.

TOPSIS takes simulation and fuzzy assessment results of alternatives and ranks them from the best to worst, which the best one is handed-off for detailed design. Hence, this framework can stand as a decision support framework at early design stages of CEPSSs.

In the next section, two case studies are presented to demonstrate the applicability of the proposed framework and its advantages compared to existing approaches in the literature.

6 Case studies

6.1 Warehouse design case

A pharmaceutical company wants to construct a warehouse that will receive three SKU types: SKU-A, SKU-B, and SKU-C. Stakeholders defined the design objectives as cost minimization, safety satisfaction, and future expansion while meeting the required throughput. The warehouse managers defined the process enablers as equipment, infrastructure, and operational policy. The warehouse managers defined DRs for process enablers and their possible design-objects. This is all part of the design knowledge used as input for the framework. Warehouse SSMM is demonstrated in Fig. 3.

Generally, the warehousing function starts by receiving consignment and continues by storing SKUs. The assignment policy determines how to allocate SKUs into the storage modules. The CBS policy allocates the high-demand SKUs in modules that require less travel-time (operation time) in storing/picking processes, while this increases the travel-time for low-demand SKUs. The random storage policy allocates SKUs to any available spot. The batch and single picking are two operational policies for the order-picking process, the former consolidates orders and make a picklist. Whereas in the single-picking policy, the orders are picked separately. The travelled distance of the operator directly affects the operation times in storing/picking processes. Having a cross-aisle between storage aisles can reduce the travel-distance, because the operator can exit/enter aisles without passing the entire aisle length, yet a warehouse with a cross-aisle requires more space.

UK	Design option	UK	Design option	UK	Design option
1	Cargomatic	9	Single stacking	17	Pallet rack
2	Crane	10	Double stacking	18	Shelve rack
3	Middle-dock	11	Forklift	19	With cross aisle
4	Corner-dock	12	Turret truck	20	Without cross aisle
5	Sunken door	13	CBS	21	Batch picking
6	Level Door	14	Random-storage	22	Single picking
7	FIFO	15	Single command		
8	LIFO	16	Double command		

Fig. 6 Defined design-objects for the warehouse case study

6.1.1 Alternative generation

The qualitative reference model of the warehouse is shown below, which includes 20 defined DRs.

WarehouseQualitative–referenceModel:
 [Unloading – equipmenttype₁, Unloading – doorposition₂, Unloading – doortype₃, Staking – timing₅, Stacking – infrastructuretype₆, Stacking – equipmenttype₇, Storing – Allocation₈, Storing – timing₉, Storing – parallel – mode₁₀, Storing – Equipment – type₁₁, Storing – storage – module – type₁₂, Storing – aisles configuration₁₃, Picking – pick – listing₁₄, Picking – parallel mode₁₅, Picking – Equipment – type₁₆, Shipping – door – type₁₇, Shipping – door – position₁₈, Shipping – timing₁₉, Shipping – Equipment – type₂₀].

Figure 6 shows the stored design-objects in the design database with their attributes (as input data such as applicability, consistency, and efficiency) to be used by feasibility checking/alternative generation algorithms, quantification models, and fuzzy framework.

The feasibility checking algorithm crossed-off almost half of the generated alternatives, showing the high risk of design failure without having an integrated design approach. Three generated alternatives are shown for demonstration.

Alternative_A = [(1)₁, (3)₂, (5)₃, (7)₄, (7)₅, (10)₆, (11)₇, (14)₈, (7)₉, (15)₁₀, (11)₁₁, (17)₁₂, (20)₁₃, (21)₁₄, (15)₁₅, (12)₁₆, (6)₁₇, (3)₁₈, (8)₁₉, (2)₂₀].

Alternative_B = [(1)₁, (3)₂, (6)₃, (7)₄, (7)₅, (9)₆, (11)₇, (13)₈, (7)₉, (15)₁₀, (11)₁₁, (17)₁₂, (20)₁₃, (22)₁₄, (15)₁₅, (11)₁₆, (6)₁₇, (3)₁₈, (7)₁₉, (11)₂₀].

Alternative_C = [(1)₁, (3)₂, (5)₃, (7)₄, (7)₅, (10)₆, (11)₇, (13)₈, (7)₉, (16)₁₀, (11)₁₁, (18)₁₂, (20)₁₃, (22)₁₄, (16)₁₅, (11)₁₆, (6)₁₇, (3)₁₈, (8)₁₉, (2)₂₀].

6.1.2 Quantification

The LP model is formulated for cost minimization and is constrained to the required throughput. The cost factors are defined as fixed and operational costs of design-objects and space cost. The cost-related data are defined as attributes of

Table 2 Warehouse case study result

Alternative	Total cost according to quantification	Fuzzy score	Total cost (validated)	TOPSIS rank
<i>Alternative_A</i>	4,660,583	0.4173	4,719,396	2
<i>Alternative_B</i>	7,130,002	0.7436	7,321,965	3
<i>Alternative_C</i>	4,788,013	0.54	4,648,492	1

design-objects. ALDeP is applied on alternatives to develop a layout according to the employed design-objects. For more details regarding quantification see Online Appendix A [31]. The quantification results for the demonstrated alternatives are given in Table 2.

6.1.3 Fuzzy assessment

The fuzzy framework assesses the alternatives against the qualitative objectives: ‘safety-satisfaction’ and ‘future-expansion’. The importance of objectives was defined by the managers and modelled as fuzzy inputs for the fuzzy models. The objectives’ importance and processes’ importance include four fuzzy sets: ‘Less-Important’, ‘Marginally-Important’, ‘Important’, ‘Very-Important’. Fuzzifications of fuzzy variables are given below:

‘Not-Good’, ‘Not-Efficient’, and ‘less-Important’:

$$f(x, -1, 0, 0.35) = \begin{cases} 1 & x < 0 \\ \frac{0.35-x}{0.35} & 0 \leq x \leq 0.35 \\ 0 & \text{Otherwise} \end{cases}$$

‘Marginally-Good’, ‘Marginally-Efficient’, and ‘Marginally-Important’:

$$f(x, 0.25, 0.4, 0.6) = \begin{cases} \frac{x-0.25}{0.15} & 0.25 \leq x \leq 0.4 \\ \frac{0.6-x}{0.2} & 0.4 \leq x \leq 0.6 \\ 0 & \text{Otherwise} \end{cases}$$

‘Good’, ‘Efficient’, and ‘Important’:

$$f(x, 0.45, 0.7, 0.8) = \begin{cases} \frac{x-0.45}{0.25} & 0.45 \leq x \leq 0.7 \\ \frac{0.8-x}{0.1} & 0.7 \leq x \leq 0.8 \\ 0 & \text{Otherwise} \end{cases}$$

‘Very-Good’, ‘Very-Efficient’, and ‘Very-Important’:

$$f(x, 0.75, 1, 1) = \begin{cases} 0 & x \leq 0.75 \\ \frac{x-0.75}{0.25} & 0.75 \leq x \leq 1 \\ 1 & 1 \leq x \end{cases}$$

Warehouse managers defined the importance of safety-satisfaction and future-expansion respectively: Important-0.6 and Marginally-Important-0.4. Managers expressed their

opinion about efficiencies of design-objects, their interactions, and process importance, which are fuzzy inputs to the fuzzy framework. Details are given in Online Appendix B.

Fuzzy assessment and simulation results of validated alternatives (after applying the feedback algorithm) are shown in Table 2.

6.1.4 TOPSIS

Two criteria were defined for the TOPSIS to assess alternatives: total validated cost and fuzzy assessment result with the relative importance of respectively 0.7 and 0.3. Table 2 shows calculated ranks by TOPSIS for each alternative.

6.1.5 Discussion on case study

Alternative_A shows a lower total cost compared to Alternative_C according to the quantitation models. These alternatives differ in their employed design-objects in storing/picking processes. Alternative_A employs ‘pallet-rack’ and ‘forklift’ in storing and ‘turret-truck’ in the picking process. Alternative_C employs ‘shelve-rack’ and ‘forklift’ in both processes. Although, ‘pallet-rack’ is more expensive than ‘shelve-rack’, storing in or picking from the ‘pallet-rack’ takes less time. The ‘turret-truck’ has a shorter operation time compared to the ‘forklift’, but the ‘turret-tuck’ is more expensive. The quantification models calculated that three ‘forklifts’ are required to perform the storing process on ‘pallet-rack’ to satisfy throughput, while four ‘forklifts’ are required if the ‘shelve-rack’ is employed. Three ‘turret-truck’ can satisfy the throughput in the picking process if the ‘pallet-rack’ is employed, while if ‘forklift’ is employed four is needed. Thus, Alternative_A demonstrated a lower total cost calculated by quantification models: employing three ‘forklifts’ in storing and three ‘turret truck’ in the picking process.

Alternative_C employs ‘CBS’ and ‘double-command’ as its operational policies in storing/picking process. ‘Double-command’ policy allows picking the ordered SKUs after storing some SKUs. Thus, the operator-equipment can cut one empty travelling activity in storing and picking processes. The combination of ‘CBS’ and ‘double-command’ policies reduces the total operation time. In Alternative_C, the utilization of equipment in storing and picking processes was

respectively 59% and 67%. This shows that the total working hours of four ‘forklift’s in Alternative_c was between the total working hours of two and three ‘forklifts’ within the available working-time. Therefore, the feedback algorithm reduced the number of ‘forklifts’ in both picking/storing processes to three in the quantification model of Alternative_c. The quantification models are called again to check the constraints satisfaction in other processes after such an update. Simulation of updated Alternative_c revealed the throughput satisfaction.

$$2 \times 5 < \text{Simulated working-hours for 'forklift'} < 3 \times 5.$$

$$\rightarrow 50\% \left(\frac{2 \times 5 \times 100}{4 \times 5} \right) < \text{Utilization} < 75\% \left(\frac{3 \times 5 \times 100}{4 \times 5} \right).$$

If a sequential design approach was followed without having an integrated design approach in addressing all design requirements at the same time including the operational policies, the results could have been different in cost minimization. It is hard to achieve such insight without having an integrated design approach as realized in this framework.

Both Alternative_A and Alternative_c received a lower fuzzy score compared to Alternative_B. This demonstrates the necessity of having a multi-aspect assessment approach. One alternative could be considered promising from the cost minimization perspective, while might not be the best with respect to qualitative objectives. Application of TOPSIS allows ranking the alternatives by considering their excellence on multiple conflicting objectives.

The novel introduced fuzzy framework has a hierarchical structure helping to reduce the number of if–then rules in fuzzy-assessment models by reducing the number of input-variables in each fuzzy-model. The number of if–then rules increases exponentially when the number of variables increases, as called ‘curse of dimensionality’ [59]. The hierarchical structure of the framework helps to break the ‘curse of dimensionality’ in design assessment of CEPSs given the scale and complexity of their design knowledge [60].

6.1.6 Lessons learnt from the case study

Simulation of all alternatives can be timely; hence this research suggests applying a supervisory approach to select and simulate a limited number of alternatives without losing an optimum alternative. However, it is hard to guarantee to find a global optimum because the analytical formulation of CEPS belongs to the NP-hard problem class.

If two subsequent processes do not share resources, if the first process operates properly such that the second can start its operation on time, the employed design-objects in the first process might not impact the operation of the latter process. If the unloading (P_{11}) and the stacking processes (P_{12}) finish their jobs within the available time, their employed design-objects do not impact operations of the storing (P_3), shipping (P_4), and picking (P_3). In this case study, the unloading can

employ different design-objects, leading to three possible configurations as shown below. The numbers in parentheses show the UKs of the employed design-objects.

$$P_{11} - \text{Configuration}_1: [(1)_1, (3)_2, (5)_3, (7)_4].$$

$$P_{11} - \text{Configuration}_2: [(1)_1, (3)_2, (6)_3, (7)_4].$$

$$P_{11} - \text{Configuration}_3: [(2)_1, (3)_2, (6)_3, (8)_4].$$

There are four feasible configurations for the stacking process, so the number of possible combinations for unloading and stacking process configurations is $3 \times 5 = 15$. Likewise, the process configurations of storing, picking, shipping lead to 320 combinations.

In total 4800 feasible alternatives are generated. This research selected 15 alternatives that covered all 15 combinations for the unloading/stacking process configurations. The selected 15 alternatives were simulated and validated particularly against their unloading/stacking processes. Then the simulation results of these two processes were written for all feasible 4800 alternatives according to their combination for unloading/stacking process configurations.

Among 15 combinations of unloading/stacking, the combination that had minimum cost while satisfying the throughput was selected. This combination was used to simulate all 320 possible combinations of [P_2 , P_3 , P_4]. However, the simulation results were only updated for storing, picking, and shipping processes. This supervisory approach was done to reduce the computational cost related to the simulation time, so instead of simulating 4800 alternatives, $15 + 320$ alternatives were simulated.

6.2 Reconfigurable manufacturing system design case

To illustrate the advantages of the proposed framework compared to the existing approaches, the framework is applied to another case study solved with other approaches in the literature [24, 61]. The used input data are from the literature belonging to the design knowledge of the case study.

This case study is briefly explained here and interested readers are referred to the literature and Online Appendix C and D for details [61]. The case is about designing a reconfigurable manufacturing system to produce two products: ANC-90 and ANC-101. The required hourly throughput for each product respectively is 120 and 180. The objective is cost minimization. Products require common and specific machining operations. Different machine tools are available with different capacities, operation times, and fixed costs. The system is to be designed as a flow line including a maximum of ten sequential and five parallel machines.

Youssef and ElMaraghy [61] used genetic algorithm for cost minimization by finding optimum configuration in terms of the number of stages, machines’ selection, and machines’

Fig. 7 Alternative₁-A possible optimal configuration

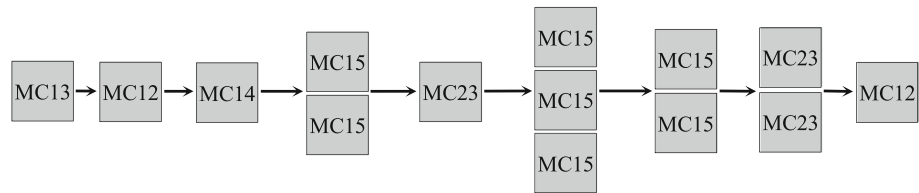


Table 3 Results of second case study

Alternative	Throughput	Total cost (validated)	Rank
1*	300	\$14,645,000 + \$1,140,000(extra machine) = \$15,785,000	1
1	302	\$14,645,000 + \$1,140,000(extra machine in process 2) + \$725,000 (extra machine in process 8) = \$16,510,000	2
2*	300	\$17,115,000	3
2	300	\$17,115,000	3

number in each stage. Figure 7 shows a found optimal configuration (called Alternative₁) with a total cost of \$14,645,000. Boxes show the machines’ names.

Wang and Dagli [24] used a combination of a genetic algorithm and simulation by PN, in which the encoded chromosome took machines’ cost and simulated throughput to choose a new configuration (population). They obtained another optimal configuration (see Fig. 8) with a higher total cost (\$17,115,000), called Alternative₂. They argued that Alternative₁ could not satisfy throughput by simulation, which this current paper also found a similar issue.

In this case study, the only enabler was defined as machines. This paper defines operational policies as another enabler to show the importance of having an integrated design approach. Accordingly, two possible policies are defined: ‘Batching’ and ‘No-batching’. The described case was aligned with the ‘No-batching’ policy, in which AN90 and ANC101 could enter the production line one by one. Under the ‘Batching’ policy, each item can enter the production line in batches consisting of the same item type. Alternative₁* and Alternative₂* are defined similar to respectively Alternative₁ and Alternative₂, except new alternatives use batching policy.

The alternative generation and feasibility checking algorithms developed feasible alternatives, which are quantified by an LP model. The alternatives are simulated for 540 working minutes. The feedback algorithm is applied to alternatives while outputs of the first 60 min are excluded to eliminate the ramp-up period effect. Table 3 shows the results. TOPSIS is applied to alternatives by considering the total cost and throughput with the relative importance of 0.7 and 0.3.

6.2.1 Discussion on case study

The LP results show that Alternative₁ have a minimum total cost. The simulation results show that Alternative₁ and Alternative₁* cannot produce 300 parts per hour, while Alternative₂ and Alternative₂* satisfy the required throughput. The reason for such a deviation is that different operation times of different machines causes desynchronization between processes and finished parts in one machine cannot immediately get service from the next, causing a queue between stations (bottleneck). Consequently, the machine after bottleneck might work below its nominal capacity, which is against the optimization formulation or assumption. Wang and Dagli [24] argued similarly about the reason for this deviation. The feedback algorithm identified the second process as a bottleneck in Alternative₁ and Alternative₁*. Thus, the quantitative models of the alternatives were updated and the number of machines in the second process was increased from ‘one’ to ‘two’.

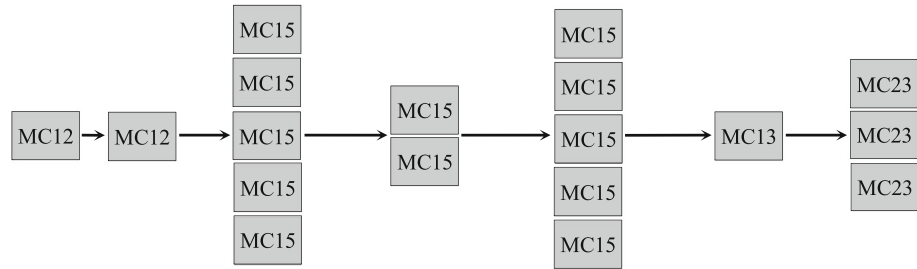
The modified Alternative₁ and Alternative₁* are simulated again, while Alternative₁ could not meet the required throughput and Alternative₁* could. The feedback algorithm found process-eight as the bottleneck, so the number of machines is increased in the process-eight from ‘two’ to ‘three’. The modified alternative is simulated again and could meet the throughput. Alternative₁* shows a better performance compared to Alternative₁ due to the selection of ‘bathing’ policy (less desynchronization).

The proposed framework can find a better solution in satisfying the objectives compared to found solutions in the literature. The scale of this improvement could be higher if the case study had more realistic assumptions. For instance, if changeover times were addressed, the impact of operational policy selection could be more significant. Moreover, the framework allows ranking the alternatives by considering their performance against qualitative/quantitative objectives. However, the reviewed approaches only considered quantitative ones.

7 Conclusion and future work

Generally, simulation-based optimisation problems face several challenges, which the computational cost is a widely accepted one. The simulation takes a long time when the

Fig. 8 Alternative₂-possible optimal configuration



number of variables is large, while the number of design variables is large at early design stages. Thus, the computational cost may limit the application of simulation-based optimization in this context. Although the simulation-based optimizations do not guarantee finding a global optimum, an investigation of the application of simulation-based optimization can be future work for this work. This research also faced the computational cost problem, which was tried to be addressed with a supervisory approach. A future research direction would be the investigation of the application of more advanced supervisory methods for searching solution space more efficiently.

The SSMM simulates the alternative behaviour under some assumptions (e.g., deterministic operation time of machines). However, the occurrence of some events that are against earlier assumptions may impact the system performance. This is called uncertainty, which can be embodied in SSMM by defining some time-events or probability functions to model possible failures or processing times variability. Accordingly, the simulation results include the impact of uncertainty on an alternative's performance.

The integrated design of CEPSs with a holistic approach allows capturing more dimensions of the solution space. As shown in Fig. 9, the optimization models seek to find an optimum solution in a solution space whose dimensions is defined by (limited to) quantifiable design requirements. When the solution space is searched by adding the operational policies dimension, a better solution might be identified. Although some limited aspects of operational policies can be modelled with complicated optimization formulation, the possibility of trapping in a local optimum might increase by adding more variables to the optimization models. This highlights the need for using simulation at early design stages.

SSMM is a key artefact in the proposed framework and embodies three aspects of a CEPS: system architecture, decisions regarding the allocation of design options to design requirements for configuring a design alternative, and dynamic behaviour of an alternative. Such a rich embodiment is one of the key contributions of this research and is the fruit of a novel application of proper modelling approach (OO), innovative application of a modelling formalism (FSM), and seamless incorporation of the framework artefacts such as

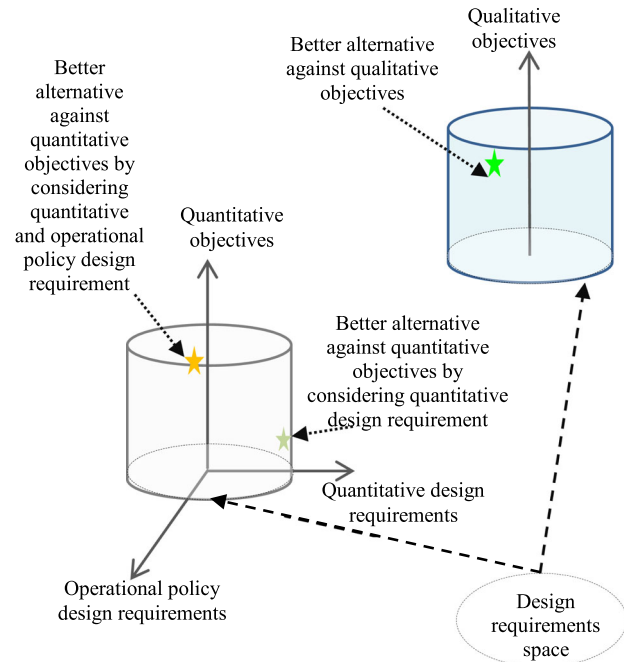


Fig. 9 Multi-dimensional solution space

SSMM, fuzzy assessment framework, TOPSIS, quantification models, and supporting algorithms.

SSMM has a hierarchal structure, which its nested states can be used as design modules for encapsulating the detailed design. Therefore, the SSMM can stand as a design platform for designers in different disciplines. The detailed design can solidify the simulation result during design iterations.

Systematic interconnection between models allows a smooth information propagation and realisation of data integrity/consistency. This is the result of the consistent application of architecting guidelines in different models and artefacts of the framework. Such integrity is a significant relief in collaborative design where the design knowledge is large scale and multi-disciplinary.

Because of having a multi-aspect design assessment, this framework allows obtaining a deep insight into the performance of alternatives and the existing trade-offs in objectives' satisfaction. Therefore, the introduced approach can stand as a decision support framework for the integrated

design of complex engineering processed based systems at their early design stages.

Supplementary Information The online version contains supplementary material available at <https://doi.org/10.1007/s12008-022-00987-x>.

Funding Open Access funding enabled and organized by CAUL and its Member Institutions. No funding was received to assist with the preparation of this manuscript.

Declaration

Conflict of interest The author has no relevant financial or non-financial interests to disclose. The authors have no competing financial or non-financial interests to declare that are relevant to the content of this article.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Reich, Y.: What is a reference? *Res. Eng. Des.* **28**(4), 411–419 (2017)
- Chen, L., Whyte, J.: Understanding design change propagation in complex engineering systems using a digital twin and design structure matrix. *Eng. Constr. Arch. Manag.* (2021)
- Sitton, M., Reich, Y.: EPIC framework for enterprise processes integrative collaboration. *Syst. Eng.* **21**(1), 30–46 (2018)
- ElMaraghy, H.A., Kuzgunkaya, O., Urbanic, R.: Manufacturing systems configuration complexity. *CIRP Ann. Manuf. Technol.* **54**(1), 445–450 (2005)
- Wits, W.W., Van Houten, F.J.: Improving system performance through an integrated design approach. *CIRP Ann.* **60**(1), 187–190 (2011)
- Zheng, C., Hehenberger, P., Le Duigou, J., Bricogne, M., Eynard, B.: Multidisciplinary design methodology for mechatronic systems based on interface model. *Res. Eng. Design* **28**(3), 333–356 (2017)
- Abdoli, S., Kara, S., Kornfeld, B.: Application of dynamic value stream mapping in warehousing context. *Mod. Appl. Sci.* **11**(1), 1913–1852 (2017)
- McNally, P., Heavey, C.: Developing simulation as a desktop resource. *Int. J. Comput. Integr. Manuf.* **17**(5), 435–450 (2004)
- Dou, K., Wang, X., Tang, C., Sullivan, K.: An evolutionary theory-systems approach to a science of the ilities. *Procedia Computer Science* **44**, 433–442 (2015)
- Tomiyama, T., Gu, P., Jin, Y., Lutters, D., Kind, C., Kimura, F.: Design methodologies: industrial and educational applications. *CIRP Ann.* **58**(2), 543–565 (2009)
- Chakrabarti, A., Blessing, L. T.: *Anthology of Theories and Models of Design*. Springer, 2016.
- Abdoli, S., Kara, S.: A review of modelling approaches for conceptual design of complex engineering systems (CESs). In: 2017 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM), 2017, pp. 1266–1270: IEEE.
- Maropoulos, P.G., Ceglarek, D.: Design verification and validation in product lifecycle. *CIRP Ann.* **59**(2), 740–759 (2010)
- Wagenhals, L.W., Haider, S., Levis, A.H.: Synthesizing executable models of object oriented architectures. *Syst. Eng.* **6**(4), 266–300 (2003)
- Meng, X.: Modeling of reconfigurable manufacturing systems based on colored timed object-oriented Petri nets. *J. Manuf. Syst.* **29**(2–3), 81–90 (2010)
- Thiers, G.: A model-based systems engineering methodology to make engineering analysis of discrete-event logistics systems more cost-accessible. Georgia Inst. Technol. (2014)
- Cabrera, A. A., Komoto, H., Van Beek, T., Tomiyama, T.: Architecture-centric design approach for multidisciplinary product development. In: *Advances in Product Family and Product Platform Design*: Springer, pp. 419–447 (2014)
- Dauby, J.P., Dagli, C.H.: The canonical decomposition fuzzy comparative methodology for assessing architectures. *IEEE Syst. J.* **5**(2), 244–255 (2011)
- Singh, A., Dagli, C. H.: Multi-objective stochastic heuristic methodology for tradespace exploration of a network centric system of systems. In: 2009 3rd Annual IEEE Systems Conference, pp. 218–223: IEEE (2009).
- Singh, R., Khilwani, N., Tiwari, M.: Justification for the selection of a reconfigurable manufacturing system: a fuzzy analytical hierarchy based approach. *Int. J. Prod. Res.* **45**(14), 3165–3190 (2007)
- Pape, L., Agarwal, S., Dagli, C.: Selecting attributes, rules, and membership functions for fuzzy SOS architecture evaluation. *Procedia Comput. Sci.* **61**, 176–182 (2015)
- Acheson, P., Dagli, C., Kilicay-Ergin, N.: Fuzzy decision analysis in negotiation between the system of systems agent and the system agent in an agent-based model. *arXiv preprint arXiv:* (2014)
- Kilicay-Ergin, N., Acheson, P., Colombi, J., Dagli, C. H.: Modeling system of systems acquisition. In: 2012 7th International Conference on System of Systems Engineering (SoSE), pp. 514–518: IEEE (2012).
- Wang, R.Z., Dagli, C.H.: Developing a holistic modeling approach for search-based system architecting (in English). 2013 Conf. Syst. Eng. Res., 16, 206–215 (2013)
- Dori, D., Renick, A., Wengrowicz, N.: When quantitative meets qualitative: enhancing OPM conceptual systems modeling with MATLAB computational capabilities. *Res. Eng. Des.* **27**(2), 141–164 (2016)
- Zhang, L.: Modelling process platforms based on an object-oriented visual diagrammatic modelling language. *Int. J. Prod. Res.* **47**(16), 4413–4435 (2009)
- Koo, H. B.: A meta-language for systems architecting. Massachusetts Institute of Technology (2005)
- Gu, P., Rao, H.A., Tseng, M.M.: Systematic design of manufacturing systems based on axiomatic design approach. *CIRP Ann. Manuf. Technol.* **50**(1), 299–304 (2001)
- Cochran, D.S., Arinez, J.F., Duda, J.W., Linck, J.: A decomposition approach for manufacturing system design. *J. Manuf. Syst.* **20**(6), 371–389 (2002)
- Kulak, O.: A decision support system for fuzzy multi-attribute selection of material handling equipments. *Expert Syst. Appl.* **29**(2), 310–319 (2005)
- Abdoli, S., Kara, S.: A modelling framework to support design of complex engineering systems in early design stages. *Res. Eng. Design* **31**(1), 25–52 (2020)

32. Matei, I., Bock, C. Modeling methodologies and simulation for dynamical systems. US Department of Commerce, National Institute of Standards and Technology (2012)
33. Silva, S. C., Alves, A.: Design of product oriented manufacturing systems. In: International Conference on Information Technology for Balanced Automation Systems, pp. 359–366: Springer (2002).
34. Buede, D.M., Miller, W.D.: The engineering design of systems: Models and methods. Wiley (2016)
35. Sasaki, M., Gen, M.: Fuzzy multiple objective optimal system design by hybrid genetic algorithm. *Appl. Soft Comput.* **2**(3), 189–196 (2003)
36. Sinha, A.K., Anand, A.: Towards fuzzy preference relationship based on decision making approach to access the performance of suppliers in environmental conscious manufacturing domain. *Comput. Ind. Eng.* **105**, 39–54 (2017)
37. Kumar, S., Barman, A.G.: Fuzzy TOPSIS and fuzzy VIKOR in selecting green suppliers for sponge iron and steel manufacturing. *Soft. Comput.* **25**(8), 6505–6525 (2021)
38. Cabrera, A.A., et al.: Towards automation of control software: A review of challenges in mechatronic design. *Mechatronic.* **20**(8), 876–886 (2010)
39. Cochran, D.S., Reinhardt, G., Linck, J., Mauderer, M.: Decision support for manufacturing system design—combining a decomposition methodology with procedural manufacturing system design. In: The Third world congress on intelligent manufacturing processes and systems—Cambridge, MA—June 28, vol. 30, p. 2000 (2000)
40. Maropoulos, P.G., Ceglarek, D.: Design verification and validation in product lifecycle. *CIRP Ann. Manuf. Technol.* **59**(2), 740–759 (2010)
41. Tomiyama, T., Gu, P., Jin, Y., Lutters, D., Kind, C., Kimura, F.: Design methodologies: Industrial and educational applications. *CIRP Ann. Manuf. Technol.* **58**(2), 543–565 (2009)
42. Shannon, R., Johannes, J.D.: Systems simulation: the art and science. *IEEE Trans. Syst. Man Cybern.* **6**(10), 723–724 (1976)
43. Schotborgh, W.O., McMahon, C., Van Houten, F.J.: A knowledge acquisition method to model parametric engineering design processes. *Int. J. Comput. Aided Eng. Technol.* **4**(4), 373–391 (2012)
44. Kossiakoff, A., Sweet, W. N., Seymour, S., Biemer, S.M.: *Systems Engineering Principles and Practice*. John Wiley & Sons (2011).
45. Pahl, G., Beitz, W., Feldhusen, J., Grote, K.H.: *Engineering Design—A Systematic Approach*, 3rd edition ed. Springer (2007)
46. Acheson, P.: Methodology for object-oriented system architecture development. In: Systems Conference, 2010 4th Annual IEEE, pp. 643–646: IEEE (2010)
47. Yaroker, Y., Perelman, V., Dori, D.: An OPM conceptual model-based executable simulation environment: implementation and evaluation. *Syst. Eng.* **16**(4), 381–390 (2013)
48. Roy, R., Hinduja, S., Teti, R.: Recent advances in engineering design optimisation: Challenges and future trends. *CIRP Ann. Manuf. Technol.* **57**(2), 697–715 (2008)
49. Osorio, C.A., Dori, D., Sussman, J.: COIM: an object-process based method for analyzing architectures of complex, interconnected, large-scale socio-technical systems. *Syst. Eng.* **14**(4), 364–382 (2011)
50. Dou, K., Wang, X., Tang, C., Ross, A., Sullivan, K.: An evolutionary theory-systems approach to a science of the ilities. *Procedia Computer Science*, 44, 433–442, (2015)
51. Yeo, S., Mak, M., Balon, S.: Analysis of decision-making methodologies for desirability score of conceptual design. *J. Eng. Des.* **15**(2), 195–208 (2004)
52. Abdoli, S., Kara, S.: Designing warehouse logical architecture by applying object oriented model based system engineering. *Procedia CIRP* **50**, 713–718 (2016)
53. Abdoli, S., Kara, S.: A modelling framework to design executable logical architecture of engineering systems. *Mod. Appl. Sci.* **11**(9), 75 (2017)
54. Abdoli, S.: Application of fuzzy-logic for design assessment of complex engineering systems in the early design stages. *J. Eng. Des.* **33**(3), 234–258 (2022)
55. Guimarães, A.C.F., Lapa, C.M.F.: Fuzzy inference to risk assessment on nuclear engineering systems. *Appl. Soft Comput.* **7**(1), 17–28 (2007)
56. Pedrycz, W., Ekel, P., Parreiras, R.: Selected design and processing aspects of fuzzy sets. In: *Fuzzy Multicriteria Decision-Making*. Wiley, (2010)
57. Bimba, A.T., et al.: Towards knowledge modeling and manipulation technologies: A survey. *Int. J. Inf. Manage.* **36**(6), 857–871 (2016)
58. Garcia-Cascales, M.S., Lamata, M.T.: Solving a decision problem with linguistic information. *Pattern Recogn. Lett.* **28**(16), 2284–2294 (2007)
59. Oduguwa, V., Roy, R., Farrugia, D.: Development of a soft computing-based framework for engineering design optimisation with quantitative and qualitative search spaces. *Appl. Soft. Comput.* **7**(1), 166–188 (2007)
60. Zeng, X.-J., Keane, J.A.: Approximation capabilities of hierarchical hybrid systems. *IEEE Trans. Syst. Man Cybernet. Part A Syst. Hum.* **36**(5), 1029–1039 (2006)
61. Youssef, A.M., ElMaraghy, H.A.: Modelling and optimization of multiple-aspect RMS configurations. *Int. J. Prod. Res.* **44**(22), 4929–4958 (2006)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.