designs

Article

An Integration Matrix for Investigating the Impact of Design Changes in Mechatronic Products

Tossaporn Assawarungsri * and Nattawut Janthong

Department of Production Engineering, Faculty of Engineering, King Mongkut's University of Technology North Bangkok, Bangkok 10800, Thailand
* Correspondence: tossaporn.as@gmail.com

Abstract: In the design process, design changes are unavoidable due to the need to meet customers' requirements and support future change through technology development. Although components are supposed to be renewed within existing designs, these changes can propagate into other parts due to their interfaces. Propagation makes it difficult for a designer to identify these changes. This study aimed to introduce the integration matrix (I-DSM), an approach to the design of mechatronic products that involves determining changes in existing products with an axiomatic design. Reverse zigzagging was used to break down the entire product to its lowest level. A design matrix (DM) was constructed and then transformed into a design structure matrix (DSM). The I-DSM consists of three layers: information technology, electrical technology, and mechanical technology. The breadth-first search (BFS) method was employed to ascertain the change propagation path in order to consider it. After this, the changing workload was analyzed, and the decision-making process was used to determine the best possible option. Finally, an automatic guided vehicle was used in a case study to demonstrate the use of this methodology by showing how changes in a product can affect it and how a designer can prioritize activities.

Keywords: engineering change; axiomatic design; design structure matrix; breadth-first search; change propagation; change decision; entropy method; decision making

Citation: Assawarungsri, T.; Janthong, N. An Integration Matrix for Investigating the Impact of Design Changes in Mechatronic Products. *Designs* **2023**, *7*, 16. https://doi.org/10.3390/designs7010016

Academic Editor: Loránd Szabó

Received: 26 November 2022
Revised: 6 January 2023
Accepted: 8 January 2023
Published: 18 January 2023

1. Introduction

In engineering design, a change occurs at every stage of a product's life cycle and development process. A design change may be initiated for many reasons, such as to improve a design, to support customers' needs, to innovative technology development, to respond to legal and regulatory policy changes, and the pressure of competitiveness. In addition, the terms of the business process should provide new products and support after-sale and maintenance services, which include repairing, retrofitting, and renewing/renovating, in order to extend the lifetime of a product, because products become obsolete over time [1]. These are seen as challenges for the engineering designer, who must manage these changes as much as possible, including customer needs, product specifications, functioning, and the component aspects of an assembly. This implies that the initial design should adapt to accommodate new changes.

Regardless of the scope of the change, an engineering change (EC) is defined in [2] as a modification of components, drawings, or software that has already been made public during the product design process. The scope of the change can vary from minor adjustments to a single component to significant changes impacting the entire product. Similarly, in [3], it was pointed out that one of the crucial characteristics of design change is that it propagates. Such propagation can be challenging to predict, diagnose, and evaluate. This is particularly true if the design consists of numerous components, if the design concerns are tightly integrated, and/or if the design knowledge is dispersed among various specialists or organizations.

Various scholars have created methods for change propagation analysis (CPA) to assist with change management. As seen in [3], there are currently several methodologies and models. When determining the effects of a design change, the authors highlighted a significant problem. Due to the issue of change propagation analysis, which may enhance the improvement of designs with respect to future potential modifications, it is likely that other elements of a design will also need to be improved in order for them to continue to function when one of them is updated [4]. Designers can find change propagation in complex products through the methods proposed by many researchers [5–7]. Furthermore, design change propagation analysis [5,8], change effect evaluation [9,10], and design change routing [11–13] are the key research methodologies used in the field of design management. These studies, however, place a significant focus on an integrated approach; a product's structural characteristics should be considered when assessing how components and change propagation paths relate to one another.

Based on this issue, the identification and evaluation of change propagation paths were proposed in this study to support the improvement of designs and support technology evolution. To achieve the objectives of this study, the types of interface relationships between the components were defined. We also identified the change propagation path and determined how it directly impacts the entire system by evaluating the change workloads and ranking them.

As already indicated, this article's aim was to present an approach that can be used to overcome the following issues and difficulties:

(1) By modeling the structure of a current product, it is possible to assist in design progression and design improvements. The difficulty is that the new design or component must satisfy the current function and be compatible with minimal effects.
(2) There are extensive and intricate relationships between a mechatronic product's components due to the complex structure of mechatronic systems. These characteristics make it easier to quantify the interactions between the components and systematically develop mathematical models for mechatronic products.
(3) Identifying a change propagation path is difficult since a mechatronic system comprises many different designs, and any modification to one of them will affect the others. Therefore, the process must be straightforward so that this can be taken into consideration.
(4) With the guidance of activities from changing paths, a designer may analyze each path that must be identified and choose which path requires more work.

This article is organized as follows: Section 2 presents the related methods and research. In Section 3, the methodology is proposed. Section 4 illustrates the methodology of the case study. Finally, in Section 5, the conclusions and future work are presented.

2. Related Methods and Research

2.1. Mechatronic Systems

Since the late 1950s, system engineering has been promoted as a multidisciplinary approach and as a means of enabling successful system connections. Figure 1 illustrates how the spiral model and V model, which have frequently been used for system engineering, are insufficient in supporting technology integration and multidisciplinary perspectives in mechatronic design. System engineering is a method that helps engineers from several disciplines to work together to solve the ever-more-difficult problems associated with system engineering [14]. However, an applicable specification for the design of mechatronic systems is VDI 2206. It incorporates a domain-specific design more methodically than the V model. In addition, in this effective collaboration of mechanical engineering, electrical engineering, and information technology, greater focus has to be placed on the links between the subsystems of different design domains.

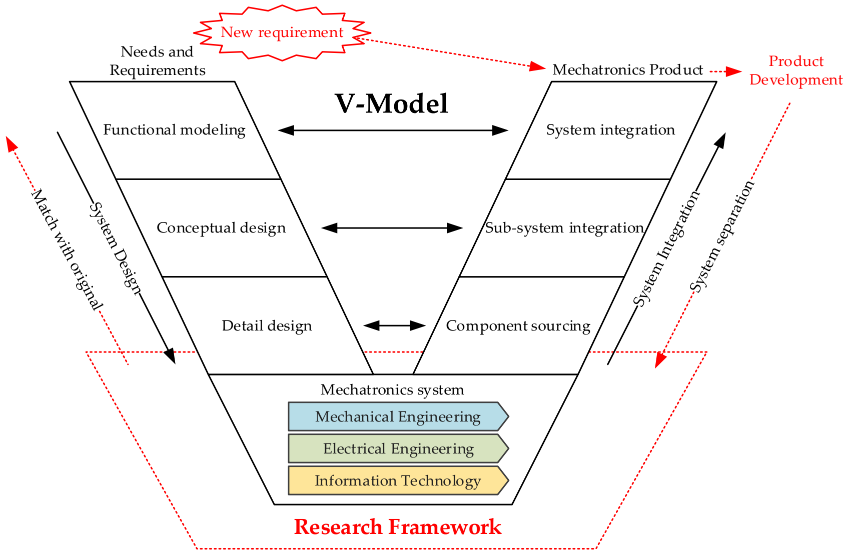


Figure 1. Modeling of mechatronic product investigation.

To solve mechatronic design challenges, a hierarchical design method is suggested by Zheng [14], in which discipline-specific design activities do not need to be integrated as a whole on the mechatronic level. In [15], the authors proposed a hierarchical model in the design process of a mechatronic system, which is a principal multidomain system using axiomatic design. It is possible to easily qualify how a product should be constructed to eliminate unnecessary iteration loops by analyzing the interconnections of the functional parameters.

2.2. Axiomatic Design Theory

The axiomatic design (AD) [16] method proceeds from a high level of abstraction to a detailed design element. A prescriptive structure of design hierarchy for the design component in each of the four domains—customer, functional, physical, and process—is produced by these activities of definition and detailing. The declaration of the design strategy at a lower level is impacted by the decisions taken at higher levels. To break down the design issue, the designer (or design team) follows a procedure wherein they zigzag between domains.

The design process is understood as a consecutive mapping between four different domains:

- Customer domain—customer attributes (CAs);
- Functional domain—functional requirements (FRs);
- Physical domain—design parameters (DPs);
- Process domain—process variables (PVs).

The AD is based on two axioms that include two aspects: (1) the independence axiom, which maintains the independence of functional requirements, and (2) the information axiom, which minimizes the information content. According to axiom 1, an ideal design preserves FR independence and states that changing one DP will satisfy a matching FR while having no impact on other FRs. On the other hand, according to AD, a design might be coupled (undesirable) or uncoupled (most preferred), depending on the design matrix produced through domain mapping. Figure 2 displays the design type, design matrix, design equations (X: influence; 0: no influence), and design procedure for the tree design characterizations.

FR and DP mapping are appropriately considered and extensively discussed in this study because these types of mapping concentrate on the design phase of a single product. Mathematically speaking, FRs and DPs may be described as matrices, and a design matrix can be used to visualize their connection. The following equation illustrates the resulting mathematical form:

$$\{FR\} = [DM]\{DP\} \tag{1}$$

$$[DM] = \begin{bmatrix} A_{11} & A_{12} & L & A_{1n} \\ A_{21} & A_{22} & L & A_{2n} \\ M & M & 0 & M \\ A_{m1} & A_{m2} & L & A_{mn} \end{bmatrix} \tag{2}$$

$$A_{ij} = \partial FR_i / \partial DP_j ; i = 1, 2, \ldots, m; \ j = 1, 2, \ldots, n \tag{3}$$

where $[A]$ is the design matrix (DM) that characterizes the design. Generally, each entry relates the *i*th FR to the *j*th DP. If the *i*th FR is affected by the *j*th DP, then A_{ij} has a finite value; otherwise, A_{ij} is zero. The matrix structure defines the design type being considered and is classified into three categories, as described in Figure 2.

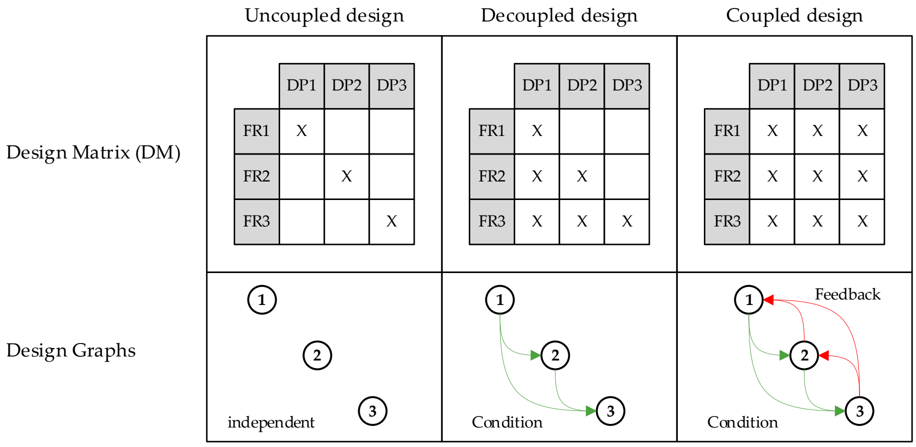


Figure 2. FR–DP relationships according to the design matrix adapted from [1].

The system interaction must be captured and analyzed as early as feasible for project management and recommendations, in order to establish a design and development process that produces high-quality goods better, quicker, and with lower costs, as suggested by [17]. However, AD cannot explain the system interactions for system integration, despite its strength in functional decomposition and mapping.

2.3. Design Structure Matrix

The design structure matrix (DSM), also known as the dependency structure matrix, has been widely utilized by researchers to express and evaluate complex system models. The DSM offers the benefits of clarity and simplicity in depiction. Additionally, it can indicate the essential patterns in system architectures (i.e., design architectures), such as modules and cycles, when supported by suitable analysis. Domain-mapping matrices (DMMs) and multidomain matrices (MDMs), which have expanded the capabilities and uses of matrix-based models of complex systems and provide additional insights, were developed more recently as a result of the use of DSMs. In this era of ever-more complex projects, goods, processes, organizations, and other systems, such competencies have come to be seen as more significant and necessary than previous models [18].

A DSM can represent a system architecture regarding the relationships between its components—for example, the modeling of- a system is decomposed into subsystems. Intelligent decomposition or partitioning is essential to managing system complexity [19].

A design structure matrix (DSM) is a system for product design, organizational structure, and project management. Steward [20] created a DSM with the aim to depict system interactions. As mentioned in [21], the individual system elements of a domain are assigned to the row and column of a square matrix to form a DSM.

A DSM is a matrix representation of a directed graph. The graph node corresponds to the column and row headings in the matrix, and the arrows correspond to the marks inside the matrix. (There are different ways to build a DSM. For a complete description of this issue, refer to the DSM website at https://dsmweb.org (accessed on 26 June 2021) http://www.DSMweb.org). The example refers to Browning [18]; there is an arrow from element 1 to elements 2, 4, 5, and 6, and a mark (such as "X" or "●") is placed next to row element 1 and column elements 2, 4, 5, and 6 (see Figure 3). Generally, diagonal elements have no significance and are usually blacked out.

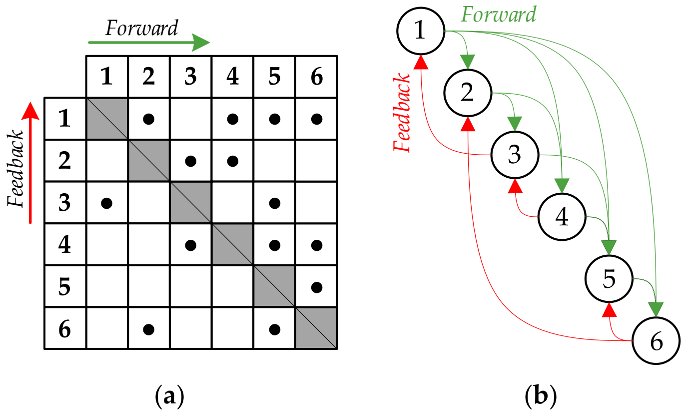


Figure 3. Example of relationship DSM, with (a) matrix relation and (b) node link diagram (directed graph).

Furthermore, the authors in [17,22] proposed that the DSM can assist in capturing the interactions between system elements. Similarly, the DSM is used to rebuild a process utilizing an integration matrix (I-DSM) that connects three layers (management, mechanical features, and control). This methodology can assist designers in analyzing the existing solutions and thus direct them toward design solutions [23].

Multiview feature modeling cannot enable product views and consistency management for company-level partnerships in which different product data might be utilized for product data views. Consequently, studies on EC management for complex engineering domains [2, 5, 24] have proposed a design structure matrix and network representation to preserve the constraints between nongeometric characteristics. Both techniques, however, exclude ECs as a change propagation pathway for consistency maintenance.

As mentioned above, the authors [17,22] proposed a transformation of the DM to DSM that can be described as the following steps:

(1) In each row of the DM, choose the dominant entry (X0 in DM);
(2) Construct a composite matrix (CM) to describe the equation relationship between FRs and DPs;
(3) To obtain the derived DSM, permute the CM by rearranging the rows and columns so that all dominating entries appear on the major diagonal. Such a conversion procedure is illustrated by the straightforward example in Figure 4.

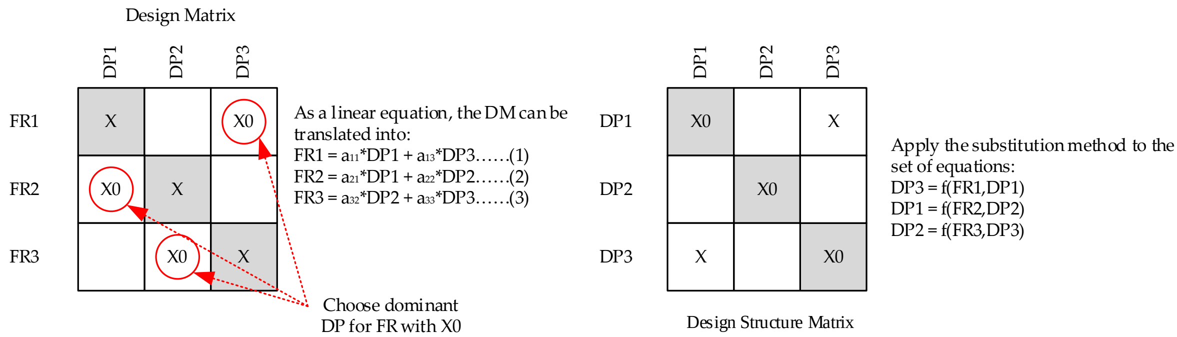


Figure 4. Conversion of DM to DSM.

2.4. Networks and Graphs

A graph mainly consists of directed or undirected nodes and edges. Nodes are entities, while edges simulate different types of relationships. Graph nodes are commonly regarded equally, which means that a system is highly abstract, and this is shown to be a significant issue in engineering applications. In this paper, a combination of a network and a matrix technique is demonstrated [25].

However, graphic techniques show advantages in viewing, statistical analysis, architectural properties, and big data. Furthermore, when different fields are considered in a graph, the level of detail and the potential explicative power of the model can be increased. In order to show the user patterns and other insights, however, better network visualization techniques are required.

Plehn [25] described the adjacency matrix A for a graph $G = (V; E)$ comprising a set of nodes V, and a set of edges E has the property $A(i, j) = 1$; if there is an edge $e_{ij} \in E$, linking nodes v_i, $v_j \in V$; otherwise, it is zero, as shown in the example in Figure 5.

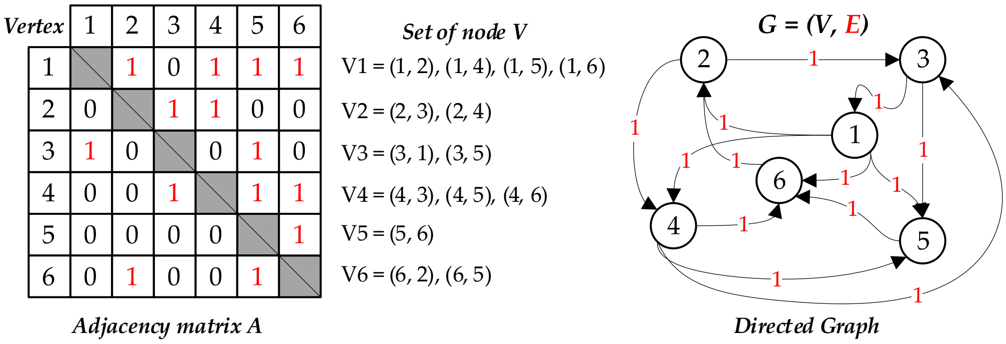
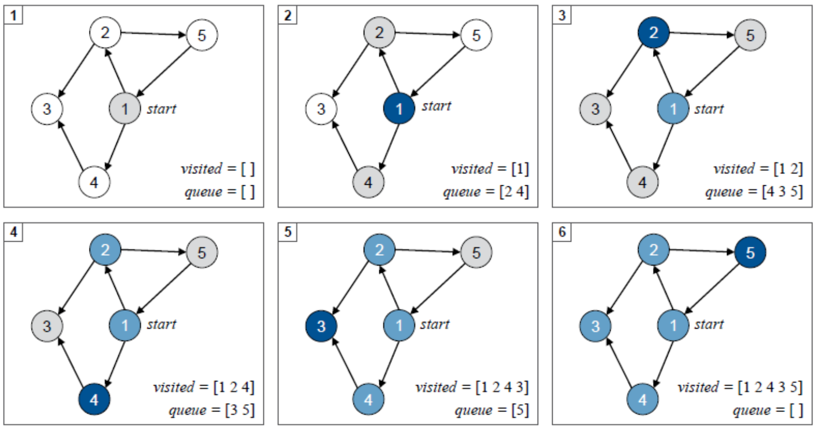


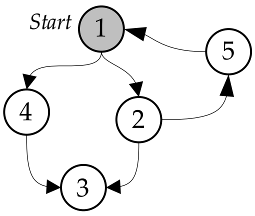
Figure 5. Adjacency matrix A and-directed graph $G = (V, E)$.

2.5. Breadth-First Graph Traversal

Breadth-first search (BFS) is a graph traversal technique invented by MOORE (1959). From the definition of graph theory, *G (V, E)* has vertices *(V)* and edge *(E)*, and all the nodes within the distance *(d)* or weight *(w)* edge traversal of the root node *s* are accessed. This indicates that the traversal starts with any vertex, and we visit every adjacent vertex of this node. Then, if this vertex has already been visited but is adjacent, we visit all adjacent vertices first. This is repeated until every vertex has been reached. BFS implementations generally employ queues to determine which nodes should be visited next, as shown in the example in Figure 6. Moreover, the pseudo-code for BFS can be found in the original work by Plehn [25].

Plehn [25] introduced BFS, before the basic idea of the CISGA was applied to discuss node visiting and propagation priority rules for the specification of change propagation behavior, as shown in the example in Figure 6.





Visit vertex	Queue
1	2 4
1 2	4 3 5
1 2 4	3 5
1 2 4 3	5
1 2 4 3 5	

Figure 6. Illustration of breadth-first graph traversal [25].

2.6. EM–TOPSIS

The multicriteria decision matrix (MCDM) method includes several techniques, one of which is the technique for order preference by similarity to the ideal solution (TOPSIS). TOPSIS attempts to rank the alternatives by calculating their distances (Euclidean distance) from the ideal and the opposite ideal solutions and then selects the best option with the shortest distance from the ideal solution and the highest distance from the opposite ideal solution. Therefore, the selection of attribute weights is a requirement when using TOPSIS. The analytic hierarchy process (AHP), the entropy method (EM), the deviation maximization method, the best–worst method, the variation coefficient method, etc., are a few methods that can be used to determine weights.

The entropy method (EM), also referred to as the entropy weight method (EWM) or Shannon entropy, is frequently used in a variety of research fields associated with TOPSIS [26–29], to make decisions or assess information, such as risk analysis, the evaluation of public blockchains, product design, performance evaluation of innovations, and real estate investment choices. The concept of EWM transforms the information data or alternative/criterion data considered in the quantitative ideal. It is established that the entropy weight index represents a value between 0 and 1 referring to the information data.

Hence, in this study, we used the EWM and TOPSIS to identify the best option because they are easy to calculate and do not require preferences to be taken into account. To calculate the weight, only objective data were needed; the calculation steps of EM–TOPSIS are provided in Section 3.2.3 (d).

3. Methodology

According to the aforementioned approach, the general demand is divided into individual requirements that correspond to each lower requirement and are followed by the operational subsystem in the functional domain. This consists of a mechanical layer that represents the requirements and solutions of a mechanical system. As indicated in Figure 7, the electrical layer represents the requirements and solutions of the electrical system, and the information technology layer represents the requirements and solutions of the information system. Furthermore, there is an interrelationship among components at each layer between the sub-solutions and the sub-requirements.

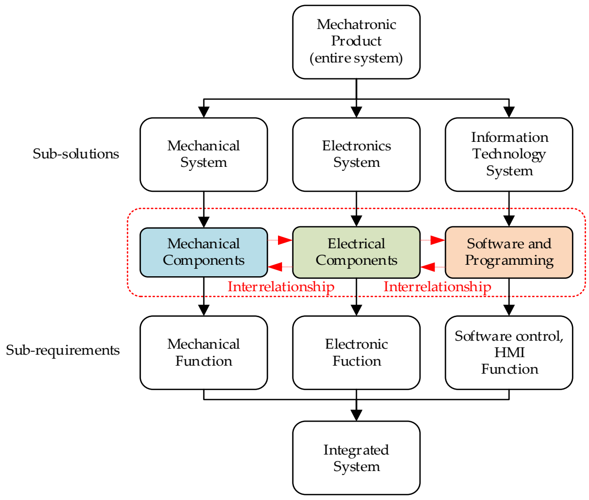


Figure 7. The mechatronic system's decomposition and the interrelationships of components.

From previous works, the AD, DSM, and design constraints were applied in the redesign process. This process helps the designer to analyze the elements that affect the changes in product design. The DPs, FRs, and their relationships leading to the identification of constraints were used for the redesign process, and axiom design constraints were also used. The constraints regulate the restructuring of the components that need to be amended to meet the new requirements. There needs to be evidence that the redesign process can thoroughly visualize the interaction between the components and change propagation.

According to research, the system of mechatronic products should be divided into sublayers and components with respect to the mechatronic discipline. It comprises systems for information technology, electronics, and mechanical components. Figure 8 illustrates how each subsystem is defined and constructed independently of the others, but, nevertheless, all subsystems must collaborate as a whole.

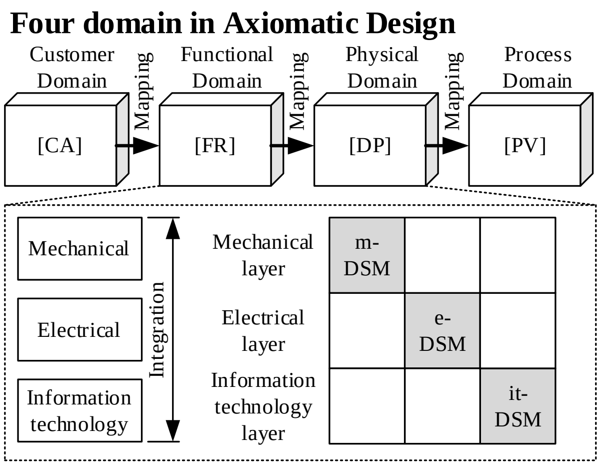


Figure 8. Four domains of AD and sublayers following the mechatronic discipline.

However, the I-DSM is not simple, compared with other technical developments, and this is a fundamental challenge. A method must be developed to satisfy the new customer's demand while maintaining compatibility with the current system. As a result, it is necessary to analyze an existing product's components in order to consider whether to update it.

This research methodology aimed to track the impact of change in mechatronic products from an existing product by applying AD and DSM to the integrated design matrix that was converted to I-DSM. Then, BFS was used to identify the change propagation path and analyze the initial components to change in the current product. The procedures are shown in Figure 9.

Methods/Tools	Research Methodology			Result
AD, DSM, WCS	Analyzing relationship of existing design			Integration Matrix
	Decomposition of the existing design	Covert to Design Structure Matrix	Construct Coherency Matrix	
Network theory, BFS, EM-TOPSIS	Evaluation of change propagation path with change workload			Optimal change propagation path
	Construct network model	Searching all change path	Qualifying change workload	

Figure 9. The framework of the proposed methodology.

3.1. Analyzing the Relationships of Existing Design

In an existing design, the relationships between components are typically established based on their roles, structures, and other properties. Exploring the functional and structural relationships between components and creating an accurate network model are crucial steps before optimizing a change propagation path, because, when a component's change parameter exceeds the tolerance of a structural or functional parameter, the adjacent nodes will also change, which is known as change propagation.

3.1.1. Decomposition of Existing Design

This approach was used to analyze and comprehend the structure of a current product in accordance with the axiomatic design theory [16]. The goal of a redesign process is expressed in terms of its functional requirements (FRs), and this is the main emphasis of axiomatic design. The distinguishing features of this approach include design parameters (DPs), the design matrix, and the breakdown. To meet the FRs, a designer determines the DPs (solutions). The most important aspect to note is that the decision regarding DPs to satisfy the FRs is guided by the axiomatic design process.

The connections between FRs and DPs are represented by design matrices. The degree of decomposition determines whether more decomposition to a higher level of FRs and DPs is required. Identifying the complex system is a straightforward approach. The highest level of the functional structure's abstraction should be followed when determining the design solution, and when higher levels of DP and FR links are broken down to the lowest level, the design solution should be identified. The decision regarding which subsystem or component implements this function will then be established by using a design matrix at the relevant level of abstraction.

Furthermore, Janthong [1] presented reverse zigzagging as an approach allowing a novice designer to examine and grasp the design rationale of an existing product. This method was developed by breaking down the product structure and design hierarchy to the lowest level of DP and FR linkages, as shown in Figure 10.

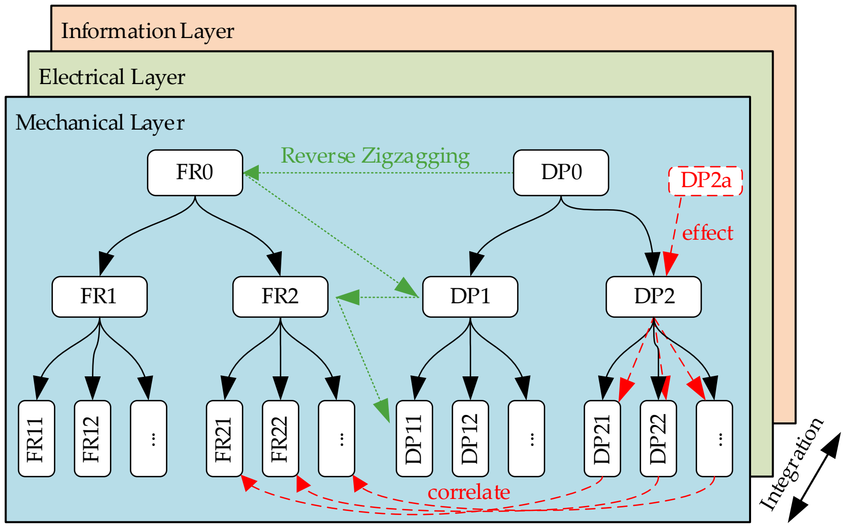


Figure 10. Reversing zigzagging method to decompose product structure, as adapted from [1].

Consequently, reverse zigzagging was used to divide a mechatronic product into three design matrices: the mechanical layer design matrix, the electrical layer design matrix, and the information technology layer design matrix.

3.1.2. Conversion of DM to DSM

The horizontal correlations of adjacent domains' information were recorded in the design matrices, with one design matrix for each node of the abstraction structure. The design matrix depicted in Figure 3 displays the identical horizontal correlations of two neighboring design domains (functional and physical). The correlations of the design matrix need to be determined by the independence axiom (as illustrated in Figure 2). The diagonal matrix represents an uncoupled design, which indicates that the elements are entirely independent of one another and may be constructed simultaneously. The decoupled design is described by a triangular matrix, which signifies that the FRs and DPs are not independent of one another, having a series of consequences on the behavior or design of one another. When the design matrix is neither triangular nor square, the design becomes linked. Any DP sequences in the linked design cannot meet the FRs. To summarize, both coupled and decoupled designs meet the independence axiom; however, uncoupled designs do not.

The DSM [20,21] was used to model the integration and connectivity (logical and physical) between the design embodiments of the system architecture and to trace the effects of this integration on the system's functionality. Dong suggested obtaining the DSM from the axiomatic design theory design matrix [24]. The author demonstrated that if the axiomatic design matrix can be analytically defined, and one design parameter (DP) is dominant in meeting a certain functional requirement (FR), the triangulated design matrix is identical to the design parameters' DSM. The researcher used this methodology to examine the interconnections between the layers in the integration matrix to promote technological evolution in the (re)design of complex products [22]. From the design matrix, each layer was transformed to obtain the design structure matrix. Tang [17] advocated for the use of DSM to improve AD in this regard.

In this section, the design matrix (DM) at each layer, i.e., the mechanical, electrical, and information technology layers, is transformed into DSM by using the principles mentioned earlier. Consequently, three DSMs are acquired. To support the design activity, the interactions between the design parameters of the three levels must now be identified and included in the model. Therefore, we propose a DSM of the mechatronic system that depicts DSM interactions at each demand level and across levels. Then, the effects of design changes are determined and summarized.

3.1.3. Construct Integration Matrix (Coherency Matrix)

The integration matrix is constructed using the three sub-DSMs. Figure 11 shows that the m-DSM, e-DSM, and it-DSM are placed on the diagonal of the integration matrix. The integration matrix is a nine-sector matrix, with the DSMs filling in only three diagonals. In most situations, particularly in industrial products, the interactions between levels follow the hierarchy of the technological level. As a result, the linkages may be characterized by four sectors.

To organize the sub-DSMs in an integration matrix into an integration matrix with a connecting matrix, {A}, {B}, {C}, and {D} are used to represent the mechanical, electronic, and information technology layers, respectively. The submatrices {A}, {B}, {C}, and {D} are generated by identifying the relationships between the design parameters (DPs) across layers or domains. The mechanical layer provides input to the electrical layer, represented by submatrix {A}, and the electronic layer then gives input to the information layer, represented by submatrix {B}. The information layer provides feedback to the electrical layer in submatrix {C}, and the electronic layer provides feedback to the mechanical layer in submatrix {D}.

Finally, Figure 11 depicts a multilayer product in the integration matrix, including a symmetric alignment of components on the axes and element groups of distinct layers. The integration matrix depicts the links between system components in a compact matrix representation of the system, allowing the visualization of interdependencies and interconnections and assisting in the exploration of the demands for information exchange. The matrix includes a list of all the interface types for each layer and the relevant information exchange and dependence qualification.

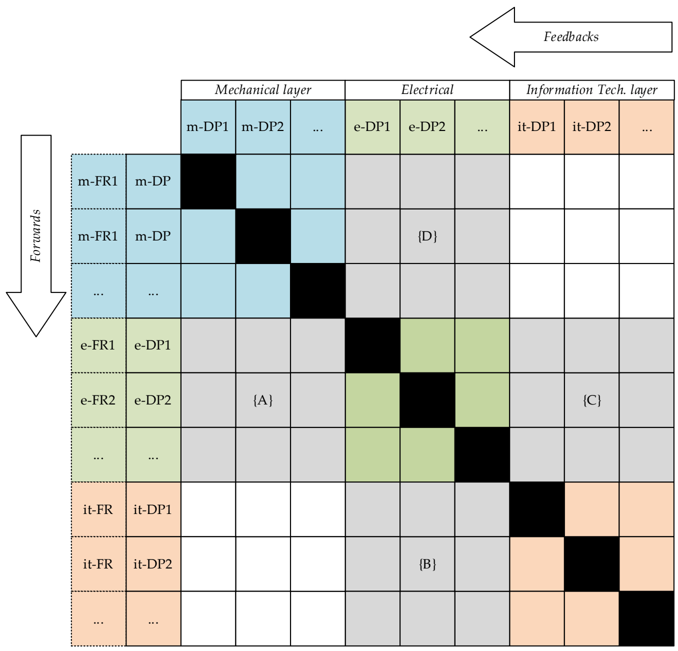


Figure 11. The integration matrix (I-DSM), adapted from [22].

To define the adjacency matrix, the I-DSM is unweighted. Thus, the I-DSM is represented as a binary matrix x, called an adjacency matrix, with all the components.

$$x_{ij} = \begin{cases} 1, \text{ if i and j are connected,} \\ 0, \text{ otherwise} \end{cases} \quad (4)$$

Thus, the adjacency matrix of the sample network is

$$x = \begin{pmatrix} x_{11} & \cdots & x_{1n} \\ \vdots & \ddots & \vdots \\ x_{m1} & \cdots & x_{mn} \end{pmatrix}_{m \times n} \quad (5)$$

Nevertheless, to define the interface types [30, 31], accurate definitions of interface types and their importance are fundamental for an accurate understanding of the product's architecture (coherency matrix). The interface type also affects the determination of the impact of design dependency. Moreover, Janthong [22] introduced a method to consider the design integration between layers and developed a scheme for the systematic identification of the interface type with four essential types of interaction between the elements from Pimmler and Eppinger [21], which include spatial, energy, information, and material components.

Therefore, in this work, we applied the classification of interface types from [30]. They classified the interface into six different types of interfaces: (1) attachment, (2) spatial, (3) power, (4) control and communication, (5) transfer, and (6) field. The definitions of the different types of interfaces are described in Table 1.

When attempting to measure design dependency, it becomes apparent that not all connections have the same intensity, and, as a result, not all connections have the same degree of design reliance. In order to determine the intensity of design dependencies utilizing the connection data from an I-DSM, we employed the idea that connections become more complicated as the number of connections between two components rises. The Weighted Complexity Score (WCS) was used to determine the relative strength of the links between components [31]. Each interface type was permitted to have a distinct weight due to the purpose measure. Equation (6) contains the resultant WCS formulation.

Table 1. Definition of interface types.

Interface Type	Notation	Definition of Interface Type
Attachment	A	A specific type of connector is needed for the structural connections between two components (e.g., bolts, screws, and rivets)
Spatial	S	Constraints relating to a component's geometry and location in relation to other components
Power	P	Contrary to the communications and control interface, the electrical connection between the two components
Control and communication	C	Communication or control of one component's state by another component through the exchange of signals or information between two components
Transfer	T	The flow of materials or power between components (e.g., water flow in a coffee maker, transfer of motion such as torque)
Field	F	The interaction between two components in which one component can generate heat, vibration, or magnetic field

$$WCS = n_{1i} + 2n_{2i} + 3n_{3i} + 4n_{4i} + 5n_{5i} + 6n_{6i} \quad (6)$$

where n_1, n_2, n_3, etc., represent the total number of interface types at complexity levels 1, 2, 3, etc., respectively.

When the quantity of interface types at complexity level 2, n_2, is multiplied by 2, the result is 2, which results in a value of WCS of 4. However, this strategy involves the assumption that all different interface types have equal value (e.g., an attachment interface is as complex as the quantity of fasteners' interface). The assumption is to identify the interface type but not assess the direction inside the system.

3.2. Evaluation of Change Propagation Path

Based on the associations between components (attachment, spatial, power, communication, transfer, and field), in this section, single-view networks are explained. The WCS then determines the network difference, edge weight, and direction of each single-view network in order to search for the change propagation path and determine the changing workload in the best possible way.

3.2.1. Construction of Network Model

As indicated in the previous section, the network model is applied from [32], in which each of the n components that compose the I-DSM is considered a set of vertices in the network model, and the set of vertices $V = \{v_1, v_2, \ldots, v_n\}$, where v_i is the ith part. In addition, a set of edges is $E = \{e_{i1}, e_{i2}, \ldots, e_{in}\}$, where e_{ij} denotes the connection between part v_i and part v_j. Finally, $W = \{w_{i1}, w_{i2}, \ldots, w_{in}\}$ are the real numbers weighted to the connections, where w_{ij} indicates the WCS between part v_i and part v_j., i.e., w_{ij} denotes the weights of edge e_{ij}.

Equations (5) and (6) show that $w_{ij} = WCN^*(x_{ij})$. Equation (7) describes the I-DSM network model. A direction-weighted network appears to be the network of the I-DSM model.

$$G_p = (V, E, W) \tag{7}$$

3.2.2. Searching All Change Paths

As mentioned above, the breadth-first search (BFS) algorithm was utilized to determine the change propagation path. The BFS algorithm performs graph traversal. All the nodes accessible from the root vertices s (start node) are visited in a "breadth-first" order; that is, all the direct neighbors of s are visited before proceeding to next-level neighbors. Thus, in a graph $G(V, E, W)$ with vertices V, edges E, and weighted W, all the nodes within the d edge traversals of the root nodes are accessed.

In this research, BFS defines the change path that the designer should consider while changing a component in the current system.

3.2.3. Qualifying Changing Workload

For this section, we referred to [33], which demonstrated the quantification of the change losses of every path. Hence, we evaluated the "changing workload" as the final comprehensive evaluation index to determine what the change path needs to consider first.

For the designer to make decisions, fewer, better solutions are identified from all possible paths based on an index called the "changing workload", which is defined as the change in all change propagation paths that began searching from the start node. This index is composed of three main indicators: the "network change rate", the "change magnification node rate", and the "change magnification rate". It measures the scope and intensity of the influence of change propagation on the network.

(a) Network change rate (NCR)

The effect on the size of the network model due to a change in customer needs is referred to as the amount of change propagation. The "network change rate (NCR)"—the ratio of the edges and nodes that are altered along a single change propagation path to the edges and nodes within the overall network—is used to quantify this impact. It is defined as follows:

$$NCR_c = \frac{M_c + N_c}{m + n} \tag{8}$$

where M_c and N_c represent the number of nodes and edges changed to satisfy the requirement in a single change propagation path, and m and n are the total number of nodes and the total number of edges in the network.

(b) Change magnification node rate (CMNR)

The degree of change propagation, which relates to the degree of influence inside the network model, is shown by the "change magnification node rate (CMNR)". The deeper the degree of change propagation, the more nodes there are in each change propagation path. As a result, the CMNR calculation equation is expressed as follows:

$$CMNR_i = \frac{CPI_i}{N_{total}} \tag{9}$$

where CPI_i represents the number of the change propagation index of the searching path starting node ith, and N_{total} is the total number of nodes implicated in this path.

The change propagation index determines the component type, i.e., absorber, carrier, or multiplier [34]. The only relationship between the change propagation index and the number of adjacency nodes is as follows:

$$CPI_i = \frac{x_{out}(i) - x_{in}(i)}{x_{out}(i) + x_{in}(i)} \tag{10}$$

The number of other nodes impacted by node i varies when $x_{out}(i)$, which is an indication that node i is out-degree. The number of nodes that can influence node i is represented by $x_{in}(i)$, which is the in-degree of node i. The ability to absorb the impact of change is improved with a propagation index that is lower and more inclined toward the absorber. In contrast, when the CPI increases, it becomes more inclined toward the multiplier and has a greater effect on the network's ability to propagate change.

(c) Change magnification rate (CMR)

The "change magnification rate (CMR)" measures the degree of CPI starting nodes; namely, it reflects the ability of a change node to propagate to all the nodes in the change path.

In the network, the degree of nodes and node strength are the main factors that determine the CMR. The degree of nodes indicates the number of other nodes directly associated with the change node. The larger the number of nodes directly associated with the change path, the stronger the node in the changing path; that is, the greater the node strength is, the higher the change magnification of the node is. Thus, the calculation of CMR is expressed as follows:

$$CMR_i = \frac{\sum w_{in}(i) + \sum w_{out}(j)}{w_{\max}} \tag{11}$$

where $w_{in}(i)$ denotes the in-degree of node strength, $w_{out}(j)$ denotes the out-degree of node strength, and w_{max} is the largest weight in each change path.

(d) Output a decision reference

In estimating the index weights, these approaches may provide different index weights for arbitrary reasons. At the same time, objective corresponding weight systems rely on the intrinsic data of indexes to generate index weights, which might eliminate human error and offer more accurate results.

The "changing workload", a comprehensive indicator, is obtained in this section using the entropy weight method (EWM) and the technique for order preference by similarity to the ideal solution (TOPSIS) from [33]. By computing the changing workload, as shown in Equations (12)–(23), a list of impacted components is generated, and a suggested order of propagation paths is defined using Equation (24).

$$A = (x_{ij}) = \begin{bmatrix} x_{11} & x_{12} & x_{13} \\ x_{21} & x_{22} & x_{23} \\ \vdots & \vdots & \vdots \\ x_{m1} & x_{m2} & x_{m3} \end{bmatrix} \tag{12}$$

the matrix A is a decision matrix (feasible alternative), which includes *NCR*, *CMR*, *CMNR*; $x_{11}, x_{21}, \ldots, x_{m1}$ are the evaluation criteria, and x_{ij} is the changing workload rating, as mentioned above.

Indices are transformed in a positive direction as follows:

$$x_{ij}^{'} = 1 - x_{ij}; i = 1, 2, \ldots, m; \ j = 1, 2, 3 \tag{13}$$

$$x_{ij}^{'} = \frac{1}{x_{ij}}; i = 1, 2, \ldots, m; \ j = 1, 2, 3 \tag{14}$$

The decision matrix is standardized as follows:

$$Z_{ij} = \frac{x_{ij}}{\sqrt{\sum_{i=1}^{m} (x_{ij})^2}}; i = 1, 2, \ldots, 3; \ j = 1, 2, 3 \tag{15}$$

$$Z = \left(z_{ij}\right) = \begin{bmatrix} z_{11} & z_{12} & z_{13} \\ z_{21} & z_{22} & z_{23} \\ \vdots & \vdots & \vdots \\ z_{m1} & z_{m2} & z_{m3} \end{bmatrix} \tag{16}$$

Entropy value of indices:

Given that there are m evaluation indexes and n evaluation problems, according to the concept of entropy, the entropy's index E_j is defined as follows:

$$E_j = -\frac{1}{\ln m} \sum_{i=1}^{m} b_{ij} \ln b_{ij}; i = 1, 2, \ldots, m; \ j = 1, 2, 3 \tag{17}$$

where

$$b_{ij} = \frac{z_{ij}}{\sum_{i=1}^{m} z_{ij}}; \ j = 1, 2, 3 \tag{18}$$

Weights of the indices:

$$\omega_j = \frac{1 - E_j}{\sum_{j=1}^{3} (1 - E_j)}; i = 1, 2, \ldots, m; \ j = 1, 2, 3 \tag{19}$$

$$W = \begin{bmatrix} \omega_1 & 0 & 0 \\ 0 & \omega_2 & 0 \\ 0 & 0 & \omega_3 \end{bmatrix} \tag{20}$$

The changing workload:

$$WP_i = \omega_1 NCR + \omega_2 CMR + \omega_3 CMNR \tag{21}$$

Determination of ideal solution:

$$H^{+} = \left\{ \max_i h_{ij} \mid i = 1, 2, \ldots, 3;\ j = 1, 2, 3 \right\}$$
$$H^{-} = \left\{ \min_i h_{ij} \mid i = 1, 2, \ldots, 3;\ j = 1, 2, 3 \right\} \quad (22)$$

Calculation of the separation measure:

Each feasible solution's separation from the ideal solution and the negative ideal solution is measured as

$$d_i^{+} = \sqrt{\sum_{j=1}^{n} \left(h_{ij} - h_j^{+} \right)^2};\ i = 1, 2, \ldots, m;\ j = 1, 2, 3$$
$$d_i^{-} = \sqrt{\sum_{j=1}^{n} \left(h_{ij} - h_j^{-} \right)^2};\ i = 1, 2, \ldots, m;\ j = 1, 2, 3 \quad (23)$$

where d_i^{+} is the separation from the ideal solution, and d_i^{-} is the separation from the negative ideal solution.

Calculation of the correlation of each change path:

$$C_i = \frac{d_i^{-}}{d_i^{+} - d_i^{-}};\ 0 \le C_i \le 1;\ \sum C_i = 1 \quad (24)$$

Finally, the optimal positive solution for the evaluation object is identified with the correlation of each change path, where C_i is closer to 1. Otherwise, the evaluation object's negative ideal solution is represented by a C_i value nearer to zero. The first change propagation path in the order is hence rather complicated when ranking the value of C_i. To change this approach, numerous procedures and additional components are required.

4. Illustration of the Methodology: A Case Study of an Automatic Guided Vehicle (AGV)

The AGV is a conventional, sophisticated mechatronic system with many different types of components and complicated interactions between them. The redesign of the AGV is necessary given the upsurge in market demand, customer demands, and supporting technological advancements. The integration matrix was employed to analyze the impact of change in an AGV to decrease the redesign complexity and product change difficulties, which satisfied the applicability requirements of the suggested technique in this study. The analysis reveals that the entire AGV model is readily impacted by the consumer demand and technology advancement, such as increased battery capacity, increased load capacity, etc., which necessitates the redesign of the AGV to satisfy consumers. Therefore, this section explains the breakdown of the current design, generates the integration matrix while also examining the interface type, and assesses which component will be most affected by changes.

4.1. Analyzing the Relationship of Existing Design

To analyze the current design, an AGV model was first used in association with the AD theory. The reverse zigzagging method was used to investigate the relationships between the DPs and FRs of the developed AGV, as illustrated in Figure 10. Later, as illustrated in Figure 11, the DPs, FRs, and interactions of components were organized and evolved into the design matrix.

There were three subsystems that contributed to the existing design of the AGV: a mechanical system, an electrical system, and an information technology system. All of the systems worked in concert to manage the vehicles used to carry materials in the manufacturing environment, which was accomplished through the information technology system. To drive the vehicle to a desired location, the electrical system was responsible for receiving the motion plan or trajectory from the master controller or information technology system. The vehicle's mechanical design included a variety of parts that were responsible for transporting the cargo to the desired location.

The AGV was arranged in a DM, composed of 50 components, and separated into 3 levels to specify each DP, FR, and their interactions. The DM showed how each layer's attributes related to one another (mechanical, electrical, and information technology). As illustrated in Figure 12 and Table 2, the details of the decomposed AGV component were defined by the mechanical layer (m-DPs and m-FRs), the electrical layer (e-DPs and e-FRs), and the information technology layer (it-DPs and it-FRs).

Then, the DMs were converted to DSMs. The relationship between the components' layers was determined by the I-DSMs, in which "1" means that a relationship exists between the design parameters, and blank means that there is no relation, as shown in Figure 13.

Each layer's FRs and DPs were constructed in order to demonstrate the AGV model. The DM and DSM both captured the links between DPs and FRs, as well as the relationships between DPs. As mentioned above, the I-DSMs were utilized to comprehend how the AGV, the electronic components, and the program statement of the AGV interacted. Attachment, spatial, power, communication, control, transfer, and field interface types were all created, as well as their relationships with the component's layers. Table 3 shows the type of interface index for each interaction between components, which had 81 linkages.

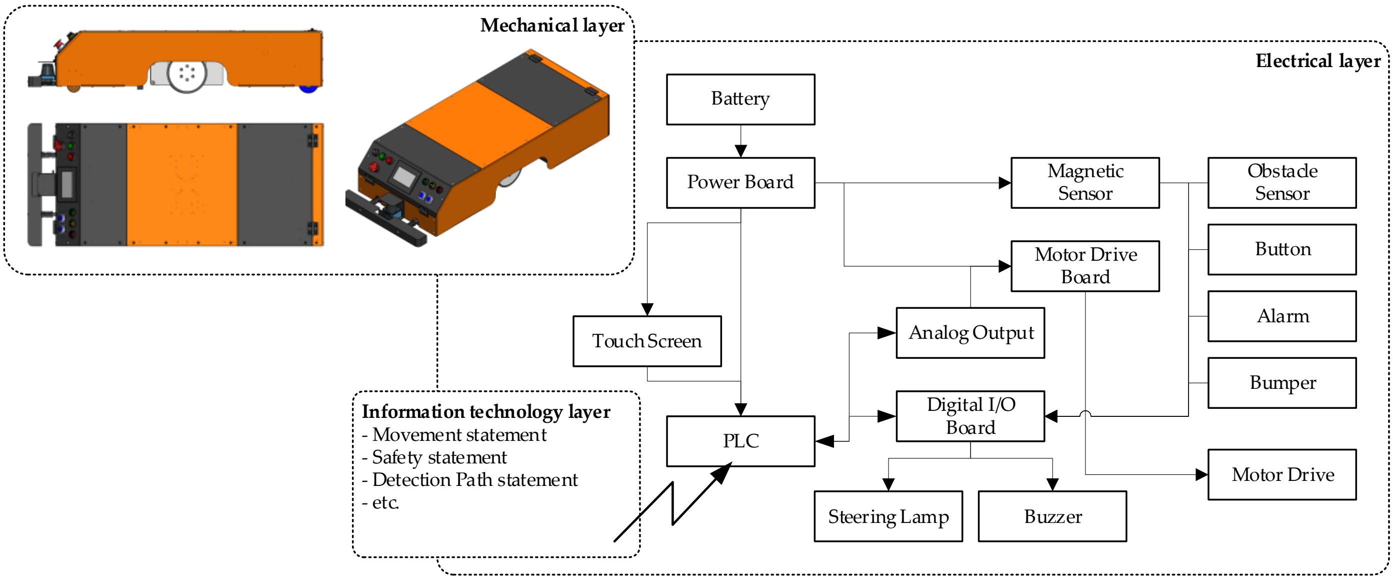


Figure 12. Automatic guided vehicle (AGV) model and mechatronic system.

Table 2. Division of components.

No.	Component Name	No.	Component Name	No.	Component Name
1.	Top plate	18.	Bearing nuts	35.	Digital I/O board
2.	Side plate	19.	C-ring	36.	Analog output
3.	Wheel bushing	20.	Washer	37.	Touch screen
4.	Magnetic guide mounting	21.	Chassis base	38.	Magnetic sensors
5.	Rib support	22.	Side cover	39.	Motor drive controller
6.	Washer lock bush	23.	Rear-wheel mounting	40.	Obstacle avoidance sensors
7.	Bottom plate	24.	Front cover	41.	Buttons
8.	Wheel	25.	Upper front cover	42.	Steering lamp
9.	Wheel outer	26.	Rear cover	43.	Alarm sensors
10.	Key	27.	Upper rear cover	44.	Bumper switch
11.	Bush rotor	28.	Middle top cover	45.	Buzzer
12.	Bearing housing	29.	Front-wheel mounting	46.	Motor
13.	End cap	30.	Sensor mounting	47.	Master controller
14.	Rotor mounting	31.	Bumper set	48	sm_Movement

15. Stopper rotation
16. Stopper cap
17. Angel bearing
32. Battery mounting
33. Battery
34. Power board
49 sm_Safety
50 sm_Detection Path

Mechanical layer: component numbers 1–32; electrical layer: component numbers 33–47; information technology layer: component numbers 48–50 (sm: program statement).

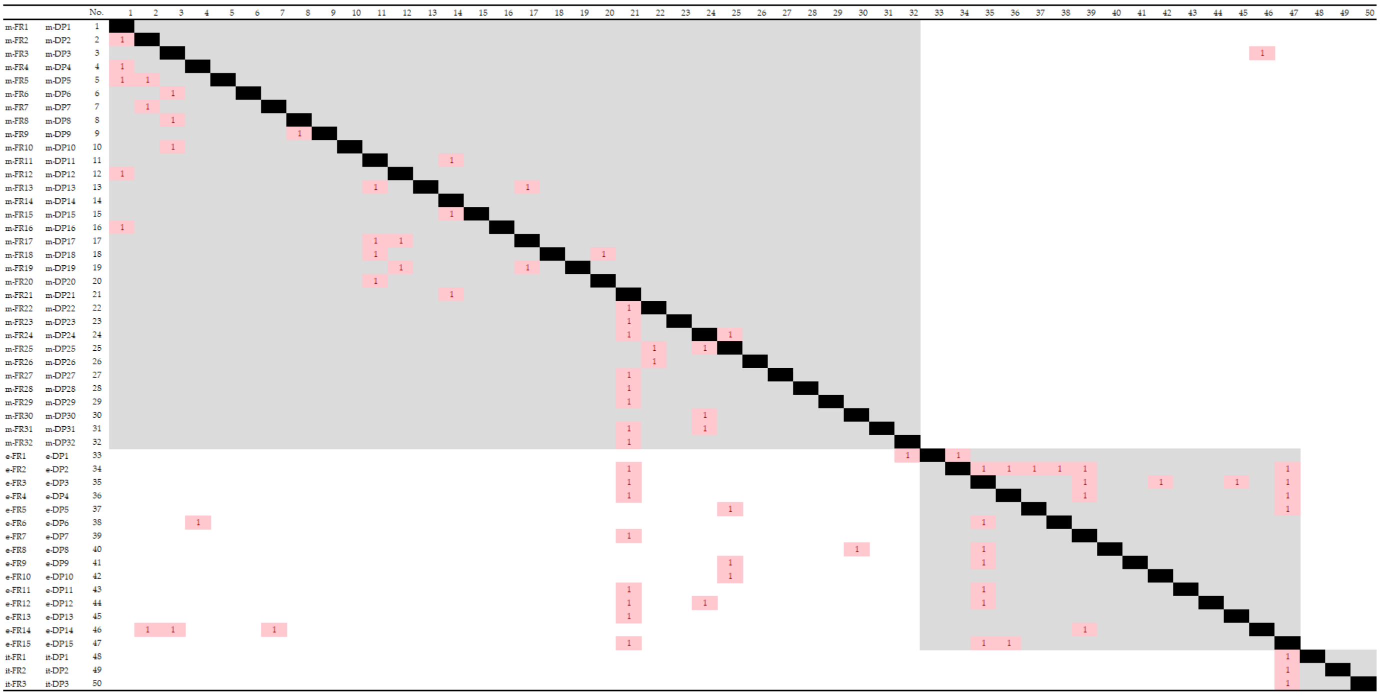


Figure 13. Design matrix of AGV.

Table 3. The type of interface index.

No.	Component Relationships		Type of Interface						Total Weight Interface	Level of Interface	WCS
	Source	Target	A	S	P	C	T	F			
1	2	1	4	1					5	2	10
2	3	46		1					1	1	1
3	4	1	2	1					3	2	6
4	5	1	2	1					3	2	6
5	5	2	2	1					3	2	6
6	6	3	1						1	1	1
7	7	2	2	1					3	2	6
8	8	3	6	1			1		8	3	24
9	9	8		1			1		2	2	4
10	10	3		1			1		2	2	4
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
73	46	3		1			1		2	2	4
74	46	7		1					1	1	1
75	46	39			1	1			2	2	4
76	47	21	4	1					5	2	10
77	47	35				1			1	1	1
78	47	36				1			1	1	1
79	48	47				1			1	1	1
80	49	47				1			1	1	1
81	50	47				1			1	1	1

As previously mentioned, this work was carefully performed by detecting the component's relationships with the interface type and analyzing the design dependencies for components' connections in the I-DSMs. The WCS method was used to assess the strength of the relationships between components. For example, the m-DP2 was connected to the m-DP1, and the interface type was attachment and spatial. As a result, the overall weight was 5, the level of interaction was 2, and the WCS was 10. The m-DP21 and m-DP24 had three interface types, i.e., attachment, spatial, and transfer, with a WCS of 18. The overall weighted I-DSM is shown in Figure 14.

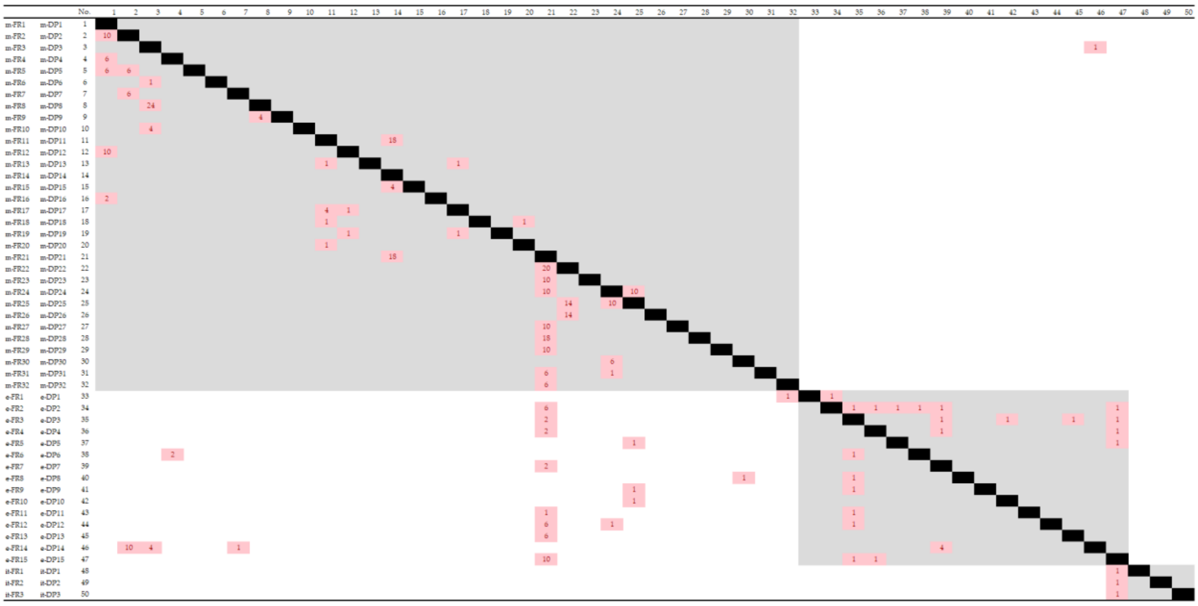


Figure 14. WCS of interface relationship between components.

4.2. Evaluation of Change Propagation Path

According to the previously mentioned network theory, the AGV model's network was implemented on a computer using MATLAB (R2022a). The networks were constructed with the I-DSMs in view. Consider the WCS association, which includes the node interface. To describe the directed graph, which contains the direction edges linking the nodes, the nodes in this network followed the components' relationships, as given in Table 3. Each edge represents a one-way relationship with the WCS, also known as a directed weighted graph. As a result, the network represented in Figure 15 was created using the components connected to the node (source and target).

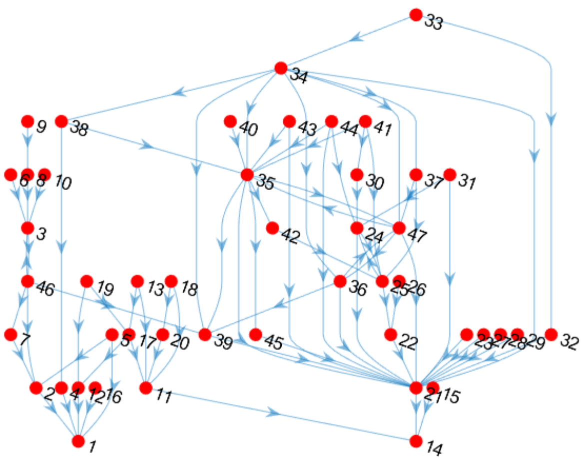


Figure 15. Network of AGV model.

By adopting the breadth-first search (BFS), which was used to identify the shortest path between the access nodes, the method for searching the change propagation path was created. A graph or a tree data structure was traversed using the BFS algorithm. The first changes in this network matched every node, as seen in Figure 16, which shows 50 alteration propagation routes.

Start Node																		
1																		
2	1																	
3	46	2	7	39	1	21	14											
4	1																	
5	1	2																
6	3	46	2	7	39	1	21	14										
7	2	1																
8	3	46	2	7	39	1	21	14										
9	8	3	46	2	7	39	1	21	14									
10	3	46	2	7	39	1	21	14										
11	14																	
12	1																	
13	11	17	14	12	1													
14																		
15	14																	
16	1																	
17	11	12	14	1														
18	11	20	14															
19	12	17	1	11	14													
20	11	14																
21	14																	
22	21	14																
23	21	14																
24	21	25	14	22														
25	22	24	21	14														
26	22	21	14															
27	21	14																
28	21	14																
29	21	14																
30	24	21	25	14	22													
31	21	24	14	25	22													
32	21	14																
33	32	34	21	35	36	37	38	39	47	14	42	45	25	4	22	24	1	
34	21	35	36	37	38	39	47	14	42	45	25	4	22	24	1			
35	21	39	42	45	47	14	25	36	22	24								
36	21	39	47	14	35	42	45	25	22	24								
37	25	47	22	24	21	35	36	14	39	42	45							
38	4	35	1	21	39	42	45	47	14	25	36	22	24					
39	21	14																
40	30	35	24	21	39	42	45	47	25	14	36	22						
41	25	35	22	24	21	39	42	45	47	14	36							
42	25	22	24	21	14													
43	21	35	14	39	42	45	47	25	36	22	24							
44	21	24	35	14	25	39	42	45	47	22	36							
45	21	14																
46	2	3	7	39	1	21	14											
47	21	35	36	14	39	42	45	25	22	24								
48	47	21	35	36	14	39	42	45	25	22	24							
49	47	21	35	36	14	39	42	45	25	22	24							
50	47	21	35	36	14	39	42	45	25	22	24							

Figure 16. The change propagation path of all start nodes and their members.

Figure 15 illustrates the network of the AGV model, and Figure 16 illustrates the search path of every node. Components 1, 2, 3, 14, 20, 21, 22, 35, 39, and 42 were classified as the absorb change nodes, components 20 and 42 belonged to the carry change nodes, and the remaining nodes of this network were multiplier change nodes, as shown in Table 4.

The NCR, CMNR, and CMR of all propagation paths were calculated with Equations (8)–(11). The results are shown in Table 5.

The optimal paths were ranked after obtaining the standardized score calculated by using Equations (12)–(24). . Each change propagation path's standardized score and ranking are shown in Table 6.

Table 4. In-degree, out-degree, and CPI of each node in the network model.

Part No.	In-Degree	Out-Degree	CPI	Part No.	In-Degree	Out-Degree	CPI	Part No.	In-Degree	Out-Degree	CPI
1.	34	0	−1.000	18.	0	2	1.000	35.	7	6	−0.077
2.	22	10	−0.375	19.	0	2	1.000	36.	2	4	0.333
3.	33	1	−0.941	20.	1	1	0.000	37.	1	2	0.333
4.	2	6	0.500	21.	125	18	−0.748	38.	1	3	0.500
5.	0	12	1.000	22.	28	20	−0.167	39.	7	2	−0.556
6.	0	1	1.000	23.	0	10	1.000	40.	0	2	1.000
7.	1	6	0.714	24.	18	20	0.053	41.	0	2	1.000
8.	4	24	0.714	25.	13	24	0.297	42.	1	1	0.000
9.	0	4	1.000	26.	0	14	1.000	43.	0	2	1.000
10.	0	4	1.000	27.	0	10	1.000	44.	0	8	1.000
11.	7	18	0.440	28.	0	18	1.000	45.	1	6	0.714
12.	2	10	0.667	29.	0	10	1.000	46.	1	19	0.900
13.	0	2	1.000	30.	1	6	0.714	47.	7	12	0.263

14.	40	0	−1.000	31.	0	7	1.000	48	0	1	1.000
15.	0	4	1.000	32.	1	6	0.714	49	0	1	1.000
16.	0	2	1.000	33.	0	2	1.000	50	0	1	1.000
17.	2	5	0.429	34.	1	12	0.846				

Table 5. Values of NCR, CMNR, and CMR of all propagation paths.

Path No.	NCR	CMNR	CMR	Path No.	NCR	CMNR	CMR	Path No.	NCR	CMNR	CMR
1	0.006	−1.000	0.238	18	0.044	0.250	0.483	35	0.132	−0.007	2.531
2	0.019	−0.188	0.462	19	0.069	0.167	0.839	36	0.132	0.030	2.531
3	0.094	−0.118	2.231	20	0.031	0.000	0.469	37	0.145	0.028	2.552
4	0.019	0.250	0.294	21	0.019	−0.374	1.280	38	0.170	0.036	2.853
5	0.031	0.333	0.545	22	0.031	−0.056	1.615	39	0.031	−0.185	1.343
6	0.107	0.111	2.238	23	0.031	0.333	1.350	40	0.157	0.077	2.594
7	0.031	0.238	0.510	24	0.057	0.011	2.140	41	0.145	0.083	2.545
8	0.107	0.079	2.427	25	0.057	0.059	2.140	42	0.069	0.000	2.154
9	0.119	0.100	2.455	26	0.044	0.250	1.713	43	0.145	0.083	2.545
10	0.107	0.111	2.259	27	0.031	0.333	1.350	44	0.145	0.083	2.587
11	0.019	0.220	0.455	28	0.031	0.333	1.406	45	0.031	0.238	1.329
12	0.019	0.333	0.322	29	0.031	0.333	1.350	46	0.094	0.113	2.231
13	0.069	0.167	0.839	30	0.069	0.119	2.189	47	0.132	0.024	2.531
14	0.006	−1.000	0.280	31	0.069	0.167	2.189	48	0.145	0.083	2.538
15	0.019	0.500	0.308	32	0.031	0.238	1.329	49	0.145	0.083	2.538
16	0.019	0.500	0.252	33	0.220	0.056	3.028	50	0.145	0.083	2.538
17	0.057	0.086	0.825	34	0.195	0.053	2.965				

Table 6. List of the standardized scores (EWM and TOPSIS) and ranking of the change propagation path.

Path No.	Standardized Score	Rank	Path No.	Standardized Score	Rank	Path No.	Standardized Score	Rank
1	0.000	50	18	0.238	43	35	0.802	14
2	0.172	48	19	0.297	36	36	0.805	12
3	0.700	22	20	0.201	47	37	0.812	11
4	0.217	46	21	0.377	35	38	0.890	3
5	0.258	39	22	0.506	27	39	0.410	34
6	0.719	18	23	0.449	29	40	0.829	4
7	0.240	42	24	0.680	25	41	0.815	6
8	0.777	16	25	0.683	24	42	0.684	23
9	0.787	15	26	0.556	26	43	0.815	6
10	0.725	17	27	0.449	29	44	0.827	5
11	0.230	45	28	0.466	28	45	0.436	32
12	0.230	44	29	0.449	29	46	0.716	19
13	0.297	36	30	0.703	21	47	0.805	13
14	0.014	49	31	0.705	20	48	0.813	8
15	0.251	40	32	0.436	32	49	0.813	8
16	0.246	41	33	0.914	1	50	0.813	8
17	0.284	38	34	0.910	2			

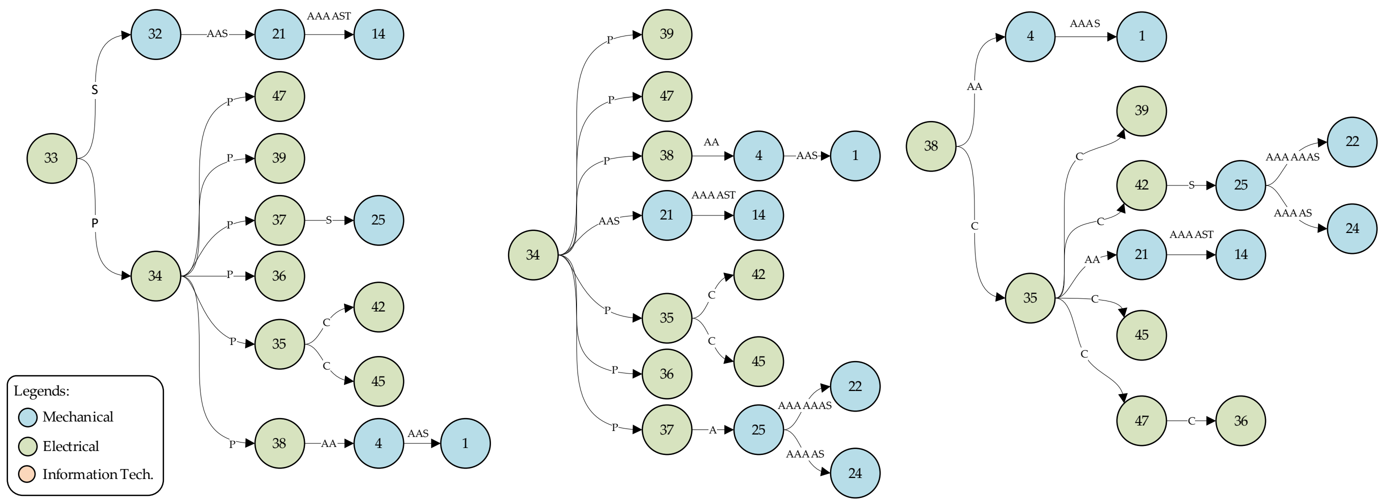
The case study's underlying premises, as shown in Table 6, indicate that component 33 (standardized score: 0.914) performed the highest among all the change propagation routes. The optimal path was compared with two more inferior alternatives, component 34 (rank 2; standardized score 0.909) and component 38 (rank 3; standardized score 0.890), as illustrated in Figure 17.

Compared with components 33 and 34 (rank 1 and 2), ~~an extra~~ the different change path included components 32, 21, and 14 (battery mounting, chassis base, and rotor mounting). The chassis (component 21) is a very important part of an AGV, and its working performance profoundly influences the AGV's operational load-bearing capacity and the installation of its support parts. The rotor mounting (component 14) element connects the chassis and the motor drive module, which is an important part of the movement of the AGV. Therefore, to avoid the challenges and risks of reacting to requirement changes throughout the design phase, these key features should not be changed.

Notably, every component in the change propagation path for ranks 1–3 was the same; the only component that differed from the others was component 32 (battery mounting) in rank 1. Additionally, although the propagation sequence could be different when component 33 was altered, another change propagation path still covered the original sequence.

The comparison of the change propagation path with different initial components is illustrated in Figure 18. Considering the change propagation path in the information technology layer, when they changed, rank 8 had a standardized score of 0.813 and was composed of the initial components 48, 49, and 50. These findings indicated that changes in information technology (programming) could impact the electrical and mechanical systems. Component 47 was affected, and lower-level fragments were transferred, as shown in component 48. Additionally, as depicted in the table, the component's interface type can be examined retrospectively. In the same way, ranks 15 to 17 (component 9, 8, 10) started with the mechanical layer's change node that affected the electrical layer but not the information technology layer, as illustrated in Figure 18.

In summary, the methodology used to choose this case study's most appropriate course of action is significant. In addition, it offers guidelines for designers to use when deciding which changes to make during the redesign process.



Rank 1; Standardized score: 0.914 Rank 2; Standardized score: 0.910 Rank 3; Standardized score: 0.890

Figure 17. Comparison of component paths 33, 34, and 38.

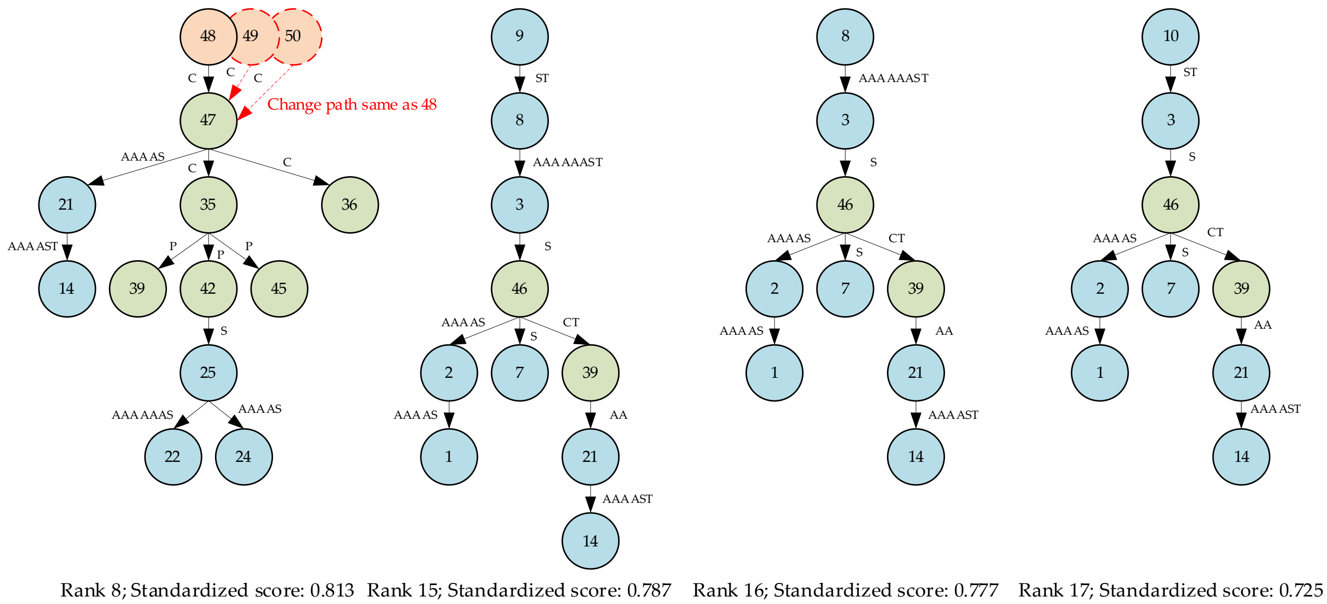


Figure 18. The initial component in other layers of the change propagation path.

4.3. Analyzing the Results and Discussion

4.3.1. Analyzing the Results

As mentioned in Sections 4.1 and 4.2, several methods were applied in this research. The effectiveness of many applications was analyzed, revealing our work's reasonable application in practice. The reverse zigzagging approach adapted from [1] was used in the analysis of the existing design stage to break down the current design of a mechatronic product and to reflect the technique of a descending order for the product structure and design hierarchy to the lowest level by applying AD (in the case of an AGV). The findings revealed three layers of multidisciplinary relationships between DPs and FRs at their fundamental level (mechanical, electrical, and information technology). Although this modeling technique does not depend on the designer's knowledge, it does result in a high workload for the designer if the existing design or product has more components.

Thus, to ascertain the relationships between components utilizing the same function, many authors [17, 22, 23] have proposed a transformation design matrix into DSM. The outcomes are displayed in Figure 13. The outcomes are displayed in the DSM, which includes three layers' DPs (m-DP, e-DP, and it-DP), and the interrelation among them is indicated with an index ("1" indicates relations). Furthermore, the DSM defines the types of interaction (spatial, energy, information, and material). Adopting this feature from [26], we expanded the categories of interaction to six (attachment, spatial, power, control and communication, transfer, and field). At this time, we found that the interrelation of members had a significant association with the attachment type, because most components are required to be installed in a specific area. Meanwhile, the other types had a single relation, as shown in Table 3. After evaluating the components' relationships, the types of interface index produced were calculated using WCS to create a weighted matrix (I-DSM).

An AGV network was built, and the BFS was used to determine the change propagation paths, as shown in Figure 16. The outcome revealed the elements of the propagation paths involved in changes in every component when propagation occurred. Designers can still assess activities in the redesign process using change propagation paths. However, these are not sufficient. The qualification of the changing workload is crucial in determining what the change path should consider first. According to Tables 4 and 5, the three indications that compose the changing workload—the network change rate, the change magnification node rate, and the change magnification rate—were determined from the change propagation path. Then, EM–TOPSIS [29] was used to obtain the changing workload for calculation. Table 6 displays the results. The path numbers were ranked based on a standardized score, which allowed the designer to perform preliminary supervision of the activities before starting the redesign process.

4.3.2. Discussion

The major topics of discussion and analysis in this section are modeling and evaluation. By comparing our method with other research approaches, the modeling aspect demonstrates how the current design was converted into information data. The purpose of the assessment aspect is to confirm the benefits of the suggested EM–TOPSIS when considering the most effective change propagation path.

Compared with two previous studies [1, 22], in terms of modeling efficiency, the modeling approach suggested in this study to analyze the information data in an original design provides a number of advantages. Due to the interconnections of the analyzed components, it is comparatively less difficult. Additionally, the consistency of the modeling developed using AD, DSM, and WCS does not require the involvement of experts, and the inaccuracies resulting from manual modeling by an assessor (designer), such as missing and erroneously filled information data, cannot be entirely avoided.

The BFS algorithm and EM–TOPSIS were used to analyze the change propagation impact and reflect the change propagation path, which is the most critical component of all mechatronic systems. This work employed techniques that help to locate the ideal solution, which can be compared with other studies [27, 29, 33]. These arguments showed that a designer can independently manage changes in design because the methods do not necessitate an expert to be involved in the evaluation step. The optimal change propagation path was demonstrated when the change initiated to a single component impacted the entire system. However, the cost and time were not included during the investigation of the impact of the design change procedure, as we mainly focused on the engineering changes.

5. Conclusions and Future Work

In this research, network theory was used to identify change propagation paths. A methodology was established to provide DMs, with a clear and simple framework for decision making during the redesign process. A list of the impacted components and the preferred sequence of propagation routes may be produced using this technique, which also enables the use of DMs to correctly and realistically estimate the changing workload. The complicated structure of an I-DSM may also be fully described by DMs using this tool.

Using axiomatic design and the design structure matrix throughout the redesign process allowed for the identification of the interface type by emphasizing the DPs, FRs, and their connections. Significantly, the interface type of the component will influence the customer's demand and means that the product must be adjusted to meet the new criteria, such as upgrading to new technology, increasing capacity, improving its efficiency, etc. Additionally, components with several functions must be developed by considering the existing components' relationships and how they affect the upgrading of new components.

Figure 17 displays the outcomes of the search path. By applying the BFS algorithm from the AGV model's network, the path of every node in all networks was revealed according to the changing workload of the optimal path. The weight evaluation (EWM and TOPSIS) of the "network change rate (NCR)", the "change magnification node rate (CMNR)", and the "change magnification rate (CMR)" comprised the "changing workload". The NCR is quantified as the propagation scale. The CMNR is quantified as the degree of CPI, and CMNR is quantified as the degree of nodes in the change path. The optimum change propagation scheme for an existing design can only be identified by design change propagation routing, as shown by comparing the optimal paths in Table 6.

This knowledge can help product designers to select the appropriate change dissemination strategy. However, it is still unable to optimize and enhance the elements that have a detrimental effect on the impact of change propagation in an existing design. Additionally, the data for the indicators in the methods suggested in this study were directly derived from the network model of the current architecture. They were independent of the designer's expertise.

In future work, we will refine and develop the methodology from this research and apply this methodology to assess an industrial mechatronic product. The relationship between components can provide a significant amount of information when applying cutting-edge data mining technology. Additionally, in this methodology, other elements still need to be determined to appropriately analyze the change components. In addition, the algorithm used to extract the information data of the design product should be considered in analyzing the relationships between the existing designs for quicker evaluation times, such as the algorithm for the extraction of the information data of a product's 3D model, which was introduced in [35]. Additionally, the evaluation procedure should consider the cost and time data.

Author Contributions: Conceptualization, T.A. and N.J.; methodology, T.A. and N.J.; software, T.A.; validation, T.A.; formal analysis, T.A.; investigation, T.A. and N.J.; resources, T.A.; data curation, T.A.; writing—original draft preparation, T.A.; writing—review and editing, T.A. and N.J.; visualization, T.A.; supervision, N.J.; project administration, T.A. and N.J.; funding acquisition, T.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Not applicable.

Acknowledgments: The authors would like to thank MINI RW COMPANY LIMITED for the case study illustrated in this research.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Janthong, N. Using Axiomatic Design and TRIZ in (re) designing of product architecture toward upgradable products. In *Advanced Materials Research*; Trans Tech Publications Ltd.: Wollerau, Switzerland, 2013; pp. 3141–3150.
2. Jarratt, T.A.W.; Eckert, C.M.; Caldwell, N.H.; Clarkson, P.J. Engineering change: An overview and perspective on the literature. *Res. Eng. Des.* **2011**, *22*, 103–124.
3. Brahma, A.; Wynn, D.C. Concepts of change propagation analysis in engineering design. *Res. Eng. Des.* **2022**, 1–35. https://doi.org/10.1007/s00163-022-00395-y
4. Eckert, C.; Clarkson, P.J.; Zanker, W. Change and customisation in complex engineering domains. *Res. Eng. Des*. **2004**, *15*, 1–21.
5. Clarkson, P.J.; Simons, C.; Eckert, C. Predicting change propagation in complex design. *J. Mech. Des*. **2004**, *126*, 788–797.
6. Cohen, T.; Navathe, S.B.; Fulton, R.E. C-FAR, change favorable representation. *Comput.-Aided Des.* **2000**, *32*, 321–338.
7. Silver, M.R.; De Weck, O.L. Time-expanded decision networks: A framework for designing evolvable complex systems. *Syst. Eng.* **2007**, *10*, 167–188.
8. Reddi, K.R.; Moon, Y.B. Simulation of new product development and engineering changes. *Ind. Manag. Data Syst.* **2012**, *112*, 520–540.
9. Wang, J.; Wei, F. Analysis and evaluation of the impacts on engineering change. *Ind. Eng. J.-GUANGZOU-* **2006**, *9*, 32.
10. Gumus, B.; Ertas, A.; Tate, D.; Cicek, I. The transdisciplinary product development lifecycle model. *J. Eng. Des.* **2008**, *19*, 185–200.
11. Li, Y.; Zhao, W.; Shao, X. A process simulation based method for scheduling product design change propagation. *Adv. Eng. Inform.* **2012**, *26*, 529–538.
12. Li, Y.; Zhao, W.; Ma, Y. A shortest path method for sequential change propagations in complex engineering design processes. *AI EDAM* **2016**, *30*, 107–121.
13. Li, M.; Sheng, J.; Li, Y.; Qi, W. Multi-objective path optimization for design change propagation on complex product. *Mach. Des. Manuf.* **2020**, *6*, 294–297.
14. Zheng, C.; Bricogne, M.; Le Duigou, J.; Eynard, B. Survey on mechatronic engineering: A focus on design methods and product models. *Adv. Eng. Inform.* **2014**, *28*, 241–257.
15. Hehenberger, P.; Poltschak, F.; Zeman, K.; Amrhein, W. Hierarchical design models in the mechatronic product development process of synchronous machines. *Mechatronics* **2010**, *20*, 864–875.
16. Suh, N.P.; Suh, N.P. *Axiomatic Design: Advances and Applications*; Oxford University Press: New York, NY, USA, 2001.

17. Tang, D.; Zhu, R.; Dai, S.; Zhang, G. Enhancing axiomatic design with design structure matrix. *Concurr. Eng.* **2009**, *17*, 129–137.
18. Browning, T.R. Design structure matrix extensions and innovations: A survey and new opportunities. *IEEE Trans. Eng. Manag.* **2015**, *63*, 27–52.
19. Browning, T.R. Applying the design structure matrix to system decomposition and integration problems: A review and new directions. *IEEE Trans. Eng. Manag.* **2001**, *48*, 292–306.
20. Steward, D.V. *Systems Analysis and Management: Structure, Strategy, and Design*; Petrocelli Books: 1981,New York, USA.
21. Eppinger, S.D.; Browning, T.R. *Design Structure Matrix Methods and Applications*. MIT Press: Cambridge, MA, USA, 2012.
22. Dong, Q.; Whitney, D.E. Designing a requirement driven product development process. In *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*; American Society of Mechanical Engineers: New York, NY, USA, 2001; pp. 11–20.
23. Janthong, N.; Brissaud, D.; Butdee, S. Integration matrix to support technology evolution in (re) design of complex products. In Proceedings of the 2010 IEEE International Conference on Management of Innovation & Technology, Singapore, 2–5 June 2010; pp. 485–490.
24. Keller, R.; Alink, T.; Pfeifer, C.; Eckert, C.M.; Clarkson, P.J.; Albers, A. Product models in design: A combined use of two models to assess change risks. In *DS 42, Proceedings of the ICED 2007, the 16th International Conference on Engineering Design, Paris, France, 28–31 July 2007*; pp. 673–674 (exec. Summ.), full paper no. DS42_P_439.
25. Plehn, C. *A Method for Analyzing the Impact of Changes and Their Propagation in Manufacturing Systems*; Herbert Utz Verlag: München, Germany, 2018.
26. Huang, J. Combining entropy weight and TOPSIS method for information system selection. In Proceedings of the 2008 IEEE Conference on Cybernetics and Intelligent Systems, Chengdu, China, 21–24 September 2008; pp. 1281–1284.
27. Li, X.; Wang, K.; Liu, L.; Xin, J.; Yang, H.; Gao, C. Application of the entropy weight and TOPSIS method in safety evaluation of coal mines. *Procedia Eng.* **2011**, *26*, 2085–2091.
28. Zhu, Y.; Tian, D.; Yan, F. Effectiveness of entropy weight method in decision-making. *Math. Probl. Eng.* **2020**, 1-5. https://doi.org/10.1155/2020/3564835.
29. Chen, P. Effects of the entropy weight on TOPSIS. *Expert Syst. Appl.* **2021**, *168*, 114186.
30. Jung, S.; Asikoglu, O.; Simpson, T.W. A method to evaluate direct and indirect design dependencies between components in a product architecture. *Res. Eng. Des.* **2018**, *29*, 507–530.
31. Dobberfuhl, A.; Lange, M.W. Interfaces per module: Is there an ideal number? In Proceedings of the International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, San Diego, CA, USA, 30 August–2 September, 2009; pp. 1373–1385.
32. Li, Y.; Chu, X.; Chu, D.; Liu, Q. An integrated module partition approach for complex products and systems based on weighted complex networks. *Int. J. Prod. Res.* **2014**, *52*, 4608–4622.
33. Dong, C.; Yang, Y.; Chen, Q.; Wu, Z. A complex network-based response method for changes in customer requirements for design processes of complex mechanical products. *Expert Syst. Appl.* **2022**, *199*, 117124.
34. Giffin, M.; de Weck, O.; Bounova, G.; Keller, R.; Eckert, C.; Clarkson, P.J. Change propagation analysis in complex technical systems. *J. Mech. Des.* **2009**, *131*, 081001.
35. Wang, S.; Li, Z.; He, C.; Liu, D.; Zou, G. An Integrated Method for Modular Design Based on Auto-Generated Multi-Attribute DSM and Improved Genetic Algorithm. *Symmetry* **2021**, *14*, 48.