

Article

A Case-Study-Based Comparative Evaluation of Functional Analysis Paradigms in Aircraft System Design

Haomin Li ¹, Meng Zhao ^{2,*}, Yong Chen ² and Youbai Xie ³¹ Shanghai Aircraft Design and Research Institute, Shanghai 201210, China; lihaomin@comac.cc² School of Aeronautics and Astronautics, Shanghai Jiao Tong University, Shanghai 200240, China;³ School of Mechanical Engineering, Shanghai Jiao Tong University, Shanghai 200240, China; aerocy@sjtu.edu.cn (Y.C.)

* Correspondence: magicmeng@sjtu.edu.cn

Abstract

Functional analysis plays a critical role in early-stage aircraft system design by defining system functions that guide downstream architectural development and verification. In practice, many design deficiencies originate not from incorrect physical realization but from incomplete or ambiguous functional definitions established at conceptual stages. This challenge is particularly pronounced in aircraft systems, where interaction- and physical-effect-induced functions tend to remain implicit and weakly justified. To address this issue, in this study, we conduct a case-study-based comparative evaluation of three functional analysis paradigms: design-theory-oriented functional decomposition, systems-engineering-based functional allocation, and scenario-driven functional analysis. Using an aircraft ground deceleration scenario as a controlled context, this comparison examines how different function-derivation mechanisms influence the identification and justification of interaction- and effect-induced functions. Through structured cross-paradigm comparison, three distinct and, in principle, reproducible derivation mechanisms, namely decomposition-driven, responsibility-driven, and physical-effect-driven, are identified. In this study, the physical-effect-driven mechanism is examined through an effect-strengthened implementation of the scenario-driven paradigm. While all paradigms consistently identify mission-oriented functions, the examined scenario-driven implementation enhances transparency in functional justification and improves sensitivity to interaction- and effect-induced functions, thereby reducing the risk of omission during conceptual design. By formalizing these derivation mechanisms and clarifying their complementary roles, this study contributes to a clearer methodological understanding of functional identification in early-stage complex system design, while providing practical guidance for methodological selection and integration in aircraft system design.

Keywords: functional analysis; systems engineering; function-derivation mechanisms; aircraft system design; scenario-driven analysis; comparative evaluation; engineering applicability

Academic Editor: Kambiz Vafai

Received: 26 December 2025

Revised: 13 February 2026

Accepted: 17 February 2026

Published: 18 February 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

In early-stage design, functional analysis plays a central role in defining and structuring system functions, thereby providing the foundation for downstream development phases [1]. For complex engineering systems, however, design rework often originates

from incomplete or ambiguous functional definitions established at early design stages, rather than from incorrect physical realizations [2]. From an engineering practice perspective, such early-stage functional deficiencies tend to persist across development phases, even when later physical implementations are technically sound, requiring increased effort for system integration, verification, and change management [3].

In aircraft systems, such early-stage functional deficiencies can often be traced to the diverse origins of system functions. While some functions originate directly from mission objectives and are specified explicitly, others emerge during the design process. For instance, expected system–environment interactions under operational scenarios, understood here as a structured description of the interactions under defined operating conditions, may introduce additional sensing and monitoring functions to capture environmental parameters. In addition, mission-oriented functions may produce residual physical effects that necessitate compensating functions, such as thermal dissipation during aircraft braking.

Unlike mission-oriented functions, these interaction- and effect-induced functions tend to remain implicit at early design stages and are often identified through expert experience, rather than systematic engineering procedures. As a result, ambiguities and omissions are most likely to arise in this category and may propagate into architectural weaknesses and downstream challenges in system integration, verification, and certification.

Existing functional analysis approaches can be broadly categorized into three paradigmatic orientations, each characterized by a distinct function-derivation mechanism. The design-theory-oriented paradigm relies on top-down decomposition within the system boundary, providing a solution-neutral means of structuring mission-oriented functions [4]. The systems-engineering-based paradigm embeds functional analysis within traceable workflows, integrating stakeholder needs, operational scenarios, and responsibility allocation to support coordination and lifecycle traceability [5]. More recently, a scenario-driven paradigm has been proposed that derives functions from structured system–environment interactions under defined operational conditions. While many scenario-driven implementations emphasize interaction modeling, some further strengthen this paradigm by explicitly tracing physical effects revealed within operational scenarios [6].

Despite their respective strengths, these functional analysis paradigms differ substantially in how system functions are identified, structured, and justified, particularly those arising from system–environment interactions and physical effects. These differences stem primarily from the distinct function-derivation mechanisms embedded within representative implementations of each paradigm. Consequently, the extent to which derived functions, which are often implicit at early design stages, are identified with sufficient completeness and transparency varies significantly across approaches. However, few studies have examined these paradigms within a shared aircraft engineering context, and a consistent basis for comparing their functional coverage and underlying function-derivation mechanisms remains lacking.

Accordingly, in this study, we do not aim to introduce a new functional analysis method or to solve a specific aircraft design problem. Instead, its scientific objective is to formalize function-derivation mechanisms as an explicit analytical construct for comparing functional analysis paradigms at the conceptual design level. By examining how different paradigms identify, justify, and structure functions, particularly those arising from system–environment interactions and physical effects, we establish a mechanism-level comparative perspective that extends beyond engineering applications toward methodological clarification.

Using a representative aircraft ground deceleration scenario as a controlled engineering context, three paradigms are applied under comparable conditions. In this study, the scenario-driven paradigm is examined in its effect-strengthened form. Through this

structured comparison, we identify decomposition- and responsibility-driven derivation mechanisms, as well as a physical-effect-driven derivation mechanism implemented within the scenario-driven paradigm. By explicitly formalizing these derivation mechanisms and structuring the comparison according to transparent evaluation criteria, we establish a transferable framework for analyzing functional completeness and derivation transparency in alternative complex systems. Accordingly, the contribution of this research lies in providing a reproducible mechanism-level basis for comparative evaluation within conceptual design research.

The remainder of this study is organized as follows. In Section 2, we review design-theory-oriented, systems-engineering-based, and scenario-driven functional analysis paradigms, with an emphasis on their respective function-derivation mechanisms and relevance to early-stage aircraft design. In Section 3, we present the operational procedure of the effect-strengthened implementation of the scenario-driven paradigm, providing the analytical basis for its mechanism-explicit examination in the comparative study. In Section 4, we apply this procedure to an aircraft ground deceleration case during runway operations. In Section 5, we conduct a structured comparative evaluation from a mechanism-level perspective. Finally, in Section 6, we summarize the main findings and outline directions for future research.

2. Literature Review

The increasing complexity of modern engineering systems highlights the need for systematic analysis methods that support function definition and traceability at early design stages [7]. To address this need, three main categories of functional analysis paradigms can be distinguished: design-theory-oriented, systems-engineering-based, and scenario-driven. Each category offers a distinct mechanism for functional identification, characterized by different function-derivation principles. In this section, we review representative methods from each category, focusing on their ability to identify functions arising from system–environment interactions and physical effects, with particular attention to aircraft systems as interaction-intensive and safety-critical applications. For clarity, the scenario-driven paradigm considered in this study refers to its effect-strengthened form, in which analysis of physical effects is explicitly incorporated into the derivation process.

2.1. Design-Theory-Oriented Functional Analysis Based on Solution-Neutral Decomposition

From a function-derivation perspective, design-theory-oriented paradigms derive system functions primarily through hierarchical solution-neutral decomposition of internal input–output transformations. These approaches provide a structured and solution-neutral foundation for early-stage functional analysis by focusing on the abstract transformation of inputs into outputs within a defined system boundary. Representative methodologies, such as the systematic design approach proposed by Pahl et al. [4], guide designers from high-level functional descriptions toward realizable technical solutions through hierarchical decomposition. In this paradigm, functions are treated as abstract transformations progressively refined without explicit reference to physical implementation details.

Related frameworks further formalize the relationships between function, behavior, and structure. The function–behavior–structure model [8,9], the structure–behavior–function model [10], and the need–function–principle–structure model [11] articulate causal links between functional intent, system behavior, and physical embodiment, supporting systematic reasoning across abstraction levels. To improve clarity and standardization of functional descriptions, Hirtz et al. [12] introduced the functional basis, which defines functions using standardized verb–object pairs. Subsequent research demonstrated its

applicability in structured concept generation and functional search, showing how abstract functional representations can support early ideation and solution exploration [13,14].

Despite these advances, later studies have highlighted limitations related to usability, completeness, and analyst subjectivity, particularly when functional models must evolve across abstraction levels [15–18]. More fundamentally, design-theory-oriented approaches derive functions primarily through hierarchical decomposition of internal input–output transformations. Their function-derivation mechanism is driven by abstract transformation logic within the system boundary, rather than by operational context, system–environment interactions, or physical effects. As a result, functions that arise from external interactions or physical effects, such as sensing, monitoring, or dissipation-related functions, are often insufficiently captured in interaction-intensive systems such as aircraft.

2.2. Systems-Engineering-Based Functional Analysis Based on Responsibility-Driven Traceable Mapping

From a derivation-mechanism perspective, systems-engineering-based paradigms identify functions through responsibility-driven and traceable mappings across stakeholder needs, operational artifacts, and architectural elements. These approaches integrate functional modeling with stakeholder needs, operational scenarios, architectural definition, and requirement traceability, and are commonly implemented using SysML-based modeling environments [19]. Within this paradigm, functional analysis is closely coupled with requirement refinement, behavioral modeling, and architectural consistency.

Representative methods include MagicGrid, which provides a matrix-based structure to relate problem-domain artifacts (e.g., stakeholder needs and use cases) to solution-domain elements (e.g., logical functions and architectures) [20]. Other formal approaches, such as Object-Process Methodology (OPM) [21], integrate structure and behavior through object–process pairs, enabling simulation and reasoning within a unified modeling framework. Recent studies have further explored SE-integrated approaches for constructing functional architectures from operational context, aiming to improve traceability and internal consistency in early design phases of cyber-physical systems [22,23].

Although systems engineering has matured as a discipline with structured processes and traceable workflows, recent research highlights that its modeling logic, particularly when mapped onto the function-behavior-structure ontology, exhibits distinct emphases compared to other design paradigms. For instance, SE models tend to prioritize requirement refinement and behavior verification over early function derivation or structural elaboration [24].

Systems-engineering-based approaches derive functions mainly through traceable mappings from stakeholder needs, use cases, and operational responsibilities to system behaviors and functional elements. Their function-derivation mechanisms emphasize *where* functions should be identified within the development process (e.g., from use cases or stakeholder responsibilities) but do not explain *how* to systematically identify functions based on physical flow mechanisms or environmental interactions. Consequently, while SE frameworks are effective in coordinating complex system interactions and maintaining traceability, the identification of derived functions, particularly those arising from environmental interactions and physical effects, often remains analyst-dependent, introducing ambiguity into early-stage functional definitions.

2.3. Scenario-Driven Functional Analysis with Effect-Strengthened Derivation

Scenario-driven functional analysis paradigms derive system functions from structured operational scenarios by explicitly modeling system–environment interactions under defined operational conditions. While many such paradigms emphasize interaction modeling, not all implementations explicitly incorporate systematic analysis of residual physical effects arising during state evolution. In this study, we focus on an effect-strengthened scenario-driven paradigm, as represented by Gui and Chen [6], in which material, energy, and information exchanges are explicitly traced to reveal physical effects within operational scenarios.

From a derivation-mechanism perspective, this effect-strengthened paradigm identifies functions when interaction processes expose physical effects that require compensation, regulation, or monitoring. By explicitly modeling unintended consequences (e.g., residual heat, energy losses, or state deviations), the paradigm enhances the systematic identification of interaction- and effect-induced functions at the conceptual design stage.

The underlying physical-effect-driven derivation mechanism is particularly suited to interaction-intensive and safety-critical systems such as aircraft, where many operationally relevant functions arise from state evolution under constrained operational conditions. However, despite its conceptual strengths, empirical comparisons with decomposition-driven and responsibility-allocation paradigms remain limited, motivating the comparative investigation conducted in this study.

2.4. Toward a Case-Study-Based Comparison of Function-Derivation Mechanisms

The review above shows that the three functional analysis paradigms adopt fundamentally different function-derivation mechanisms. Design-theory-oriented paradigms derive functions through hierarchical decomposition of internal input–output transformations, providing structural clarity but limited sensitivity to operational interactions and residual physical effects. Systems-engineering-based paradigms derive functions through traceable mappings from stakeholder needs, use cases, and operational responsibilities, offering strong lifecycle traceability while relying largely on descriptive constructs for function identification. The scenario-driven paradigm with effect-strengthened derivation, in contrast, derives functions through explicit analysis of system–environment interactions and physical effects revealed within operational scenarios, enabling interaction- and effect-induced functions to be systematically identified.

Although each paradigm offers distinct advantages, existing studies have rarely examined their implications within a shared engineering context using consistent comparison criteria. In particular, how different function-derivation mechanisms influence functional completeness, transparency, and traceability at the conceptual design stage remains insufficiently understood. This gap is especially relevant for aircraft systems, where many safety- and performance-critical functions emerge from system–environment interactions and physical effects rather than from mission objectives alone.

To address this gap, we conduct a case-study-based comparative investigation of the three paradigms using an aircraft ground deceleration scenario. This scenario involves both mission-oriented functions and derived functions arising from system–environment interactions and physical effects, providing a balanced and realistic testbed for comparison. By applying representative methods from each paradigm to the same scenario, we compare their function-derivation mechanisms and evaluate their effects on functional completeness and traceability. Through this focused comparison, the study aims to clarify the complementary roles and practical trade-offs of different functional analysis paradigms in early-stage aircraft system design.

3. Scenario-Driven Functional Analysis: Procedure and Derivation Mechanism

In this section, we present the operational procedure and associated function-derivation mechanism of the scenario-driven paradigm, examined in its effect-strengthened form [6]. The purpose is to explicate the mechanism through which this implementation derives and justifies functions, particularly those arising from system–environment interactions and residual physical effects, thereby establishing the analytical basis for subsequent mechanism-level comparison.

3.1. Analytical Inputs and Assumptions

The scenario-driven functional analysis procedure is intended for use in early-stage system design, where detailed physical architectures and numerical models are not yet available. The system under analysis is described at a conceptual level, focusing on operational behavior, interaction conditions, as well as material, energy, and information exchange characteristics, rather than on component-level implementations.

Physical behavior is represented in terms of material, energy, and information input/output together with associated state parameters that characterize the system and its interaction with the operational environment. These representations are qualitative or semi-quantitative and are sufficient to support function identification without requiring detailed simulations or sizing calculations.

The main inputs to the procedure include:

- an operational scenario describing how the system is expected to operate within its environment;
- a definition of context entities interacting with the system;
- a set of relevant state parameters that evolve during scenario execution;
- high-level descriptions of material, energy, and information associated with system operation.

3.2. Scenario Definition and Context Analysis

The first step in scenario-driven functional analysis is to establish the operational scenario and its context. The scenario describes the system's operation over time, including relevant operational phases, interaction conditions, and boundary situations that may influence system behavior.

Context entities interacting with the system—such as human operators, external systems, and environmental elements—are identified explicitly. For each context entity, relevant state parameters are defined for characterizing how the entity influences or is influenced by system operation.

By explicitly linking context entities to state parameters, the procedure establishes a structured representation of system–environment interactions. This representation provides the basis for identifying physically grounded input and output in subsequent steps, rather than treating operational context as an implicit background condition.

3.3. Identification of Inputs and Outputs

This step aims to explicitly reveal the physical consequences of system operation under a given operational scenario by identifying the dominant input and output across the system boundary. Given a defined scenario, the material, energy, and information exchanged between the system and its context entities are identified based on operational interactions and fundamental physical principles.

These inputs and outputs describe how physical quantities and signals are transformed, accumulated, or dissipated during scenario execution. By tracing their evolution,

the resulting physical effects are derived as inevitable consequences of system operation, such as energy dissipation, heat generation, load transfer, or information delay.

At this stage, physical effects are identified in a descriptive and physics-grounded manner only. They represent what necessarily occurs when the system operates as intended, regardless of whether these effects are already regulated or accommodated by the existing set of system functions. No judgment is made yet regarding functional sufficiency or the need for additional functions. This separation ensures that functional assumptions are not introduced prematurely and that subsequent function derivation is grounded in explicit physical reasoning.

3.4. Function Identification and Derivation

Function derivation in the scenario-driven approach is conducted in a hierarchical and iterative manner. Starting from the overall system objective defined by the operational scenario, an initial functional decomposition is first performed based on logical and causal relationships between intended effects and required transformations, following solution-neutral principles as in classical approaches, such as Pahl's systematic design method.

Logical decomposition alone, however, does not guarantee functional completeness. For each derived function and sub-function, its material, energy, and information input–output are explicitly identified, and the corresponding physical effects are analyzed by repeating the procedure described in Section 3.2. This repetition ensures that physical consequences emerging at intermediate functional levels are not implicitly absorbed into higher-level descriptions.

A new function is introduced if and only if a physical effect leads to state parameters whose values or trends cannot be maintained within acceptable bounds by the existing set of functions under the given operational conditions. State parameters denote observable variables characterizing system behavior and system–environment interaction, and unbalanced state parameters serve as the explicit trigger for function derivation. This explicit trigger condition differentiates the approach from purely descriptive or experience-based function introduction.

Physical effects that give rise to such unbalanced state parameters are regarded as residual physical effects at the corresponding functional level. The notion of “residual” is therefore level-dependent, reflecting insufficiency of the existing functional set rather than an intrinsic property of the effect itself. When residual physical effects are identified, additional functions are introduced to regulate, dissipate, monitor, or compensate for the associated state imbalance.

3.5. Outputs and Traceability Artifacts

Applying the scenario-driven functional analysis procedure results in a structured functional model consisting of mission-oriented and derived functions. Each identified function is explicitly associated with:

- the operational scenario and relevant operational phases in which it arises;
- the context entities involved in the interaction;
- the material, energy, or information input–output that motivates the function;
- the state parameters whose evolution or imbalance triggers its introduction.

This explicit linkage establishes causal traceability from operational context and physically grounded input–output transformations to functional definitions. As a result, the rationale for each function is transparent and inspectable, and the distinction between mission-oriented and derived functions is preserved.

3.6. Role of Analyst Judgment and Method Limitations

As with other early-stage design methods, analyst judgment is inevitably involved in applying the scenario-driven functional analysis procedure. Such judgment is required primarily when defining scenario boundaries, selecting relevant context entities and state parameters, and determining appropriate levels of abstraction for representing physical effects and their associated state evolution within the operational scenario. The resulting function set serves as a functional baseline that can be used independently or as a common reference for the comparative evaluation against other functional analysis approaches that we present in Section 5.

However, the role of analyst judgment through this approach is constrained by explicit derivation rules established in Sections 3.1–3.4. In particular, functions are not introduced solely based on prior design experience or heuristic expectations, but only when justified by observable physical effects and unbalanced state parameters revealed through scenario- and effect-driven analysis. This constraint reduces arbitrariness in function identification and improves transparency and reproducibility compared with purely descriptive or experience-driven approaches.

The effectiveness of this approach is, therefore, dependent on both the completeness and representativeness of the operational scenarios considered, as well as the analyst's ability to identify dominant physical effects and meaningful state parameters at the conceptual level. If critical scenarios or interaction conditions are omitted, corresponding functions may remain undiscovered. Similarly, excessively coarse abstraction may obscure relevant physical effects, while overly detailed modeling may exceed the intended scope of early-stage analysis.

Accordingly, the scenario-driven functional analysis approach is particularly suited to conceptual design phases in which operational scenarios can be reasonably defined and dominant physical effects can be qualitatively or semi-quantitatively characterized, while detailed component-level architectures are not yet fixed. Within this scope, the method provides a structured basis for relating operational interactions and residual physical effects to function identification, while clearly delineating the conditions under which its derivation mechanism remains valid.

4. Application to an Aircraft Ground Deceleration Case

This section applies the functional analysis procedure that we present in Section 3 to a case study of aircraft ground deceleration during runway operations. Beyond serving as an illustrative example, this case is selected because it exhibits strong coupling between operational phases, system–environment interactions, and residual physical effects. The case study, therefore, provides a suitable and representative context for examining how functions can be explicitly identified and justified through physically grounded analysis.

4.1. Scenario Definition and Context Analysis for Aircraft Ground Deceleration

The operational scenario that we consider in this study corresponds to aircraft ground deceleration following touchdown during landing. The system of interest is defined as the aircraft during the landing roll, with particular emphasis on the braking-related subsystems operating in interaction with the airframe, landing gear, runway surface, and surrounding environment. Human operators, onboard control systems, and external environmental elements are treated as relevant context entities whose interactions influence system behavior and functional requirements.

To capture phase-dependent behavior, the ground deceleration scenario is divided into several key operational phases, including touchdown, brake engagement, active deceleration, and transition to taxiing. For each phase, relevant state parameters are identified to characterize the evolving system behavior and interaction conditions. These parameters include aircraft ground speed, wheel rotational speed, braking force, normal

load on the landing gear, and accumulated kinetic and thermal energy. All parameters represent observable or measurable quantities that evolve as the scenario progresses and serve as the basis for the subsequent input and output identification.

Based on the scenario definition and the context analysis procedure that we describe in Section 3.2, the following context entities are explicitly identified for the ground deceleration scenario:

- Flight crew, acting as the primary human operators who issue braking and deceleration-related commands;
- Aircraft systems, including wheel braking systems, ground spoilers, and engines, which directly contribute to deceleration and energy dissipation;
- External entities, such as air traffic control, runway surface conditions, and prevailing weather, which impose operational constraints and influence interaction conditions;
- Passengers and cabin crew, who are indirectly affected by the deceleration process and whose safety must be ensured under all operational conditions.

For each context entity, relevant state parameters are identified to characterize system behavior and interactions during ground deceleration:

- Aircraft state parameters, including the aircraft ground speed, wheel rotational speed, braking force, normal load, and accumulated kinetic and thermal energy;
- Braking and control system parameters, including brake command signals, actuator response states, and feedback signals used for braking modulation and stability control;
- Environmental state parameters, including runway friction conditions, wind effects, and external operational constraints communicated by air traffic control;
- Human-related state parameters, including pilot braking commands and mode selections that influence system actuation and control logic.

By explicitly linking operational entities to phase-dependent state parameters, the scenario and context analysis establishes a structured basis for subsequently identifying material, energy, and information exchanges. This linkage enables operational interactions to be translated systematically into functional requirements, rather than being treated implicitly or inferred retrospectively.

4.2. Identification of Inputs and Outputs for Aircraft Ground Deceleration

Following the identification procedure that we describe in Section 3.3, the analysis at this stage focuses on identifying material, energy, and information that cross the system boundary during the execution of the ground deceleration scenario, without assessing functional adequacy.

From a material-exchanging perspective, interactions between the aircraft and the surrounding air, as well as between the landing gear and the runway surface, are identified as dominant contributors to deceleration-related behavior. These interactions generate aerodynamic drag and frictional forces that directly influence the aircraft's motion and load distribution.

From an energy-exchanging perspective, the aircraft's kinetic energy constitutes the primary input to the system during ground deceleration. This energy is transformed and dissipated through multiple physical mechanisms, including frictional heating in the wheel braking system, aerodynamic drag induced by ground spoiler deployment, and reverse thrust generated by the engines under specific operational conditions.

From an information-exchanging perspective, control commands and feedback signals are continuously exchanged between human operators, onboard control systems, and sensing elements. These inputs and outputs include pilot braking commands, automatic braking and spoiler deployment signals, as well as sensor feedback, such as wheel speed,

wheel load and brake temperature, which govern system actuation, monitoring and coordination.

In accordance with Section 3.3, the identification of material, energy, and information is performed descriptively and based on fundamental physical principles. The resulting physical effects, such as energy dissipation, thermal accumulation, load transfer and information latency, are treated as inevitable consequences of scenario execution, without yet judging whether the existing system functions are sufficient to regulate or accommodate them.

4.3. Function Identification and Derivation for Aircraft Ground Deceleration

Following the procedure defined in Section 3.4, system functions for the ground deceleration scenario are identified through two complementary steps. Logical functional decomposition is first used to establish the primary functions required to achieve the overall operational objective, while effect-driven evaluation is then applied to identify residual physical effects and unbalanced state parameters that trigger additional design-derived functions.

The process begins with the overall functional objective of the scenario, namely, *to decelerate the aircraft on the ground in a safe and controlled manner*. Based on logical causality, this objective is decomposed into a set of primary functions, each corresponding to a distinct physical mechanism contributing to the required state transition from high speed to low speed.

Based on the dominant input and output identified above, three mission-oriented functions are identified: wheel braking, ground spoiler deployment and engine reverse thrust generation. Each represents a non-substitutable transformation of material, energy or information that directly contributes to aircraft deceleration under specific operational conditions.

For each primary function, the associated input-output are identified, and the resulting physical effects are examined by repeating the effect-driven analysis described in Section 3.3. This step ensures that physical consequences arising at intermediate functional levels are not implicitly absorbed into higher-level functional descriptions.

When physical effects give rise to state parameters whose values or trends cannot be maintained within acceptable bounds by the existing functional set, additional functions are introduced in accordance with the trigger in Section 3.4. This criterion is applied directly in the ground deceleration scenario, as illustrated by the following examples.

During wheel braking, a substantial portion of the aircraft's kinetic energy accumulates as thermal energy in the brake assemblies. As this accumulation cannot be adequately regulated by the primary braking function alone, a design-derived function, namely, *dissipate wheel brake heat*, is introduced to maintain brake temperature within acceptable operating limits.

Similarly, safety-critical state parameters such as wheel speed and wheel load require continuous sensing and feedback. When delayed or unbalanced state information is revealed through effect-driven analysis, functions such as *generate anti-skid warning*, *provide wheel speed signals* and *output wheel load signals* are introduced to monitor and regulate these parameters.

Through this iterative combination of logical decomposition and effect-driven evaluation, both mission-oriented and derived functions are systematically identified and justified. Each function corresponds to a specific transformation or regulation of material, energy or information under the defined operational scenario, ensuring transparent and reproducible function derivation.

4.4. Functional Analysis Results as a Baseline for Comparative Evaluation

Table 1 summarizes the set of system functions identified for the aircraft ground deceleration case study using the functional analysis procedure in Section 3. The resulting function set spans multiple functional layers, ranging from primary deceleration mechanisms, such as wheel braking, ground spoiler deployment, and reverse thrust generation, to supporting functions related to sensing, monitoring, thermal management, and feedback control.

Table 1. Results of function identification using the scenario-driven paradigm for the aircraft ground deceleration case.

No.	Function	Input	Output
1	Provide braking force	pilot brake command automatic braking command hydraulic/electrical actuation energy	frictional braking force reduced wheel rotational speed heat energy
2	Regulate wheel slip	wheel speed information aircraft ground speed information	modulated braking force stabilized wheel slip
3	Detect wheel lock tendency	wheel speed information wheel acceleration information	anti-skid activation signal
4	Provide anti-skid feedback to crew	anti-skid system status information	warning or status indication to cockpit
5	Dissipate brake heat	accumulated thermal energy in brake assembly	heat transfer to surrounding air and structure
6	Supply electrical power for cooling	electrical energy from onboard power system	energy delivered to cooling or ventilation devices
7	Provide brake temperature information	brake temperature data	brake temperature to cockpit
8	Deploy ground spoilers	spoiler deployment command (manual/automatic) hydraulic/electrical actuation energy	increased aerodynamic drag increased wheel normal load
9	Adjust spoiler deflection according to wheel load	wheel load information aircraft deceleration state	adjusted spoiler deflection angle
10	Provide spoiler deployment status	spoiler position feedback	spoiler deployment status indication to cockpit
11	Generate reverse thrust	thrust reverser command engine power	reverse thrust force

Beyond the primary deceleration functions, several additional functions emerge as direct consequences of physical constraints and state parameter evolution during scenario execution. For example, the accumulation of thermal energy during wheel braking necessitates a dedicated brake heat dissipation function, while variations in wheel speed and wheel load trigger sensing and feedback functions that support anti-skid operation and load-dependent spoiler adjustment.

The identified functions are not limited to direct force or energy transformation processes. They also include information acquisition, signal generation, and status indication functions that support pilot awareness, system coordination, and safety assurance. Functions such as wheel speed sensing, brake temperature monitoring and spoiler deployment status indication arise naturally from interaction between system components, human operators and the operational environment.

Importantly, all functions listed in Table 1 can be traced back to specific context entities, state parameters, and input/output identified in the preceding analysis steps. Each function corresponds to a distinct transformation or regulation of material, energy, or

information within the ground deceleration scenario, ensuring that no function is introduced without an explicit operational trigger or physical necessity.

Overall, the results of the functional analysis provide a structured and transparent functional representation of the aircraft ground deceleration process. The identified function set serves as a mechanism-explicit baseline for the comparative evaluation presented in Section 5, where differences in functional coverage and function-derivation mechanisms across alternative functional analysis paradigms are systematically examined.

5. Comparative Evaluation and Methodological Discussion

5.1. Functional Analysis Results Using Pahl's Systematic Design Approach

Following Pahl's systematic design approach, a top-down functional decomposition is applied for analyzing the aircraft ground deceleration case study and establishing a solution-neutral functional structure as a baseline for comparison. The primary objective of this approach is to clarify what the system must accomplish, independent of specific physical implementations, thereby providing a structured functional basis for early-stage concept development and subsequent comparison with other functional analysis paradigms.

The analysis begins with a black-box representation of the aircraft ground deceleration system. At this level, the overall function is defined as follows: "to reduce the aircraft's ground speed during the landing roll phase in a safe and controlled manner." The inputs to the black box include the aircraft's initial ground speed and pilot or automatic control commands, while the outputs consist of a reduced ground speed and corresponding status information indicating successful deceleration. Internal mechanisms and physical realization details are intentionally abstracted at this stage to preserve solution neutrality.

Based on this black-box definition, the overall function is refined through hierarchical decomposition into a set of primary functions that represent abstract transformations required to achieve the desired system-level objective. In this case, three primary functions are identified, as illustrated in Figure 1: (i) transmit control signals, which receive and process pilot or automatic braking commands; (ii) increase resistance to motion, which represents the transformation of aircraft kinetic energy into other forms to achieve deceleration; and (iii) sense system status, which provides information on key state variables relevant to safe and controlled operation. It should be noted that the functions shown in Figure 1 are identified through solution-neutral decomposition logic, rather than being triggered by physical effects.

Each primary function is further decomposed into solution-neutral sub-functions. For example, the function increase resistance to motion is decomposed into sub-functions associated with wheel braking, reverse thrust generation, and aerodynamic drag increase via spoiler deployment. Similarly, the control transmission branch is decomposed into functions related to command acceptance and distribution, while the sensing branch includes functions such as detecting landing gear status and transmitting status information. These sub-functions are organized sequentially to form a coherent functional chain that ensures logical completeness within the defined system boundary.

The outcome of this process is a hierarchically structured functional model that provides structural clarity and preserves solution neutrality, as shown in Figure 1. By construction, functions are identified through abstract transformation logic within the system boundary, rather than through explicit analysis of operational context, system–environment interactions, or physical effects. As a result, while mission-oriented functions are reliably captured, functions that arise from interaction effects or physical effects are not systematically triggered by the decomposition process.

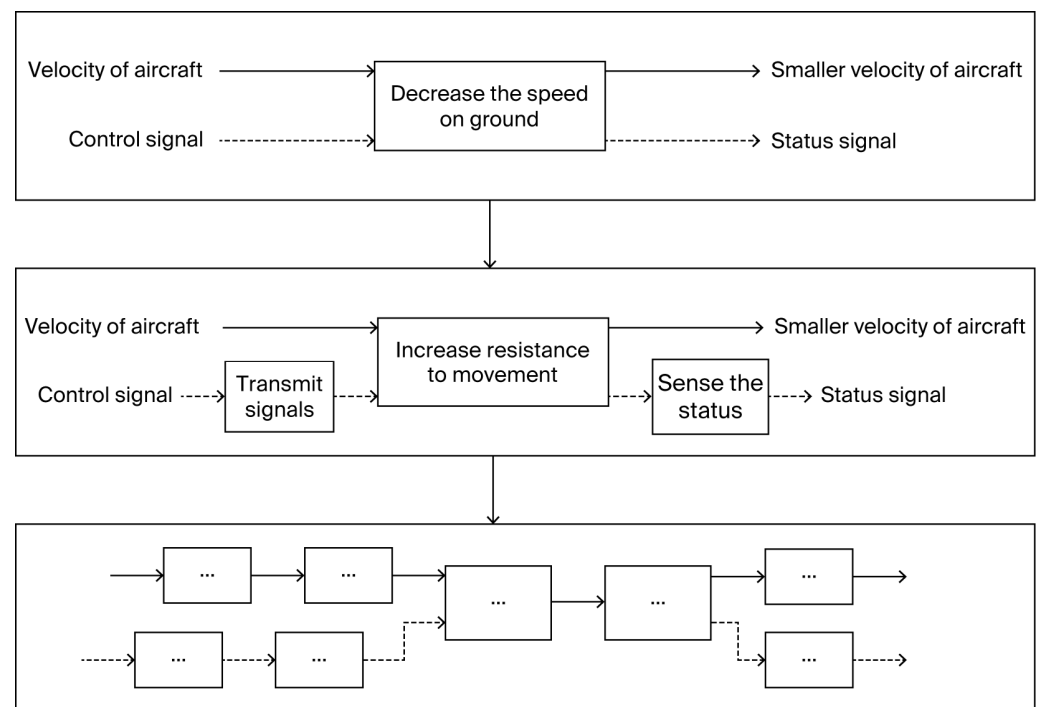


Figure 1. The functional structure derived using Pahl’s systematic design approach for the aircraft ground deceleration case study. (Solid arrows represent energy or material, while dashed arrows represent information.)

This characteristic defines both the strength and the limitation of Pahl’s systematic design approach in the context of aircraft ground deceleration. The resulting functional structure serves as a clear and consistent baseline for comparison, while also illustrating why interaction- and physical-effect-induced functions remain implicit when using solution-neutral decomposition alone. These observations form an important input to the comparative evaluation presented in Section 5.3.

5.2. Functional Analysis Results Using the MagicGrid-Based Framework

Following a systems-engineering-based paradigm, the aircraft ground deceleration case study is represented and functionally analyzed using the MagicGrid framework to capture stakeholder responsibilities, operational interactions, and information-centric traceability at the system level. To ensure methodological symmetry with the other paradigms discussed in this study, the MagicGrid-based analysis is formulated here as a structured derivation process with explicit function-identification logic.

Step 1: Operational context framing and actor definition

The analysis begins with defining the operational scenario and system context corresponding to aircraft ground deceleration during the landing roll. The system of interest is defined as the aircraft, while external entities include the flight crew, cabin crew, air traffic control, and passengers. At this stage, the focus is on capturing who interacts with the aircraft and under what operational conditions.

Step 2: Black-box functional expectation and interaction modeling

Based on the defined operational context, the aircraft is first represented as a black-box system. At this level, functional requirements are formulated in terms of externally observable operational behavior and information interaction. As illustrated in Figure 2, the black-box model captures what the aircraft is expected to accomplish during ground deceleration and how it exchanges control commands, status information, and feedback signals with external actors. This step establishes a system-level functional view oriented toward traceability and operational accountability.

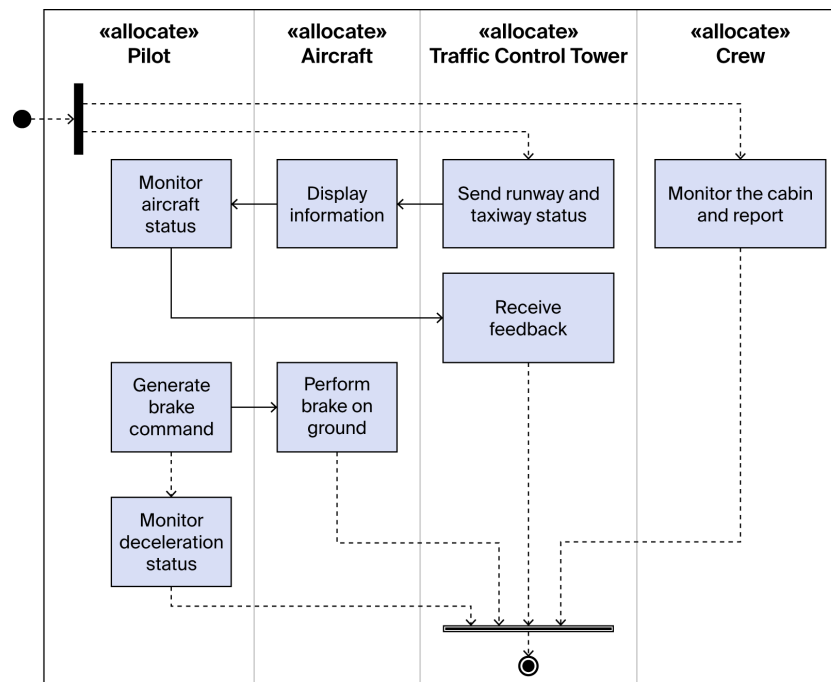


Figure 2. A black-box functional and responsibility-oriented analysis of aircraft ground deceleration based on the MagicGrid framework. (Solid arrows represent energy, material, or information, while dashed arrows control tokens.)

Step 3: Responsibility allocation and subsystem-level refinement

The black-box functions are subsequently refined through allocation to major onboard subsystems, resulting in a white-box representation of system behavior. Using swimlane-based activity modeling, functional responsibilities are distributed across subsystems such as the wheel braking system, flight control/spoiler system, engine system, and indication system, as illustrated in Figure 3. This allocation clarifies how different subsystems contribute to ground deceleration, coordinate control actions, and provide system status information, without requiring specific physical solutions or component-level designs.

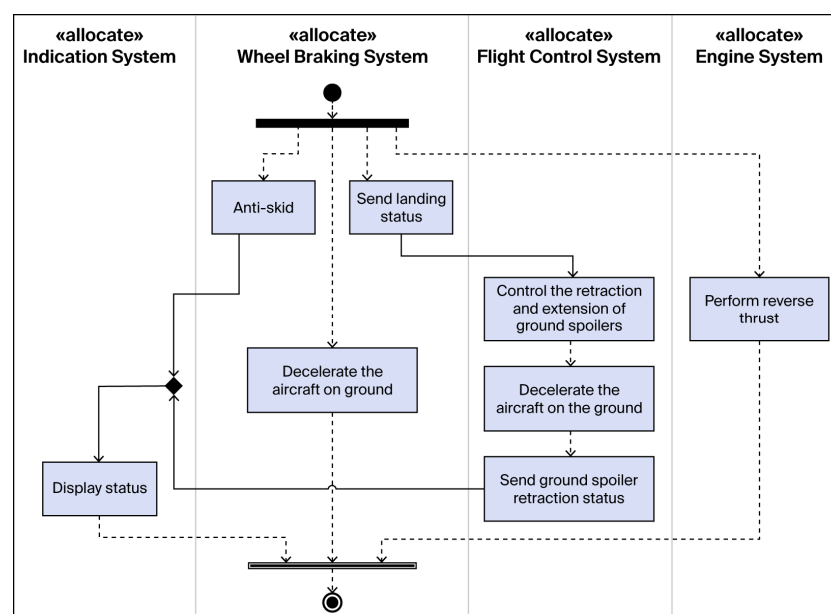


Figure 3. White-box allocation of functional responsibilities for aircraft ground deceleration based on the MagicGrid framework. (Diamonds represent decision nodes.)

Step 4: Function consolidation and traceability articulation

To support direct comparison with the scenario-driven functional analysis, the functions identified through the MagicGrid-based approach are consolidated in Table 2, using a structured format consistent with Table 1.

The resulting function set reflects MagicGrid's emphasis on stakeholder interaction, responsibility allocation, and information-centric traceability. Each function is explicitly linked to an operational need and assigned to a responsible subsystem, providing a clear justification from a systems engineering perspective.

Table 2. Results of function identification using the MagicGrid framework for aircraft ground deceleration.

No.	Function	Input	Output
1	Provide braking force for ground deceleration	pilot brake command system status	brake torque aircraft deceleration
2	Indicate anti-skid system status	wheel speed anti-skid logic	warning signal to cockpit
3	Transmit landing status	weight-on-wheels detection	landing status
4	Deploy ground spoilers	spoiler command (manual/auto)	aerodynamic drag vertical load transfer
5	Control spoiler deflection	wheel load control input	adjusted spoiler angle
6	Indicate spoiler deployment status	spoiler position sensor	status signal to cockpit
7	Generate reverse thrust on command	thrust lever input	reverse engine thrust
8	Display system status to crew	brake/spoiler/reverse thrust status	visual feedback on cockpit interface

It can be found that functions related to residual physical effects or unbalanced state parameters, such as thermal dissipation, energy loss management, or constraint-driven compensation mechanisms, are not included in Table 2. This is not due to an inability of the MagicGrid framework to represent such functions, but rather because they are not systematically derived within the responsibility-driven derivation mechanism in this case. Their inclusion would require explicit analyst introduction as derived requirements prior to responsibility allocation.

This characteristic reflects the distinct derivation focus of the responsibility-driven mechanism. The MagicGrid-based analysis therefore yields a system-level functional representation emphasizing stakeholder interaction and traceability, providing a complementary perspective to the scenario-driven approach and forming a symmetric basis for the comparative discussion in Section 5.3.

5.3. Added Value of the Scenario-Driven Approach and Methodological Implications

While all three paradigms support the identification of primary deceleration functions, they differ fundamentally in how functions are derived and justified, leading to systematic differences in functional completeness and traceability. To enable a transparent and mechanism-level comparison, the evaluation is conducted using explicitly defined criteria grounded in the formalized function-derivation mechanisms established in the preceding sections.

Specifically, the comparison is structured along the following criteria:

- Primary mode of function introduction—the underlying logic through which new functions are introduced and justified (e.g., hierarchical decomposition, responsibility-based allocation, or physical-effect-driven derivation);

- Role of system–environment interaction—whether interactions act as explicit drivers within the derivation mechanism or are incorporated descriptively after function identification;
- Treatment of information exchanges—how sensing, monitoring, and feedback functions are systematically incorporated and justified;
- Treatment of physical effects—whether residual physical effects systematically lead to additional function identification within the derivation process;
- Typical sources of omission—systematic blind spots associated with each derivation mechanism;
- Traceability structure—the form of causal, hierarchical, or responsibility-based linkage supporting functional justification.

These criteria characterize different structural attributes of the derivation mechanisms and provide a consistent basis for cross-paradigm comparison. By applying these criteria consistently, the comparison emphasizes mechanism-level distinctions rather than case-specific outcomes. This structured evaluation framework provides a reproducible basis for analyzing functional completeness and derivation transparency in alternative complex systems.

Design-theory-oriented approaches primarily identify functions through the top-down decomposition of an overall transformation, ensuring solution neutrality and structural clarity. In the ground deceleration case study, this enables the reliable identification of core functions such as wheel braking, spoiler deployment, and reverse thrust generation. However, because the underlying function-derivation mechanism is driven by abstract transformation logic within the system boundary, functions induced via system–environment interactions, including sensing, monitoring, and secondary physical effects, are not inherently generated by the decomposition process. Consequently, functions such as brake heat dissipation, thermal management, and feedback-based control adaptation tend to remain implicit or are introduced by the analyst based on engineering judgment rather than being derived systematically at the conceptual design stage. This limitation follows directly from the solution-neutral, decomposition-driven mechanism.

Systems-engineering-based approaches such as MagicGrid strengthen functional analysis by explicitly modeling stakeholders, operational scenarios, and system responsibilities, thereby enhancing lifecycle traceability and supporting certification-oriented development. In the examined case, this facilitates the identification of monitoring-, indication-, and control-related functions associated with pilot interaction and system supervision. Nevertheless, from a mechanism perspective, function identification in such frameworks is primarily guided by traceable mappings from needs, use cases, and operational responsibilities to system behaviors. Residual physical effects accompanying mission-oriented functions are not systematically used as derivation drivers. As a result, compensating or regulation-related functions associated with residual physical effects may depend on analyst interpretation rather than emerging from an explicit derivation rule.

In contrast, the examined effect-strengthened scenario-driven implementation introduces a physical-effect-driven mechanism for function identification and justification. By analyzing state evolution within operational scenarios and identifying residual physical effects that lead to unbalanced state parameters, functions such as brake heat dissipation, electrical power supply for cooling, environmental sensing, and load-dependent spoiler adjustment emerge as necessary responses to unresolved physical conditions. In this approach, function introduction occurs when residual physical effects cannot be accommodated by the existing function set under defined operational conditions, thereby systematically exposing interaction-induced and residual-effect-induced functions that tend to remain implicit in other paradigms.

Beyond improvements in functional completeness, the scenario-driven approach also enhances traceability by linking each identified function to operational context entities, measurable state parameters, and the residual physical effects that justify its introduction. Compared with requirement-centered traceability in systems engineering frameworks, this effect-based causal linkage provides clearer justification for functions arising from thermal limits, energy constraints, environmental perception, and dynamic feedback behavior.

Table 3 summarizes the comparative evaluation according to the explicitly defined criteria introduced above, emphasizing methodological capabilities rather than case-specific implementations. The results indicate that the scenario-driven approach does not replace existing functional analysis paradigms, but complements them by systematically exposing interaction-induced and residual-physical-effect-induced functions at the conceptual design stage. This integration-oriented role aligns with contemporary systems engineering perspectives that emphasize methodological complementarity rather than reliance on a single universal approach when addressing early-stage system complexity [25].

Table 3. Mechanism-level comparison of the three functional analysis paradigms based on the defined comparison criteria.

Comparison Criterion	Design-Theory-Oriented (Top-Down Decomposition)	Systems-Engineering-Based (Responsibility Allocation)	Scenario-Driven (Effect-Strengthened Implementation)
Primary mode of function introduction	Top-down decomposition from overall mission objectives within the system boundary	Allocation and refinement of functions through responsibility assignment across system elements	Introduction of functions triggered by physical effects under defined operational conditions
Role of system–environment interaction	Generally abstracted as boundary condition rather than explicit derivation driver	Considered through stakeholder and operational views but not systematically modeled as a causal derivation source	Explicitly modeled as structured interaction context that exposes additional functional requirements
Treatment of information exchanges	Embedded within decomposed functional hierarchy	Formalized through allocation, traceability, and interface traceability	Modeled dynamically within operational scenarios to reveal interaction-induced functions
Treatment of physical effects	Typically implicit unless explicitly decomposed	Addressed if captured within system requirements or allocation artifacts	Explicit tracing of physical effects leading to compensating functions
Typical sources of function omission	Interaction- and effect-induced functions beyond mission decomposition	Interaction effects insufficiently reflected in allocation workflows	Dependent on scenario completeness but less prone to omission of interaction- and effect-induced functions
Traceability structure	Hierarchy decomposition tree	Multi-view traceable architecture linking requirements, functions, and system elements	Causal linkage between operational interactions, physical effects, and introduced functions

6. Conclusions

In this study, we contribute to clarifying the methodological foundations of functional analysis through a mechanism-explicit and case-based comparative evaluation of three paradigms within a shared aircraft engineering context, including design-theory-oriented decomposition, systems-engineering-based allocation, and a scenario-driven paradigm examined in its effect-strengthened form. Rather than proposing a new modeling technique, this study focuses on formalizing function-derivation mechanisms as an explicit analytical dimension for systematically comparing functional analysis

approaches. Through this structured comparison, we identify decomposition- and responsibility-driven derivation mechanisms, as well as a physical-effect-driven mechanism implemented within the scenario-driven paradigm.

Within this mechanism-based comparative framework, the evaluation reveals systematic differences in how these mechanisms introduce, justify, and structure functions under early-stage design conditions. Decomposition-driven approaches primarily ensure structural completeness of mission-oriented functions but may overlook interaction-induced and residual-physical-effect-induced functions. Responsibility-driven allocation enhances traceability and coordination across subsystems but does not inherently expose residual physical effects accompanying mission-oriented functions. In contrast, physical-effect-driven derivation systematically reveals functions associated with residual physical effects, thereby strengthening the transparency and causal grounding of function identification.

From a methodological standpoint, the results support an integration-oriented view of functional analysis. Functional completeness is therefore not solely a matter of hierarchical decomposition or responsibility mapping but is strongly influenced by the underlying derivation mechanism embedded in the analytical method. By making these mechanisms explicit and comparable, this research contributes to a clearer methodological positioning of functional analysis approaches within early-stage design research.

In light of these mechanism-level distinctions, the findings suggest that the examined scenario-driven implementation does not replace established paradigms but complements design-theory-oriented and systems-engineering-based methods by systematically exposing interaction- and physical-effect-induced functions that may otherwise remain implicit. Importantly, the added value identified in this study concerns not the number of functions generated, but the transparency, causal grounding, and reproducibility of the function-identification process.

This study is subject to several limitations. The comparative evaluation is based on a single aircraft case study, and all approaches were applied by the same analyst. Although this controlled setting strengthens internal comparability, broader validation across multiple systems and analyst teams would further strengthen generalizability. Future research will therefore extend the mechanism-based comparative framework to additional complex systems and explore structured metrics for quantitatively assessing functional coverage and derivation robustness.

Author Contributions: Methodology, H.L., Y.C. and Y.X.; investigation, H.L. and M.Z.; writing—original draft preparation, H.L.; writing—review and editing, M.Z., Y.C. and Y.X.; project administration, Y.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the Civil Aircraft Special Program of the Ministry of Industry and Information Technology of the People's Republic of China, and the COMAC-SJTU Joint Foundation for Systems Engineering of Commercial Aircraft, a joint initiative between the Commercial Aircraft Corporation of China, Ltd. and Shanghai Jiao Tong University.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data are contained within this article.

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

References

1. Ulrich, K.T.; Eppinger, S.D. *Product Design and Development*; McGraw-Hill: New York, NY, USA, 2016.
2. Cash, P.; Daalhuizen, J.; Hekkert, P. Evaluating the efficacy and effectiveness of design methods: A systematic review and assessment framework. *Des. Stud.* **2023**, *88*, 101204. <https://doi.org/10.1016/j.destud.2023.101204>.
3. Vizitiu, C.; Dinculescu, A.C.; Marin, M.G.; Nistorescu, A.I.; Dominey, K.A.; Kristály, D.M. Optimizing early-stage efficiency in complex system development via model-based systems engineering (MBSE) and concurrent engineering (CE) integration. *Res. Eng. Des.* **2025**, *36*, 12.
4. Pahl, G.; Beitz, W.; Feldhusen, J.; Grote, K.H. *Engineering Design: A Systematic Approach*, 3rd ed.; Springer Science & Business Media: London, UK, 2007.
5. INCOSE. *INCOSE Systems Engineering Handbook*; John Wiley & Sons: Hoboken, NJ, USA, 2023.
6. Gui, F.J.; Chen, Y. A scenario-driven approach for functional design of smart systems. *AI EDAM* **2021**, *35*, 165–179.
7. Jiao, R.; Luo, J.; Malmqvist, J.; Summers, J. New design: Opportunities for engineering design in an era of digital transformation. *J. Eng. Des.* **2022**, *33*, 685–690.
8. Gero, J.S.; Kannengiesser, U. The situated function–behaviour–structure framework. *Des. Stud.* **2004**, *25*, 373–391. <https://doi.org/10.1016/j.destud.2003.10.010>.
9. Gero, J.S.; Kannengiesser, U. A Function–Behavior–Structure Ontology of Processes. *AI EDAM* **2007**, *21*, 379–391.
10. Goel, A.K.; Rugaber, S.; Vattam, S. Structure, behavior, and function of complex systems: The structure, behavior, and function modeling language. *AI EDAM* **2009**, *23*, 23–35.
11. Chen, Y.; Zhao, M.; Xie, Y.B.; Zhang, Z.N. A new model of conceptual design based on Scientific Ontology and intentionality theory. Part II: The process model. *Des. Stud.* **2015**, *38*, 139–160.
12. Hirtz, J.; Stone, R.B.; McAdams, D.A.; Szykman, S.; Wood, K.L. A functional basis for engineering design: Reconciling and evolving previous efforts. *Res. Eng. Des.* **2002**, *13*, 65–82.
13. Nagel, J.K.S.; Nagel, R.L.; Stone, R.B.; McAdams, D.A. Function-based, biologically inspired concept generation. *AI EDAM* **2010**, *24*, 521–535.
14. Wu, C.; Wang, X.; Chen, T.; Li, H.; Wang, H.Q. Conceptual design of smart product-service system based on function-oriented search. *J. Eng. Des.* **2025**, *36*, 976–1010.
15. Tomiyama, T.; Van Beek, T.J.; Cabrera, A.A.A.; Komoto, H.; D’Amelio, V. Making function modeling practically usable. *AI EDAM* **2013**, *27*, 301–309.
16. Vermaas, P.E. The coexistence of engineering meanings of function: Four responses and their methodological implications. *AI EDAM* **2013**, *27*, 191–202.
17. Aurisicchio, M.; Bracewell, R.; Armstrong, G. The function analysis diagram: Intended benefits and coexistence with other functional models. *AI EDAM* **2013**, *27*, 249–257.
18. Booth, J.W.; Reid, T.N.; Eckert, C.; Ramani, K. Comparing functional analysis methods for product dissection tasks. *J. Mech. Des.* **2015**, *137*, 081101.
19. Friedenthal, S.; Moore, A.; Steiner, R. *A Practical Guide to SysML: The Systems Modeling Language*, 3rd ed.; Morgan Kaufmann: Waltham, MA, USA, 2014.
20. Aleksandraviciene, A.; Strolia, Z.; Morkevicius, A. Systems Architecture Meta-Model for the MagicGrid Framework. In Proceedings of the INCOSE’s 34th Annual International Symposium, Dublin, Ireland, 2–6 July 2024.
21. Dori, D. *Model-Based Systems Engineering with OPM and SysML*; Springer Science & Business Media: New York, NY, USA, 2016.
22. Chen, Q.; Chen, Y.; Liu, Y. A systems engineering-integrated approach for functional architecture design of cyber-physical products. *J. Eng. Des.* **2025**, 1–30. <https://doi.org/10.1080/09544828.2025.2569021>.
23. Eichmann, O.C.; Lamm, J.G.; Melzer, S.; Weilkiens, T.; God, R. Development of Functional Architectures for Cyber-Physical Systems Using Interconnectable Models. *Syst. Eng.* **2024**, *27*, 993–1011.
24. Kannengiesser, U.; Gero, J.S. What distinguishes a model of systems engineering from other models of designing? An ontological, data-driven analysis. *Res. Eng. Des.* **2022**, *33*, 129–159. <https://doi.org/10.1007/s00163-021-00382-9>.
25. Madni, A.M.; Sievers, M. Systems integration: Key perspectives, experiences, and challenges. *Syst. Eng.* **2014**, *17*, 37–51.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.