APPLIED SCIENCE

The International Journal of Applied Sciences.
Vol. 1, No. 2, 2025, pp. 139-160
ISSN 3104-8935
https://international-journal-of-applied-sciences.jo

Multidimensional Architecture Design for Cross-Domain Operational Systems: A Framework for Robustness and Adaptability Optimization

Farin zakir A. Khan [1]

[1] **School of computer science and engineering. bangladesh university. Email: farin.a.khan@bu.edu.bd**

*** Corresponding Author**: *Farin zakir A. Khan*, **farin.a.khan@bu.edu.bd**

DOI: https://doi.org/

ARTICLE INFO

Article history
Received June 08, 2025
Revised June 17, 2025
Accepted Oct 25, 2025

Keywords
cross-domain operational systems; multidimensional architecture; conceptual architecture; physical architecture; logical architecture; robustness; adaptability; complex networks; architecture optimization; systems engineering

This is an open-access article under the CC–BY-SA license.

ABSTRACT

Cross-domain operational systems are characterized by heterogeneous components, dynamic operational environments, complex organizational relationships, and rapidly changing information requirements. These characteristics make conventional architecture design approaches insufficient for representing the structural, functional, and adaptive properties of such systems in a unified manner. This study proposes a multidimensional architecture design framework for cross-domain operational systems that integrates conceptual architecture, physical architecture, and logical architecture within a unified systems-engineering process. In addition to conventional architectural dimensions, the proposed framework introduces two additional design variables, namely architectural robustness and architectural adaptability, to evaluate the ability of a system to maintain essential functions under disturbances and to reconfigure its structure in response to changing environmental and mission conditions. The conceptual architecture dimension focuses on environmental and threat-context analysis, stakeholder and mission analysis, capability requirements, and operational concept generation. The physical architecture dimension addresses component selection, interconnection structures, resource allocation, and deployment configuration. The logical architecture dimension describes organizational relationships, functional processes, information exchanges, temporal dependencies, and information-quality propagation. A quantitative evaluation layer is further introduced by defining robustness and adaptability indices and incorporating them into a multi-objective architecture optimization model. The proposed model seeks to maximize mission effectiveness, robustness, adaptability, and information connectivity while minimizing structural complexity, resource consumption, and reconfiguration cost. A directed complex-network representation is employed to characterize relationships among heterogeneous system elements, while graph-based metrics are introduced to evaluate connectivity, centrality, redundancy, network density, and structural resilience. The resulting framework provides a systematic approach for mapping capability requirements to architectural structures and for evaluating the extent to which alternative architectures can tolerate component degradation, accommodate environmental changes, and support dynamic reconfiguration. The proposed approach establishes a more general and measurable basis for architecture design and provides a foundation for subsequent model-based simulation, architecture evaluation, and optimization of complex cross-domain operational systems.

1. Introduction

Cross-domain operational systems represent a new generation of complex systems in which heterogeneous resources, functional capabilities, information infrastructures, and organizational entities operate across multiple interconnected domains. Unlike conventional systems that are generally designed around a relatively stable operational environment and a clearly defined organizational structure, cross-domain systems must operate under conditions characterized by uncertainty, dynamic interactions, heterogeneous resources, incomplete information, and rapidly changing mission requirements.

From a systems-engineering perspective, the fundamental characteristic of a cross-domain operational system is not simply the coexistence of multiple domains, but rather the **continuous interaction and dynamic coupling among heterogeneous system elements**. Consequently, system effectiveness increasingly depends on the quality of interconnections, information exchange, organizational coordination, resource allocation, and structural adaptability.

The architecture of a complex operational system represents its highest-level structural description. It determines, to a considerable extent, the composition of system elements, their relationships, deployment configurations, organizational structures, information flows, and functional processes. An inappropriate architecture may result in excessive structural complexity, insufficient redundancy, weak information connectivity, poor resource utilization, and limited adaptability. Therefore, architecture design should be regarded as a fundamental stage preceding detailed component design and system implementation.

Several architecture description frameworks have been developed internationally, including the Department of Defense Architecture Framework (DoDAF), NATO Architecture Framework (NAF), MODAF, UPDM, and TOGAF. These frameworks provide structured approaches for representing complex systems from different viewpoints. Model-based systems engineering (MBSE), agent-based modeling, and simulation-based systems engineering have subsequently extended these approaches by connecting architecture descriptions with executable models and simulation environments.

However, conventional architecture design methods still face several limitations when applied to highly dynamic cross-domain systems. First, architecture descriptions frequently emphasize structural relationships but provide insufficient quantitative representation of **robustness**. Second, adaptability is often treated as a qualitative property rather than as an explicit design variable. Third, conceptual, physical, and logical architectures are sometimes developed as relatively independent views, which may result in inconsistencies between capability requirements, physical resources, organizational structures, and information processes.

To address these limitations, this study proposes an enhanced multidimensional architecture design framework in which **robustness and adaptability are explicitly incorporated into architecture design and evaluation**.

The main contributions of this study are as follows:

1. A unified multidimensional architecture framework integrating conceptual, physical, and logical architectures is established.

2. **Architectural robustness** is introduced as an explicit variable for measuring the ability of an architecture to maintain essential functions under component degradation, communication disruption, and environmental uncertainty.
3. **Architectural adaptability** is introduced as an explicit variable for measuring the ability of an architecture to modify its structure, resources, and information relationships in response to changing requirements.
4. A directed complex-network model is established to represent heterogeneous relationships among system elements.
5. A multi-objective architecture optimization model is proposed to jointly consider effectiveness, robustness, adaptability, complexity, and resource consumption.
6. A closed-loop architecture design process is established, connecting requirements analysis, architecture construction, evaluation, simulation, optimization, and iterative refinement.

Accordingly, the proposed framework extends conventional multidimensional architecture design from a primarily descriptive methodology toward a **quantitative, evaluative, and optimization-oriented architecture design approach**.

---

2. Multidimensional Architecture Design Framework

2.1 Characteristics of Cross-Domain Operational Systems

Cross-domain operational systems possess several characteristics that distinguish them from conventional systems.

First, system elements are highly heterogeneous. Different elements may possess different functions, performance characteristics, communication protocols, organizational affiliations, and information-processing capabilities.

Second, the operational environment is dynamic. System requirements may change because of environmental disturbances, mission transitions, resource availability, or changes in system states.

Third, system relationships are strongly coupled. The failure or degradation of one component may influence multiple other components through information, functional, organizational, or resource dependencies.

Fourth, system-level effectiveness cannot always be obtained through the simple aggregation of individual component performance. Emergent behavior may arise from interactions among components.

Therefore, architecture design should simultaneously consider:

- functional capability;
- structural connectivity;
- organizational relationships;

- information exchange;
- resource allocation;
- robustness;
- adaptability;
- complexity;
- reconfiguration cost.

2.2 Overall Architecture Design Process

The proposed architecture design process consists of six major stages:

Requirement analysis → Conceptual architecture → Physical architecture → Logical architecture → Robustness/adaptability evaluation → Multi-objective optimization

Unlike a purely sequential design process, the proposed framework adopts an iterative architecture-development mechanism.

Let the architecture state at iteration **(k)** be represented as

$$A^{(k)} = \left[A_C^{(k)}, A_P^{(k)}, A_L^{(k)}\right]$$

where:

- (A_C) represents the conceptual architecture;
- (A_P) represents the physical architecture;
- (A_L) represents the logical architecture.

The architecture is evaluated according to

$$\mathbf{F}(A) = [E(A), R(A), D(A), C(A), Q(A)]$$

where:

- **(E)** = system effectiveness;
- **(R)** = robustness;
- **(D)** = adaptability;
- **(C)** = architecture complexity;
- **(Q)** = resource or reconfiguration cost.

The design process seeks an architecture that satisfies required capability constraints while achieving an appropriate balance among these objectives.

3. Conceptual Architecture Design

3.1 Definition

The conceptual architecture represents the highest-level description of the system. It does not focus on detailed component implementation but establishes the relationship between environmental conditions, mission requirements, capability requirements, and operational concepts.

The conceptual architecture therefore provides the basis for subsequent physical and logical architecture development.

The conceptual architecture includes four principal activities:

1. environmental analysis;
2. mission and stakeholder analysis;
3. capability requirement analysis;
4. operational concept generation.

3.2 Environmental and Context Analysis

The first stage is the identification of the operational environment.

Instead of representing the environment as a static condition, the proposed framework describes it using a state vector:

$$X_E = [x_1, x_2, \dots, x_n]^T$$

where x_i represents an environmental variable relevant to system operation.

The environment can therefore be represented as

$$X_E(t)$$

rather than as a constant state.

The architecture should remain functional under a range of plausible environmental states:

$$X_E(t) \in \Omega_E$$

where Ω_E denotes the expected operational environment space.

This formulation provides a basis for evaluating architecture robustness under environmental uncertainty.

APPLIED SCIENCE

The International Journal of Applied Sciences.
Vol. 1, No. 2, 2025, pp. 139-160
ISSN 3104-8935
https://international-journal-of-applied-sciences.jo

4. Capability Requirement Analysis

Capability requirements are derived from mission objectives, environmental conditions, functional requirements, performance requirements, and system constraints.

Let the capability vector be

$$C = [c_1, c_2, \ldots, c_m]^T$$

where each c_i represents a required system capability.

A requirement matrix can subsequently be defined as

$$M_{RC} = (m_{ij})$$

where

$$m_{ij} = \begin{cases} 1, & \text{if requirement } r_i \text{ depends on capability } c_j \\ 0, & \text{otherwise.} \end{cases}$$

This matrix establishes traceability between requirements and capabilities.

An important advantage of this approach is that subsequent physical and logical architecture elements can be traced back to their originating requirements.

5. Operational Concept Generation

Based on the capability requirements, an operational concept is generated to describe how the system is expected to achieve its required functions.

The operational concept establishes:

- major functional activities;
- information dependencies;
- temporal relationships;
- organizational coordination;
- resource interactions;
- transition conditions between different system states.

The operational concept should not be treated as a fixed sequence. Instead, multiple feasible operational paths should be considered.

The International Journal of Applied Sciences.
Vol. 2, No. 1, 2025, pp. 139-160
ISSN 3104-8935
https://international-journal-of-applied-sciences.jo

Let

$$P = \{p_1, p_2, \ldots, p_k\}$$

represent the set of feasible functional pathways.

The architecture should maintain sufficient alternatives to prevent excessive dependence on a single pathway.

This characteristic directly contributes to architectural robustness.

---

6. Physical Architecture Design

6.1 Definition

The physical architecture describes the actual system elements, their functional characteristics, interconnections, quantities, and deployment configurations.

The physical architecture can be represented as

$$A_P = (V, E, L)$$

where:

- **(V)** represents physical elements;
- **(E)** represents relationships among elements;
- **(L)** represents deployment locations or configuration states.

The elements may include sensing, processing, communication, command, support, and other functional entities, depending on the specific application.

---

7. Physical Element Selection

Physical elements should be selected based on capability requirements rather than simply according to existing equipment inventories.

Let

$$P_i = \{p_{i1}, p_{i2}, \ldots, p_{in}\}$$

$.c_i$ represent candidate physical elements capable of contributing to capability (c_i).

The selection problem can therefore be expressed as:

$$\max \sum_{i=1}^{m} w_i C_i(A_P)$$

subject to:

$$\sum_{j} x_j Cost_j \leq B$$

where:

- w_i is the importance weight of capability i
- x_j indicates whether element (j) is selected;
- $Cost_j$ represents its resource requirement;
- B represents the available resource budget.

This converts physical architecture construction into a constrained architecture-selection problem.

---

8. Complex-Network Representation

Because a cross-domain architecture consists of numerous heterogeneous elements and relationships, graph theory provides an appropriate mathematical representation.

The architecture can be represented as a directed graph:

$$G = (V, E)$$

where

$$V = \{v_1, v_2, \dots, v_N\}$$

represents system elements and

$$E = \{e_{ij}\}$$

represents directed relationships.

The adjacency matrix is defined as

$$A_G = (a_{ij})_{N \times N}$$

where

$$a_{ij} = \begin{cases} 1, & \text{if element } i \text{ can transmit information or perform a defined interaction with } j \\ 0, & \text{otherwise.} \end{cases}$$

For weighted relationships,

$$a_{ij} = w_{ij}$$

where w_{ij} may represent communication reliability, latency, bandwidth, dependency strength, or another architecture-specific parameter.

This weighted representation is particularly useful for robustness and adaptability analysis.

---

9. Architecture Connectivity and Complexity

Several network metrics can be used to evaluate the structural characteristics of an architecture.

9.1 Connectivity

The normalized connectivity index can be expressed as

$$C_N = \frac{E}{N(N-1)}$$

for a directed architecture.

Higher connectivity generally improves information accessibility, but excessive connectivity may increase structural complexity and resource consumption.

9.2 Average Path Length

$$L_G = \frac{1}{N(N-1)} \sum_{i \neq j} d_{ij}$$

where d_{ij} represents the shortest path between nodes i and j.

9.3 Network Density

$$D_G = \frac{E}{N(N-1)}$$

which provides a measure of the relative density of system relationships.

9.4 Centralization

Architecture centralization can be evaluated using node centrality measures. Excessive dependence on a small number of highly central nodes may increase vulnerability to component degradation.

Therefore, high network connectivity should not automatically be interpreted as optimal architecture quality.

---

10. Architectural Robustness

10.1 Definition

The first new variable introduced by this study is **architectural robustness**.

Robustness refers to the ability of an architecture to maintain essential functions when subjected to disturbances without requiring fundamental structural redesign.

Disturbances may include:

- component degradation;
- communication interruption;
- resource reduction;
- environmental uncertainty;
- temporary information loss;
- partial functional failure.

Let the nominal system effectiveness be

$$E_0$$

and the effectiveness after disturbance (s) be

$$E_s.$$

A basic robustness index can be defined as

$$R_s = \frac{E_s}{E_0}.$$

For multiple disturbance scenarios:

The International Journal of Applied Sciences.
Vol. 2, No. 1, 2025, pp. 139-160
ISSN 3104-8935
https://international-journal-of-applied-sciences.jo

$$R = \sum_{s=1}^{S} p_s \frac{E_s}{E_0}$$

where p_s represents the probability or importance weight of disturbance scenario **(s).**

A higher **(R)** indicates that the architecture is less sensitive to disturbances.

---

11. Architectural Redundancy

Robustness is strongly associated with functional redundancy.

For a function f_i, let n_i denote the number of independent elements capable of supporting the function.

A simple redundancy index can be defined as

$$Red_i = n_i - 1.$$

$$Red = \frac{1}{M} \sum_{i=1}^{M} Red_i.$$

The overall redundancy can then be expressed as

However, excessive redundancy increases cost and complexity. Therefore, redundancy should be optimized rather than maximized.

This establishes an important design principle:

The objective is not maximum redundancy, but sufficient redundancy under acceptable resource and complexity constraints.

---

12. Architectural Adaptability

12.1 Definition

The second major variable introduced in the proposed framework is **architectural adaptability**.

Adaptability describes the ability of a system architecture to modify its organizational structure, information relationships, resource allocation, or functional configuration in response to changes in mission requirements and environmental conditions.

Let

The International Journal of Applied Sciences.
Vol. 1, No. 2, 2025, pp. 139-160
ISSN 3104-8935
https://international-journal-of-applied-sciences.jo

$$A_0$$

represent the initial architecture and

$$A_1$$

represent the reconfigured architecture.

The architectural distance between the two configurations can be defined as

$$D_A(A_0, A_1)$$

while the corresponding reconfiguration cost is

$$C_R.$$

The adaptability index can therefore be expressed conceptually as

$$D = \frac{\Delta E}{C_R}$$

where ΔE represents the improvement or preservation of system effectiveness following reconfiguration.

A high value indicates that the architecture can achieve significant functional adaptation with relatively low reconfiguration cost.

---

13. Dynamic Reconfiguration Model

The architecture can be considered as a dynamic state:

$$A(t) = [A_C(t), A_P(t), A_L(t)].$$

When environmental or mission conditions change from

$$X_E(t_0)$$

to

$$X_E(t_1),$$

the architecture may need to transition from

The International Journal of Applied Sciences.
Vol. 2, No. 1, 2025, pp. 139-160
ISSN 3104-8935
https://international-journal-of-applied-sciences.jo

$$A(t_0)$$

to

$$A(t_1).$$

The reconfiguration process can be represented as

$$A(t_1) = \mathcal{R}[A(t_0), X_E(t_1), C(t_1)]$$

where **R** is the architecture-reconfiguration operator.

This formulation makes adaptability an explicit architectural property rather than an informal qualitative description.

---

14. Logical Architecture Design

Logical architecture describes the functional and organizational relationships that cannot be adequately represented by physical architecture alone.

It includes:

1. organizational hierarchy;
2. functional processes;
3. information interaction;
4. temporal relationships;
5. information-quality propagation.

The logical architecture can be represented as

$$A_L = (F, I, T, O)$$

where:

- **(F)** = functional relationships;
- **(I)** = information relationships;
- **(T)** = temporal relationships;
- **(O)** = organizational relationships.

---

15. Organizational Structure

The International Journal of Applied Sciences.
Vol. 1, No. 2, 2025, pp. 139-160
ISSN 3104-8935
https://international-journal-of-applied-sciences.jo

Traditional centralized architectures generally adopt hierarchical structures:

$$Root \to Level_1 \to Level_2 \to \cdots \to Leaf.$$

Such structures provide clear authority and relatively simple coordination mechanisms, but may become less flexible under rapidly changing conditions.

A distributed architecture instead permits multiple nodes to interact through lateral relationships.

The proposed framework therefore considers organizational structure as a variable rather than a fixed design assumption.

The degree of organizational decentralization can be represented as

$$O_D = f(Centrality, Connectivity, Redundancy).$$

The optimal value of **(O_D)** depends on mission requirements, information latency, resource constraints, and adaptability requirements.

---

16. Information Interaction Architecture

Information interaction is represented by an information-flow graph:

$$G_I = (V_I, E_I).$$

Each edge represents an information exchange between two functional elements.

For each information link e_{ij}, a vector can be assigned:

$$Q_{ij} = [q_{ij}^{acc}, q_{ij}^{tim}, q_{ij}^{rel}]$$

where:

- q^{acc} = information accuracy;
- q^{tim} = timeliness;
- q^{rel} = reliability.

An overall information quality function may then be expressed as

$$Q_I = w_1 q^{acc} + w_2 q^{tim} + w_3 q^{rel}$$

with

The International Journal of Applied Sciences.
Vol. 2, No. 1, 2025, pp. 139-160
ISSN 3104-8935
https://international-journal-of-applied-sciences.jo

$$w_1 + w_2 + w_3 = 1.$$

This provides a quantitative mechanism for evaluating the information dimension of the architecture.

---

17. Information Propagation and Quality Degradation

Information quality may decrease as information passes through multiple processing and transmission stages.

For an information chain consisting of (n) stages, the resulting quality can be modeled as

$$Q_n = Q_0 \prod_{i=1}^{n} (1 - \delta_i)$$

where:

- Q_0 = initial information quality;
- δ_i = degradation coefficient associated with stage (i).

This model makes it possible to identify architectures that contain excessive information-processing or transmission dependencies.

Reducing unnecessary information-processing stages can improve both efficiency and robustness.

---

18. Multi-Objective Architecture Optimization

The central extension of the proposed research is to transform architecture design into a multi-objective optimization problem.

The architecture quality vector is defined as

$$F(A) = [E(A), R(A), D(A), Q(A), C(A)].$$

The optimization problem is:

$$\max F(A)$$

subject to:

The International Journal of Applied Sciences.
Vol. 1, No. 2, 2025, pp. 139-160
ISSN 3104-8935
https://international-journal-of-applied-sciences.jo

$$g_i(A) \leq 0,$$
$$h_j(A) = 0.$$

A normalized weighted objective function can be expressed as

$$J(A) = w_E E(A) + w_R R(A) + w_D D(A) + w_Q Q(A) - w_C C(A) - w_K K(A)$$

where:

- **(E)** = effectiveness;
- **(R)** = robustness;
- **(D)** = adaptability;
- **(Q)** = information quality;
- **(C)** = architecture complexity;
- **(K)** = reconfiguration cost.

The weights satisfy

$$\sum_i w_i = 1.$$

This formulation allows the architecture designer to adjust priorities according to mission requirements.

---

19. Robustness–Adaptability Trade-off

An important finding of the proposed framework is that robustness and adaptability should not be treated as completely independent variables.

A highly redundant architecture may exhibit strong robustness but suffer from increased complexity.

Conversely, a highly flexible architecture may exhibit strong adaptability but incur greater reconfiguration costs.

Therefore, the architecture design problem can be regarded as a trade-off:

$$\max\,(R, D)$$

subject to:

$$Cost \leq B$$

and

$$Complexity \leq C_{max}.$$

The optimal architecture is therefore not necessarily the architecture with the maximum individual robustness or adaptability. Rather, it is an architecture located on an appropriate **Pareto-optimal frontier**.

---

20. Architecture Evaluation Framework

The proposed architecture can be evaluated using five dimensions:

Dimension	Representative indicators
Effectiveness	Capability fulfillment, functional coverage
Robustness	Failure tolerance, redundancy, performance degradation
Adaptability	Reconfiguration cost, response time, structural flexibility
Information quality	Accuracy, timeliness, reliability
Complexity	Number of nodes, edges, dependencies, coordination burden

A comprehensive architecture score can be calculated as

$$S_A = \sum_{i=1}^{5} w_i S_i.$$

This provides a standardized mechanism for comparing alternative architectures.

---

21. Architecture Iteration and Optimization

The proposed framework adopts an iterative process:

Requirements → Conceptual Architecture → Physical Architecture → Logical Architecture → Simulation → Evaluation → Optimization → Architecture Update

If the evaluated architecture does not satisfy the required performance:

$$S_A < S_{min},$$

the architecture returns to an earlier design stage.

This feedback mechanism is important because architecture quality cannot always be determined during the initial design phase.

The iterative process can be expressed as

$$A^{(k+1)} = \mathcal{O}\left[A^{(k)}, F(A^{(k)})\right]$$

where $\mathcal{O}$ represents the architecture optimization operator.

---

22. Discussion

The proposed multidimensional architecture framework extends conventional architecture design in three important ways.

First, it establishes explicit relationships among conceptual, physical, and logical architectures. The conceptual architecture determines the capability requirements, the physical architecture maps capabilities to system resources, and the logical architecture defines how these resources interact and perform functions.

Second, the proposed approach transforms robustness and adaptability from descriptive concepts into measurable architecture variables.

Third, the framework supports iterative optimization rather than one-time architecture construction.

From a systems-engineering perspective, this is particularly important because architecture quality emerges from interactions among multiple dimensions rather than from any single structural characteristic.

A system may contain highly capable individual components but still exhibit poor system-level performance if its information connectivity is weak. Similarly, a highly connected architecture may exhibit poor robustness if it depends excessively on a limited number of central nodes.

Consequently, architecture evaluation should focus on the **relationship structure among components**, rather than simply on individual component performance.

---

23. Proposed Architecture Optimization Algorithm

The overall optimization procedure can be summarized as follows.

Step 1: Requirement initialization

Define mission requirements and system constraints.

Step 2: Conceptual architecture generation

Construct alternative conceptual architectures according to environmental and capability requirements.

Step 3: Physical architecture generation

Select candidate components and construct alternative physical configurations.

Step 4: Logical architecture generation

The International Journal of Applied Sciences.
Vol. 2, No. 1, 2025, pp. 139-160
ISSN 3104-8935
https://international-journal-of-applied-sciences.jo

Construct organizational, functional, information, and temporal relationships.

Step 5: Network modeling

Convert the architecture into a directed weighted graph.

Step 6: Robustness evaluation

Apply representative disturbance scenarios and calculate

$$R(A).$$

Step 7: Adaptability evaluation

Apply representative environmental or mission changes and calculate

$$D(A).$$

Step 8: Multi-objective optimization

Calculate

$$J(A)$$

and identify Pareto-optimal architectures.

Step 9: Simulation and validation

Evaluate candidate architectures under multiple representative scenarios.

Step 10: Iterative refinement

Return to the architecture generation stage if the required performance has not been achieved.

---

24. Conclusions

This study proposed a multidimensional architecture design framework for cross-domain operational systems and extended the conventional conceptual–physical–logical architecture methodology by incorporating **robustness and adaptability as explicit design variables**.

The principal conclusions are as follows.

(1) A unified multidimensional architecture framework was established. The framework integrates conceptual architecture, physical architecture, and logical architecture into a continuous architecture-development process, thereby improving traceability between requirements, system elements, functions, and information relationships.

(2) Architectural robustness was introduced as a quantitative variable. By considering component degradation, communication disruption, environmental uncertainty, and resource reduction, the proposed robustness index can characterize the ability of an architecture to maintain essential functions under disturbances.

(3) Architectural adaptability was introduced as a second major variable. Adaptability measures the ability of an architecture to modify its configuration, organizational relationships, functional pathways, and information interactions when environmental or mission conditions change.

(4) A directed complex-network representation was developed to describe relationships among heterogeneous architecture elements. Network connectivity, centrality, density, path length, and redundancy can therefore be incorporated into architecture evaluation.

(5) A multi-objective optimization model was proposed to jointly optimize system effectiveness, robustness, adaptability, information quality, structural complexity, and reconfiguration cost. This approach avoids optimizing a single architecture characteristic at the expense of overall system performance.

(6) Robustness and adaptability should be considered jointly rather than independently. Excessive redundancy may improve robustness while increasing complexity, whereas excessive structural flexibility may improve adaptability while increasing reconfiguration costs. Consequently, the desired architecture should be selected from a Pareto-optimal solution set according to specific requirements and constraints.

(7) The proposed framework establishes a foundation for subsequent MBSE-based architecture modeling, simulation, quantitative evaluation, and optimization. Future research can further introduce dynamic network models, uncertainty quantification, machine-learning-assisted architecture optimization, and digital-twin-based validation to improve the accuracy and efficiency of architecture evaluation.

Overall, the proposed methodology transforms multidimensional architecture design from a primarily descriptive activity into a **quantitative, iterative, and optimization-oriented systems-engineering process**. By explicitly integrating robustness and adaptability into conceptual, physical, and logical architecture design, the framework provides a general methodological basis for designing complex cross-domain systems capable of maintaining essential functions under uncertainty while dynamically adapting to changing operational requirements.

Ethical Considerations

Not applicable. This study did not require ethical approval because it does not include human or animal subjects and does not involve any personal or sensitive data.

List of Abbrevations:

None.

Acknowledgment:

The International Journal of Applied Sciences.
Vol. 2, No. 1, 2025, pp. 139-160
ISSN 3104-8935
https://international-journal-of-applied-sciences.jo

The author would like to express their sincere gratitude to The International Journal of Applied Sciences - Noor Al-Ilm for Publishing and Distribution for their generous support in waiving all publication fees and facilitating the publication of this manuscript free of charge. Their commitment to promoting scientific research and supporting researchers is highly appreciated.

Author Contribution:

All authors contributed equally to the main contributor to this paper. All authors read and approved the final paper.

Declaration of generative AI and AI-assisted technologies in the writing process

The authors hereby declare that no generative artificial intelligence or AI-assisted technologies were used at any stage during the preparation of this manuscript, including language editing, proofreading, or content development. The authors take full responsibility for the originality and integrity of the work presented in this publication.

Funding:

This research received no external financial funding. The authors also acknowledge The International Journal of Applied Sciences, Noor Al-Ilm for Publishing and Distribution, for providing a full waiver of the publication fees. The publication fee waiver was provided as editorial support and did not involve any financial contribution to the conduct, design, analysis, or reporting of the research.

Conflicts of Interest:

"The authors declare no conflict of interest."