SECTION 8. SYSTEMS APPROACH: ADVANCED METHODOLOGIES DESIGN

DOI https://doi.org/10.30525/978-9934-26-597-6-80

INTEGRATING A NEW ELEMENT INTO THE SYSTEM: BELATED ALGORITHMIZATION

Cobzaru Alex[1], Mihalache Andreea[2*]
[1] Alexandru Ioan Cuza University of Iași, Carol I Boulevard, No. 11, 700506, Iași, Romania
[2]Alexandru Ioan Cuza University of Iași, Carol I Boulevard, No. 11, 700506, Iași, Romania
**Corresponding author's e-mail: acobzaru88@gmail.com*
**Corresponding author's e-mail:mhc.andreea03@gmail.com*
Received 20-25 April 2025, www.isma.lv

Abstract

Adding new elements into established systems is one of computer system design's principal issues. This paper will investigate the incorporation of new components into existing systems using the concept of belated algorithmization, highlighting its effect on software system stability. Through the systems theory approach, will be illustrated how changes to input data at late stage, without corresponding algorithmic updates, contravene emergent system properties and degrade performance. Comparing system models of ROI evaluation, it can be demonstrated how the addition of minor elements can create notable changes in system behaviour. The key of maintaining resilience is the balance between physical and logical database independence, which allows systems to progress without affecting functionality. Based on the conceptual framework formulated based on the principles of independence, superposition, and emergence will be created a guide for sustainable system development. The study highlights the importance of adaptive maintenance practice that support scalability and stability where technology changes very fast. This approach ensures continued performance and integrity, even when systems undergo frequent updates and structural changes.

Keywords: belated algorithmization, system stability, emergent properties, database independence, adaptive maintenance.

1 Introduction

Today's software systems operate in a state of perpetual change, with top priorities like sustainability and adaptability. This thesis begins with three key observations. Firstly, the questionnaire, which is commonly used to identify suitable candidates for organizational roles, reveals deeper insights reflecting an ongoing industry-wide need to identify professionals who can sustain complex software systems across their full life cycles. Secondly, the quality growth potential of software in 2025 is linked to the leader's ability to manage system maintenance, an ongoing process that is more important than initial development. Thirdly, reflecting on the 2/98 rule, where only 2% of effort yields 98% of results, and most of the output tends to come from a small percentage of effort, initiated by a skilled expert [1-2].

The responses to certain questions reveal a more authentic reality: successful system design relies on understanding the emergence property, where system behavior is a result of interactions among components. Over 90% of system design is a management process, and implementing ROI often takes over half a year.

2 Overview

System maintenance depends on physical and logical database independence, which gives room for system changes [3]. But if it is done without respecting the superposition principle, the new models added upon existing layers of them, emergence may be disallowed and provoke instability. This research is built around the triangle of Independence, Superposition, and Emergence principles. It explores how this triad can be preserved and balanced to support long-term system resilience, especially in environments where continuous updates are unavoidable [4]. From theory to practice, this paper proposes a design approach for systems that can evolve and thrive in the presence of complexity and change. In doing so, it contributes to our knowledge of sustainable software system management and the role of strategic decision-making in ensuring systemic integrity. In practical terms, we make the transition to the scheme:

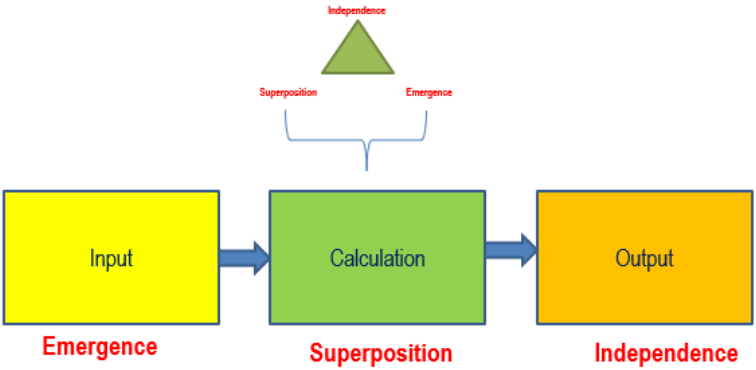

Figure 1. Emergence–Superposition–Independence: A Framework for Adaptive System Design

The studyr considers an example of assessing the effectiveness of an organization using the ROI methodology. During the assessment, the effect of adding a new element to the system was tested. Six critical elements must be addressed in this methodology (See Table 1).

Table 1

The concept of embedding a new element into the system

	Elements	**Content**
1	**Contradiction**	A fundamental tension emerges in system maintenance practices. Frequent software modifications often lead developers to alter input data sets while neglecting corresponding algorithmic updates, creating systemic imbalances.
2	**Problem**	The main challenge is declared by the axiom: "System maintenance stability cannot be guaranteed without verifying emergent properties while preserving database independence effects." This principle underlies all subsequent design decisions.
3	**Object**	The research focuses on a specialized ROI assessment system within computer system design frameworks. This system serves dual purposes: evaluating financial claims and optimizing organizational decision-making processes.

4	**Impact of New Elements:**	The integration of novel elements affects system behavior through: – Algorithmic modifications that alter ROI calculation methodologies – Evolving performance indicator requirements – Changing data integration paradigms
5	**Implementation Tasks:**	Successful system integration requires: – Database Independence: Ensuring ROI model compatibility across diverse database architectures with flexible financial indicators. – Algorithm Adjustments: Implementing dynamic, data-driven computational models that maintain operational efficiency. – System Stability: Conducting comprehensive stress testing to preserve performance integrity during transitional phases.
6	**Outcomes Achieved:**	The implemented solution demonstrates: – Enhanced System Resilience: ROI models now accommodate financial metric evolution without structural overhauls. – Improved Adaptive Capacity: Support for customized financial modeling scenarios. – Optimized Maintenance: Streamlined integration of new elements with minimal operational disruption

Unlike static computation models, this approach preserves the property of emergence, ensuring that the behavior of the system develops holistically with the data it receives.

The results obtained demonstrate how the system architecture can remain stable under continuous changes when the principles of independence and superposition are observed. As a result, it was established that the system supports scalable ROI assessment and dynamic adaptability of the system, thereby restoring system balance.

4 Conclusions

The long-term support of intricate software systems in contemporary organizations relies on the prudent addition of new features while maintaining system stability, as demonstrated by our three-model case study. Ensuring the stability of modern software systems requires not just a strategic balance between adaptability and structure, but a rigorous application of the core principles examined in this work: database independence, emergent property preservation, and superposition-aware design.

By maintaining both physical and logical database independence while respecting the principle of emergence organizations can support continuous system evolution without compromising performance. Our findings confirm that system updates, when properly managed through the structured maintenance strategy outlined in this study, are indeed vital for sustaining long-term efficiency, resilience, and growth. This is particularly crucial in our current technological environment where change is continuous and inevitable.

The practical validation through ROI assessment systems provides concrete evidence that applying systems theory to design and revision processes represents a reliable pathway to keeping systems both strong and prepared for long-term growth. Future work could explore automating the verification of emergent properties during system updates, building on the foundation established here.

References

1.Chroust, G. (2002). EMERGENT PROPERTIES IN INFORMATION SYSTEMS, /10th Interdisciplinary Information Management Talks, Sept, 2002, Zadov

2.Christopher W. Johnson, (2005). What are Emergent Properties and How Do They Affect the Engineering of Complex Systems?

3. Date, C.J. (2004) An Introduction to Database Systems , 8th Edition, Pearson Education

4. Kossiakoff, F., Swee, N., Seymor, S., Bier, S. (2011) Systems Engineering Principles and Practice. John Willey & Sons