DOI: 10.63550/ICEIP.2025.54.61.015

Development of Automated Systems Based on the Natural-smart Interface

A. I. Razumowsky[a]
Trapeznikov Institute of Control Sciences of Russian Academy of Sciences, 65 Profsoyuznaya street, Moscow 117997, Russia
razumowsky@yandex.ru

Keywords: Automated system, dual representation, nature-smart interface, experimental validation.

Abstract: This paper proposes a paradigm where software design is guided by an external control framework capable of simultaneously managing task algorithmization/structuring and enabling autonomous analysis of system elements beyond their contextual dependencies. Central to this approach is dual representation — a method allowing arbitrary coordination of structural components through a natural-intelligent interface. This interface facilitates dual perception of information space elements, reconciling their roles as interconnected structural entities and contextually independent units. It will increase adaptability in the design of complex systems by maintaining structural consistency and facilitating creative, out-of-context exploration of components.

[a] https://orcid.org/0000-0003-1769-0178

DOI: 10.63550/ICEIP.2025.54.61.015

1 INTRODUCTION

When developing complex specialized automated systems, the interplay between formal and informal components in the development environment plays a key role. Such systems are best classified as automated design and software lifecycle management systems (ASD/ALM). Their foundation remains the "human factor" — the informal component that defines the conceptual core of the process. The use of such systems relieves developers from routine coordination of details, allowing them to focus on the creative aspects of design (Booch, 2006). At the same time, automated systems can be viewed as tools for "accumulating computational knowledge" about machine methods in control and regulation theory (Floridi, 2017). Thus, the very concept of automation implies that a system's lifecycle must incorporate elements of creative oversight. This is underscored by the need to ensure not only the quality of outcomes but also the reliability of their evaluation. As Floridi and Cowls rightly point out, "The problem of solution validity lies at the intersection of the real world and theoretical knowledge, with humans acting as the mediator" (Floridi & Cowls, 2022). The contradiction between a task's formal description and its qualitative interpretation leads to a gradual degradation of the software product's characteristics over its lifecycle. This manifests as reduced operational reliability and deteriorating interface ergonomics, disrupting the seamless interaction between humans and software components in automated processes.

2 LITERATURE REVIEW

The principles of orderliness — including logical sequencing, structured hierarchies, cyclical process organization, and systematic prioritization — underpin the effectiveness of automated systems designed for qualitative decision-making processes. By embedding these organizational frameworks, developers ensure that complex evaluations are conducted with consistency, transparency, and alignment to predefined objectives, thereby enhancing the reliability and trustworthiness of the system's outcomes. As noted by C. Rovelli: "the discovery of order in nature fosters the intellectual confidence necessary for scientific inquiry" (Rovelli, 2018), a notion that aligns with the need to control the manageability of a project's informational content, which directly impacts its successful implementation. The necessity to structure inherently chaotic and heterogeneous data has driven the emergence of foundational concepts like data encapsulation (ensuring element isolation) and polymorphism (flexible use of data structures while preserving their integrity) (Stroustrup, 2013).

Developing plans for complex CAD (Computer-Aided Design) systems involves overcoming barriers, with knowledge management and decision-making standing out as central challenges. Study (Huang et. al., 2018) proposes a knowledge graph for visualizing ideas in solving geophysical tasks, providing a structural representation of connections between concepts, while research (Keil et. al., 2000). analyzes risk factors in multicultural project teams. The success of CAD development hinges on strategies for motivating participants, where project advancement decisions are tied to developers' mindsets (Akar et.al., 2019). A key tool for reducing programming errors is dual-process theory, which integrates intuitive and analytical thinking (Hadar, 2013). As highlighted in (Paivio, 2013), the fundamental contribution to this field lies in distinguishing cognitive mechanisms into automatic and controlled processes.

3 DEFINITION PROBLEM

For developing a reliable CAD algorithm, two key tasks must be addressed: creating autonomous algorithmic solutions and establishing clear, manageable connections between functional elements. Each of these tasks requires analysis from the perspective of how the algorithm's informational components — including its local blocks and results — are perceived. Special attention should be given to the fact that control over algorithmization and structuring of tasks in complex deterministic software solutions for automated systems must transcend formal constraints tied to the sequence of operators and data.

The delineation of procedural and structural connections should not be rigidly bound to a project's formal requirements or entrenched CAD design traditions. Instead, software component development demands external control that encompasses both algorithmization and structuring, as well as independent analysis of each element outside its interactions with other system components.

This approach is grounded in the principle of dual representation — the free interplay of the system's structural elements. A bridging mechanism is the intelligent interface, which enables a duality of perception: elements of the information space must be viewed both within structural connections and in

isolation. This duality avoids fixation on explicit dependencies between task parameters, overcomes artificial "OR" dichotomies, and moves beyond template-driven associations.

The core objective is to account for all factors, even those indirectly influencing outcomes, through maximum isolation of informational elements. Analyzing components outside their interdependencies grants designers the freedom to redefine their roles in the system, as existing connections lose rigid constraints. This simplifies the discovery of new solution combinations, enhances design flexibility, and instills a sense of control over the structure and content of the information space across all stages of the automated system's lifecycle.

4 METHOD

4.1 Information Space Intellectualization

The factor of dual representation of elements in the system's information space can be illustrated as follows. Figure 1 depicts the traditional approach to perceiving and controlling structural dependencies between algorithmic elements (AE1 and AE2). A natural intellectual interface monitors existing connections between these elements. This control yields both general parameters characterizing the aggregate of AE1 and AE2, as well as specifics of their interrelations.

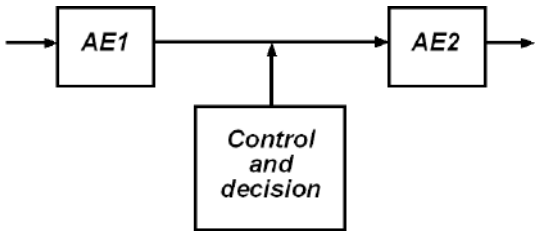

Figure 1: Traditional model of control over algorithm elements.

The scheme of natural-intellectual dual representation is illustrated in Figure 2. It is based on separating the perception of AE1 and AE2 into distinct components. Autonomous algorithmic elements (AAE1 and AAE2) act as virtual projections of AE1 and AE2, which must be perceived in isolation — both from one another and outside the parameters of their interconnections within the algorithm's structure. At the same time, control over connections within the algorithmic structure, consistent with the classical algorithmization approach, is preserved.

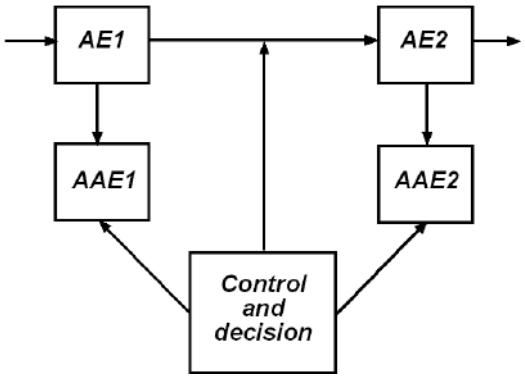

Figure 2: Dual representation model of algorithmic elements.

Thus, dual representation enables simultaneous observation of the information space in two fundamentally distinct planes: the classical structural-algorithmic and the autonomous-chaotic. Under this approach, any studied element — whether an operation or operand — gains the ability to function outside predefined structural constraints and semantic frameworks. This opens the possibility of overcoming even categorical boundaries, where, for example, a function may be interpreted as a static quantity, and a data structure as a full-fledged functional element — redefining traditional design paradigms.

The proposed strategy offers several key advantages. First, elements that in classical algorithmic or class hierarchy schemes lack rigid sequencing or direct dependencies acquire potential for joint analysis and interaction. Second, it becomes possible to establish new connections — both parallel and auxiliary — integrating components into the system that were not initially specified in the project's design. Such connections may include updated components or parts fulfilling supplementary roles, enhancing system adaptability without compromising its base architecture.

4.2 Working Experiment

For the experimental verification of the dual-representation method, based on the natural-intellectual splitting of perception, it is necessary to confirm its effectiveness in practice. Successful application of the method must demonstrate two key aspects. First, it must ensure the complete isolation of elements within the designed structure, meaning

their analysis under the "as-is" principle—without considering the mechanisms of their integration into the system. Second, it must investigate the properties of element connections within the context of the software solution, revealing their dynamics and adaptability. The experiment involves introducing an additional component with conditionally predictable behavior into an ordered set of algorithmic elements. As the "known aspect" of such an element, it is advisable to select its semantic interpretation: hypotheses about its purpose, application scenarios, assessment of functional potential, and other cognitive criteria.

During the experiment, the hypothesis regarding the necessity of arbitrary coordination of elements in the algorithmic structure is expected to be confirmed. This will not only enable a holistic analysis of the informational design space but also create conditions for the mutual interpenetration of element properties, free from traditional formal constraints. Such coordination opens the possibility of overcoming rigid frameworks, for example, combining functional and structural characteristics in non-standard configurations.

The dual-representation method emphasizes the priority of informal and natural descriptions, as they simplify the perception and manipulation of elements. As some researchers note: "although the steps we follow in formalizing the strategy may appear mechanical, it is not an automatic procedure... [it] requires a great deal of real world knowledge and intuitive understanding of the problem " (Abbott, 1983).

At the same time, the key objective of the method is revealed in the following: the generation, dissemination, and implementation of new ideas, as well as a comparative analysis of the potential of the system's structural elements.

Consider an algorithm for reading data, followed by re-encoding and placing the transformed structural elements of the file into a database (DB). Its structure is quite standard (Fig. 3). However, developers rarely focus on elements that are not directly connected, as the fundamental rule of formal "input-output" linkage, underlying the synthesis of algorithmic blocks, limits interaction between components. This minimizes the influence of elements that are not adjacent in the algorithmic space.

In other words, the direct "File" ↔ "DB" connection (Fig. 3) is absent and, consequently, remains outside the designer's analysis. Unlike this, the dual-representation method allows for arbitrary connections between any elements. This makes it meaningful to separately study potential pairwise interactions, such as: "File" ↔ "DB"; "File" ↔ "Converter"; "File" ↔ "Costs".

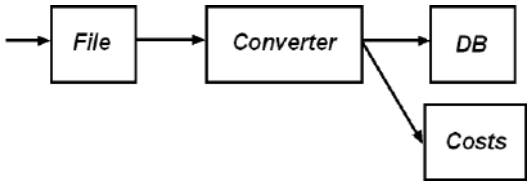

Figure 3: Simplified data converter algorithm structure.

Analyzing the interaction "File" ↔ "DB" outside the standard algorithmic framework allows for proposing the idea of designing an additional element—"Write to File". As seen in the algorithm's structure in Figure 3, the linkage "File ↔ Converter ↔ DB" is traditionally perceived as a sequence: read file → transform data → write to DB. However, such a linear approach limits the ability to detect functional elements that extend beyond the algorithm's core logic.

The dual-representation method demonstrates its effectiveness here: associative thinking tied to the concept of "File" leads to identifying a "mirror" element—"Write to File". This element not only complements the standard process but also expands the system's functionality. For example, it can be used to implement data quality checks in the DB by comparing the original data (D) and twice-transformed data (D''):

After reading and conversion (D → D').

After reverse conversion and writing to a file (D' → D'') (Fig. 4).

This correspondence ensures data consistency across all stages, eliminating discrepancies between file primitives and DB elements.

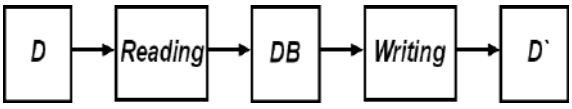

Figure 4: Information model of harmonization of original and transformed data.

Now it is possible to directly compare the original data D and the transformed data D', which not only identifies the need for corrections in the "Converter" module but also prioritizes the importance of internal file elements. First, this approach enables the processing of isolated, standalone data—for example, primitives—which is particularly critical when working with undocumented or poorly structured formats. Second, it allows evaluating the algorithm's results at early

design stages rather than post hoc during debugging and testing, when making changes becomes resource-intensive.

In both cases, the focus on isolated data elements, free from contextual dependencies with other components of the information space, provides developers with fundamentally new advantages. This allows concentrating on the internal properties of the studied functionality by assessing the results of executing a locally extracted program, removed from the algorithm's broader context. Such an autonomous element—whether an elementary operation or an isolated data primitive of a specific format—loses properties of interdependence, i.e., artificial constraints imposed by the system's synthesis. As a result, the designer's mind is freed from the confines of formally defined templates and gains the ability to generate arbitrary ideas and implementation methods previously blocked by traditional decision-making patterns.

Furthermore, the dual-representation method establishes a foundation for informal correlation of algorithmic elements. This opens a path to flexible system modification: additional modules can be introduced into the project, components can be replaced, and interaction logic can be refined. The method seeks to produce "pure thinking" beyond formal abstraction, which is related to the desire to connect the certainty of the thinkable with the ideality of thinking.

Consistent application of the dual-representation principle to the perception of individual elements in the design space leads to structural changes in the algorithm. As shown in Figure 5, its architecture now reflects not only formal connections but also associative dependencies revealed through the analysis of autonomous components.

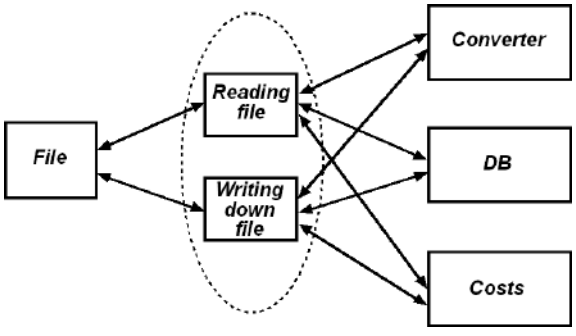

Figure 5: Structure of the data converter algorithm applying the dual representation method.

This enables the creation of a more adaptive and reliable system, where every element can be reimagined and modified without compromising the integrity of the entire algorithm.

Clearly, the result of expanding the perception of the system's properties is the formation of a flexible and productive algorithm structure, which implements not only the original analytically and technically justified actions but also additional operations. These additions provide three key advantages: first, functional completeness of the algorithm through expanding the system's capabilities; second, the possibility of a priori debugging, i.e., testing solutions during the design phase before the project's final implementation; and third, incorporating algorithm execution results in real time by integrating feedback directly into the development process.

Summarizing the transformation process of perception under the dual-representation method, a sequence of stages can be outlined. It begins with the initial traditional perception of the structure, where the algorithm is analyzed through the lens of formal connections and templates. Next comes the semantic splitting of perception—for example, dividing the "File" object into opposing aspects like "Reading" and "Writing"—which uncovers hidden functional possibilities. The subsequent step involves expanding connections between elements, such as establishing new relationships between the "Converter" and data verification. The final stage is intuitive-associative supplementation, where new elements are introduced based on informal assumptions, such as the DB data quality check function. This process transforms not only the algorithm's structure but also the very approach to design, shifting the focus from formal logic to a synthesis of analytical and creative thinking. This enables the creation of more adaptive and reliable systems, where every component can be reimagined without compromising the integrity of the entire architecture.

5 RESULTS AND DISCUSSION

Within the dual-representation method, any element introduced into the structure of an algorithm or software system can attain experimental status. This is because its emergence is not initially predefined by the design context, meaning its creation process is spontaneous. Paradoxically, such an element becomes more manageable. Its parameters depend primarily on the designer's natural-intellectual perception and only secondarily on formal interactions with surrounding operations and

operands, which define its behavior and properties within the system. This is where the essence of the dual-representation strategy unfolds: the cognitive and structural aspects of the element are synthesized under the influence of its categorical attributes and functional needs, forming a holistic entity that is simultaneously distributed and balanced.

Moreover, the designer's perception is not confined to analytical and formal requirements for the element. Every algorithm component becomes the result of a creative process—associations, intuitive assumptions, subjective evaluations. This can lead to unconventional, even whimsical structures, as their synthesis is based not on classical criteria (execution speed, resource intensity, reliability) but on creative principles. These include:

- ease of formalization,
- convenience of selecting options,
- flexibility of structural modification.

Thus, the dual-representation method ensures the natural-intellectual integrity of the design information space. Its value for developing automated systems lies in the simultaneous control over automation processes and their comprehension—assessing the necessity, importance, and responsibility of applying solutions within a comfortable, intuitively understandable interaction with the system.

Elements not connected by algorithmic adjacency are rarely considered in relation to each other. This creates an illusion of their independence from other components, increasing the risk of erroneous decisions. At the early stages of algorithm development, it is critically important to obtain its operational results as quickly as possible. These results are necessary to compare with algorithmic elements and adjust them subsequently. Incorporating supplementary elements like detailed documentation, paired associations, execution results, and debugging data into the development workflow greatly enhances comprehension of the algorithm's core logic. Decisions regarding algorithm modification or correction using the dual-representation method become simpler and more reliable, as this approach relies on the principles of dual-process psychological theory.

6 CONCLUSIONS

The dual representation method demonstrates its relevance and universality in modern contexts, finding application not only in CAD development but also in fields such as recognition algorithm design, robotics, and unmanned aerial vehicle (UAV) control. Its use enables a comprehensive exploration of a task's challenges and potential solutions, transcending traditional formal-technical analysis confined to standard approaches for synthesizing local algorithms and automated system structures. Dual representation, acting as a synthesis of multidimensional perception of the information space, facilitates the identification of new algorithmic expansion pathways, simplifies modifications to the system's architecture, and allows parallel analysis of multiple algorithmization options. This ensures the selection of optimal solutions based on creative criteria rather than template-driven rules.

A key advantage of the method remains its ability to adapt to a task's unique characteristics and account for a designer's individual preferences — particularly valuable when addressing complex, weakly deterministic challenges in developing software modules for automated systems. Such flexibility overcomes the limitations of traditional methods, where formal requirements often stifle innovative ideas, and enables the creation of solutions that blend analytical rigor with creative freedom. Ultimately, the method transforms the very approach to design, turning it into a process where technical precision is complemented by intellectual adaptability, and algorithms become not just tools but "co-authors" in discovering non-obvious solutions. This opens new horizons for developing adaptive and reliable systems capable of evolving under uncertainty.

REFERENCES

Abbott, R. J., 1983. Program design by informal English descriptions. *Communications of the ACM*, 26(11), 882-894.

Akar, E., Mardikyan, S., Dalgic, T., 2019. User roles in online communities and their moderating effect on online community usage intention: An integrated approach. *International Journal of Human–Computer Interaction*, 35(6), 495-509.

Booch, G., 2006. Object oriented analysis & design with application, Pearson Education India.

Floridi, L., Cowls, J., 2022. A unified framework of five principles for AI in society. *Machine learning and the city: Applications in architecture and urban design*, 535-545.

Floridi, L., 2017. The logic of design as a conceptual logic of information. *Minds and Machines*, 27(3), 495-519.

Hadar, I., 2013. When intuition and logic clash: The case of the object-oriented paradigm. *Science of Computer Programming*, 78(9), 1407-1426.

Huang, X, Wang, L, Ruan, J, Wu, S., Li, D., 2018. Exploration research of thinking innovation based on PCST theory and knowledge graph. *Chinese Automation Congress (CAC)* (pp. 1040-1045). IEEE.

Keil, M., Tan, B. C., Wei, K. K., Saarinen, T., Tuunainen, V., Wassenaar, A., 2000. A cross-cultural study on escalation of commitment behavior in software projects. *MIS quarterly*, 299-325.

Paivio, A., 2013. *Imagery and verbal processes*. Psychology Press.

Rovelli, C., 2018. *Reality is not what it seems: The journey to quantum gravity*. Penguin.

Stroustrup, B., 2013. *The C++ programming language*. Pearson Education.