

# Cognitive Effectiveness in Requirements Engineering: A Comparative Framework for Visual Notation Evaluation

Daria Maria Meseşan<sup>1</sup>

Research Methodologies and Academic Writing, Business Modeling and Distributed Computing Master's Degree, Babeş Bolyai University, Cluj-Napoca, Romania

**Abstract.** Visual notations represent the building base of diagrams in Requirements Engineering (RE). They simplify the representation of complex systems and improve communication between stakeholders by assessing cognitive effectiveness. However, the visual quality of modeling languages varies significantly, impacting their usability and adoption. Building on previous studies on Physics of Notations (PoN) and evaluation of i\*, this paper introduces a comparative framework for evaluating visual notations based on five key principles: Semiotic Clarity, Perceptual Discriminability, Semantic Transparency, Visual Expressiveness, and Graphic Economy. We apply the method to four widely used visual notations: i\*, KAOS, UML and BPMN, using criteria derived from prior studies. Results revealed significant variations: BPMN demonstrated the highest cognitive effectiveness with consistently high scores across all dimensions, while i\* faced challenges in clarity, transparency, and discriminability. UML showcased adaptability and transparency, but struggled with graphic economy, and KAOS achieved moderate performance across all principles. The trade-offs between principles, such as balancing graphic economy with expressiveness, underscore the importance of context-specific notation selection. This study highlights the need for future work in integrating notations to leverage their complementary strengths, fostering more effective tools for RE tasks.

**Keywords:** Requirements Engineering · Visual Notation · Physics of Notation · Cognitive Effectiveness · Comparative Analysis.

## 1 Introduction

In the context of building complex engineering systems and technologies driven by Artificial Intelligence (AI), **requirements engineering (RE)** has become a critical step in the early stages of system development. The primary objective of RE is to clearly define the problem a system aims to solve while aligning proposed features with stakeholder needs. This process not only minimizes persistent and costly errors but also ensures that the system fulfills its intended purpose efficiently [1].

Requirements Engineering heavily relies on visual diagrams to represent concepts, relationships, and processes, making complex systems easier to understand. These diagrams play a vital role in facilitating communication between technical teams and stakeholders. [13] The **visual syntax**, which defines the set of graphical elements used to construct these diagrams, is a key factor in their cognitive effectiveness [10]. However, despite its importance, visual syntax remains an under-explored area in RE research, with much of the focus traditionally placed on the semantics of notations—the meaning associated with constructs—rather than their perceptual representation [10].

The aim of this paper is to develop a comparative framework for evaluating visual notations using the principles of the Physics of Notation. We use the research by Moody et al. [10] on *i\** as a starting point and extend it to other visual modeling languages in a comparative way. Specifically, the research addresses the following questions:

- How do *i\**, KAOS, UML, and BPMN perform in terms of cognitive effectiveness?
- What trade-offs exist between these notations in their support for different RE tasks?
- How can these insights guide the selection or combination of notations in RE practice?

To answer these questions, we employ a theoretical evaluation methodology, scoring each notation against five PoN principles: Semiotic Clarity, Perceptual Discriminability, Semantic Transparency, Visual Expressiveness and Graphic Economy. The results provide actionable insights into the strengths and limitations of each notation, offering an overview on what might be the best visual notation for different situations.

## 2 Related Work

**Visual notations** are structured constructions of graphical symbols, compositional rules, and semantics designed to represent complex systems and processes [3, 10]. They bridge communication gaps between stakeholders by providing intuitive, easy-to-understand representations of technical concepts. The primary goal of visual notation is to improve **cognitive effectiveness** – the ease, speed, and precision with which users process information [10].

### 2.1 Frameworks for Visual Notation Evaluation

Evaluating visual notations requires systematic frameworks that assess their usability and effectiveness. Over the years, several frameworks have been proposed, each with its strengths and limitations.

**Cognitive Dimensions of Notations (CDs)** Cognitive Dimensions of Notations (CDs) framework is one of the earliest tools for evaluating cognitive artifacts. It defines 13 dimensions, such as *error-proneness* and *abstraction gradient*,

providing a vocabulary for analyzing usability [5]. However, CDs lack specificity for visual notations, as they do not include detailed metrics for evaluating graphical representations [10].

**Semiotic Quality (SEQUAL)** Semiotic Quality (SEQUAL) framework evaluates modeling languages based on the semiotic ladder, addressing levels such as syntax, semantics, and pragmatics [6]. While comprehensive, SEQUAL primarily focuses on conceptual and theoretical quality rather than visual syntax, limiting its application to visual notation evaluation [9].

**The Physics of Notations Framework** Among the most prominent approaches for improving visual syntax is the **Physics of Notations (PoN)** framework, introduced by Moody [9]. PoN establishes nine principles for designing cognitively effective visual notations. These principles emphasize the importance of designing symbols and diagrams that are intuitive, visually distinct, and manageable in complexity. Studies have demonstrated the positive impact of PoN in evaluating and refining RE notations, particularly for addressing cognitive load and enhancing usability [3, 4, 10].

## 2.2 Key Visual Notations in Requirements Engineering

Visual notations represent the modeling language in Requirements Engineering (RE), aiding in the specification, analysis, and validation of complex systems. In this section we provide an overview of four widely used notations, emphasizing their features, applications, and challenges as addressed in the literature.

**i\*** is a goal-oriented visual notation introduced by Yu [13] which focuses on modeling requirements engineering systems by capturing dependencies and rationales among stakeholders. It uses two models: the Strategic Dependency (SD) model to outline relationships and the Strategic Rationale (SR) model to explore motivations behind goals. **KAOS**, a goal-oriented language, specializes in hierarchical goal decomposition and traceability, particularly suited for safety-critical systems. **UML** provides a versatile framework with multiple diagram types, such as class, sequence, and activity diagrams, offering broad applicability across system design tasks. Lastly, **BPMN** supports process modeling with standardized symbols, bridging technical and business stakeholders through its focus on workflows.

## 2.3 Applications of PoN in Visual Notations

Several studies have applied PoN principles to evaluate and refine visual notations, providing insights into their cognitive effectiveness:

**i\*:** Moody et al. [10] conducted a detailed analysis of i\* using PoN, identifying critical issues such as symbol overload and limited perceptual discriminability. i\* remains a robust tool for capturing stakeholder goals but struggles with large-scale diagram readability. Recommendations included simplifying the notation's visual vocabulary and improving semantic transparency [10].

**BPMN:** Genon et al. [4] applied PoN to BPMN 2.0, highlighting its strengths in standardization but also its deficiencies in dual coding and perceptual discriminability. Their findings emphasized the need for improved semantic transparency (more intuitive symbols) and better textual integration.

**UML:** Evaluations of UML underscore its flexibility but critique its graphic economy. Studies by Moody et al. [11] suggest limiting the number of diagram types in specific contexts to reduce cognitive load. Moody and van Hillegersberg assessed UML diagrams, revealing that while UML excels in integration of diagrams across different domains, its extensive symbol set often compromises graphic economy and the lack of color reduces visual expressiveness. [11].

**KAOS:** Studies on KAOS, including those by Dardenne et al. and Nwokeji et al. [2, 12], highlight its strengths in goal decomposition and formal validation but emphasize the need for improving its visual syntax to enhance accessibility for novice users. In addition, Matulevicius et al. [7] analyzed KAOS’s visual syntax, identifying areas for improvement in modularity and integration. Their recommendations include enhanced structuring techniques and better visual clues for cognitive integration.

Despite extensive research, gaps remain in the systematic evaluation of visual notations using PoN principles. Few studies provide comparative analyses across multiple notations, and trade-offs between principles, such as balancing semantic transparency with graphic economy, are rarely explored. Additionally, the combined use of notations like  $i^*$ , KAOS, UML, and BPMN for comprehensive RE tasks has not been sufficiently investigated. Integrating notations, such as using  $i^*$  for goal modeling and BPMN for process modeling, could address specific RE challenges, but this approach requires further investigation. This paper addresses these gaps by providing a structured comparison of these notations, offering actionable recommendations.

### 3 Framework Methodology

This section outlines the detailed methodology we used to evaluate the cognitive effectiveness of visual notations in Requirements Engineering (RE), based on the Physics of Notations (PoN) framework. We propose a structured approach to assess visual notations against five principles: **Semiotic Clarity, Perceptual Discriminability, Semantic Transparency, Visual Expressiveness, and Graphic Economy**.

#### 3.1 Criteria Application

**Semiotic Clarity:** The evaluation of semiotic clarity will focus exclusively on Symbol Overload, defined as the number of semantic constructs represented a single symbol representing multiple constructs [9, 10]. Symbol overload introduces perceptual ambiguity and challenges comprehension. To assess this, we calculate the average number of constructs per overloaded symbol and document specific instances where multiple constructs share one graphical symbol.

**Perceptual Discriminability:** Involves evaluating the uniqueness of visual variables across symbols and measure the perceptual distance between similar constructs [10]. We calculate the number of symbols that look too much alike.

**Semantic Transparency:** For this principle, we assess the degree to which symbols visually suggest their meanings [10]. The evaluation is done by counting the symbols that intuitively align with their semantic purpose.

**Visual Expressiveness:** We evaluate visual expressiveness by counting the distinct visual variables used in the visual notation [10]. There are 8 visual variables (Orientation, Texture, Shape, Size, Color, Color hue, Arrangement, Saturation, Value) [11] so we consider a number of 5 or more to be a high adherence to the principle.

**Graphic Economy:** Graph economy is evaluated by counting the total number of unique symbols in the notation and the score is given based on the ease of creating simpler diagrams. The human capacity to distinguish between different symbol categories is about 6 [8].

### 3.2 Scoring System

We propose a 1-to-3 scoring system, where each principle is scored on a scale of **1 (Low)**, **2 (Moderate)**, and **3 (High)**, with specific thresholds for each metric. The analysis involves generating specific metrics and scoring visual notations based on the outlined thresholds. The scoring system is summarized in Table 1.

**Table 1.** Scoring System for Framework Evaluation

Principle	Metric	Low-1	Moderate-2	High-3
<b>Semiotic Clarity</b>	Symbol Overload	$> 10$	3–10	$\leq 2$
<b>Perceptual Discriminability</b>	Similar Symbols	$\geq 6$	3–5	$\leq 2$
<b>Semantic Transparency</b>	Constructs Aligned	$\leq 2$	3–4	$\geq 5$
<b>Visual Expressiveness</b>	Visual Variables	$\leq 2$	3–4	$\geq 5$
<b>Graphic Economy</b>	Unique Symbols	$\geq 16$	7–15	$\leq 6$

### 3.3 Trade-Off Analysis

To address potential conflicts between principles, a trade-off matrix is constructed to visualize interactions. The trade-off matrix presented in Figure 1 illustrates the interactions between the five evaluation principles and it replicates Moody’s et al. [10] methodology. Positive interactions (+) indicate that improving one principle supports the other, while negative interactions (−) highlight conflicts where enhancing one principle might compromise another. Neutral interactions (+/−) suggest that the principles have little to no direct effect on each other. For example, improving **semantic transparency** often enhances **perceptual discriminability**, as intuitive symbols are easier to differentiate. Similarly, increasing **visual expressiveness** by using diverse visual variables

supports both **semantic transparency** and **perceptual discriminability**. Conflicts arise when enhancing one principle compromises another. For example, increasing **graphic economy**—by reducing the number of unique symbols to minimize cognitive load—can conflict with **visual expressiveness**. Neutral interactions (+/-) depend on context; for instance, improving **semiotic clarity** can either support or conflict with **graphic economy**, depending on whether redundant symbols are removed or additional ones are introduced.

		EFFECT								
		Semiotic Clarity	Perceptual Discriminability	Semantic Transparency	Complexity Management	Cognitive Integration	Visual Expressiveness	Dual Coding	Graphic Economy	Cognitive Fit
CAUSE	Semiotic Clarity							+		
	Perceptual Discriminability					+			+	
	Semantic Transparency		+						+	
	Complexity Management							-	+	
	Cognitive Integration	-		+				-		
	Visual Expressiveness		+					+	+	
	Dual Coding								+	
	Graphic Economy		+	+	-				+	
	Cognitive Fit									

**Fig. 1.** Interactions between principles: + (green cell) indicates a positive effect, - (red cell) indicates a negative effect, and (orange cell) indicates a neutral effect depending on the situation. Source: Moody et al. (2010), p.146 [10].

## 4 Results of Framework Application

This section presents our findings regarding the cognitive effectiveness evaluation for the four visual notations: *i\**, *KAOS*, *UML*, and *BPMN*, using the comparative framework proposed in the previous section. Table 2 summarizes the scores for each notation, categorized by the five principles of cognitive effectiveness from the Physics of Notations framework. Each score (Low, Moderate, or High) is supported by detailed reasoning and concrete examples.

**General results of evaluation:** **BPMN** scored the highest in cognitive effectiveness, particularly excelling in perceptual discriminability and semantic transparency, making it ideal for process modeling. **KAOS** stood out in graphic

economy, with a minimal and efficient symbol set that prioritizes simplicity and clarity. **UML** demonstrated strengths in perceptual discriminability and extensibility, offering flexibility for diverse use cases, though its extensive symbol set contributed to lower graphic economy. **i\*** performed well in visual expressiveness for a requirements engineering modeling language, utilizing multiple visual variables effectively, but it struggled with semiotic clarity and perceptual discriminability, indicating areas for improvement in symbol design and standardization.

**Table 2.** Overall Evaluation Scores by Principle

Principle	i*	KAOS	UML	BPMN
Semiotic Clarity	1-Low	2-Moderate	1-Low	3-High
Perceptual Discriminability	1-Low	2-Moderate	2-Moderate	3-High
Semantic Transparency	1-Low	1-Low	2-Moderate	2-Moderate
Visual Expressiveness	2-Moderate	2-Moderate	1-Low	3-High
Graphic Economy	1-Low	2-Moderate	1-Low	2-Moderate
Total Score	6	9	7	13

**Semiotic Clarity** - We used number of overloaded symbols as evaluation metric *i\**: Scored **Low (1)** due to **17 overloaded symbols**, including 22 relationships represented by only 5 arrow types. For example, *actor associations*, *dependencies*, and *contribution links* use identical arrow shapes, creating confusion [10].

**KAOS**: Scored **Moderate (2)** with **8 overloaded symbols**, such as 4 types of parallelograms for *goals*, *softgoals*, *requirements*, and *expectations*, and 4 arrow types. Subtle variations like border thickness and orientation are used for disambiguation. [7].

**UML**: Scored **Low (1)** due to **over 20 overloaded symbols** in Class Diagrams. For instance, dashed arrows can represent dependencies, interface realizations, or package imports, relying heavily on context [11].

**BPMN**: Scored **High (3)** with only **2 overloaded symbols**, such as gateways represented by plain or crossed diamonds. This minimal overload ensures strong semiotic clarity [4].

**Perceptual Discriminability** - We calculated the number of symbols that are hard to distinguish.

*i\**: Scored **Low (1)** with **23 similar symbols**, including 4 *actor* types, 2 *belief* symbols, 2 *dependency rationale* symbols, 6 *actor association* arrows and 9 *contribution* arrows. Association links and contribution links are visually indistinguishable without textual labels [10].

**KAOS**: Scored **Moderate (2)** with **5 similar symbols**, such as parallelograms for *requirements* and *soft goals*. Visual differentiation relies on bor-

der thickness or shading, which can be difficult to distinguish in complex diagrams [7].

*UML*: Scored **Moderate (2)** with **5 similar symbols**, including association, aggregation, and composition links. Distinctions depend on small variations like hollow or filled diamonds, which are not easily perceptible [11].

*BPMN*: Scored **High (3)** with only **2 similar symbols**, where a *collapsed sub-process* might resemble a *task*. Also, in some cases, *Pools* and *Lanes* could be easily confused by novice users. But overall, the use of distinct shapes and markers ensures high discriminability [10].

**Semantic Transparency** - Scored the number of constructs that visually represent their meaning.

*i\**: Scored **Low (1)** due to abstract, semantically opaque symbols that require prior learning. For example, *actors* are represented by non-intuitive shapes, and only the cloud symbol for *belief* achieves semantic transparency as a thought bubble. Improvements like stick figures for actors could enhance intuitiveness [10].

*KAOS*: Scored **Low (1)** with only two transparent symbols, the house-shaped symbol for *domain/property* or lightning symbol for *goals conflict*. The parallelogram shapes are not semantic transparent. [7].

*UML*: Scored **Moderate (2)**, as symbols like *actors*, *classes*, *components* and *interfaces* are intuitive, but relationships rely on similar arrows, requiring textual clarification for novice users [11].

*BPMN*: Scored **Moderate (2)** due to semantically opaque symbols like the *data file* symbol (resembling a sticky note) and gears (used for *service tasks*). As mentioned before, it is hard to distinguish and to guess the meaning of *Pools* and *Lanes* just by looking at their symbols. Some icons, like colored events, icons for different kind of tasks and diamonds for gateways, are intuitive, but in general transparency is inconsistent [4].

**Visual Expressiveness** - scored the number of visual variables used.

*i\**: Scored **Moderate (2)** as it uses **3 visual variables**: shape, brightness, and orientation. While it exceeds most RE notations relying on a single, or two variables, it lacks color, texture, or arrangement, limiting its adherence to the principle [10].

*KAOS*: Scored **Moderate (2)**, employing **4 visual variables**: shape, size, orientation, and element alignment. However, the absence of effective use of color or texture reduces its overall expressiveness [7].

*UML*: Scored **Low (1)**, with most diagrams using only **2 visual variables**: shape and value. Activity Diagrams stand out by utilizing spatial arrangement (x, y), but the lack of color or texture limits expressiveness [11].

*BPMN*: Scored **High (3)**, using **5 visual variables**: shape, color, arrangement, grain, and brightness. While color usage could be expanded even more, the existing variables provide a visually enriched representation [4].



**Graphic Economy** - Calculated the number of unique symbols

*i\**: Scored **Low (1)** with **16 unique symbols**, exceeding the threshold for moderate simplicity. While providing flexibility, the large symbol set increases complexity and risks cognitive overload for users [10].

*KAOS*: Scored **Moderate (2)** with **9 unique symbols**, striking a balance between simplicity and expressiveness. Its symbol set includes constructs like goals, obstacles, agents, and domain properties, making it efficient without being overly complex [12].

*UML*: Scored **Low (1)** with **31 unique symbols**, far exceeding the threshold for simplicity. The large symbol set risks overwhelming users, particularly novices, and complicates diagram interpretation [11].

*BPMN*: Scored **Moderate (2)** with **10 unique symbols** in practical usage, despite having a nominal complexity of 171 symbols. Practical usage includes core constructs like events, activities, gateways, and data objects, making it more accessible for novices [4].

## 5 Limitations and Future Work

The proposed framework has a series of limitations that offer opportunities for future research. While we applied the Physics of Notations (PoN) framework, we did not include all nine principles. Principles such as cognitive fit and cognitive integration are particularly challenging to measure quantitatively, as they rely on specific contexts and personal interpretations. Additionally, we assessed perceptual discriminability and semantic transparency using prior studies of the analyzed visual notations, but further research should involve investigations with real participants to gather statistical evidence on their effectiveness. We also limited the evaluation to four visual notations—*i\**, *KAOS*, *UML*, and *BPMN*—and applied only the PoN framework, without combining it with other evaluation methods. Furthermore, while PoN effectively evaluates visual syntax, it does not address practical factors like tool support or the adoption of notations in real-world applications.

Future work will focus on addressing the limitations identified in this study. We plan to expand the framework to include all nine principles from the Physics of Notations (PoN) and broaden its scope to evaluate additional notations and modeling languages, such as SysML and ArchiMate. To provide a more detailed understanding of subjective principles like perceptual discriminability and semantic transparency, we aim to conduct studies involving user experiments and statistical analysis. We also intend to extend the evaluation of semiotic clarity to include symbol redundancy, balance, excess, and deficit, rather than focusing solely on symbol overload. By integrating principles from other evaluation frameworks, we seek to enhance the robustness and comprehensiveness of the framework. Additionally, we will explore ways to automate the framework, making it adaptable for use across a wider range of notations and contexts.

## 6 Conclusion

This study extended the work of Moody et al. [10] on  $i^*$  by applying the PoN methodology to evaluate four widely used visual notations. We implemented a comparative evaluation framework, highlighting the importance of balancing cognitive effectiveness principles in RE. The analysis revealed that BPMN is the most effective, excelling across multiple principles, while  $i^*$  demonstrated areas for refinement, particularly in semiotic clarity and perceptual discriminability. KAOS was notable for its simplicity and graphic economy, and UML demonstrated versatility, but faced challenges with its extensive symbol set. These findings underscore the unique strengths and limitations of each notation, highlighting the importance of aligning notation choice with project requirements.

## References

1. Dominique Blouin. *Modeling Languages for Requirements Engineering and Quantitative Analysis of Embedded Systems*. PhD thesis, 12 2013.
2. A. Dardenne, A. van Lamsweerde, and S. Fickas. Goal-directed requirements acquisition. In *Science of Computer Programming*, volume 20, pages 3–50, 1993.
3. Randa El-Ghafar, Abeer Ghareeb, and Eman Nasr. Designing user comprehensible requirements engineering visual notations: A systematic survey. 12 2014.
4. Nicolas Genon, Patrick Heymans, and Daniel Amyot. Analysing the cognitive effectiveness of the bpmn 2.0 visual notation. In *Proceedings of the International Conference on Software Language Engineering*, pages 377–396, 2011.
5. Thomas Green and Marian Petre. Usability analysis of visual programming environments: A ‘cognitive dimensions’ framework. *Journal of Visual Languages and Computing*, 7(2):131–174, 1989.
6. O. I. Lindland, G. Sindre, and A. Solvberg. Understanding quality in conceptual modeling. *IEEE Software*, 11(2):42–49, 1994.
7. Raimundas Matulevicius and Patrick Heymans. Visually effective goal models using kaos. In *Advanced Information Systems Engineering Workshops (CAiSE)*, pages 94–108. Springer, 2007.
8. George Miller. The magical number seven, plus or minus 2: Some limits on our capacity for processing information. *Psychological review*, 63:81–97, 03 1956.
9. Daniel L. Moody. The physics of notations: Toward a scientific basis for constructing visual notations in software engineering. *IEEE Transactions on Software Engineering*, 35(5):756–779, 2009.
10. Daniel L. Moody and Patrick Heymans. Visual syntax does matter: Improving the cognitive effectiveness of the  $i^*$  visual notation. *Requirements Engineering*, 15(2):141–175, 2010.
11. Daniel L. Moody and Jan van Hilleghersberg. Evaluating the visual syntax of uml: An analysis of the cognitive effectiveness of the uml suite of diagrams. In *Proceedings of the 1st International Conference on Software Language Engineering (SLE)*, pages 16–34, 2008.
12. Joshua C. Nwokeji, Tony Clark, and Balbir S. Barn. Towards a comprehensive meta-model for kaos. In *2013 3rd International Workshop on Model-Driven Requirements Engineering (MoDRE)*, pages 30–39, 2013.
13. Eric Yu. Towards modeling and reasoning support for early-phase requirements engineering. pages 226–235, 02 1997.