

Designing and Validating Affordant Interfaces for Hybrid Intelligence Systems

Maximilian Kretzschmar^{1*}, Bernhard Saske¹, Jan Mehlstäubl², Kristin Paetzold-Byhain¹

¹Dresden University of Technology, Chair of Virtual Product Development, 01069 Dresden

²MAN Truck & Bus SE, AI Center of Enablement, 1050-053 Lisbon

*Corresponding author. E-Mail address: maximilian.kretzschmar@tu-dresden.de; Tel.:0351/46335360

Abstract: The integration of Large Language Models (LLMs) into engineering workflows is transforming automation from solely deterministic execution into hybrid intelligence, where humans and AI agents jointly reason and act. Compound LLM systems expand automation possibilities, however there is also a risk of obscuring their decision-making processes, challenging engineers' ability to validate results. Transparency therefore becomes a prerequisite for trust and accountability in designing such systems for engineering tasks. We explore how transparency can be both embedded in and empirically tested in such AI based expert systems. Drawing on Leonardi's affordance theory as well as the triangle of design responsibility, an industrial prototype of a compound LLM engineering system is developed and verified. The prototype integrates reasoning and tool agents with transparency mechanisms, enabling engineers to observe actions, data sources, and decision pathways. Based on empirical insights and combined with a literature analysis, a concept for validation and transparency evaluation is derived to assess whether hybrid intelligence systems are not only able to deliver value by solving required tasks but also remain interpretable and accountable. Building upon established verification and validation strategies the concept provides a foundation for the evaluation of transparency as a distinct quality dimension in AI-driven engineering systems.

Keywords:

Large Language Models, Multi-Agent Systems, Agentic AI, Human-AI Interaction

1 Introduction

Automation in engineering design aims to support engineers by embedding information and knowledge into reusable systems and tools [1]. As such systems evolve and more complex problems are tackled, a recurring trade-off emerges: deterministic, rule-based automation offers repeatability and control but becomes increasingly difficult to extend as design variability grows. In contrast, more adaptive or knowledge-rich systems can accommodate a wider range of situations but require substantial expert input and development effort. This practical trade-off between reliability and flexibility has been repeatedly observed in design automation research [2,3].

The introduction of Large Language Models (LLMs) has changed this dynamic. They can interpret natural language to reason across domains, coordinate other software tools and be designed to act with varying degrees of autonomy within Multi-Agent Systems (MAS), enabling hybrid configurations that combine deterministic computation with probabilistic reasoning. Their integration into engineering design introduces what Pradas Gómez et al. [4] describe as a shift in design responsibility – from a dual relationship between human and deterministic tools to a triadic configuration involving the designer, traditional tools, and LLMs. This “*triangle of design responsibility*” marks a transition toward hybrid intelligence systems where human and technological agencies continuously share control and decision-making.

As automation systems ascend this triangle towards systems that plan and act semi-autonomously the challenge is no longer limited to developing systems that solve engineering tasks efficiently, but also to ensuring that their reasoning and outputs remain transparent and accountable. Consequently, the interface layer between humans and such AI Systems becomes central to the human-AI relationship. From an affordance perspective [5], the interface not only transmits information but shapes how human and AI agencies interact. Figure 1 illustrates how human agency is expressed through system users, while technological agency is divided between the mediating interface layer where affordances are perceived and enacted as well as the infrastructural compound LLM system which provides the pre-structured space for action and determines which components exist, their interactions with one another and in turn defines which tasks are solvable. In this work we adopt the term compound LLM system [6–8] to denote an architecture in which LLMs are embedded within and orchestrate additional subsystems (e.g. retrieval and data access, tools via APIs, external ML models, multi-agent architectures).

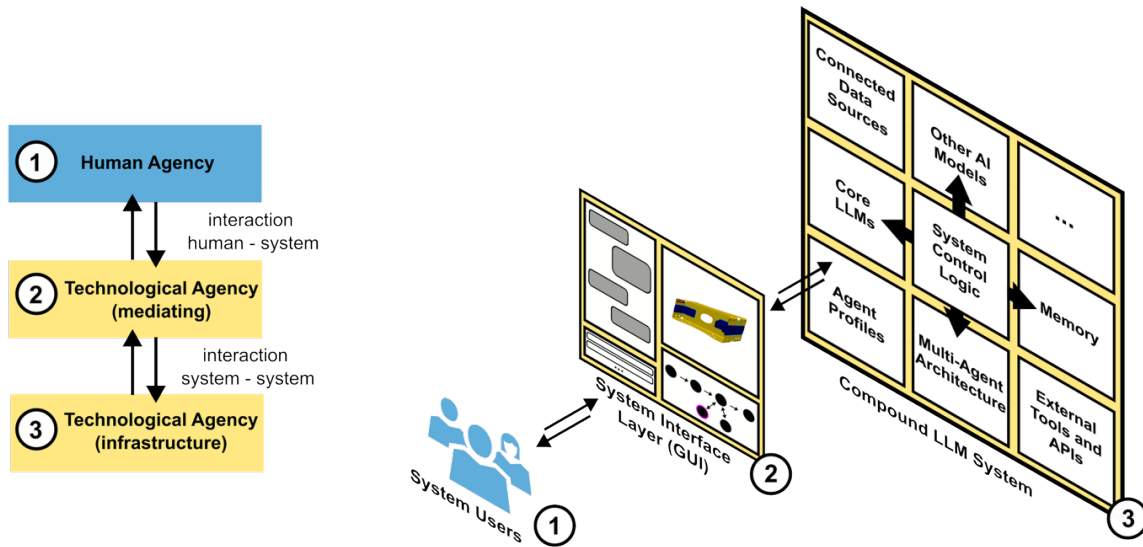


Figure 1: Human and technological agency across system users, system interface and compound LLM system layers

Interaction occurs at multiple levels: humans and the interface (frontend), where goals are translated into system-readable actions as well as between the interface and the compound LLM system (backend), where actions are operationalized through computational exchanges within the system infrastructure. The LLM’s generative and orchestrative capabilities are leveraged to make use of other components, yielding a cohesive multi-part system that can plan, call tools, interpret results and iterate toward goals in ways standalone LLMs cannot.

The combination of deterministic and non-deterministic methods in a system creates new operational challenges. Unlike conventional automation, where logic chains are explicit and traceable, LLM-based reasoning is stochastic and context-dependent. Users must therefore rely on interface-level transparency mechanisms to understand how conclusions are reached and which data, tool calls, or prompts contributed to them. Without such visibility, validation of results, essential in regulated engineering domains, becomes increasingly difficult. Motivated by this fact we pursue two complementary research aims, embodied by the following two research questions:

RQ1: How can compound LLM-based automation systems be designed to support complex engineering workflows while maintaining transparency of their decision-making processes to users?

RQ2: How can hybrid intelligence systems be validated and systematically tested for transparency once developed, and what criteria determine whether such systems meet transparency and efficiency requirements?

2 Method

We adopt a two-stage research design aligned with the above research questions connecting a designed artefact in a real-world context to a knowledge base of existing theories and models to develop an evaluation concept. The first stage (RQ1) covered in section 3 focuses on the development of a compound LLM system with an industrial case study serving as an example, while the second stage (RQ2) covered in section 4 builds upon this and carries out a theoretical analysis based on established literature to determine necessary evaluation dimensions. Together, these stages address both the practical and assessment-oriented dimension of transparency in hybrid intelligence systems. These findings are used in section 5 to conceptualize an evaluation approach for validating usefulness and assessing transparency in such systems.

2.1 Method for RQ1 – Compound LLM System Development

Prototype development was conducted in collaboration with an industrial partner whereby the case company specializes in the design of complex products and employs extensive finite element analysis workflows for structural and dynamic evaluation of automotive components. These processes involve partly repetitive post-processing tasks as well as tasks which require engineering judgment and result interpretation. The compound LLM system is based on a multi-agent architecture in which individual agents combine deterministic computation (through pre-programmed simulation tools) and probabilistic reasoning (through the LLMs themselves) to partially automate workflows with a human-in-the-loop. System development followed an iterative action research cycle of design, deployment, observation, and reflection [9]. Each iteration was carried out jointly by researchers and simulation engineers, ensuring the prototype evolved under real operational conditions. After gathering requirements from stakeholders and translating these into design decisions to develop the prototype, observations from live usage were complemented by semi-structured interviews and questionnaires to identify points where the users found the system’s reasoning unclear or difficult to follow. Subsequently existing features were tweaked, removed or new features added.

2.2 Method for RQ2 – Evaluation Dimensions of Compound LLM Systems

The development of the validation concept is based on a literature review focusing on determinants for validating expert systems. In addition, metrics were sought that describe efficiency both for method use in general and for the use of expert systems in particular. Since the compound LLM system presented here integrates non-deterministic elements into an expert-system-like architecture, a further literature search was conducted to identify determinants by which transparency of AI methods in development contexts can be described and measured.

3 Compound LLM System Development – Industry Case Study

3.1 Motivation and Context

The compound LLM automation system targets the post-processing stage of simulation-driven product development in which engineers analyse large sets of result artefacts such as Ansys .rst and .cdb files to extract, compare, and communicate insights including stress distributions, eigenfrequencies, material mappings and displacements (Figure 2a). These tasks, typically performed through a mix of scripts and manual GUI operations in specialized engineering software, are repetitive, time-consuming, and prone to inconsistency across projects. The developed compound LLM system interface acts as a layer between the engineer on one side and simulation data, material databases, other AI models and computational tools on the other (Figure 2b). It enables engineers to interact with simulation results via natural language queries and integrated validated Ansys functions and thereby provides personalized analytical assistance with higher automation as well as knowledge management benefits.

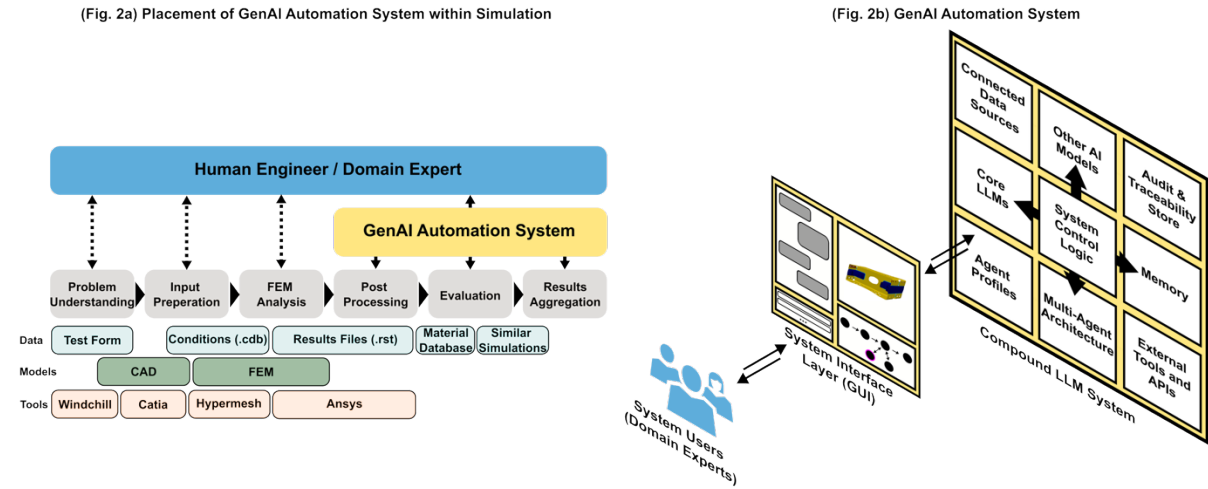


Figure 2: Placement of compound LLM system within the engineering simulation process

3.2 System Architecture

The implemented system follows a multi-agent architecture with three strategic reasoning agents and several specialized tool agents. We use a mixed methods approach combining both deterministic and non-deterministic elements depending on the type of task. A *Planner Agent* interprets user queries, decomposes them into subtasks, and generates an execution plan. An *Information Agent* validates files, identifies relevant components, and ensures consistency between user terms and database identifiers. A *Manager Agent* coordinates workflow execution, error handling, and user communication. Specialized *Tool Agents* perform domain-specific analyses such as Stress, Modal, Material, and Mesh. Agents are orchestrated using LangGraph [10], enabling graph-based workflow coordination. Claude Sonnet 4 [11] is used as the foundation model across the board via the AWS Bedrock API while custom tools interacting with simulation files are implemented in python and utilize Ansys PyDPF [12]. The system interface layer (frontend) is built as a react application hosted in a custom streamlit component.

3.3 Workflow and Interaction

In a typical session, a user issues a natural-language query such as “Compare stress distribution between chassis part X in Simulation Y and Z and determine any critical load cases.” The *Planner Agent* decomposes this request into subtasks (e.g. file and part validation, load case mapping, stress extraction). The *Information Agent* confirms file integrity and component mappings. The *Manager Agent* orchestrates the execution, invoking relevant tool agents sequentially. Specialized domain agents perform their tasks through validated DPF functions. Results can be integrated into composite reports and visualizations, accompanied by traceability metadata.

3.4 Transparency-Oriented System Design

Transparency is achieved by linking outputs to prompts, data sources, and model versions for traceability. The system combines LLM-based reasoning for planning and explanation, deterministic tool execution for computations, as well as other AI models such as custom fastText embedding models [13] for resolving domain-specific terminology and component identifiers. The core challenge lies in coupling these heterogeneous reasoning and computation layers into a coherent architecture that automates tasks while ensuring that each decision, data source, and analytical pathway remains transparent. The interface shown in Figure 3 presents the conceptual intermediary layer between engineer and system, offering an interactive overview of actions, intermediate results, and representations of decision pathways.

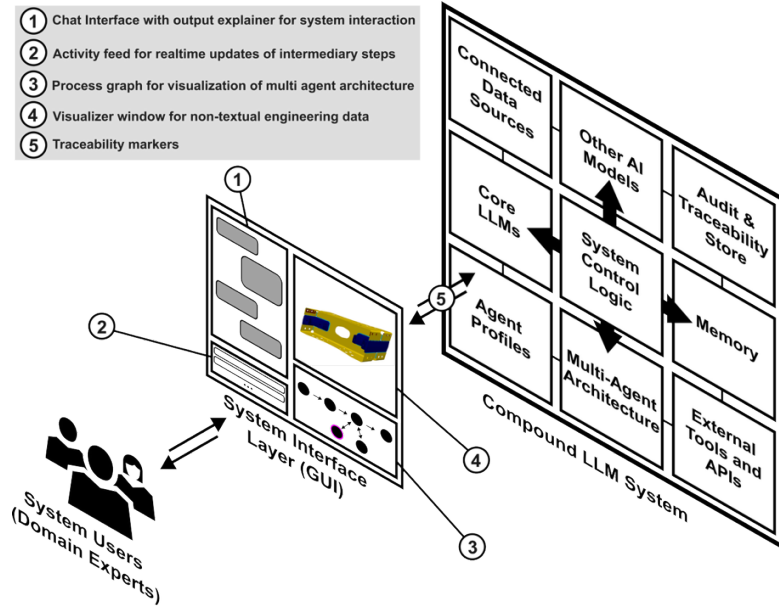


Figure 3: Conceptual interface layer for the proposed compound LLM system

Resulting from iterative development combined with user feedback, the following five categories of interface mechanisms are embedded to support transparency and numbered accordingly in Figure 4:

- **(1) Chat with output explainer:** main chat interface for the user to pose queries (1a) whereby each system output is accompanied by an expandable section (1b) showing tool calls with input parameters as well as any errors that arose for that specific system output.
- **(2) Activity feed:** chronological log of system actions and intermediate outputs, displaying validation steps such as file checks, load-case mappings, or material identification.
- **(3) Process graph:** real-time visual representation of agent coordination, illustrating dependencies and the flow of information among LLM components.
- **(4) Visualizer Window:** interactive representation of non-textual engineering data specific to this use case such as stress and material distributions or modal animations.
- **(5) Traceability markers:** hashes linking metadata annotations and results to their prompts, data sources, model versions, and timestamps, enabling reproducibility and auditability (not presented to users).

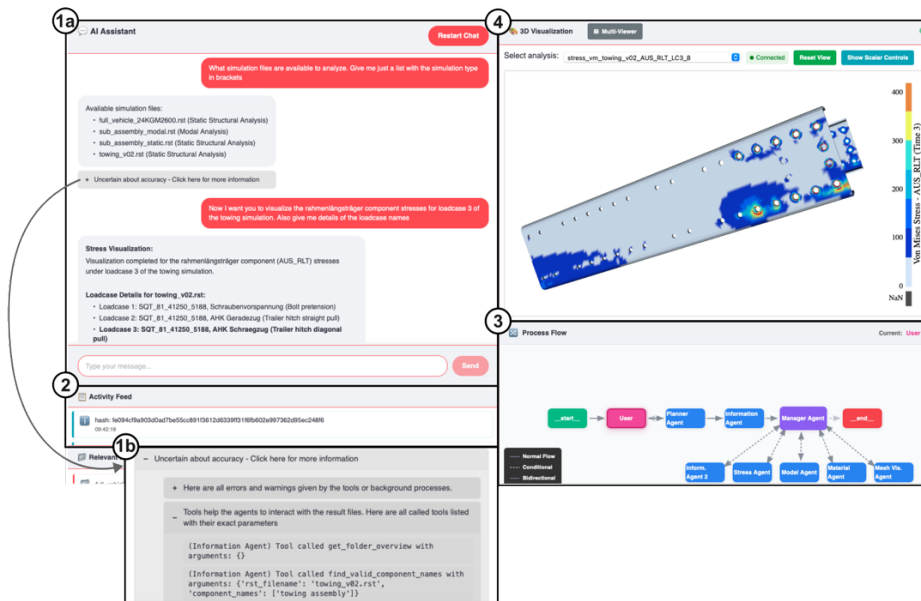


Figure 4: Example of prototype interface layer including annotation of embedded transparency mechanisms 1-4

4 Evaluation Dimensions of Compound LLM Systems

The compound LLM system described above can be interpreted as a method that supports designers in specific activities, rather than solely as a software artefact. Following Kroll and Weisbrod [14], such methods should be evaluated with respect to their applicability, their ability to convey understanding and support process compliance, and their effectiveness for the targeted development task. The specific criteria depend on the task to be supported and on the intended purpose of the method. Further literature also highlights several factors that can negatively affect such a method's performance or weaken its potential. These include incomplete information and inconsistencies between method elements [15].

To be useful, such methods need to satisfy four basic properties [16]: they must be defined (logically complete and coherently described, with a clear scope of application), predictable (internally understandable and behaviourally foreseeable), usable (accessible, easy to apply, and credible), and desirable (producing results that are appropriate and valuable for the task at hand).

Traditional expert systems exemplify such methods in the context of complex problem solving that requires substantial domain expertise. Knowledge is encoded via rules and facts from which conclusions are drawn [17]. However cognitive work that was previously reserved for human experts is increasingly taken over by information systems [18] which leads directly to the automation dilemma discussed in the introduction. This makes clear that validation of expert systems must go beyond classical validation. A holistic perspective on verification and validation (V&V) is required.

With the introduction of AI methods, and particularly LLMs into expert systems, an additional dimension emerges. Alongside deterministic mechanisms, non-deterministic, probabilistic methods are now used for data analysis, planning, and task completion. Automated conclusions are based on statistical relationships and are therefore associated with uncertainty. This can lead to misinterpretation or misplaced trust if users are not aware of the limits of the models or the quality of underlying data. Consequently, V&V of AI based expert systems must also incorporate criteria for transparency.

From an affordance perspective, the GUI is the layer where technological agency becomes perceptible and actionable for users; it is here that system outputs are exposed in a form that engineers can interact with. To evaluate and contextualize results, users must be able to build an appropriate mental model of how the system approaches a task [18], and that mental model is shaped by what the interface makes visible. At the same time, the interface due to being a tool for the human-in-the-loop to interact with the compound LLM system is also part of a broader socio-technical system [19].

Design decisions about what to surface, at what level of detail, and with which interaction patterns all influence the workload, required expertise, and the balance between usability and flexibility of the system. Highly automated interfaces that hide too much may encourage over-reliance and deskilling, while excessively detailed ones can overwhelm users and erode efficiency [2]. These trade-offs are directly linked to transparency: they determine whether engineers can meaningfully oversee and challenge AI decisions. Consequently, our evaluation concept treats transparency as something that is realized through the interface but evaluated regarding its effects on both human understanding and overall system performance.

4.1 Verification, Validation and Transparency

Synthesising the findings from literature as well as practical experience from the compound LLM case study, three related but distinct evaluation dimensions which build upon one another in multiple levels can be distinguished:

1. **Verification** (Does the system do things correctly?) – Focuses on technical correctness, robustness, and internal consistency across all deterministic and probabilistic components.
2. **Validation** (Does the system do the correct things?) – Focuses on the benefits for engineers and fit to the intended task, including efficiency gains and quality of decision support.
3. **Transparency** – (Is it understandable how and why the system does things?) – Focuses on interpretability, explainability, and traceability of the system’s reasoning and actions from the user’s perspective.

These dimensions form a hierarchical foundation for evaluating compound LLM systems: verification is the base layer, validation builds on verified functionality to assess usefulness, and transparency sits on top as a property that determines whether engineers can appropriately assess the system’s behaviour. In compound LLM systems, each of these dimensions must be addressed separately for deterministic and non-deterministic methods.

The starting point for evaluation is **verification** of the compound LLM system whereby the focus lies on the technical system. In a first step, it must be ensured that the compound LLM system produces correct results. Classical software tests can be used for deterministic elements. In order to identify and assess uncertainties or fluctuations in the results caused by probabilistic elements, an additional, more detailed analysis is required to understand how the non-deterministic components affect the overall result. This includes examining reproducibility of results using for example human or LLM-as-a-judge techniques using a sufficiently large, unbiased, diverse test dataset covering anticipated tasks. Once the system is deployed monitoring through constant evals is advisable. Furthermore, the general capability of the chosen or trained AI model for its tasks needs to be determined for which processes such as DASC [20] or numerous domain specific existing benchmarks [21–23] can be used as guidance if available.

Within the scope of **validation**, the focus extends beyond the technical system to include the users of the compound LLM system. The socio-technical system becomes the object of investigation. The primary concern is the usefulness. Validation examines whether the system’s results address the engineers’ queries and whether they provide added value for decision-making. For example: is the functionality of the deterministic tools sufficiently useful for the tasks? And on the non-deterministic side: are the tools used correctly and do the non-deterministic elements provide additional value?

To ensure acceptance and valid decisions when using the compound LLM system, it is necessary that **transparency** is evaluated to make sure that users can reconstruct the process by which results are generated. In deterministic expert systems, acceptance and trust in the result are often based on existing knowledge and experience of how similar tasks are usually solved. This knowledge is encoded as rules in the expert system and at the same time forms the basis for the mental models engineers build up [18]. It can be assumed that a similar mechanism also apply when non-deterministic elements are used within a compound LLM system. Regardless of this, users must be given opportunities to confirm and further develop their mental models. The GUI of the compound LLM system is considered an essential element for this. To assess transparency, special attention must therefore be paid to the GUI and to whether the information that users expect is provided in a way that helps them complete and adjust their mental model.

The individual aspects of verification, validation, and transparency, as described above, are shown as a model in Figure 5. Crucial for productive deployment of a compound LLM system in an organisation is that it offers efficiency advantages compared to conventional approaches. Evaluation is therefore always linked to an efficiency analysis, which can be operationalised at the proposed levels and translated into concrete metrics. The right side of Figure 5 provides an overview of how efficiency is interpreted in this context and which measures can be derived from it.

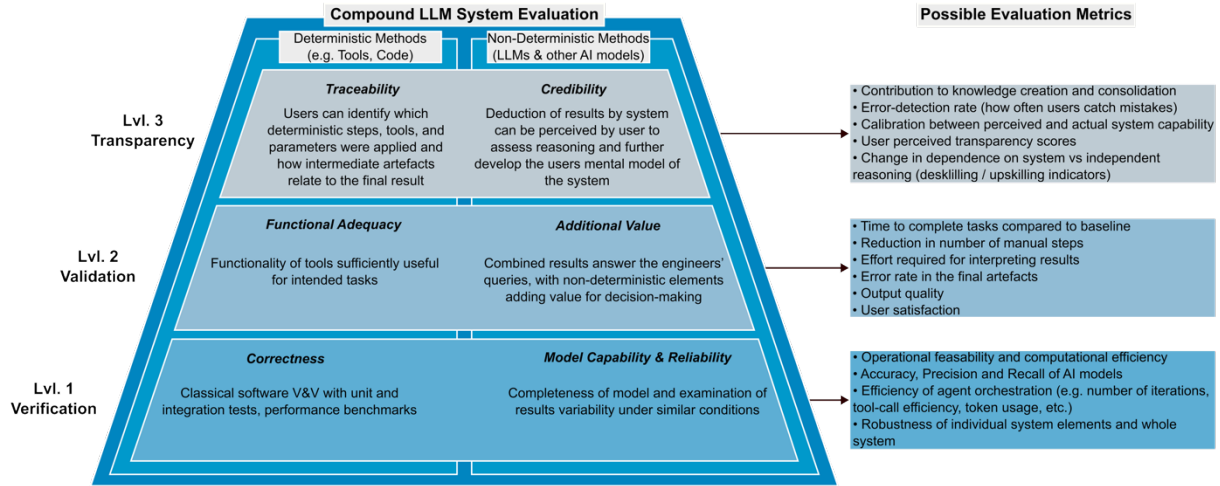


Figure 5: Verification, validation and transparency pyramid for compound LLM systems

5 Validation & Transparency Evaluation Concept for Compound LLM System

Building on the criteria and aspects that must be considered in the evaluation of validation and transparency levels as well as a framework investigating AI and human interaction [24] a concept is derived that also captures different facets of efficiency. A key premise for the following considerations is that verification (e.g. checks for correctness, completeness, and reproducibility of the technical system) has already been carried out. The evaluation concept presented here therefore focuses on the socio-technical system consisting of the users and the compound LLM system, with a particular emphasis on the transparency dimension.

The evaluation is designed to be primarily qualitative. This is partly because the available sample size is relatively small, with a maximum of around ten engineers, and partly because individual perceptions are likely to lead to considerable variability in how response options are interpreted, making a purely quantitative analysis less meaningful. Although a sample size of ten participants is still on the low side for a qualitative study, it is deemed appropriate here because a comparatively homogeneous target group is being studied within a clearly defined context.

At the core of the evaluation concept is an experiment in which participants are given a typical development task to solve. The sample is split: one third of the participants solve the task using the conventionally used simulation workbench, providing baseline values for comparison with the compound LLM system. The remaining participants are given access to the compound LLM system for task completion. Beforehand, they receive a brief training session on the system's functionality and use while the design of the training aims to minimize the possibilities of priming.

During task execution, participants are observed by the experimenter. In addition, they are encouraged to verbalize their thoughts following the think-aloud method. The task execution and think-aloud protocols are recorded for later anonymized analysis. The purpose of this approach is to capture and understand how participants use the tools, with particular attention to how they engage with the non-deterministic elements, how they handle the implications of these elements for judging solutions, and whether and how they recognize uncertainties. From this, it should be possible to infer the extent to which the information provided via the interface influences efficiency in task completion and supports or hinders the formation of accurate mental models. This analysis also allows estimation of how much effort and time are required for information retrieval and task completion in each condition.

To realize an efficiency assessment at the transparency level, it is necessary to evaluate how the information provided by the compound LLM system is processed by the engineers and how it

contributes to building experience and competence. This is done using questionnaires that participants complete after the experiment. A pre-experiment questionnaire collects demographic data (experience with the task, tenure in the company, age, gender, etc.) and captures expected difficulties and challenges related to task completion, both in terms of content and organization. A five-point Likert scale is used for the formulated questions.

The concluding questionnaire is intended to provide insights into knowledge building and perceived transparency. Since this is closely tied to individual and subjective factors, it is difficult to measure directly. It must therefore be captured indirectly by correlating responses with observed patterns of information use and processing. Accordingly, the questionnaire includes items that address the way information is used and interpreted, and how uncertainties or ambiguities are perceived and handled. Participants are also asked to rate how well the tool supported them in different aspects of the task and how the effort for information retrieval compares to their previous approach. Global questions on the perceived transparency of the compound LLM system complete the questionnaire.

Taken together, this mixed qualitative design combining think-aloud protocols, observational data, and targeted questionnaires aims to capture the multidimensional construct of transparency and its relationship to efficiency in the use of the compound LLM system.

6 Conclusion & Outlook

The increasing integration of automation into engineering workflows has historically focused on efficiency, reducing manual effort through deterministic algorithms and rule-based configurations. While this ensures reproducibility, it also constrains adaptability and contextual awareness: deterministic systems operate within predefined boundaries and scale poorly to more ambiguous multi-step processes. In hybrid intelligence settings, the interface between the users and the system therefore no longer just relays commands but must act as an affordant mechanism that enables the visualisation, constraining, and communication of system reasoning. Its design determines whether users can trace AI generated outputs, interpret decision pathways, and retain meaningful authority over system actions, thereby making transparency-oriented interaction design central to trust and accountability in human-AI collaboration.

This paper showed how transparency can be designed and evaluated in compound LLM systems that blend deterministic tools with probabilistic reasoning. Addressing RQ1, an industrial case study of a multi-agent LLM assistant for simulation post-processing demonstrated how an affordant interface can surface planning steps, tool calls, data sources, and intermediate results. The transparency mechanisms operationalise Leonardi's affordance perspective and the triangle of design responsibility by making the system's technological agency inspectable and governable for engineers. Addressing RQ2, we derived three interconnected evaluation dimensions: verification, validation, and transparency, for both deterministic and non-deterministic components, and proposed a transparency evaluation concept combining experimental comparison, think-aloud protocols, observation, and targeted questionnaires.

Although developed for a single engineering context and not yet tested at scale, the work offers an initial, structured way to treat transparency as a distinct quality dimension rather than a by-product of accuracy and usability. As compound LLM systems spread across domains increasingly enabled by tool integration protocols like the Model Context Protocol [25], a mixed methods character of automation will become standard, and scalable systematic transparency evaluation will be essential. Future work will extend this perspective into a development framework spanning technical foundations, interface design, and organisational embedding supporting practitioners in implementing effective hybrid intelligence systems.

References

- [1] Cederfeldt M, Elgh F. Design Automation in SMEs—Current State, Potential, Need and Requirements. DS 35: Proceedings ICED 05, Melbourne: 2005.
- [2] Verhagen WJC, Bermell-Garcia P, van Dijk REC, et al. A critical review of Knowledge-Based Engineering: An identification of research challenges. *Advanced Engineering Informatics* 2012;26:5–15. <https://doi.org/10.1016/j.aei.2011.06.004>.
- [3] Vidner O, Wehlin C, Wiberg A. Design Automation Systems for the Product Development Process: Reflections from Five Industrial Case Studies. *Proc Des Soc* 2022;2:2533–42. <https://doi.org/10.1017/pds.2022.256>.
- [4] Pradas Gómez A, Kretzschmar M, Paetzold-Byhain K, et al. A team of three: the role of generative AI in the development of design automation systems for complex products. *Proc Des Soc* 2025;5:309–18. <https://doi.org/10.1017/pds.2025.10045>.
- [5] Leonardi PM. When Flexible Routines Meet Flexible Technologies: Affordance, Constraint, and the Imbrication of Human and Material Agencies. *MIS Quarterly* 2011;35:147–67. <https://doi.org/10.2307/23043493>.
- [6] Chen J, Ye J, Wang G. From Standalone LLMs to Integrated Intelligence: A Survey of Compound AI Systems. *ArXiv* 2025. <https://doi.org/10.48550/arXiv.2506.04565>.
- [7] Chen L, Davis J, Hanin B, et al. Are more LM calls all you need? towards the scaling properties of compound AI systems. *Proc 38th NeurIPS* 2024:45767–90.
- [8] IBM, Stryker C. What are compound AI Systems? <https://www.ibm.com/think/topics/compound-ai-systems> 2024. <https://www.ibm.com/think/topics/compound-ai-systems> (accessed September 3, 2025).
- [9] Hevner AR, March ST, Park J, et al. Design Science in Information Systems Research. *MIS Quarterly* 2004;28:75–105.
- [10] langchain-ai. LangGraph GitHub. <https://github.com/langchain-ai/langgraph> (accessed August 29, 2025).
- [11] Anthropic. Claude Sonnet 4 2025. <https://www.anthropic.com/news/claude-4> (accessed January 7, 2026).
- [12] Ansys Inc. Ansys Data Processing Framework (DPF) for developers 2025. <https://developer.ansys.com/docs/dpf> (accessed August 28, 2025).
- [13] Bojanowski P, Grave E, Joulin A, et al. Enriching Word Vectors with Subword Information. *ArXiv* 2017. <https://doi.org/10.48550/arXiv.1607.04606>.
- [14] Kroll E, Weisbrod G. Testing and evaluating the applicability and effectiveness of the new idea-configuration-evaluation (ICE) method of conceptual design. *Res Eng Des* 2020;31:103–22. <https://doi.org/10.1007/s00163-019-00324-6>.
- [15] Cash P, Daalhuizen J, Hekkert P. Evaluating the efficacy and effectiveness of design methods: A systematic review and assessment framework. *Des Stud* 2023;88:101204. <https://doi.org/10.1016/j.destud.2023.101204>.
- [16] Daalhuizen J, Cash P. Method content theory: Towards a new understanding of methods in design. *Des Stud* 2021;75:101018. <https://doi.org/10.1016/j.destud.2021.101018>.
- [17] Feigenbaum EA. Expert systems in the 1980s. *State of the Art Report on Machine Intelligence Maidenhead: Pergamon-Infotech*, 23 1981.
- [18] Hacker W. Menschengerechtes Arbeiten in der digitalisierten Welt. vdf Hochschulverlag an der ETH Zürich; 2018. <https://doi.org/10.3218/3938-2>.
- [19] Gericke K, Eckert C, Stacey M. Elements of a design method – a basis for describing and evaluating design methods. *Design Science* 2022;8:e29. <https://doi.org/10.1017/dsj.2022.23>.
- [20] Schulz M, Neuhaus U, Kaufmann J, et al. DASC-PM v1.1: A Process Model for Data Science Projects 2022. <https://doi.org/10.25673/91094>.
- [21] SWE-bench Team. SWE-Bench Leaderboard 2025. <https://www.swebench.com> (accessed August 28, 2025).
- [22] Wang A, Pruksachatkun Y, Nangia N, et al. SuperGLUE: A Stickier Benchmark for General-Purpose Language Understanding Systems. 2019. <https://doi.org/10.48550/arXiv.1905.00537>.
- [23] Everingham M, Van Gool L, Williams CKI, et al. The pascal visual object classes (VOC) challenge. *Int J Comput Vis* 2010;88:303–38. <https://doi.org/10.1007/s11263-009-0275-4>.
- [24] Hillebrand L, Raisch S, Schad J. Managing with Artificial Intelligence: An Integrative Framework. *Academy of Management Annals* 2025;19:343–75. <https://doi.org/10.5465/annals.2022.0072>.
- [25] Anthropic. Introducing the Model Context Protocol 2024. <https://www.anthropic.com/news/model-context-protocol> (accessed January 7, 2026).