Design Orchestrator

An Agent-Based, AI-First Framework for Multidisciplinary Design Workflows

Aditya Soman[1], Jun Wen Loo[2]
[1]Tp Bennett Architects, [2]EOC Engineers
[1]adityaasoman@gmail.com, [2]loojunwen1995@gmail.com

Abstract. *This paper presents an Agentic AI platform Design Orchestrator capable of building multi-disciplinary agentic workflows in a shared project context. The platform combines a node-based visual programming interface for constructing multi-capability agents with persistent project memory and integrations with computational design and analysis tools.This platform is evaluated through an early-stage design case study for a timber bridge limited to architectural and structural engineering disciplines. Two agents are developed on the platform one for site analysis and other for model exchange and structural analysis. The Bridge design options are iteratively transferred, validated, structurally analysed, and evaluated for structural performance, embodied carbon, and relative cost. The case study demonstrates the benefits of integrating agentic design generation, analysis, evaluation, and documentation into a continuous workflow, enabling rapid design iterations while retaining human-in-the-loop validation at critical stages. It also reveals challenges related to agent reliability, requirement of precise agent instructions, performance optimization, context and token management, reinforcing the need for human oversight and more transparent agentic workflows .*
Keywords. *AI Agents, Multi-Disciplinary design, Agentic Collaboration, Data Context portability, Visual Programming.*

INTRODUCTION

The rapid maturation of AI-first methodologies driven by advances in generative artificial intelligence and computational design is transforming contemporary architectural practice (Brynjolfsson et al., 2017). Emerging studies suggest that smaller, agile design teams adopting integrated AI workflows gain a competitive advantage as manual tasks become automated, data-rich, and interdependent. However, these advances often remain confined within disciplinary silos, limiting their impact on project delivery. Since the Architecture, Engineering, and Construction (AEC) sector is inherently collaborative, the full benefits of AI integration can only be realised through collective, cross-disciplinary adoption rather than isolated innovation. Historically, architects have occupied the role of orchestrator coordinating diverse consultants, disciplinary expertise, and project knowledge to shape the overarching design intent. This paper introduces Design Orchestrator, an agent-based framework that extends this orchestration paradigm into a design ecosystem. Within this environment, all project disciplines

operate through a shared set of AI-enabled tools and workflows, enabling integrated, cross-disciplinary collaboration and more coherent decision-making throughout the design process.

BACKGROUND

In the computational design research community, recent work has demonstrated a growing convergence between agent-based systems and large language models (LLMs). These approaches extend earlier paradigms of parametric and rule-based design by introducing adaptive, language-driven intelligence capable of mediating between design intent, data-rich models, and executable logic. In this context, AI is increasingly framed not as a standalone optimisation tool, but as an active participant in design reasoning, coordination, and decision-making.

Recent studies have explored multi-agent frameworks that distribute design responsibilities across specialised AI agents operating over shared BIM representations. Projects such as Archi-Agents (Tang et al., 2025) and Building-Agent (Zhong et al., 2024) illustrate how role-based agents can collaborate to negotiate architectural intent, spatial configuration, and performance criteria, demonstrating the potential of LLMs to support interdisciplinary design processes beyond isolated analytical tasks. Complementary research has shown how discrete spatial representations and agent-based reasoning can support generative design across multiple disciplines. For example, DeciGenArch (Soman et al., 2022) formulated architectural zoning as a multi-agent, multi-criteria decision process in voxelised space, enabling spatial configurations to emerge from collaborative agent behaviours guided by stakeholder and environmental criteria. Importantly, the integration of AI is not limited to textual outputs or coarse, approximate geometric representations. Projects such as Raven (Rietschel et al., 2024) demonstrate how natural language input can be translated directly into parametric and visual programming logic, lowering the barrier to algorithmic design and aligning computational workflows more closely with designers' cognitive modes of thinking.

This Agentic Design approach is also not limited to architecture alone. Recent work in generative structural engineering also demonstrates how task decomposition and specialised agents can reliably automate complex engineering workflows, including geometric reasoning, model validation, and code generation for finite element analysis (Geng et al., 2025). Topogen (Soman, A., 2023) extended similar generative logic into structural engineering by translating architectural models into voxelised analytical representations from which structural systems and finite-element models could be generated and evaluated automatically. This highlights the broader applicability of agent-based orchestration beyond architecture alone. Despite these advances, most systems remain discipline-specific and task specific. This paper addresses this gap by framing AI integration as an orchestration problem, proposing a shared design environment that enables cross-disciplinary collaboration throughout the design process.

PLATFORM

The Design Orchestrator platform organises AI workflows along two complementary dimensions: disciplinary domain and task domain. While the framework is intended to support the full spectrum of AEC disciplines, this paper focuses on architectural and structural engineering workflows, as these currently represent the most mature and validated implementations within the system. The platform is primarily developed as a cloud-based application, enabling a centralized and persistent project state accessible across multiple environments. It is designed to be accessed through a range of client interfaces, all of which connect to this shared project context. A browser-based client serves as the primary enviornment, providing the main interface for interaction and workflow configuration. This is complemented

by application-level extension clients embedded within existing design and analysis software, allowing practitioners to engage with the platform directly within their established tools.

A native extension has been implemented for Rhino3D, reflecting its widespread adoption in computational design practice; however, the platform architecture is designed to support similar integrations with other software. This approach enables the execution of software-specific workflows while maintaining synchronisation with the shared, cloud-based project environment. This multi-client approach prioritises augmenting the capabilities of existing design and engineering software rather than replacing them. By connecting these tools to agentic workflows, the platform enables established disciplinary practices to remain intact while being coordinated through a shared, AI-driven layer. This avoids the need to reimplement validated workflows and instead enhances them through integration and orchestration.

Interfaces

At a system level, the platform is structured around a set of modality-specific interfaces, namely conversation, content generation, configurator, and canvas each supporting distinct categories of design interaction. These range from language-driven queries and multimodal content creation to parametric modelling, BIM integration, and automated documentation. Together, these interfaces provide accessible, preconfigured workflows while maintaining a shared project context across all interactions. At the core of this system is a visual, node-based Agent Builder, which enables users to compose custom, multi-capability agents capable of orchestrating workflows across one or more interfaces. (Figure 1)

1. **Agent Builder:** The primary interface for constructing customised agents. It provides a visual, no-code environment based on node-based composition, allowing users to assemble workflows on a unified canvas. Components from multiple interfaces can be combined to support tasks ranging from simple operations to complex, multi-stage processes.
2. **Agent Garden Interface:** Supports conversational and natural language-driven workflows, with the custom or pre-configured agents on the platform.
3. **Content Generation Interface:** Enables AI-driven content production across multiple modalities, including text, imagery, video, and 3D assets. It integrates state-of-the-art generative models within a unified inference layer, allowing users and agents to generate, modify, and iterate design content within the shared project environment.
4. **Configurator Interface:** Connects agents to computational design environments and scripts, enabling the execution of parametric models, simulations, and geometry-based workflows. It facilitates interaction with tools such as Grasshopper, BIM platforms, and analysis software within the shared context (Figure 5).
5. **Canvas Interface:** Provides an infinite canvas style workspace for documentation and presentation. Users can create reports, presentation decks, and design boards, either manually or with assistance from agents. The interface supports AI-assisted composition, layout generation, and automated documentation workflows, enabling seamless translation of design processes into communicable outputs (Figure 2).

Project Creation

Project configuration establishes the shared context within which agents operate. Users initialise a project by defining core metadata which includes location, building typology, participating disciplines, and team structure, which becomes part of the persistent project memory accessible to all agents. Discipline-specific environments are then activated, providing access to preconfigured interfaces and

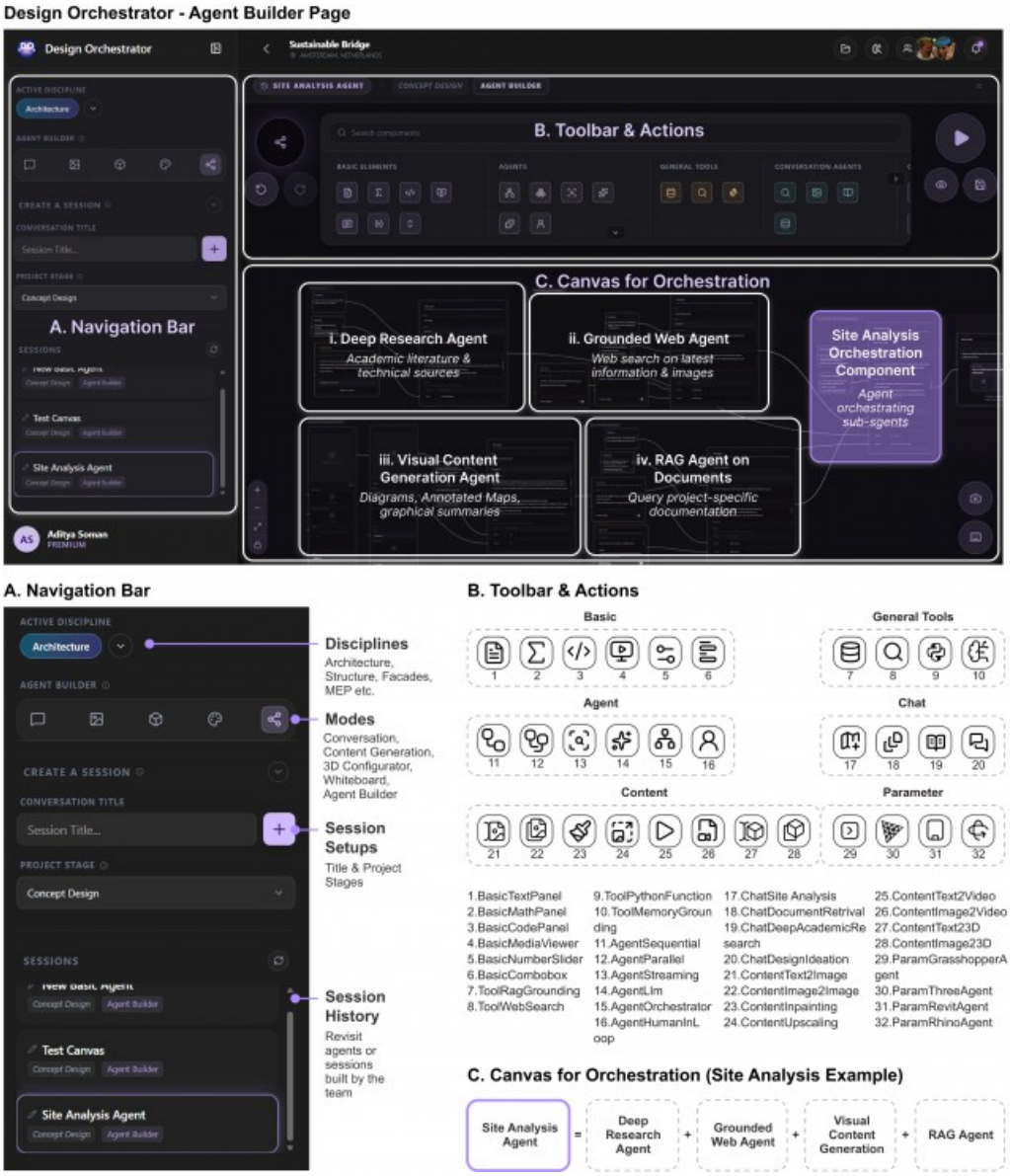
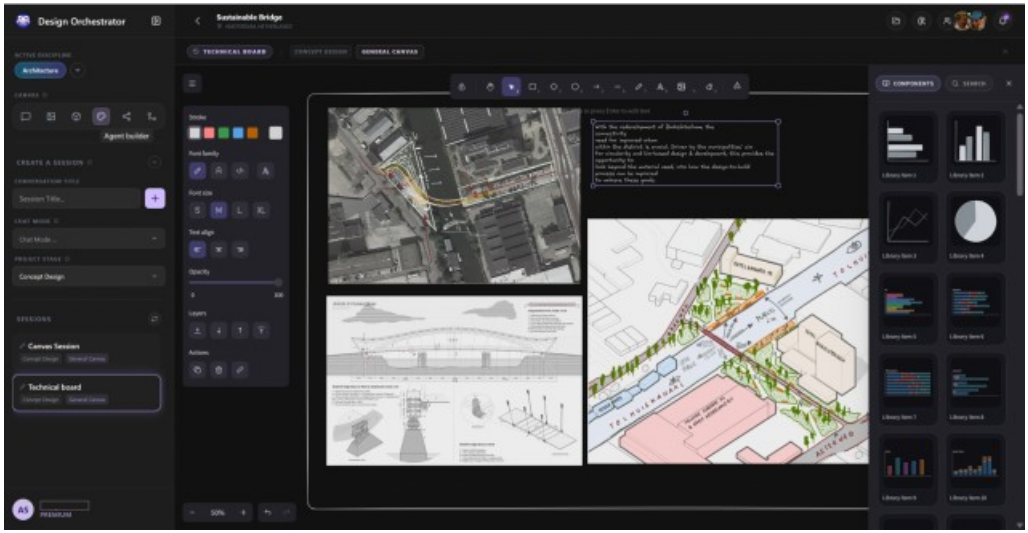


Figure 1
Design Orchestrator UI Breakdown offeatures & tools available on Agent Builder visual programming interface, including the example of Site Analysis agent configuration mentioned in the case study section.

Figure 2
Canvas interface offering an infinitecanvas to create presentation boards and documentation



agentic workflows aligned with the selected domain. These default configurations support immediate use while ensuring consistency of assumptions and data structures across collaborators.

AGENT BUILDER

The Agent Builder is the primary interface of the Design Orchestrator platform and serves as the main environment for creating customised AI agents. It is implemented as a visual, node-based programming interface, conceptually similar to computational design tools such as Grasshopper, where users assemble workflows through drag-and-drop components on an infinite canvas. This approach enables practitioners to define agent behaviours without requiring extensive programming expertise, while still supporting complex, multi-stage logic. The visual programming canvas is automatically parsed into Python code, which executes the configured agent within the platform. The system leverages the Google Agent SDK as the underlying agentic framework used to instantiate, configure, and run agents. The sandboxing, authentication, scaling ,deployment and other cloud infrastructure configuration is also bundled behind the visual scripting components.

The interface is organised around a structured palette of components (Figure 1), broadly categorised into basic elements, agent types, and tools. Basic elements provide fundamental input, output, and data manipulation capabilities, including text panels, mathematical operators, code panels for Python scripting, media viewers for visual outputs, and interactive controls such as number sliders and dropdown selectors. These elements enable users to parameterise workflows and manage intermediate data within the agent pipeline. Agent components define the core computational logic of the system. A general-purpose large language model (LLM) agent allows users to specify model selection, instructions, and tool access. In addition, the platform provides specialised agent types that support different execution logics, including sequential agents for stepwise task execution, parallel agents for concurrent processing, and orchestration agents that coordinate multiple

sub-agents. A human-in-the-loop component can be incorporated at any stage, enabling user validation and intervention within automated workflows.

The Agent Builder further integrates a suite of tools that extend agent capabilities. These include retrieval-augmented generation (RAG) mechanisms for querying project-specific document databases, Web-based grounding through search and crawling, and custom Python functions. Additionally, pre-configured agents from other platform interfaces such as research, site analysis, and design intent agents (conversation interface), generative media agents (content generation interface), and computational design agents (configurator interface, e.g., Grasshopper agents) can be embedded and recomposed within custom workflows. To construct a functional workflow, users must include at least one agent component within the canvas. Once assembled, the visual graph is executed through a "play" operation, which translates the node-based configuration into executable code and provides real-time previews of intermediate outputs. Upon validation, workflows can be deployed as packaged agents to the cloud infrastructure, making them accessible across all platform interfaces and client environments. The deployed agents can be accessed through a hybrid conversation interface capable of producing ouputs from all modalities in the chat responses.

DATA CONTEXT PORTABILITY

Data context portability is one of the most significant advantages of working with the Design Orchestrator. The system preserves design knowledge continuously, ensuring that context is retained not only within a single discipline but also across multiple disciplines throughout the project lifecycle.

The memory bank service strokes both short term and long term memories across the user interactions. The tool at the moment works with sessions of interaction. Each session can be tagged as per the stage of the project. The RIBA plan of work *(RIBA, 2020)* is used as the primary source to map out the key project details required at each stage. The short term memory stores the whole conversation and all the generated content. Then with certain callback functions based on the stage of the session certain relevant factors in the conversations are saved in the memory as per the design stage and discipline. The agents in subsequent interactions can answer by grounding their responses with memories. New users in the project also have the memory already present in their interactions with the system.

This portability functions both within and across disciplines. Within a discipline such as architecture, the system maintains continuity between tasks including briefing, massing studies, façade exploration, and detailing, allowing each stage to build directly on earlier decisions. Across disciplines architecture, structural engineering, MEP, sustainability, and interiors agents remain aware of one another's activities through a shared, vectorized memory database. Knowledge produced in one domain becomes accessible to others, enabling agents to retrieve relevant context and support coordinated decision-making rather than isolated workflows.

Design Orchestrator also provides transparency and user control over this shared memory. Designers have full access to the memory bank databse and can write custom functions to tailor the stored information. This prevents unnecessary or sensitive data from propagating while still enabling meaningful collaboration. As a result, context portability transforms the design process into a shared multidisciplinary intelligence, where agents operate with a coherent understanding of the evolving project.

CASE STUDY

To showcase the application of Design orchestrator on a project we selected a bridge design case study in early design stages where inter-disciplinary collaboration between architects and structural

engineers is key in driving the design. The 38m span bridge features a tied arch bridge with catenary glulam arches topped with an intricate timber shell structure supporting an ETFE canopy, based in the north of Amsterdam (Figure 3). Given the complexity and early development stage of the design, close collaboration between architect and structural engineer across site analysis, design model exchange and documentations is critical. Two primary agents were developed for this case study a Site Analysis Agent and a Model Exchange & Structural Analysis Agent. Both are implemented as instances of an orchestration agent capable of orchestrating multiple specialised sub-agents.



Figure 3
Architectural rendering of the sustainable timber bridge designed for the Case Study

Agent 1: Site Analysis

The Site Analysis Agent incorporated four key LLM based sub-agents (Figure 1). The first was a deep research agent, capable of retrieving and synthesising information from academic literature and technical sources. The second was a grounded web agent, combining Google Search and web crawling capabilities to collect up-to-date information and image references from online sources. The third was a retrieval-augmented generation (RAG) agent, which enabled querying of project-specific documents, including uploaded standards and guidelines. The fourth was a content generation agent, which utilised text-to-image and image-to-image models (Google Nano Banana) to produce visual outputs such as diagrams, annotated maps, and graphical summaries of research findings. Two variants of the Site Analysis Agent were configured: one tailored to architectural workflows and the other to structural engineering. Both agents operated within a shared project context, ensuring consistent access to core project metadata such as location, typology, and participating disciplines. This shared context enabled cross-disciplinary awareness while allowing each agent to specialise in domain-specific tasks. The architectural variant focused on qualitative and contextual aspects of the site involving cultural, environmental, historical context, and identify architectural precedents & innovations of timber bridge design within Amsterdam. In contrast, the structural variant addressed technical and performance-driven considerations such as material properties, relevant clauses from Eurocode standards (European Committee for Standardization, 2004) and IStructE guidelines (The Institution of Structural Engineers, 2020) as well as material procurement and construction sequences. These combined generates a comprehensive site analysis encompassing both technical and non-technical aspects within a single integrated environment, eliminating the need to switch between external tools or platforms.

Agent2: Model Exchange & Structural Analysis

To enable rapid evaluation of design iterations based on structural performance, cost, and embodied carbon, a Model Exchange and Structural Analysis Agent was implemented as an orchestration agent, supporting coordinated, multidisciplinary workflows within a unified digital environment. This agent operated in conjunction with the Architectural Design Agent, representing the structural counterpart within the broader dual-agent framework. The Architectural Design Agent generated shell geometries as design variants, informed by spatial, material, and contextual considerations, which were then transmitted to the Structural Engineering Design Agent for

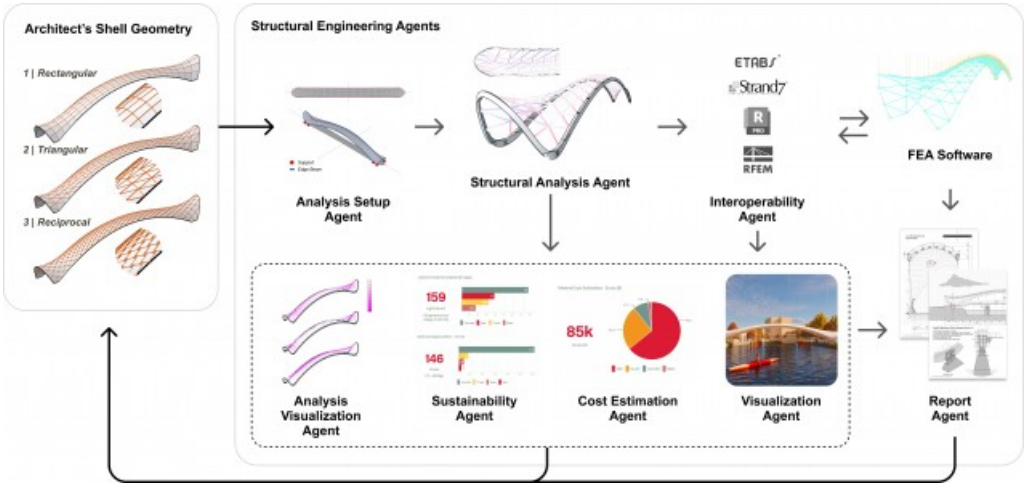


Figure 4
Model exchange and Structural analysis agentic workflow diagram.

validation and analysis. The structural agent coordinated a series of specialised sub-agents (Figure 4). An Analysis Setup Agent first validated and rationalised incoming geometry, ensuring topological consistency, connectivity, and correct definition of support conditions. Verified models were then processed by a Structural Analysis Agent, which executed finite element (FE) simulations within the Rhino/Grasshopper environment using a preconfigured Karamba workflow. An Interoperability Agent enabled validation and transfer of analytical models to external engineering tools. Following analysis, a set of parallel agents processed the results. An Analysis Visualisation Agent generated graphical representations of structural performance, while a Sustainability Agent estimated embodied carbon based on material quantities and project context. In parallel, a Cost Estimation Agent evaluated relative construction costs using predefined datasets, and a Visualisation Agent produced rapid renderings incorporating structurally informed geometry. These outputs were synthesised by a Report Generation Agent into structured technical documentation within the platform's canvas interface, providing integrated feedback on structural integrity, environmental impact, and cost.

This workflow operated as an iterative loop, where each design iteration generated by the Architectural Design Agent triggered a new cycle of structural evaluation. Through this bidirectional Agent-To-Agent interaction, the architectural and structural variants remained continuously aligned, enabling informed, performance-driven decision-making during early-stage design.

LIMITATIONS

The platform showed promising results but also

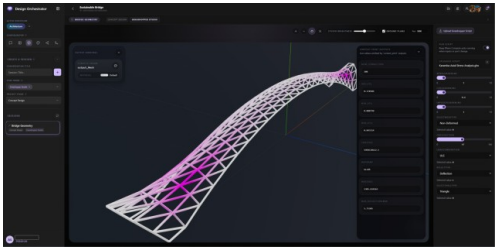


Figure 5
The configurator Interface showing the Karamba structural analysis script results.

revealed certain limitations. The effectiveness of agent-based workflows remained closely tied to the precision of user-defined instructions, requiring ongoing refinement to achieve an appropriate balance between response speed and depth of reasoning. In practice, configuring agents and defining tool selection logic could be complex, often necessitating iterative adjustments and a degree of prior understanding of model capabilities, which may limit accessibility for some users.

Additionally, agent behaviour is not always fully predictable, particularly in multi-step workflows, and can require manual intervention to ensure consistent outcomes. From a technical standpoint, interoperability remains an area for further development; the current implementation primarily supports a Rhino-based connector, while integrations with BIM and analytical software are still in progress. Token limitations can also constrain long-context reasoning and deep research tasks, particularly in complex, multidisciplinary scenarios. Furthermore, while grounding mechanisms are incorporated, the possibility of hallucination in LLM-generated outputs persists, highlighting the continued importance of human-in-the-loop validation to ensure accuracy and reliability within professional design workflows.

CONCLUSIONS

This study demonstrates the potential of an agent-based platform to support multidisciplinary design within a unified environment. The proposed framework significantly reduces the need to switch between disparate tools, enabling a more streamlined and efficient design process. A key outcome of the case study is the substantial improvement in iteration speed. Design alternatives could be generated, analysed, and evaluated in rapid cycles through agent-to-agent communication, allowing teams to focus more on interpreting results and making informed decisions rather than performing time-intensive procedural tasks.

Despite this automation, critical stages such as structural validation, interpretation of analysis outputs, and decision-making remained human-driven, ensuring reliability and maintaining professional oversight. The integration of human-in-the-loop verification also introduced transparency, as outputs exchanged between agents were consistently reviewed before progressing across disciplinary boundaries. The use of a shared project context further enhanced collaboration. By maintaining a persistent, centralised memory of project data including site analysis, design iterations, analysis scripts, and standards the platform enabled continuity across workflows and disciplines. This approach reduces information fragmentation and supports onboarding, allowing new participants to quickly understand the project state through agent-assisted queries and summaries. In this sense, the framework contributes to breaking down traditional disciplinary silos, fostering a more integrated and data-driven design process. Additionally, interoperability mechanisms allowed outputs to be transferred to established analysis software, ensuring compatibility with industry-standard verification processes. This bridging of AI-driven workflows with conventional tools highlights the practical applicability of the approach within existing professional contexts.

FUTURE WORKS

Future development will address several limitations identified in this study. First, expanding interoperability with widely used design and analysis tools remains critical. The current integrations are still at a prototype stage and rely on custom workflows. A more robust, API-driven ecosystem would allow agents to interact more seamlessly with industry-standard platforms.The framework would also benefit from improved agent autonomy and reasoning capabilities. At present, most agent actions are constrained by predefined scripts and workflows. Moving towards more adaptive, context-aware agents that can handle ambiguity and

incomplete information will be necessary for the platform to progress beyond assistive automation

Another important area for development is the implementation of explainable AI principles (Gaddam, R. R., 2024). In grounded agent interactions, for example, drops in confidence below a defined threshold could be reflected in the tone or framing of the response. For sequential agents, exposing the sequence of actions and reasoning steps within the user interface would make it easier to trace how responses are generated. These reasoning cycles could also help users debug and refine agent instructions when building agents in the Agent Builder. Finally, further research is required to evaluate the framework across a wider range of project types and disciplines. In its current form, Design Orchestrator represents a promising but still evolving approach. It highlights both the potential and the unresolved challenges of agent-based design systems in architecture and engineering.

REFERENCES

Brynjolfsson, E., Rock, D., & Syverson, C. (2017). *Artificial intelligence and the modern productivity paradox: A clash of expectations and statistics* (No. 24001). https://doi.org/10.3386/w24001

European Committee for Standardization. (2004). *Eurocode 5: Design of timber structures – Part 1-1: General – Common rules and rules for buildings*.

Gaddam, R. R. (2024). Vertex AI Agent Builder for Regulated Environments. *American International Journal of Computer Science and Technology*, *6*(2), 50–62. https://doi.org/10.63282/3117-5481/AIJCST-V6I2P106

Geng, Z., Liu, J., Cao, R., Cheng, L., Wang, H., & Cheng, M. (2025). *A Lightweight Large Language Model-Based Multi-Agent System for 2D Frame Structural Analysis*. http://arxiv.org/abs/2510.05414

RIBA. (2020). *RIBA Plan of Work 2020 Overview RIBA Plan of Work*. Royal Insitute of British Architects. www.ribaplanofwork.com

Rietschel, M., Guo, F., & Steinfeld, K. (2024). *Mediating Modes of Thought: Large Language Models for Design Scripting*. University of California, Berkeley. https://doi.org/10.48550/arXiv.2411.14485

Soman, A. (2023). TOPOGEN: A generative design methodology for Building engineering. *41st Conference on Education and Research in Computer Aided Architectural Design in Europe*, 519–528. https://doi.org/10.52842/conf.ecaade.2023.2.519

Soman, A., Nourian, P., & Azadi, S. (2022). DeciGenArch - A generative design methodology for participatory architectural configuration via multi-criteria decision analysis. *40th Conference on Education and Research in Computer Aided Architectural Design in Europe*, 459–468. https://doi.org/10.52842/conf.ecaade.2022.1.459

Tang, C., Tam, K.-M. M., & Herr, C. M. (2025). Archi-Agents Approximating the architectural design process with collaborative specialized multi-agent system llm and building information modeling. *30th Conference on Computer-Aided Architectural Design Research in Asia*, 141–150. https://doi.org/10.52842/conf.caadria.2025.3.141

The Institution of Structural Engineers. (2020). *Manual for the design of timber building structures to Eurocode 5*. The Institution of Structural Engineers.

Zhong, X., Liang, J., & Li, Y. (2024). Building-Agent A 3D generation agent framework integrating large language models and graph-based 3D generation model. *42nd Conference on Education and Research in Computer Aided Architectural Design in Europe*, 291–300. https://doi.org/10.52842/conf.ecaade.2024.2.291