Duale Hochschule
Baden-Württemberg
Ravensburg

Multi-Agent Architecture Design

Workflow Automation in Compliance with Operational Excellence

Bachelor's Thesis

Wirtschaftsinformatik – Business Engineering
Duale Hochschule Baden-Württemberg
Ravensburg

Francisco Rodriguez Müller, on September 3, 2025
Mat. Nr.: 2775857, Course: RV-WWIBE122
Supervisor: Prof. Dr. Paul Kirchberg

Declaration of Authenticity

I hereby declare that I have independently written my bachelor's thesis with topic

Multi-Agent Architecture Design
Workflow Automation in Compliance with Operational Excellence

and that I have not used any sources or aids other than those indicated.

Ravensburg, September 3, 2025

(Place and Date) *(Signature)*

Abstract

The literature suggests three design-relevant insights: (1) OpEx increasingly hinges on adaptability, decision quality, and compliance—not only efficiency; (2) automation is moving from deterministic scripts (RPA) to adaptive, tool-using agents; and (3) MAS provide a natural organizing structure for such agents in complex enterprises but necessitate robust governance, monitoring, and integration. These insights inform the requirements (e.g., traceability, policy constraints, observability) and the architecture choices (agent roles, interaction patterns, orchestration) developed later in this thesis.

Contents

1 Introduction

Organizations across industries continue to face persistent challenges in achieving operational excellence. Fragmented processes, manual interventions, and inconsistent data quality undermine efficiency and decision-making. Legacy workflows and siloed systems exacerbate these inefficiencies, while traditional automation approaches often lack the adaptability needed in dynamic business environments. For companies, this translates into slower response times, higher compliance risks, and limited scalability—issues that directly threaten competitiveness.

Generative artificial intelligence (GenAI) opens new possibilities to extend automation beyond deterministic scripts, enabling adaptive, tool-using agents that support governance, decision quality, and agility. However, despite this potential, organizations and the academic literature alike lack structured strategies and conceptual frameworks for embedding such agentic capabilities into operational workflows in a scalable, value-driven way. This gap motivates the present research.

This thesis investigates how multi-agent systems (MAS) can serve as a reference architecture for integrating GenAI into enterprise workflow automation. The central research question is:

How can a multi-agent architecture be designed to integrate GenAI capabilities into workflow automation, in order to enhance agility, compliance, & decision quality to achieve operational excellence?

To answer this question, the study addresses the following sub-questions:

- *What are the strengths & limitations of GenAI in the context of workflow automation using a multi-agent architecture?*
- *Which design requirements & agent roles are necessary to align a multi-agent architecture with the goals of operational excellence?*
- *Under which conditions is deploying a generative multi-agent architecture justified over traditional automation approaches?*

Methodologically, the thesis applies Design Science Research (DSR) to develop a conceptual reference architecture. The approach synthesizes requirements from academic literature and operational excellence principles, models agent roles and interactions, and derives applicability conditions for real-world deployment.

The core contribution of this work is a conceptual design of a multi-agent system that leverages GenAI to support operational excellence in enterprise workflows.

Specifically, it delivers:

1. A structured synthesis of system requirements derived from academic literature and operational excellence principles.
2. A conceptual architecture detailing agent roles, interactions, and integration points
3. A set of applicability conditions and design considerations to guide future deployment and evaluation of generative multi-agent architectures in practice.

The scope is limited to conceptual design; formal evaluation and technical implementation are proposed as future work. The approach remains industry-agnostic but draws illustrative examples from the financial services sector, given its regulatory complexity and reliance on legacy systems.

The thesis is structured as follows: Section 2 outlines the research methodology, including DSR and supporting methods. Section 3 presents a literature review on operational excellence, automation paradigms, and multi-agent systems. Section 4 develops applicability conditions and use case illustrations. Section 5 introduces the conceptual architecture design, and Section 6 concludes with reflections and directions for future research.

2 Methodology

This research adopts the DSR methodology, which is particularly suitable for investigating the application of information systems (IS) in business contexts. In support of the different phases of the study, qualitative content analysis, requirements engineering (RE), and process modeling are additionally employed.

The literature review follows the qualitative content analysis approach proposed by Mayring (2022) to extract and synthesize relevant knowledge. However, since this method is considered standard practice in a thesis, it has no dedicated subsection.

2.1 Design Science Research Methodology

This thesis applies DSR methodology to develop a conceptual architecture for a generative MAS aimed at enterprise workflow automation. DSR is well-suited for producing prescriptive artifacts—constructs, models, methods, and instantiations—that solve defined organizational problems (Hevner et al. 2004).

In this work:

- *Constructs* represent the MAS-related concepts identified (e.g., Orchestration Agent, Compliance Verification Agent, Data Integration Agent).
- The *Model* is the conceptual architecture integrating these constructs into a coherent system design (Section 5).
- The *Method* (algorithm or guideline) gives life to the model by explaining the steps toward performing the desired task. March and G. F. Smith (1995) use the example of data structures, which combine different data types with algorithms to store and retrieve data.
- An *Instantiation* is the actual realization of an artifact in the real world. It requires the translation of constructs, models, and methods into working software or systems that address the business problem.

In contrast to natural science, design science is more concerned with utility than truth. Researchers build an artifact to demonstrate feasibility, either by proving that a viable solution exists for an unaddressed problem or by improving upon existing solutions. This artifact then becomes the object of study. In many business contexts, artifacts are created out of necessity without a formal methodological process. DSR can be used retrospectively to reconstruct the underlying constructs, models, and methods in order to evaluate and share their utility (cf. March and G. F. Smith 1995, pp. 256–259).

This work follows the DSR process defined by Peffers et al. (2007), consisting of the following activities:

1. *Problem identification*: Identifying a problem or set of problems that need to be addressed. Proper fragmentation of the problem into concrete issues helps define the project scope.
2. *Objective identification*: Inferring objectives from the problem definition and literature. These are later transformed into criteria for evaluating the success of the artifact. Objectives can be qualitative (e.g., introducing a new paradigm) or quantitative (e.g., meeting a KPI threshold).
3. *Design & development*: Designing and creating the artifact, which may consist of constructs, models, and methods. This thesis focuses on conceptual design rather than instantiation.
4. *Demonstration*: Applying the artifact in a practical scenario to illustrate its utility. This can take place in a real-world setting or controlled experiment.

5. *Evaluation*: Assessing the artifact using criteria defined in phase two. Evaluation tools may include test scenarios or simulations. Iterations between phases two to five are common until the artifact reaches satisfactory maturity.
6. *Communication*: Communicating the artifact and findings with scientific rigor to both technical experts and decision-makers.

Although these steps represent the nominal sequence, the DSR process is not strictly linear. Peffers et al. (2007 2007, pp. 52–56) describe four entry points into the process: problem-centered, objective-centered, design- and development-centered, and client/context-initiated. This thesis follows a *design-centered approach*, meaning the research is guided by the capabilities and constraints of existing but emerging technologies. The goal is to derive design requirements from relevant literature and operational excellence principles, and to conceptually design a multi-agent architecture that leverages these technologies. Illustrative use cases—drawn from the financial services sector—are used to ground the discussion, while maintaining the architecture's *industry-agnostic* and *client-centric* scope. The scope is limited to the conceptual design of a method and architecture for a MAS; development, demonstration, and evaluation are proposed as future work. To support this design, the next section outlines how relevant system requirements were identified and structured.

DSR was selected over alternative research strategies such as case study or action research because it is well-suited to the creation of prescriptive artifacts aimed at solving defined organizational problems. This aligns with the thesis objective of producing a conceptual architecture rather than solely observing or intervening in an existing setting.

2.2 Requirements Engineering & System Analysis

RE represents the initial practical phase in designing an information technology system. It translates the abstract definition of a problem into concrete specifications—referred to as requirements—that describe what an IT system should do before it exists. A requirement is a condition or feature that the system must meet to assist the user in performing a task or achieving a goal. A contract, service-level agreement, or government regulation is also a form of requirement (cf. IEEE 1991, p. 62).

In the context of this thesis, requirements are formulated for a conceptual MAS intended to support enterprise workflow automation. These requirements can be classified into three main categories. *Functional requirements* describe the essential characteristics needed for the system to achieve its intended purpose. *Quality requirements* improve system performance or usability (e.g., resource optimization, user experience, or processing speed). *Constraints* define external boundaries such as budget, time, dependencies, and system integration (cf. Glinz et al. 2020, p. 8).

In this thesis, requirements are synthesized from academic literature, technical documentation, and case studies. This aligns with the artifact-centered nature of DSR, where requirements emerge from the problem context and guide the conceptual design. The RE process follows three main stages:

1. *Contextual Framing*: Analyzing the problem space by identifying recurring challenges, architectural needs, and strategic goals related to operational excellence and workflow automation in enterprise environments.
2. *Requirements Extraction*: Extracting relevant requirements through qualitative literature review and document analysis. Sources include publications on agentic systems, enterprise architecture frameworks, ERP documentation, and industry use cases.
3. *Requirements Structuring & Traceability*: Clustering and prioritizing requirements and linking them to the system architecture. This ensures design decisions are grounded in the problem definition and overall research objectives.

The outcome of the RE process directly informs the system design, laying the foundation for the conceptual MAS architecture presented in the next subsection. The resulting requirements are explicitly linked back to the central research question and sub-questions, ensuring traceability from problem identification through to the proposed architecture design. This ensures ongoing aligment between the research objectives and the artifact's conceptual structure.

2.3 Information System Design

Within the DSR process, this stage corresponds to the *design & development* activity as defined by Peffers et al. (2007). While Section 2.1 outlined the overall research cycle, the focus here is on the methodological approach used to transform the structured requirements from Section 2.2 into a formal conceptual artifact. The outcome of this stage is a *model* in the DSR sense (Peffers et al. 2007), representing a prescriptive solution that can be evaluated, communicated, and eventually instantiated.

From a methodological perspective, information system design (ISD) is not an ad-hoc creative process but a structured, model-driven activity. Hevner et al. (cf. 2004, p. 68) define ISD as both a process (comprising expert activities that build and evaluate artifacts) and a product (represented by constructs, models, methods, or instantiations aimed at solving an identified problem). ISD translates an abstract, requirement-oriented view of a system into a structured specification that can be implemented. Situated between requirements analysis and implementation, ISD ensures that all functional, quality, and constraint requirements are accurately reflected in the proposed solution concept.

By explicitly linking each design element back to its originating requirement, the approach supports both *validation* (ensuring the right system is built) and *verification* (ensuring the system is built right), while maintaining adaptability to changing operational conditions. In a research context, the design step produces conceptual models that form the basis for communication among stakeholders, evaluation against requirements, and eventual realization (**wand1995ontology**).

In this thesis, the conceptual design process follows *model-based systems engineering* (MBSE) principles, employing the Systems Modeling Language (SysML) as the primary modeling notation. SysML, an Object Management Group standard derived from UML and adapted for systems engineering (Object Management Group 2023), supports both *structural* views (e.g., block definition diagrams, internal block diagrams) and *behavioral* views (e.g., activity, sequence, and state machine diagrams). It also provides dedicated requirement diagrams that enable explicit traceability between requirements and design elements through mechanisms such as the ≪`satisfy`≫ and ≪`verify`≫ relationships (**friedenthal2014practical**). This traceability ensures that, for example, a "Compliance Verification Agent" modeled in a block definition diagram can be directly linked to quality and constraint requirements concerning auditability and regulatory compliance.

SysML was selected over alternative notations such as UML, ArchiMate, or BPMN for three primary reasons: (1) it offers an integrated approach to structural, behavioral, and requirement modeling within a single framework; (2) it is vendor-neutral and widely adopted in both academia and industry, ensuring long-term relevance; and (3) it is well-suited to multi-domain systems that combine software agents, business processes, and IT infrastructure—critical for the architecture envisioned in this research. While UML excels in software-level specification, it lacks native requirement modeling; ArchiMate is optimized for enterprise architecture abstraction but offers less granularity in behavioral modeling; and BPMN focuses on process flows without structural integration. The integrated nature of SysML is especially advantageous for operational excellence initiatives, where governance, compliance, and continuous improvement must be considered alongside technical design.

In the context of this thesis, the application of SysML is scoped to the conceptual level. The modeling activities include:

- *Requirement Diagrams* to link each modeled component to specific functional, quality, and constraint requirements from Section **??**.
- *Block Definition Diagrams* to represent the high-level system structure and main agent types (e.g., Orchestration Agent, Compliance Verification Agent, Data Integration Agent).
- *Activity Diagrams* to illustrate core interaction patterns between agents and external systems, ensuring alignment with orchestration and governance principles.

By adopting a formalized, traceability-driven modeling approach, the ISD stage ensures that the proposed artifact is rigorously specified, grounded in the problem definition, and evaluable against explicit criteria. The resulting architecture views are presented in Section 5.3.

3 Literature Review

This section synthesizes academic and industry perspectives on three pillars underpinning this research: (i) the evolving definition of operational excellence, (ii) the shift in enterprise automation from rules-based RPA to goal-oriented generative agents, and (iii) the role of multi-agent systems in complex enterprise settings. The review follows a qualitative content analysis approach (Mayring 2022) and remains problem-oriented: each subsection surfaces implications for the requirements engineered in Section 2.2 and the architecture designed in Section 5.

3.1 The Evolving Definition of Operational Excellence

Operational excellence (OpEx) emerged from manufacturing disciplines—Lean and Six Sigma—where the primary levers were waste elimination, variance reduction, and standard work (**Womack1990**; Harry and Schroeder 1998). Contemporary excellence models broaden this focus to services and knowledge work, emphasizing culture, strategy alignment, governance, and learning. Representative references include the Shingo Model and the EFQM Model, both of which integrate people, process, and result dimensions while stressing continuous improvement and adaptability (Shingo Institute 2014; EFQM 2020).

In data-rich and highly regulated domains (e.g., finance), the OpEx constraint is increasingly cognitive rather than purely procedural: human attention, handoffs, and decision latency limit throughput and quality (Hammer 2004; Davenport 2018). Yet most excellence frameworks still presuppose human-led improvement cycles, offering limited guidance on embedding autonomous, machine-driven capabilities for sustained optimization, monitoring, and compliance. This gap motivates evaluating whether agentic, tool-using AI can extend OpEx from periodic human interventions toward continuous, machine-assisted improvement and control. For this thesis, the implications are twofold: requirements should (1) include decision-quality, compliance, and auditability as first-class non-functional goals, and (2) anticipate organizational enablers (roles, governance) that allow AI systems to participate in improvement cycles.

3.2 Automation Paradigms: From RPA to GenAI Agents

Robotic Process Automation (RPA) has proven effective for stable, rule-based back-office tasks and has delivered measurable cost, speed, and quality benefits (Lacity and Willcocks 2016). However, RPA's brittleness under interface or policy changes and its limited handling of unstructured inputs constrain scalability in dynamic environments (Syed et al. 2021). "Intelligent" variants (often termed IPA) augment RPA with OCR, NLP, and ML components, broadening applicability but remaining predominantly workflow- and rule-driven.

GenAI agents mark a paradigm shift: powered by large language models and tool orchestration, agents can decompose goals into sub-tasks, call external tools/APIs, reflect on intermediate outputs, and adapt strategies during execution (Park et al. 2023; Rodriguez Müller 2025). Conceptually, this elevates automation from *process* *mimicry* (predefined steps) to *goal-oriented problem solving* (contextual reasoning under uncertainty). The literature highlights open challenges central to enterprise adoption—reliability under distribution shift, controllability, explainability, data protection, and governance of autonomous actions (Bommasani et al. 2022). For system design, this motivates requirements for guardrails (policy constraints, approvals), observability (telemetry, traces), and life-cycle management (versioning, evaluation) that exceed what is typical for RPA.

3.3 Multi-Agent Systems in Enterprise Contexts

MAS comprise autonomous, interacting agents that perceive, decide, and act within a shared environment (Wooldridge 2009). MAS have long been proposed for enterprise-scale coordination problems—supply chains, scheduling, and distributed control—leveraging decomposition, local decision-making, and coordination protocols (Parunak 1999; Jennings and Bussmann 2003). Established patterns include the blackboard architecture for shared problem solving (Nii 1986) and market/negotiation mechanisms such as the Contract Net Protocol for dynamic task allocation (R. G. Smith 1980). Methodologies like Gaia provide organizational abstractions (roles, interactions, norms) useful for analysis and design (Zambonelli et al. 2003).

For heterogeneous enterprise landscapes (ERP, data warehouses, SaaS, legacy), MAS offer modularity and fault isolation: specialized agents encapsulate capabilities (e.g., document understanding, policy checking, posting to ERP), coordinate via protocols, and scale horizontally. However, integration and governance remain hard problems: ensuring interoperability with legacy systems, maintaining compliance and audit trails, and making agent decisions transparent enough for risk and regulatory stakeholders (Luck et al. 2005). The recent infusion of GenAI amplifies MAS potential—agents can reason with unstructured artifacts, learn task patterns, and collaborate—but simultaneously raises the bar for safety, observability, and evaluation.

4 Applicability Conditions for Agentic Automation

4.1 Workflow Sustainability Criteria

4.2 Use Case Illustration: Finance Domain

4.3 Use Case Validation of the Evaluation Framework

5 Conceptual Architecture Design

5.1 Agents Roles and Capabilities

5.2 Agents Roles, Behaviors, & System Embedding

5.3 Multi-Agent Interaction & Orchestration Model

6 Conclusion

References

Bommasani, Rishi et al. (2022). "On the Opportunities and Risks of Foundation Models". *Journal of Artificial Intelligence Research* 76, pp. 191–201.

Davenport, Thomas H. (2018). *The AI Advantage: How to Put the Artificial Intelligence Revolution to Work*. Cambridge, MA: MIT Press.

EFQM (2020). *The EFQM Model 2020*. Brussels: EFQM.

Glinz, Martin et al. (2020). *Handbook for the CPRE Foundation Level according to the IREB Standard*. International Requiremenst Engineering Board.

Hammer, Michael (2004). "Deep Change: How Operational Innovation Can Transform Your Company". *Harvard Business Review* 82, pp. 84–93.

Harry, Mikel and Richard Schroeder (1998). *Six Sigma: The Breakthrough Management Strategy Revolutionizing the World's Top Corporations*. New York: Currency/Doubleday.

Hevner, Alan R. et al. (2004). "Design Science in Information Systems Research". *MIS Quarterly* 28, pp. 75–105.

IEEE (1991). *Standard glossary of software engineering terminology*. Corrected ed. New York: Institute of Electrical and Electronics Engineers.

Jennings, Nicholas R. and Stephen Bussmann (2003). "Agent-based Control Systems: Why Are They Suited to Engineering Complex Systems?" *IEEE Control Systems Magazine* 23, pp. 61–73.

Lacity, Mary C. and Leslie P. Willcocks (2016). *Robotic Process Automation at Telefónica O2*. Tech. rep. London: London School of Economics and Political Science.

Luck, Michael et al. (2005). *Agent Technology: Computing as Interaction – A Roadmap for Agent-Based Computing*. Tech. rep. Southampton: AgentLink III.

March, Salvatore T. and Gerald F. Smith (1995). "Design and natural science research on information technology". *Decision Support Systems* 15, pp. 251–266.

Mayring, Philipp (2022). *Qualitative content analysis: a step-by-step guide*. London Thousand Oaks, California New Delhi Singapore: SAGE.

Nii, H. Penny (1986). "The Blackboard Model of Problem Solving and the Evolution of Blackboard Architectures". *AI Magazine* 7, pp. 38–53.

Object Management Group (2023). *OMG Systems Modeling Language (SysML)*.

Park, Joon Sung et al. (2023). "Generative Agents: Interactive Simulacra of Human Behavior". Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology (UIST), pp. 1–22.

Parunak, H. Van Dyke (1999). "Industrial and Practical Applications of Distributed AI". Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence. Ed. by Gerhard Weiss. Cambridge, MA: MIT Press, pp. 377–421.

Peffers, Ken et al. (2007). "A design science research methodology for information systems research". *Journal of Management Information Systems* 24, pp. 45–77.

Rodriguez Müller, Francisco (2025). *A Review of the Literature on Generative Agents: Capabilities and Limitations*. Zenodo.

Shingo Institute (2014). *The Shingo Model for Operational Excellence*. Logan, UT: Shingo Institute.

Smith, Reid G. (1980). "The Contract Net Protocol: High-level Communication and Control in a Distributed Problem Solver". Proceedings of the 1st Annual National Conference on Artificial Intelligence (AAAI), pp. 357–366.

Syed, Rizwan et al. (2021). "Robotic Process Automation: Contemporary Themes and Challenges". *Computers in Industry* 125, p. 103389.

Wooldridge, Michael (2009). *An Introduction to MultiAgent Systems*. 2nd ed. Chichester: John Wiley & Sons.

Zambonelli, Franco et al. (2003). "Developing Multiagent Systems: The Gaia Methodology". *ACM Transactions on Software Engineering and Methodology* 12, pp. 317–370.

Appendix

Citation Styles

- **Standard citation (parenthetical)** Use when the source supports your point but isn't central to the sentence. *AI is foundational in modern computing (Rodriguez Müller 2025).*
- **With page number** Use when referring to a specific passage or detail. *AI systems raise ethical questions (Rodriguez Müller 2025, p. 45).*
- **With page range** Use when summarizing or referencing a larger section. *Example:* Search algorithms are discussed in detail (Rodriguez Müller 2025, pp. 15–24).
- **Indirect citation with "cf."** Use when comparing or referencing related work (not quoted directly). *Other researchers have a similar view (cf. Rodriguez Müller 2025, p. 112).*
- **Narrative citation** Use when the author is part of your sentence. *Rodriguez Müller (2025) discuss the impact of generative AI.*
- **Narrative citation with page number** Combine author-as-subject with specific location in the text. *Rodriguez Müller (2025, p. 112) emphasize model transparency.*
- **Narrative citation with "cf."** Used in comparative framing, with the author mentioned. *Rodriguez Müller (cf. 2025, p. 112) provide a related perspective.*
- **Multiple citations with page numbers** Use to support a general claim with multiple sources. *AI in finance is widely explored (Rodriguez Müller 2025, pp. 15–17; Rodriguez Müller 2025, p. 112).*
- **Quote with citation** Use for exact wording with mandatory page reference. *"Artificial intelligence is the new electricity" (Rodriguez Müller 2025, p. 7).*

SysML Diagrams

Interview Guide & Transcripts

Extended Tables and Results