Towards Agentic Orchestration for Business Processes in the Insurance Industry

Christin Schulze
Arne Koschel
Andreas Hausotter
Hochschule Hannover
University of Applied Sciences & Arts Hannover
Faculty IV, Department of Computer Science
Hanover, Germany
Email: {christin.schulze, arne.koschel, andreas.hausotter}@hs-hannover.de

Tim van Dorp
Lennart Vermehr
Donika Mirena
Hochschule Hannover
University of Applied Sciences & Arts Hannover
Faculty IV, Department of Computer Science
Hanover, Germany
Email: {tim.van-dorp, lennart.vermehr, donika.mirena}@stud.hs-hannover.de

***Abstract*—The growing complexity of business processes poses a challenge to traditional, deterministic orchestration approaches, especially in highly regulated, event-driven fields like insurance. Although standards such as the Business Process Model and Notation (BPMN) allow for the precise specification and execution of workflows, they often lack the flexibility necessary to handle uncertainty and exceptional situations. This paper examines Agentic Orchestration approaches in insurance business processes using Camunda as a workflow engine. From a practical perspective, we explore how intelligent agents can be incorporated into BPMN processes to facilitate non-deterministic decision-making and adaptive behavior. We implement and evaluate two agentic approaches in realistic insurance scenarios, allowing for a comparative analysis of flexibility, integration effort, and practical applicability. Based on these results, the paper discusses the strengths and limitations of these approaches, as well as the implications and future research directions for Agentic Orchestration in enterprise process management.**

Keywords*—*Business Processes; AI; Agentic Orchestration; BPMN.

I. Introduction

The accurate implementation of business processes is a fundamental requirement in software development, as inaccuracies can adversely affect system reliability and organizational performance in digital environments. To address this challenge, Business Process Model and Notation (BPMN) [1] has been established as a standardized modeling language for specifying business processes. BPMN facilitates a common understanding between technical developers and domain experts, thereby supporting the correct and consistent implementation of complex business workflows.

In this context, companies often use workflow engines such as Camunda [2] to automate BPMN process execution. These engines automate, control, and monitor business processes. Despite their advantages, classic BPMN processes are inherently deterministic. Decision logic, process paths, and responses to events must all be modeled in advance. This results in limited flexibility, particularly in knowledge-intensive or event-driven domains, where unforeseen situations, incomplete information, and case-specific characteristics cannot be adequately represented. To overcome these limitations, traditional process orchestration is being expanded to include a concept known as Agentic Orchestration. This approach integrates intelligent software agents directly into BPMN processes, enabling them to perform non-deterministic tasks, make context-dependent decisions, and generate recommendations for action. This approach adds adaptive components to process execution while maintaining governance, transparency, and control mechanisms.

In domains such as the insurance industry, where processes are often characterized by uncertainty, exceptions, and unique case scenarios, Agentic Orchestration offers new opportunities for evaluating cases dynamically, structuring complex decisions in advance, and reducing the workload of human employees.

This paper introduces Agentic Orchestration. The paper is structured as follows. First, we review existing related work and platforms for workflow automation in Section II, narrowing the scope to Camunda and its native support for AI agent integration within BPMN processes. In Section III, we give a short introduction to Agentic Orchestration and its meaning. Section IV shows a BPMN process from our insurance partners, in which we attempt to implement Agentic Orchestration. Section V provides an overview of two approaches to using Agentic Orchestration in Camunda, and we compare these two approaches. In Section VI, we summarize our findings and provide an outlook.

II. Related Work

The orchestration of services, including microservices, constitutes a central paradigm for managing distributed systems. When coordinated control becomes necessary, the question arises of how services should be structured, executed, and monitored. Orchestration introduces a dedicated control component that manages workflow logic, coordinates service interactions, and handles failures. While practitioner literature highlights orchestration as a key design option for microservices [3]–[5], academic research has extensively examined workflow execution and scheduling in distributed and cloud environments, emphasizing challenges such as resource optimization and scalability [6].

BPMN [1] has emerged as a de facto standard for modeling orchestrated workflows and for bridging business and technical perspectives. Model-driven approaches demonstrate how BPMN models can be transformed into executable artifacts and integrated with behavioral specifications [7]. Research on flexible workflow architectures proposes adaptive mechanisms, including rule-based extensions, to support runtime variability and dynamic process changes [8]. Moreover, BPMN has proven suitable for structuring complex and data-intensive workflows, for example, in Big Data systems [9].

Concurrently, AI-assisted modeling and AI-driven execution components are gaining momentum. Recent studies evaluate the structural quality of AI-generated BPMN models and highlight the necessity of systematic validation mechanisms [10]. However, existing research primarily focuses on workflow flexibility and scheduling, BPMN model transformation and quality assessment, or AI-supported model generation. A systematic investigation of how AI agents can be integrated into BPMN-based orchestration within microservice architectures while preserving process transparency, structural quality, and governance requirements remains largely unaddressed. In particular, there is limited empirical evidence comparing architectural integration strategies that combine deterministic workflow control with non-deterministic, agent-based decision-making.

In industrial practice, BPMN-based workflow engines such as Camunda [2] are widely used to automate orchestrated processes. Camunda's concept of Agentic Orchestration [11] introduces AI agents into BPMN-controlled workflows and enables context-sensitive decisions within structured processes. Yet, architectural implications, integration patterns, and decision criteria for such hybrid approaches have not been systematically evaluated in domain-specific settings.

Building on prior work in the insurance domain [12]–[14], this paper addresses this gap by analyzing and implementing two alternative approaches for integrating Agentic Orchestration into BPMN-based business processes within microservice architectures. Using an insurance-specific use case, we systematically compare both approaches with respect to flexibility, maintainability, and architectural implications. The results provide structured guidance for selecting suitable integration strategies under different process and governance conditions.

III. Meaning of Agentic Orchestration

BPMN [1], and processes executed by workflow engines such as Camunda, are inherently deterministic. Predefined workflow paths and decisions limit their ability to respond to unforeseen events, reducing flexibility in modern business environments. Agentic Orchestration was developed to overcome these limitations [11]. This approach integrates intelligent agents into BPMN processes, enabling them to perform specific tasks, support or automate non-deterministic activities, and make preliminary decisions. As a result, deterministic process orchestration is enhanced with adaptive, context-aware behavior while preserving governance and control.

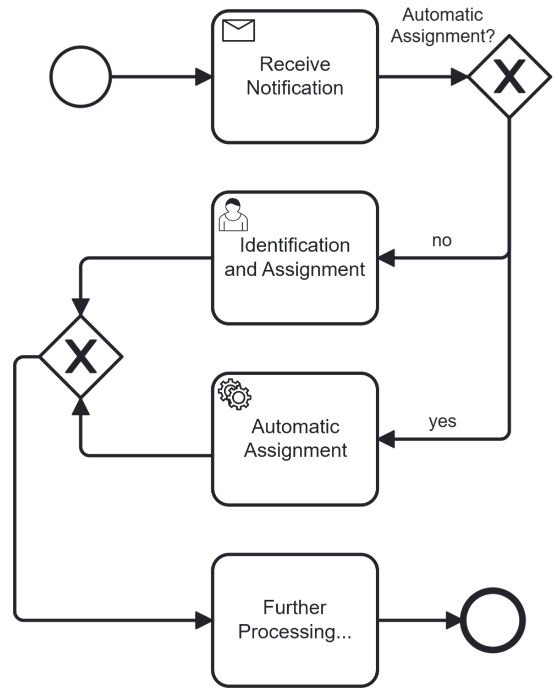

Figure 1. Simplified example - Claim assignment for liability insurance without an agent.

In the insurance domain, Agentic Orchestration fundamentally improves the handling of complex, event-driven processes such as claims management, underwriting, and fraud detection. Intelligent agents allow insurers to dynamically assess cases, pre-classify claims, request missing information, and propose real-time decisions. This leads to increased process speed and flexibility while maintaining regulatory compliance, auditability, and human oversight.

IV. Context and Use Case

In collaboration with our insurance partners, we are exploring the application of Agentic Orchestration to their business processes. For this purpose, we use Camunda Agentic Orchestration and test Camunda-specific tools [2].

The technical environment is based on Camunda 8 as the process orchestration platform, including the Agentic Orchestration components and corresponding connectors. As a Large Language Model, Azure OpenAI Service is used, providing access to GPT-based models via secured enterprise endpoints. The LLM is integrated into the process landscape through Camunda connectors and custom tools, enabling controlled interaction between BPMN processes and AI-based agents [11].

Our goal is to demonstrate how various types of agents can support insurance processes. In our example, a message regarding liability insurance should either be assigned to an

Copyright (c) IARIA, 2026. ISBN: 978-1-68558-372-9

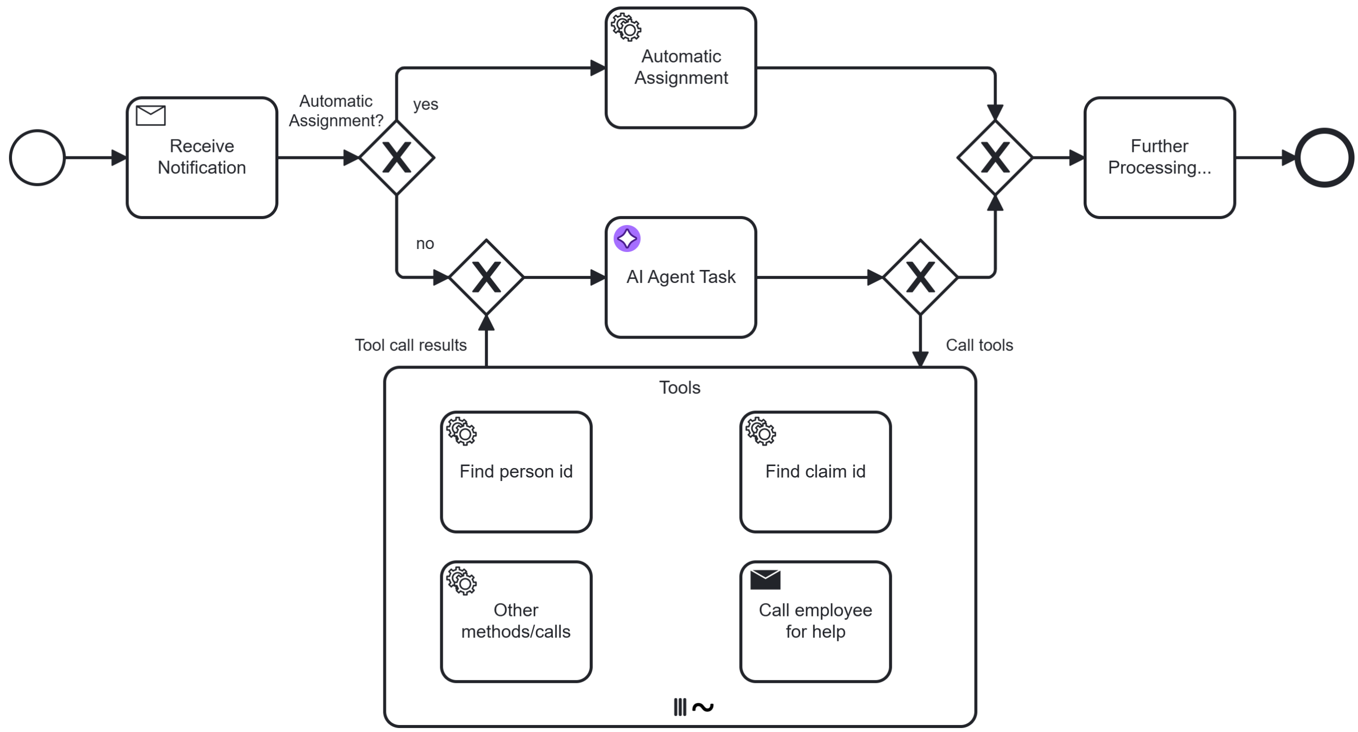

Figure 2. Claim assignment for liability insurance with Agent Task.

existing claim or a new claim should be created for the customer.

Figure 1 shows a simplified version of the process. As soon as a notification is received, it is checked to see if it can be automatically assigned. If not, it is forwarded to an employee. The employee then attempts to assign the report by searching for the customer or the damage in various systems and databases. They also determine whether the claim is new or has been reported previously. Once assigned, further processing can take place, which is irrelevant in the context of this paper.

This search and assignment is now to be taken over by an agent. The agent has access to systems and databases and is responsible for finding as much information as possible about the damage or customer to determine an assignment.

At this stage, the agent does not make any final, autonomous decisions. Instead, it makes recommendations that are reviewed by an employee. Once a certain level of reliability has been achieved through several tests and experiments, decisions will also be automated.

To determine the optimal integration approach for the agent, we evaluate the available options based on the following key criteria:

- **Control of the loop:** Refers to the mechanisms governing task iteration and the specific conditions under which the loop terminates.
- **Modeling effort:** Indicates the complexity and time required to explicitly define the desired logic within the process model.
- **Monitoring per iteration:** Specifies the granularity with which intermediate results and execution states can be tracked during each loop iteration.
- **Event subprocesses:** Describes the capability to handle asynchronous events (e.g., messages, cancellations, or timers) concurrently with the main task.
- **Errors and escalation logic:** Details the mechanisms for capturing and resolving technical exceptions and business-level faults within the workflow.
- **Use Cases:** Identifies the specific operational contexts and practical scenarios for which the respective construct is most suitable.

V. Approaches

So far, we have tried two implementation approaches in our work: Agent Task and Agent Subprocess. The two approaches are introduced briefly below and illustrated using the example of liability insurance. Finally, the two approaches are compared.

A. Agent Task

An Agent Task describes a service task in Camunda and serves as a connector to artificial intelligence. For example, it can interface with a Large Language Model (LLM). This approach is characterized by a loop containing any number of iterations that successively solve the task.

In a modeled subprocess, the agent has various tools at their disposal to complete the task. The agent decides which tool

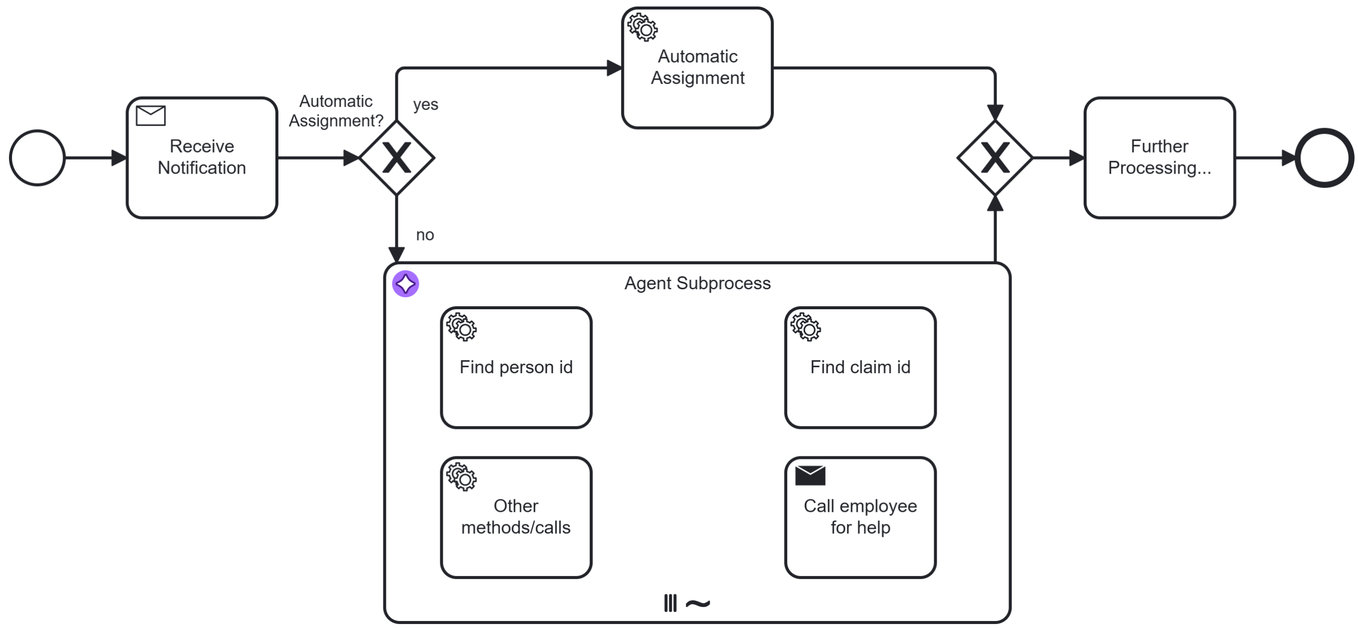

Figure 3. Claim assignment for liability insurance with Agent Subprocess.

to use. After using a tool, the task returns to the agent. This constitutes another call to the LLM. The result is then used to determine whether to finish the task or call the subprocess again.

As Figure 2 shows, the AI connector is separate from the AI Agent Task. In this case, the LLM is accessed using the available information and the means to further determine it. After one iteration, the connector is used again to determine if another iteration is necessary and, if so, which method to use. Employees can be notified of errors or unexpected results.

This approach is suitable for precise step-by-step monitoring. After each iteration, it is clear what decisions the agent will make and what steps it will take to arrive at the result.

B. Agent Subprocess

The Agent Subprocess takes a different approach. In this case, the connector, or the connection to the LLM, is inserted directly as a subprocess into the process, bypassing the loop. Within the service task, the LLM is called multiple times as needed, and the agent decides which tool to use. There is also a termination condition for transitioning from the subprocess to the main process. This approach may also include mechanisms for detecting errors or transferring them to a human employee.

Figure 3 shows the Agent Subprocess implementation. There is no loop here because the connector decides internally how many iterations to perform. Various methods are also available to determine further information, and employees can be notified in the event of problems.

This approach contains significantly fewer BPMN elements and structures, making it simpler. The connector in the subprocess controls the entire interaction, thereby reducing the load on the process model. Additionally, these subprocesses can respond to events, such as timers.

C. Comparison

According to initial assessments, it is difficult to make a blanket statement about the use of these approaches. The following criteria (see Table I) can help select an approach. As shown in the table, the two approaches can be used for different use cases, but there is no one-size-fits-all solution. To compare the two approaches objectively, structural complexity (modelling effort) and execution robustness (error susceptibility) were used as evaluation metrics. Runtime metrics, such as token consumption, do not differ significantly by nature of the concept.

Structural complexity was measured based on the required BPMN elements and variable overhead. The AI Agent Task requires explicit modelling of the tool loop, including gateways, as well as complex manual variable mapping for data feedback. In contrast, the Agent Subprocess massively reduces this effort, as the connector automates the orchestration of multiple method calls internally, and no explicit BPMN loop is required. This results in significantly lower modelling effort and reduced diagram complexity.

Execution robustness is measured by system stability and the AI's reliable use of the tools. Although the AI Agent Task offers a great deal of manual control, it is highly prone to errors due to the mandatory, rigid JSON specifications for BPMN routing. If the AI is overly constrained by this high process complexity, it tends to ignore tool calls and instead outputs pure text that breaks the process. The AI Agent Subprocess proves to be significantly more robust here. This is because the connector handles the loop control internally,

TABLE I. Comparison of Agent Task and Agent Subprocess.

Criterion	Agent Task	Agent Subprocess
Control of the loop	Fully modeled in BPMN	Automatically within the Connector
Modeling effort	High, due to loops, gateways, etc.	Comparatively low
Monitoring per iteration	Very high, possible after each iteration	Less, only after the run or explicitly implemented
Event subprocesses	Not included	Contained
Errors and escalation logic	Must be modeled completely in BPMN	Partially executed by the connector
Use Cases	Specialized logic, auditing, approvals, complex loops	Standardized Agentic Orchestration, simple and fast modeling
Recommendation by Camunda	Special cases	Standard recommendation for most processes

error-prone JSON parsing at the process level is completely eliminated. The AI can work directly, which drastically minimizes the workflow termination rate.

For the liability insurance example (see Figures 2 and 3), the Agent Subprocess is the more appropriate approach. This use case is not critical because it only involves gathering information, and the agent has not yet made any decisions. Therefore, it is not necessary to monitor the iterations as strictly as with the Agent Task approach.

The example could also work with the Agent Task approach, but it would be significantly more complex and require more initial effort. To improve monitoring in general, iteration steps can be logged to track decisions later, even when using the Agent Subprocess.

VI. Conclusion and Future Work

This paper examined initial implementation possibilities for Agentic Orchestration in the insurance industry. Two initial approaches were tested as prototypes. Both approaches are suitable for our use case, but the Agent Subprocess is slightly more effective due to its streamlined design.

However, there are (as yet) no standards governing when to use which approach. Our comparison provides guidance for decision-making. However, decisions must be made on a case-by-case basis.

Nevertheless, this work is only a preliminary study. Future work will include many different issues.

First of all, we will provide a more technical description of our prototype, including its architecture, design, connectors, and data sources.

From a business-related point of view, the question arises where the decision-making normally takes place, i.e., are the decisions predominantly made by an agent, or are they forwarded to an employee? For this purpose, a set of metrics will be designed to address business process performance, including latency, the number of iterations per case, and the human escalation rate. A series of test cases for the claim assignment will be used to generate measured values for the metrics listed above.

Potential challenges, such as bias and hallucinations, coming with the use of AI agents must be carefully examined. Again, the above tests may help to assess the risks.

References

[1] OMG, “Business process model and notation (BPMN), version 2.0,” p. 538. [Online]. Available: https://www.omg.org/spec/BPMN/2.0/About-BPMN

[2] Camunda. Camunda platform. [Online]. Available: https://camunda.com/

[3] M. Folwer and J. Lewis. Microservices. [Online]. Available: https://martinfowler.com/articles/microservices.html

[4] S. Newman, *Building Microservices*, 1st ed. O’Reilly Media, Inc.

[5] C. Chen. Choreography vs orchestration. [Online]. Available: https://medium.com/ingeniouslysimple/choreography-vs-orchestration-a6f21cfaccae

[6] S. M. Jaybhaye and V. Z. Attar, “A review on scientific workflow scheduling in cloud computing,” in *2017 2nd International Conference on Communication and Electronics Systems (ICCES)*, pp. 218–223. [Online]. Available: https://ieeexplore.ieee.org/document/8321269

[7] I. Ben Fraj, Y. B. Hlaoui, and L. J. Ben Ayed, “A modeling approach for flexible workflow applications of cloud services,” in *2017 IEEE 41st Annual Computer Software and Applications Conference (COMPSAC)*, vol. 2, pp. 175–180, ISSN: 0730-3157. [Online]. Available: https://ieeexplore.ieee.org/document/8029914

[8] Y. Gao and F. Guan, “The design of flexible workflow in scientific research management system,” in *2018 2nd International Conference on Data Science and Business Analytics (ICDSBA)*, pp. 61–64. [Online]. Available: https://ieeexplore.ieee.org/document/8588888

[9] S. Rabaaoui, M. S. Naceur, and K. Barkaoui, “A BPMN-based approach for modeling complex workflows in big bio data systems,” in *2025 IEEE International Conference on Advanced Systems and Emergent Technologies (IC_ASET)*, pp. 1–6. [Online]. Available: https://ieeexplore.ieee.org/document/11232130

[10] A. Kopp, L. Cibák, M. Godlevskyi, M. Kozulia, D. Orlovskyi, and O. Ivashchenko, “Software tool for evaluation of AI-generated BPMN models using structural measures,” in *2025 IEEE 6th KhPI Week on Advanced Technology (KhPIWeek)*, pp. 1–5, ISSN: 3064-9579. [Online]. Available: https://ieeexplore.ieee.org/document/11288602

[11] Camunda. Agentic orchestration: Model, deploy & manage AI agents seamlessly into your end-to-end processes with camunda. [Online]. Available: https://camunda.com/en/solutions/agentic-orchestration/

[12] A. Koschel, A. Hausotter, R. Buchta, A. Grunewald, M. Lange, and P. Niemann, “Towards a microservice reference architecture for insurance companies.” [Online]. Available: https://www.thinkmind.org/library/SERVICE_COMPUTATION/SERVICE_COMPUTATION_2021

[13] A. Koschel, A. Hausotter, R. Buchta, C. Schulze, P. Niemann, and C. Rust, “Towards the implementation of workflows in a microservices architecture for insurance companies - the coexistence of orchestration and choreography.” [Online]. Available: https://www.thinkmind.org/library/SERVICE_COMPUTATION/SERVICE_COMPUTATION_2023

[14] A. Koschel, A. Hausotter, C. Schulze, H. Meyer, A. Link, and T. van Dorp, “A technical reference architecture for microservices-based applications in the insurance industry.” [Online]. Available: https://www.thinkmind.org/library/SERVICE_COMPUTATION/SERVICE_COMPUTATION_2024