Delegation, Substitution, and Escalation Mechanisms in Workflow Management Systems: A Structured Literature Review

Bachelorarbeit

im Studiengang Informationsmanagement und Unternehmenskommunikation
in der Fakultät Informationsmanagement
der Hochschule für angewandten Wissenschaften Neu-Ulm

Erstbetreuer:	Professor Dr. Thomas Bauer
Zweitbetreuer:	Professor Dr. Jörg-Oliver Vogt

Verfasserin:	Selin Dikme
Matrikelnummer:	2948321
Adresse:	Wileystraße 1, 89231 Neu-Ulm
E-Mail:	Selin.Dikme@student.hnu.de
Studiengang:	IMUK

Datum der Abgabe: 03.06.2026

Abstract

Workflow Management Systems (WFMS) support the execution of business processes by coordinating activities, managing information, and allocating tasks to organizational actors. However, workflow execution rarely follows fully predictable conditions. Resource unavailability, workload fluctuations, and execution delays may require task assignments to be adapted during process execution. In response to such situations, WFMS employ runtime adaptation mechanisms such as delegation, substitution, and escalation to maintain workflow continuity. This thesis examines how delegation, substitution, and escalation mechanisms are modeled and implemented in WFMS and how escalation scenarios arising from deadline violations or resource unavailability should be handled. To address this objective, a structured literature review was conducted to analyze existing approaches to workflow resource allocation, runtime task reassignment, and adaptive workflow execution.

The findings indicate a broader transition from static, role-based resource allocation toward more context-sensitive and runtime-oriented approaches. Contemporary research increasingly incorporates factors such as workload, availability, qualifications, execution history, and case-specific context into allocation decisions. Delegation, substitution, and escalation emerge as complementary mechanisms for maintaining workflow continuity under changing execution conditions. At the same time, the literature reveals persistent tensions between flexibility, authorization consistency, and process control. The analysis further identifies several limitations in current WFMS approaches. Resource allocation often remains strongly dependent on predefined organizational structures, while runtime adaptation capabilities remain limited in operational environments. In particular, substitution mechanisms remain comparatively underdeveloped and insufficiently formalized. Furthermore, limited research addresses the coordinated interaction of delegation, substitution, and escalation during workflow execution. The thesis concludes that future WFMS should support more adaptive, context-sensitive, and integrated runtime intervention mechanisms while balancing organizational flexibility with authorization and process integrity requirements.

Keywords: Workflow Management Systems, WFMS, resource allocation, delegation, substitution, escalation, runtime adaptation, workflow resource management, business process management

Table of Contents

List of Abbreviations

BPM	Business Process Management
WFMS	Workflow Management System
RBAC	Role-Based Access Control
PAIS	Process-Aware Information System
PASS	Parallel Activity Specification Schema
OQL	Organizational Query Language
SoD	Separation of Duties
BoD	Binding of Duties

List of Figures

List of Tables

1 Introduction

Workflow Management Systems (WFMS) support business processes by coordinating activities, managing information, and allocating tasks to organizational actors. As organizational environments become increasingly dynamic, process execution frequently needs to adapt to changing operational conditions such as fluctuating workloads, resource unavailability, or execution delays. This thesis examines how WFMS support adaptive workflow execution under changing runtime conditions, with particular emphasis on delegation, substitution, and escalation. The chapter introduces the background and motivation of the study, formulates the research problem and objectives, and outlines the structure of the thesis.

1.1 Background and Motivation

Organizations increasingly rely on business processes to coordinate operational activities across departments, systems, and organizational actors. To support such processes, WFMS formalize and automate process execution by coordinating activities, managing information, and assigning work items to appropriate resources during execution (van der Aalst & van Hee, 2004; Dumas et al., 2018). As a result, WFMS contribute to greater process transparency, process consistency, and operational efficiency.

A central aspect of workflow execution concerns the allocation of tasks to actors. Determining who should perform a specific activity is commonly referred to as resource allocation or actor assignment, and directly influences process efficiency, execution time, workload distribution, and resource utilization (Cabanillas, 2016; Arias et al., 2018). Traditional WFMS frequently rely on predefined organizational structures such as roles, organizational units, or authorization models to determine eligible task performers (Stroppi et al., 2015; Vanderfeesten & Grefen, 2015).

In practice, however, workflow execution rarely follows a perfectly predictable path. Employees may become unavailable, workloads may fluctuate, organizational priorities may shift, and activities may fail to meet expected deadlines. Consequently, task assignments defined during process design may no longer be appropriate during execution, creating the need for mechanisms that enable workflows to adapt to changing runtime conditions (van der Aalst et al., 2007; Bellaaj et al., 2017). Static assignment mechanisms may therefore become insufficient when workflow continuity depends on flexible reassignment or intervention.

To address such situations, WFMS employ runtime adaptation mechanisms that support workflow continuity under changing execution conditions. Delegation enables the reassignment of responsibilities or execution authority, substitution facilitates the temporary replacement of unavailable actors, and escalation mechanisms support corrective intervention when workflow execution becomes endangered (Atluri et al., 2004; Bauer, 2009; Panagos & Rabinovich, 1997). Given the growing importance of runtime adaptation in workflow execution, understanding how WFMS support these mechanisms has become increasingly relevant.

1.2 Problem Statement

Although modern WFMS provide support for assigning tasks to actors, adapting workflow execution to changing runtime conditions remains challenging. Traditional workflow approaches commonly rely on predefined organizational structures and static assignment rules. While such mechanisms provide clarity and control, they may become insufficient when resources become unavailable, workloads fluctuate, or process conditions change unexpectedly.

To address such situations, WFMS require mechanisms for transferring responsibilities (delegation), identifying replacement actors (substitution), and responding to endangered process execution (escalation). However, existing approaches differ considerably regarding how these mechanisms are modeled, implemented, and integrated into workflow execution. Furthermore, the literature discusses delegation, substitution, and escalation from different perspectives, making it difficult to obtain a consolidated understanding of their role in workflow resource management. A structured literature review is therefore required to compare existing approaches and clarify how WFMS support adaptive task reassignment and runtime intervention under changing execution conditions.

1.3 Research Objectives and Questions

The objective of this thesis is to analyze how WFMS support delegation, substitution, and escalation as mechanisms for adaptive workflow execution. The analysis focuses particularly on the identification of substitute actors, the modeling of reassignment mechanisms, escalation responses, and the limitations of current approaches.

The study is guided by the following research question:

“How should workflow management systems model and implement delegation and substitution mechanisms, and how should escalation scenarios arising from deadline violations or resource unavailability be handled? “

The following sub-questions further structure the analysis:

1. How can substitute actors be identified and selected at design time and runtime?
2. What modeling constructs and allocation rules can be applied to define delegation and substitution?
3. How should workflow systems react to deadline violations and resource unavailability through escalation mechanisms?
4. What limitations exist in current approaches?

1.4 Structure of the Thesis

This thesis is structured into seven chapters. Following this introduction, Chapter 2 presents the research methodology and explains the literature selection and analysis process. Chapter 3 introduces the theoretical and conceptual foundations of the thesis by defining key concepts related to workflow management systems, workflow resource management, delegation, substitution, and escalation.

Chapter 4 analyzes existing approaches identified in the literature. Chapter 5 synthesizes recurring developments and remaining research gaps. Based on these findings, Chapter 6 discusses the findings in relation to the research questions. Finally, Chapter 7 concludes the thesis by summarizing the key findings, outlining limitations, and identifying directions for future research.

2 Methodology

This chapter outlines the methodological approach used in this thesis, including the research design, literature search, selection criteria, and synthesis process applied to investigate delegation, substitution, and escalation mechanisms in workflow management systems.

2.1 Methodological Foundation

This thesis adopts a structured literature review to examine delegation, substitution, and escalation mechanisms in WFMS. Because the thesis examines conceptual mechanisms and modeling approaches rather than measurable empirical outcomes, a literature-based research design was considered appropriate. The methodological approach is informed by recommendations from vom Brocke et al. (2015) and Durach et al. (2017). Vom Brocke et al. (2015) provide guidance for literature searches in information systems research, particularly regarding search scope, database selection, search strategies, and documentation. Durach et al. (2017) emphasize structured conceptual synthesis in interdisciplinary domains where recurring concepts, allocation constructs, and system mechanisms vary across studies.

Following Webster and Watson (2002), the review adopts a concept-centric perspective, focusing on recurring concepts, modelling constructs, and system mechanisms instead of reviewing studies individually. The review process followed an iterative search strategy in which additional concepts and terminology were refined during analysis (vom Brocke et al., 2015).

2.2 Literature Search and Selection

The literature search was conducted using the ACM Digital Library and the HNU University Library search system, which provided access to academic databases and publishers such as IEEE Xplore and SpringerLink. Google Scholar was used as a supplementary search tool to identify additional literature and citation links. Because the topic intersects workflow management, Business Process Management (BPM), organizational modeling, and information systems engineering, literature had to be identified across multiple sources (vom Brocke et al., 2015).

Where peer-reviewed literature provided limited coverage of highly specific technical or conceptual aspects, relevant Bachelor's and Master's theses were consulted as supplementary sources. Initial searches were conducted using combinations of keywords related to the central research topic. Examples of search terms included:

Category	Example Search Terms
Foundational Workflow Concepts	“workflow management systems”, “workflow management system”, “business process management”, “process-aware information systems”, “workflow execution”
Delegation	“workflow delegation”, “task delegation workflow”, “delegation in workflow systems”, “dynamic task reassignment“
Substitution	“workflow substitution”, “substitute actor selection”, “resource substitution workflow”, “replacement performer workflow”, “deputyship workflow” “task reassignment wfms“
Escalation	“workflow escalation”, “business process escalation”, “deadline violation workflow”, “workflow deadline management”
Related Workflow Concepts	“exception handling workflow systems”, “dynamic task allocation”, “role-based assignment workflow”, “workflow resource allocation”

Table 1: Search Terms used to find Literature for this Thesis
Source: Own Illustration

Keyword combinations were varied iteratively to account for inconsistent terminology across disciplines and publication outlets. No strict publication period was defined because foundational WFMS research from earlier decades remains conceptually relevant. To ensure relevance and consistency, literature was selected according to inclusion and exclusion criteria.

Inclusion Criteria	Exclusion Criteria
a) Literature related to workflow management systems, business process management, or process-aware information systems b) Studies addressing delegation, substitution, escalation, task reassignment, resource allocation, or exception handling c) Publications discussing modeling constructs, allocation rules, or implementation approaches relevant to workflow execution d) Peer-reviewed journal articles and conference papers e) Relevant Bachelor's and Master's theses containing substantive conceptual or technical contributions f) Publications written in English or German	a) Studies unrelated to workflow execution or task allocation mechanisms b) Publications without conceptual or implementation relevance to delegation, substitution, or escalation c) Purely domain-specific workflow applications without transferable mechanisms d) Non-accessible or incomplete sources

Table 2: Inclusion and Exclusion Criteria used for Literature
Source: Own Illustration

2.3 Literature Analysis and Synthesis

Following the initial search, the identified literature was screened based on titles, abstracts, and keywords to assess relevance to the research scope. Where relevance remained unclear, full-text screening was conducted.

Additional sources were identified through backward searching (reference list analysis) and forward searching (citation tracking) (Webster & Watson, 2002; vom Brocke et al., 2015).

The search process was concluded when newly identified literature contributed only limited conceptual additions to the emerging thematic categories. No formal saturation criterion was applied.

Following Webster and Watson (2002), the literature was analyzed using a concept-centric synthesis approach. Rather than reviewing studies individually, the literature was grouped according to recurring concepts, modeling approaches, and workflow mechanisms relevant to adaptive workflow execution in WFMS.

The analysis focused on four thematic dimensions derived from the research questions:

Analytical Dimension	**Focus of Analysis**
Substitute actor identification	Design-time and runtime approaches for determining alternative task performers
Delegation and substitution modeling	Allocation rules, authorization constraints, reassignment mechanisms, and modeling constructs
Escalation mechanisms	Responses to deadline violations, resource unavailability, and workflow interruptions
Limitations of current WFMS approaches	Constraints regarding runtime flexibility, integration, security, and practical implementation

Table 3: Analytical Dimensions of the Literature Synthesis
Source: Own illustration

The objective of the synthesis was to identify recurring modeling patterns, compare implementation approaches, and develop a structured understanding of how WFMS support adaptive workflow execution through delegation, substitution, and escalation.

3 Theoretical and Conceptual Foundations

This chapter establishes the theoretical and conceptual basis for analyzing delegation, substitution, and escalation in workflow management systems. It introduces key concepts related to business process management, workflow execution, workflow management systems, and workflow resource management as foundations for analyzing adaptive task reassignment and runtime intervention in workflow execution.

3.1 Business Process Management

Business Process Management (BPM) is a discipline concerned with the systematic design, analysis, execution, monitoring, and continuous improvement of business processes. BPM aims to improve business processes with regard to efficiency, effectiveness, quality, and flexibility (Dumas et al., 2018).

Rather than focusing on isolated tasks or functional units, BPM adopts an end-to-end perspective, emphasizing the management of entire processes that generate value for customers and stakeholders. In this context, a business process can be defined as a collection of interrelated activities, events, and decision points involving various actors and resources, which collectively produce an outcome of value to at least one customer (Dumas et al., 2018). Business processes therefore represent the operational realization of organizational goals by transforming inputs into valuable outputs.

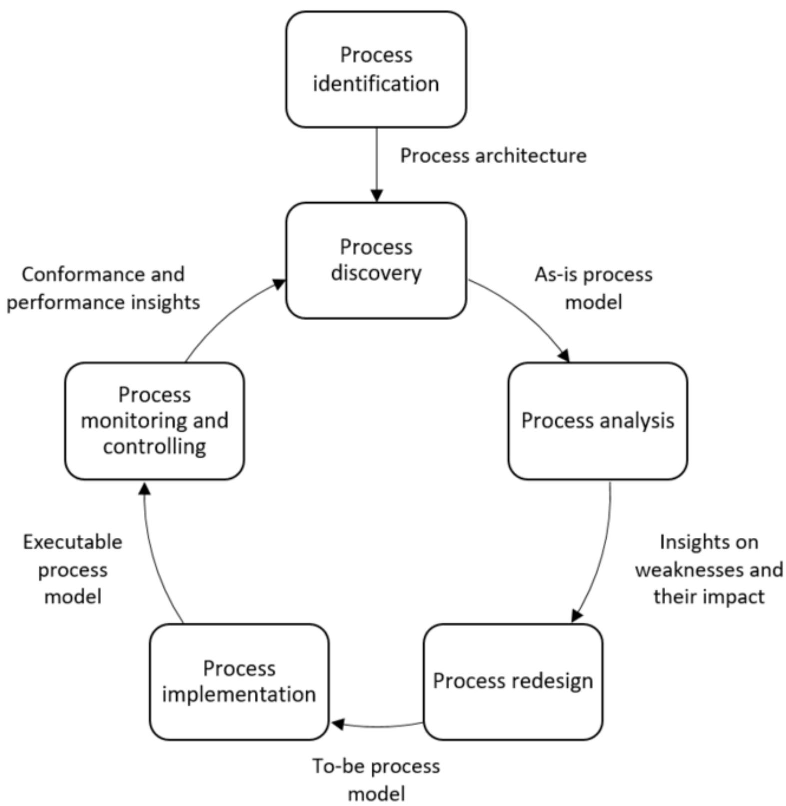

Figure 1: Business Process Management Lifecycle
Source: Adapted from Dumas et al. (2018)

BPM follows a lifecycle-oriented perspective comprising process identification, modeling, analysis, redesign, implementation, and monitoring (Dumas et al., 2018). This lifecycle provides a structured approach for continuously improving business processes over time. Central to this lifecycle are formal process models, which serve as conceptual representations of processes and form the basis for both organizational analysis and technical implementation. While early BPM research primarily focused on control-flow modeling, additional perspectives such as data, temporal, and organizational aspects have become increasingly relevant due to their impact on process performance and robustness.

While BPM provides a holistic and conceptual view on the management and improvement of business processes, a more technical and execution-oriented perspective is required to understand how processes are enacted within information systems. This perspective is provided by workflow concepts, which focus on the structured execution of process instances.

3.2 Business Processes and Workflow Concepts

From a workflow perspective, business processes are operationalised through structured sequences of activities, events, and execution conditions that determine how process instances progress within information systems. Workflow concepts therefore focus on the enactment of process instances and provide a technical realization of business processes. In this context, a process defines how a particular category of cases is executed and thereby operationalizes organizational work. A central concept in Workflow Management Systems (WFMS) is the case. A case represents a concrete instance of a process, such as an insurance claim or a customer order. Each case has a unique identity and a finite lifecycle that begins with its creation and ends upon completion. At any point in time, a case is characterized by its state, which includes relevant data attributes and the progress of execution reflected in fulfilled conditions.

Within this framework, it is essential to distinguish between three closely related concepts. Figure 2 provides a simplified overview of this relationship.

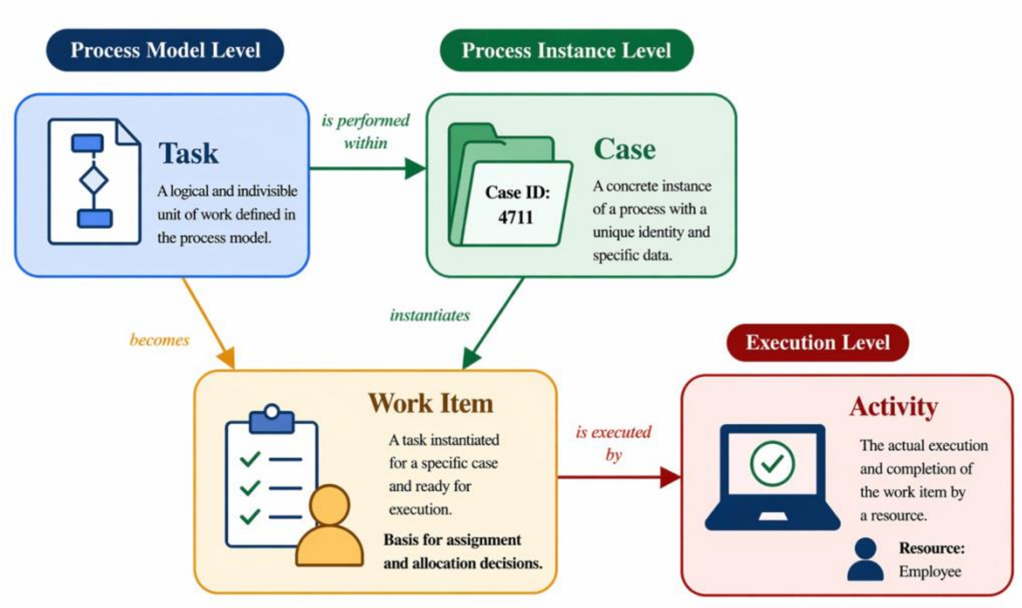

Figure 2: Relationship between Task, Case, Work Item and Activity in WFMS
Source: own illustration based on van der Aalst and van Hee (2001)

A task is defined at the process model level as a logical and indivisible unit of work. A work item refers to a task instance associated with a specific case that is available for execution. Once execution starts, the work item becomes an activity performed by a resource (van der Aalst & van Hee, 2001). An activity denotes the actual execution of a work item by a resource. Distinguishing between tasks, work items, and activities is important because WFMS allocate and monitor work at work-item level rather than at the level of abstract process tasks. Process execution is governed by routing structures that define the lifecycle of cases. According to van der Aalst and van Hee (2004), four basic routing constructs can be distinguished: sequential routing, parallel routing, selective routing, and iteration. These routing constructs determine how workflow instances progress and define possible execution paths within a process.

3.3 Workflow Management Systems

While BPM provides the methodological foundation for analyzing and improving business processes, Workflow Management Systems (WFMS) provide the technological infrastructure required for their operational enactment. A WFMS can be described as a generic software system that supports the modeling, execution, and monitoring of workflows by managing the flow of work between participants and applications (van der Aalst & van Hee, 2004). Rather than executing business activities themselves, WFMS coordinate process execution by ensuring that work items are generated, routed, and assigned in accordance with the underlying process definition.

At the core of a WFMS lies the workflow engine, which interprets process definitions and governs workflow enactment. It manages the creation and termination of cases, evaluates execution conditions, generates work items, allocates them to resources, invokes supporting applications, and records execution data (van der Aalst & van Hee, 2001).

In addition to the workflow engine, WFMS typically provide process definition tools for modeling workflows, workflow client applications that present worklists to users, as well as administrative and monitoring components for system management. The separation of process control logic from application functionality enables flexibility and allows process changes to be implemented without modifying underlying application systems.

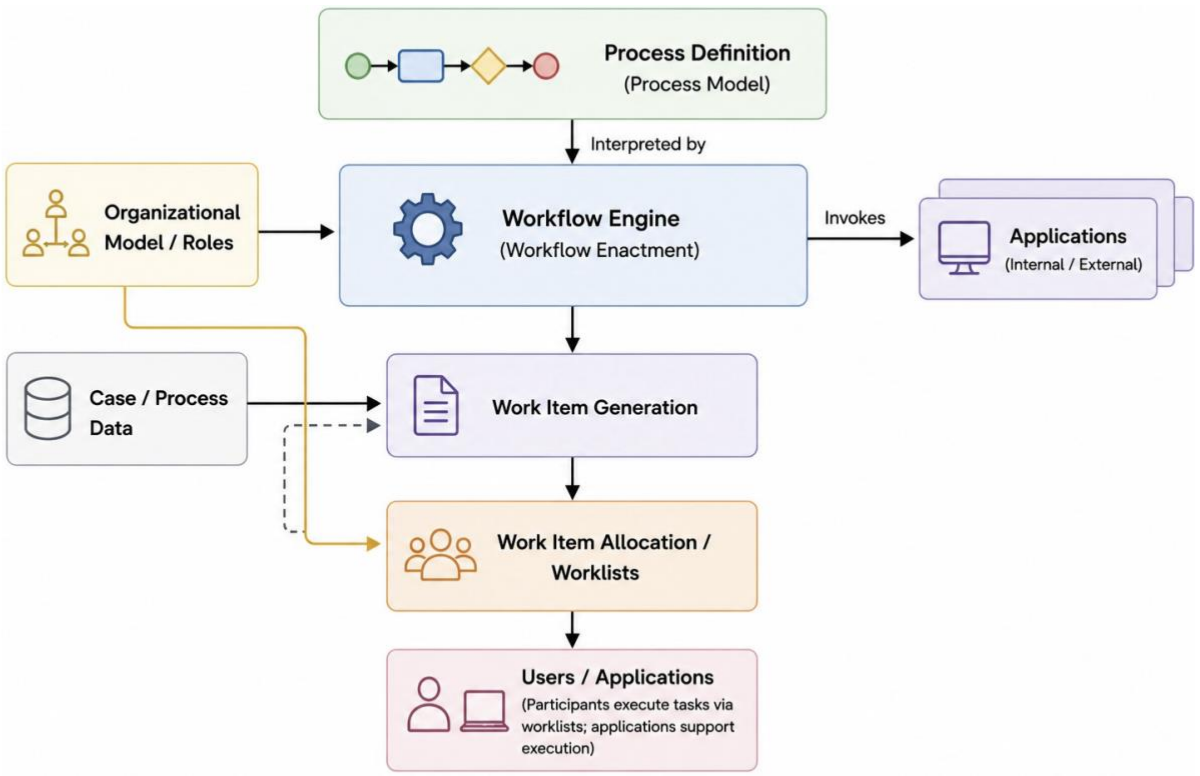

Figure 3: Overview of the WFMS Architecture
Source: own illustration inspired by van der Aalst and van Hee (2004)

Figure 3 illustrates the simplified operational architecture of a WFMS and the interaction between process definitions, workflow enactment, and work item allocation.

One of the core operational functions of WFMS concerns the allocation of work items to resources. When the state of a case satisfies the preconditions of a task, the workflow engine generates a corresponding work item that is made available for execution. Different allocation strategies can be distinguished (van der Aalst and van Hee, 2004). In push-based allocation, the system assigns a work item directly to a specific resource. In pull-based allocation, work items are offered to a set of eligible resources who may claim them autonomously. In both cases, work items are typically presented through worklists, which serve as the primary interface between the system and its users by displaying the tasks that a resource is authorized and eligible to execute.

In most WFMS, tasks are associated with roles or resource classes at design time rather than with specific individuals. At runtime, the system determines eligible resources by evaluating organizational structures and allocation rules. Role-based allocation primarily relies on structural matching between task requirements and organizational roles, although some systems additionally consider contextual factors such as qualifications or availability during execution (van der Aalst & van Hee, 2004; Vanderfeesten & Grefen, 2015). From a broader systems perspective, WFMS can be understood as a class of Process-Aware Information Systems (PAIS). PAIS are software systems that manage and execute operational business processes based on explicitly defined process models, involving human actors, applications, and information sources. They coordinate the execution of activities, control the flow of information, and support the allocation of tasks to resources during process execution. WFMS and Business Process Management Systems (BPMS) represent common implementations of PAIS (Cabanillas, 2016). Beyond workflow execution, resource modeling represents an additional dimension for understanding how tasks are allocated and performed in practice.

3.4 The Resource Perspective in WFMS

The resource perspective explains how resources are represented and how work items are allocated during workflow execution. It provides the conceptual basis for understanding delegation, substitution, and escalation mechanisms.

3.4.1 Foundations of Resource Management

Workflow research has traditionally prioritized control-flow and data perspectives, while organizational resource management received comparatively less attention. Role-based assignment structures remain a common approach to resource modeling in WFMS.

Since resources are typically limited and costly, ineffective allocation decisions may result in bottlenecks, increased waiting times, and higher operational costs (Arias et al., 2018; Djedović et al., 2018; Wibisono et al., 2016).

Resource management in business processes can be structured into three operations: resource assignment, resource allocation, and resource analysis (Cabanillas, 2016). Resource assignment defines the conditions under which resources are eligible to perform specific tasks, typically based on organizational models and authorization constraints. These include Separation of Duties (SoD), which prevents conflicting tasks from being performed by the same actor, and Binding of Duties (BoD), which requires certain activities to be executed by the same actor. Resource allocation determines which concrete resource is selected to execute a specific work item at runtime, often based on criteria such as availability or workload. Resource analysis evaluates resource utilization across different phases of the BPM lifecycle and supports the identification of execution-related issues during process execution. Related concepts such as task assignment and role resolution reflect the relationship between design-time specifications and runtime decisions (Vanderfeesten, 2024).

3.4.2 Core Aspects of the Resource Perspective

Within PAIS, the resource perspective represents a distinct analytical dimension that complements control-flow and data perspectives. It encompasses both the representation of organizational resources and the mechanisms governing the distribution and authorization of work during process execution (Stroppi et al., 2015).

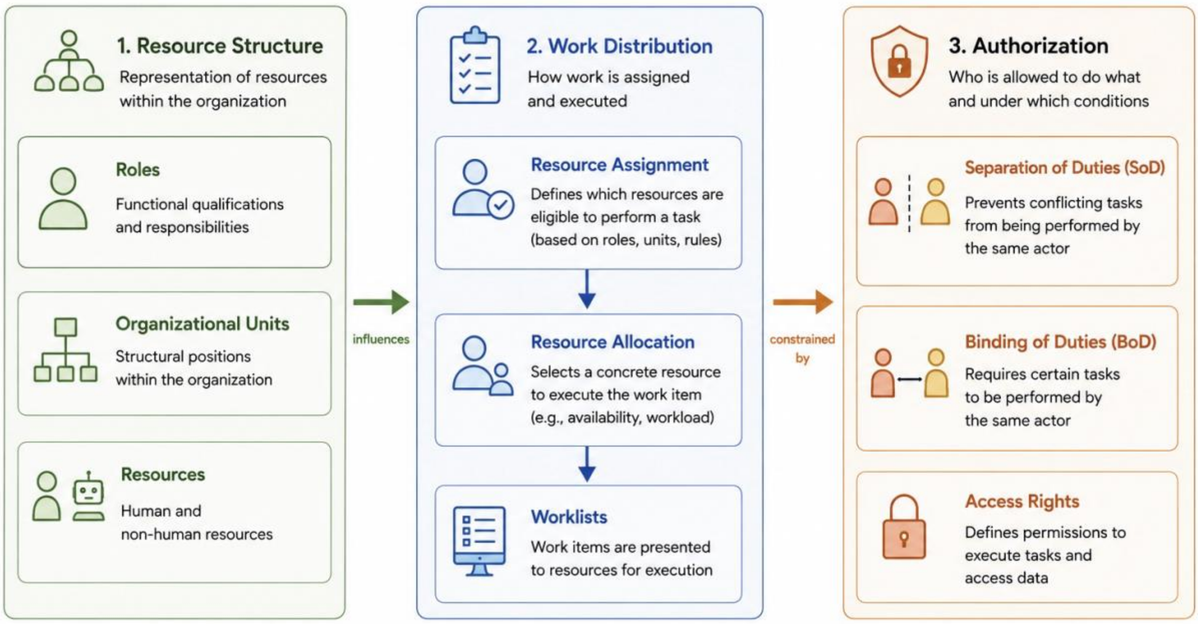

Figure 4: Overview of the Resource Perspective in WFMS
Source: Own illustration inspired by Stroppi et al., (2015) and van der Aalst & van Hee, (2001)

This perspective can be structured into three interrelated aspects: resource structure, work distribution, and authorization. The resource structure describes how resources are represented and organized within an organizational model. Resources may include both human and non-human entities; however, in most business contexts, the focus lies on human resources (van der Aalst & van Hee, 2004). These are typically classified through concepts such as roles or organizational units. This enables tasks to be assigned to groups of eligible resources rather than to specific individuals, thereby increasing execution flexibility. Roles classify resources according to functional qualifications, whereas organizational units reflect structural positions within an organization (van der Aalst & van Hee, 2001).

The work distribution aspect concerns how work items are allocated and executed by resources. Allocation decisions are made at work-item level and are constrained by organizational and security requirements such as SoD and BoD. The authorization aspect defines which operations a resource is permitted to perform within a workflow. It ensures that only authorized resources can execute specific tasks or access certain information. Authorization may be predefined or evaluated dynamically during execution, depending on workflow context (Stroppi et al., 2015). Authorization mechanisms are often closely integrated with organizational models and may rely on centralized infrastructures to ensure consistency across systems (Lawall et al., 2015; Schaller & Fleischmann, 2025).

3.4.3 Advanced Considerations for Resource Allocation

Beyond formal role-based assignment, advanced allocation approaches incorporate actor-specific and contextual characteristics into resource selection. Relevant factors include qualifications, experience, workload, availability, and resource capabilities. In addition, contextual and case-specific characteristics, such as process state, task complexity, or execution urgency, may further influence allocation decisions. Resource preferences may also play a role, including preferred tasks, collaboration partners, or working contexts. Some advanced allocation approaches additionally incorporate resource preferences into runtime decision-making (Bidar et al., 2019). Advanced approaches therefore complement role-based assignment with contextual and actor-specific information (Vanderfeesten, 2024).

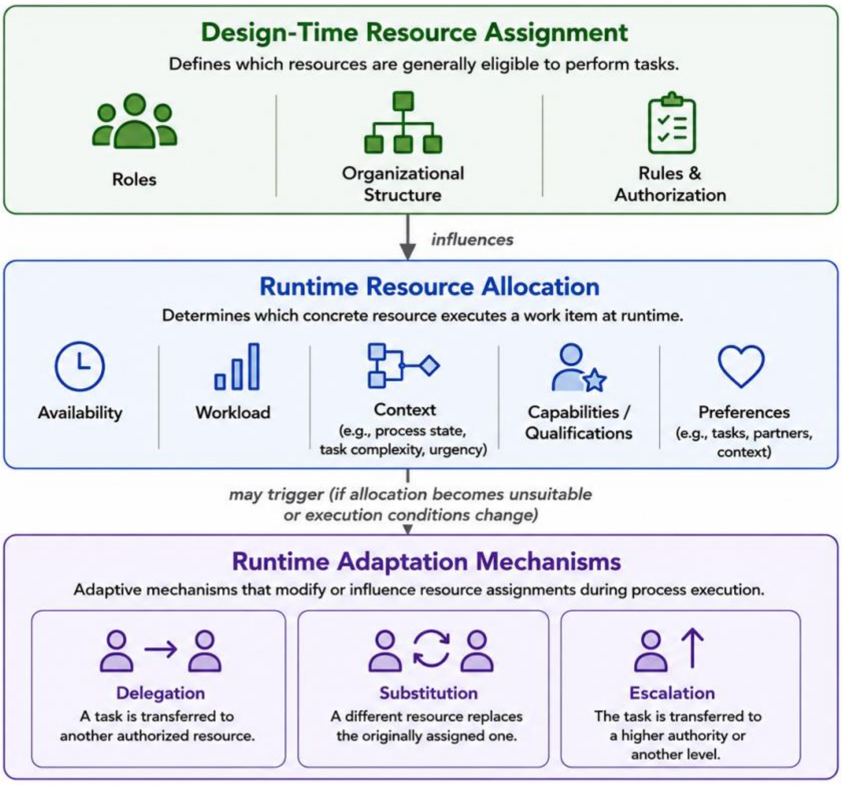

Figure 5: Conceptual relationship between resource allocation and runtime task reassignment mechanisms in WFMS

Source: Own illustration based on Cabanillas (2016), Stroppi et al. (2015), Vanderfeesten (2024), and Zeng & Zhao (2005).

Beyond allocation mechanisms, WFMS additionally employ runtime adaptation mechanisms such as delegation, substitution, and escalation to address changing execution conditions.

3.5 Runtime Adaptation in WFMS

Runtime adaptation refers to the ability of WFMS to respond to changing execution conditions during process enactment. While design-time resource assignment defines which resources are generally eligible to perform tasks, runtime conditions such as workload fluctuations, resource unavailability, or deadline violations may require adjustments to predefined assignments. In such situations, WFMS rely on runtime adaptation mechanisms that modify or influence work-item allocation during process execution. Delegation, substitution, and escalation represent key mechanisms through which workflow systems support adaptive task execution and maintain process continuity under changing operational conditions.

3.5.1 Static vs Runtime Role Resolution

Resource allocation in WFMS reflects a trade-off between formalization and runtime flexibility (Antunes & Murão, 2011). This tension becomes particularly visible when workflow systems must balance predefined organizational structures with changing execution conditions. In traditional WFMS, resource allocation is often defined at design time based on static process information, such as roles or organizational structures. However, such design-time allocation mechanisms often struggle under changing runtime conditions (Delias et al., 2011). As a result, many systems rely on simple allocation strategies, such as round-robin assignment (alternating task assignment across available resources), random allocation, or shortest-queue selection, in which tasks are assigned to the resource with the lowest current workload.

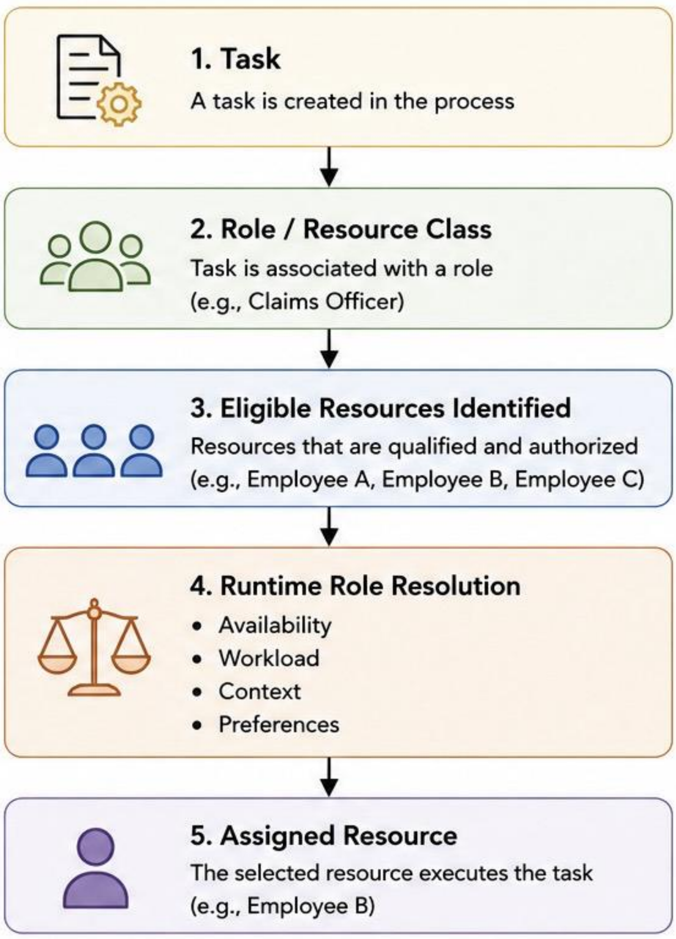

Figure 6: Simplified distinction between design-time role assignment and runtime role resolution in WFMS

Source: Own illustration based on Zeng & Zhao (2005) and Schaller & Fleischmann (2025)

A key distinction in this context is between the definition of eligible resources and the actual selection of a specific resource at runtime. Workflow models typically associate tasks with roles, thereby defining which resources are authorized to execute a task. However, this role-based specification does not determine which individual resource will perform a concrete work item. Zeng and Zhao (2005) conceptualize this runtime decision as role resolution, emphasizing its operational significance. They distinguish between strategic, offline decisions that assign resources to roles, and operational, online decisions that assign incoming work items to individual resources. Even when the same set of qualified resources is available, different role-resolution strategies can lead to significantly different process outcomes.

From a system perspective, this distinction is reflected in different approaches to subject–resource binding. For example, the Parallel Activity Specification Schema (PASS) meta-model differentiates between static mappings, in which process subjects are bound to predefined agents at design time, and dynamic mappings, in which the assignment is resolved during execution based on organizational information (Schaller & Fleischmann, 2025). Dynamic role resolution can be implemented through mechanisms such as Organizational Query Language (OQL). In this approach, queries over organizational data are evaluated at runtime to determine eligible resources for a given task instance. This enables more flexible and context-aware allocation decisions compared to static mappings.

This distinction between static and runtime allocation is particularly relevant for this thesis, as delegation, substitution, and escalation mechanisms operate on top of runtime decision-making. Rather than defining eligibility conditions, these mechanisms modify or influence the allocation of work items during process execution. For instance, these mechanisms may alter resource assignments or influence runtime allocation decisions during workflow execution. Runtime resource selection therefore plays a central role in adaptive workflow execution.

3.5.2 Delegation in WFMS

The term delegation is used inconsistently across academic disciplines. In management and organizational literature, delegation is primarily understood as a leadership practice in which authority, responsibility, or decision-making power is assigned from a superior to a subordinate to facilitate managerial effectiveness and organizational coordination. In contrast, workflow and access control research conceptualize delegation as a formalized and security-sensitive mechanism embedded in PAIS, enabling the reassignment of task execution authority, permissions, roles, or duties during workflow execution (Atluri et al., 2004;

Crampton & Khambhammettu, 2008). This thesis adopts the latter understanding and focuses exclusively on delegation in WFMS.

Delegation in WFMS refers to a controlled mechanism through which task-related authority is reassigned from one actor to another to enable workflow execution under changing organizational or runtime conditions. The actor initiating the delegation is commonly referred to as the delegator, while the actor receiving the delegated authority is referred to as the delegatee (Venter, 2003; Crampton & Khambhammettu, 2008). In workflow contexts, delegation is not limited to the reassignment of work itself, but typically also involves the authorization required to perform the delegated task, such as task task-specific permissions, role-based authorizations, or duties associated with a task (Crampton & Khambhammettu, 2008; Schefer-Wenzl & Strembeck, 2014).

Rather than determining an initial assignment, delegation modifies an existing authorization or task assignment by allowing another actor to execute a task on behalf of the delegator (Atluri et al., 2004; Crampton & Khambhammettu, 2008). Delegation may apply to different workflow elements, including permissions, roles, or task-related responsibilities (Schefer-Wenzl & Strembeck, 2014; Hsu & Wang, 2010).

In WFMS, delegation serves as an adaptive mechanism for maintaining workflow continuity when the originally assigned or authorized actor is unavailable, overloaded, lacks required resources or cannot complete a task within time constraints (Venter, 2003; Gaaloul & Charoy, 2009; Bakkali et al., 2013). However, delegation must be constrained by workflow security and process logic. Delegation decisions must remain consistent with authorization requirements and workflow constraints, including security policies such as SoD and BoD (Atluri & Warner, 2005; Schefer-Wenzl & Strembeck, 2014). For the purpose of this thesis, delegation is understood as a security-aware and context-dependent runtime mechanism through which task-related authority, permissions, or duties are reassigned from a delegator to a delegatee in order to preserve workflow continuity while maintaining consistency with organizational, authorization, and process-related constraints.

3.5.3 Substitution in WFMS

The term substitution is used inconsistently across workflow and process management literature. German workflow research explicitly conceptualizes substitution (*German: Stellvertretung or Stellvertreterregelung)* as a distinct organizational mechanism for temporarily replacing unavailable task performers *(Taskbearbeiter)*. In contrast, English-language workflow literature often addresses comparable situations through related concepts such as replacement, reassignment or resource substitution (Bauer & Laue, 2021). As a result, substitution currently lacks a standardized conceptualization in WFMS research.

This thesis adopts the understanding proposed by Bauer (2009), while also considering conceptually related approaches addressing comparable workflow challenges. In WFMS, substitution refers to a runtime mechanism through which originally assigned task performers are temporarily replaced when they become unavailable, while workflow continuity must nevertheless be maintained (Bauer, 2009). The substitute temporarily assumes task execution during the absence period of the originally assigned actor. In contrast to ordinary task allocation, substitution occurs after an initial assignment has already been established and is typically triggered by temporary actor unavailability rather than regular process execution.

Substitution primarily serves as a continuity mechanism within workflow execution. It is commonly triggered by circumstances such as illness, vacation, or other forms of temporary actor absence that may prevent the originally assigned performer from completing a task (Bauer, 2009). By enabling alternative actors to temporarily take over affected activities, substitution helps preserve workflow continuity and reduces the risk of process interruptions.

A defining characteristic of substitution in WFMS is its contextual dependency. Suitable substitutes may depend on factors such as the process type, task characteristics, project context, or application-specific case data (Bauer, 2009). Consequently, substitution is generally understood as a context-sensitive mechanism rather than a purely static reassignment of predefined replacement actors.

Bauer (2009) further highlights that substitution mechanisms may involve multi-level replacement structures, allowing additional substitute actors to be identified when initially designated substitutes are likewise unavailable. Furthermore, substitution mechanisms may differ regarding their activation and revocation behavior, for example concerning when substitute assignments become active and whether they are revoked once the original actor returns (Bauer, 2009). Although substitution and delegation both involve changes in actor assignment during workflow execution, they differ conceptually. Delegation is typically initiated intentionally to transfer task execution authority or responsibility to another actor, whereas substitution primarily addresses temporary actor unavailability in order to maintain process continuity (Bauer, 2009; Russell et al., 2005).

For the purpose of this thesis, substitution is understood as a context-dependent and organizationally constrained runtime mechanism through which unavailable regular task performers are temporarily replaced by alternative actors in order to preserve workflow continuity while respecting process-related and organizational requirements.

3.5.4 Escalation in WFMS

WFMS are typically designed under the assumption that process activities are completed within predefined execution conditions. Activities are often associated with deadlines, performance expectations, or other constraints defining how and when tasks should be completed. In practice, however, workflow execution may deviate from planned execution due to delays, unavailable resources, or changing runtime conditions (van der Aalst et al., 2007). To address such situations, WFMS employ escalation mechanisms.

In WFMS, escalation refers to a predefined intervention mechanism that is triggered when workflow execution deviates, or is expected to deviate, from predefined process constraints, particularly temporal requirements such as activity deadlines (Chan et al., 2009; Si et al., 2014). Escalation mechanisms aim to restore or maintain workflow execution when normal process progression becomes endangered and corrective intervention is required (Panagos & Rabinovich, 1996). Unlike unforeseen runtime disruptions, escalations represent anticipated situations for which responses can be specified in advance.

Escalations may therefore be understood as a specific category of workflow exceptions. While workflow exceptions generally refer to deviations from planned execution, escalation mechanisms concern predictable situations for which predefined intervention strategies can be embedded into workflow logic (van der Aalst et al., 2007). The concrete form of escalation depends on workflow and organizational requirements. Escalation mechanisms may involve notifying responsible actors, involving supervisors, reassigning work items, initiating corrective actions, or otherwise adapting workflow execution in order to restore process progress (Winkler et al., 2008; Panagos & Rabinovich, 1996).

Although delegation, substitution, and escalation all represent adaptive runtime mechanisms in WFMS, they differ regarding their primary trigger and objective. Delegation concerns the intentional transfer of execution authority or responsibility, substitution addresses the temporary replacement of unavailable actors, whereas escalation responds to situations in which workflow execution is endangered and corrective intervention becomes necessary (Bauer & Laue, 2021).

For the purpose of this thesis, escalation is understood as a predefined and context-sensitive workflow mechanism that is triggered when workflow execution deviates, or is expected to deviate, from predefined process constraints in order to restore process continuity through corrective intervention.

Dimension	Delegation	Substitution	Escalation
Primary trigger	Intentional reassignment	Actor unavailable	Endangered workflow execution
Primary objective	Transfer execution authority or responsibility	Maintain workflow continuity through temporary replacement	Restore workflow continuity through corrective intervention
Nature of mechanism	Authority transfer mechanism	Replacement mechanism	Intervention mechanism
Typical runtime condition	Overload, expertise, organizational policies	Illness, vacation, temporary absence	Delays, deadline risks, execution disruptions
Typical intervention	Delegate tasks, permissions, or duties	Assign substitute performer	Notify, reassign, adapt workflow execution

Table 6: Key Differences Between Delegation, Substitution, and Escalation in WFMS
Source: Own illustration based on Bauer and Laue (2021), Panagos and Rabinovich (1996), van der Aalst et al. (2007), Chan et al. (2009), and Si et al. (2014).

As summarized in Table 6, delegation, substitution, and escalation represent distinct but closely related mechanisms for adaptive workflow execution. While each mechanism addresses different runtime conditions and follows different operational objectives, all contribute to maintaining workflow continuity under changing execution circumstances. Understanding these distinctions provides an important conceptual foundation for the following literature analysis, which examines how current WFMS approaches model and implement these mechanisms in practice.

4 Conceptual Analysis of the Literature

The literature on workflow resource management primarily focuses on how tasks are assigned and adapted during process execution. While initial actor assignment constitutes the baseline of resource allocation, changing execution conditions often require additional mechanisms such as delegation, substitution, or escalation. This chapter examines these mechanisms and analyses how workflow research addresses task assignment and reassignment in WFMS.

4.1 Baseline: Static Role-Based Assignment

Existing workflow approaches commonly assign work items based on predefined organizational descriptors such as roles, positions, or business units (Russell et al., 2004).

Within this perspective, resource allocation is primarily organized through formal organizational structures that determine which actors are eligible to perform specific activities. Therefore, predefined assignment rules that allocate work items according to role membership or organizational responsibility are often relied on. In collaborative worklist environments, tasks may be offered to groups of eligible actors rather than assigned directly to specific individuals, leaving responsibility for task uptake distributed across multiple performers (Illibauer et al., 2016).

More recent research additionally considers expertise, experience, collaboration history, and execution-related conditions when assigning actors (Abdulhameed et al., 2016; Ihde et al., 2022; Vanderfeesten, 2024). Situations in which assignments must be adapted after execution has started are addressed through runtime adaptation mechanisms. The following sections therefore examine delegation, substitution, and escalation as approaches for handling changing execution conditions.

4.2 Delegation Mechanisms in WFMS Literature

Workflow research increasingly treats delegation as a mechanism for adapting task execution under changing operational conditions. Delegation enables runtime reassignment of already allocated work items. The reviewed literature differentiates delegation across several dimensions, including the delegated object (e.g., tasks, task instances, roles, or permissions), the initiating actor, the delegation target, and the organizational setting in which reassignment occurs (Kumar et al., 2002). The literature mainly differentiates delegation according to delegation scope, organizational level, and runtime behavior. Delegation therefore functions as an adaptive reassignment mechanism that enables workflows to respond to operational disruptions while preserving process performance (Gaaloul & Charoy, 2009; Schefer-Wenzl & Strembeck, 2014). In collaborative settings, reassignment may also extend beyond organizational boundaries when external actors participate in workflow execution (Bessai & Charoy, 2016).

At the same time, delegation extends beyond the simple redistribution of work, as reassignment directly affects workflow execution and authorization structures. Since delegation modifies existing assignment relations, inappropriate decisions may compromise workflow feasibility, violate authorization requirements, or create inconsistencies during execution. Delegation mechanisms must therefore remain compatible with workflow constraints and organizational policies to ensure that reassignment remains executable and compliant with process requirements (Crampton & Khambhammettu, 2008).

4.2.1 Forms and Scope of Delegation

The literature differentiates delegation according to the organizational level and scope at which reassignment takes place. A widely adopted distinction differentiates between user-to-user, role-to-role, and user-to-role delegation (Hsu & Wang, 2010). User-to-user delegation represents the most common form and involves transferring task execution responsibility directly to another actor. By contrast, role-to-role delegation reallocates responsibilities between organizational roles, whereas user-to-role delegation assigns work to a role rather than a specific individual, allowing WFMS to identify an appropriate performer dynamically.

Literature additionally differs regarding the object of delegation. Some approaches delegate permissions or access rights, whereas others focus on delegating workflow roles, task execution responsibilities, or specific task instances. The delegated object influences both the degree of control retained by the delegator and the extent to which workflow execution and authorization structures are affected (Schefer-Wenzl & Strembeck, 2014; Hsu & Wang, 2010).

Delegation also differs regarding execution scope. Hasebe and Mabuchi (2010) distinguish between abstract and concrete task delegation. Abstract delegation authorizes a delegatee to execute a task type across workflow instances, whereas concrete delegation applies only to a specific task instance during execution. Likewise, delegation may be temporary or permanent, depending on whether reassignment is restricted to a particular workflow instance or extends to future process executions (Schefer-Wenzl & Strembeck, 2014).

4.2.2 Constraints and Requirements of Delegation

The literature consistently emphasises that delegation must remain compatible with workflow security and process constraints. In particular, delegation mechanisms must preserve SoD and BoD requirements to ensure that reassignment does not violate organizational security policies or task dependencies (Gaaloul & Charoy, 2009; Schefer-Wenzl & Strembeck, 2014).

Beyond individual authorization constraints, delegation may also affect the overall executability of workflows. Crampton and Khambhammettu (2008) argue that delegation decisions must preserve workflow satisfiability, meaning that valid task assignments remain possible under the specified authorization model. This issue becomes particularly relevant in transfer delegation, where permissions are temporarily removed from the delegator and reassigned to another actor, potentially invalidating previously feasible task allocations.

Atluri et al. (2004) propose integrating delegation rules directly into authorization models to ensure consistency between reassignment decisions and security requirements. Atluri and

Warner (2005) further show that delegation decisions may depend on contextual conditions such as time restrictions, workload, task dependencies, role activation constraints, delegation chains, or revocation requirements.

Delegation decisions may additionally depend on workload, qualifications, availability, or organizational limits (Bessai & Charoy, 2016; Djedović et al., 2018). Likewise, Djedović et al. (2018) demonstrate that resource reallocation remains bounded by organizational limits, including minimum and maximum staffing requirements for individual activities. Delegation therefore reflects the interaction between authorization requirements, process logic, and operational feasibility.

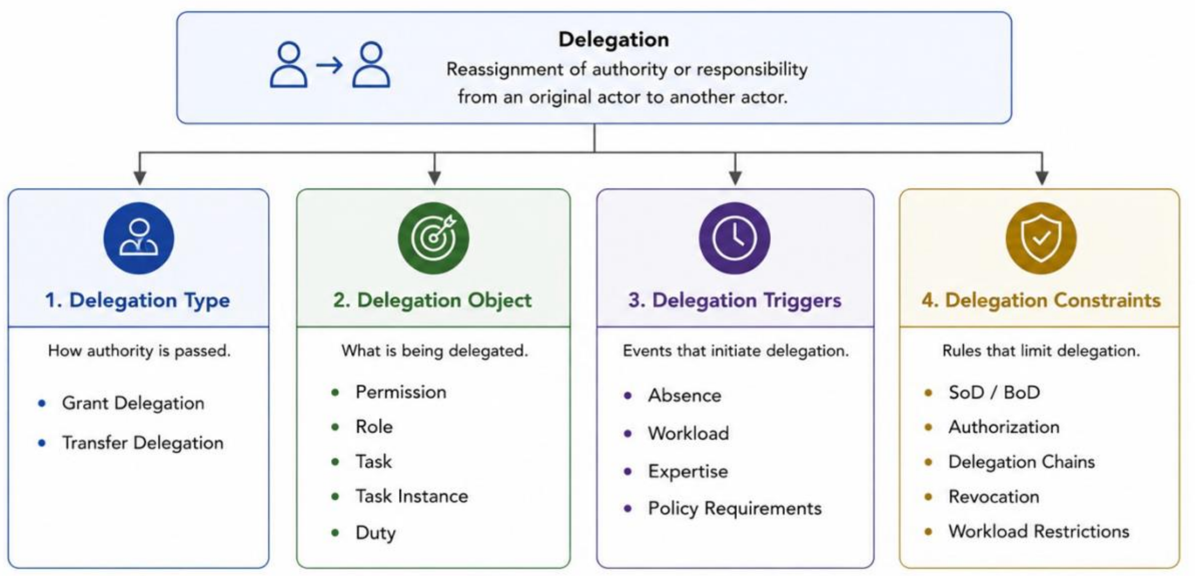

Figure 7: Analytical dimensions of delegation mechanisms in WFMS literature

Source: Own illustration based on Crampton & Khambhammettu (2008), Hsu & Wang (2010), Atluri et al. (2004), Gaaloul & Charoy (2009), and Schefer-Wenzl & Strembeck (2014).

As illustrated in Figure 7, delegation mechanisms in WFMS differ across several analytical dimensions, including delegation type, delegated object, triggering conditions, and applicable constraints. While these dimensions explain how delegation can be structured and governed, delegation also differs regarding when and how reassignment is enacted during workflow execution. The following section therefore examines runtime delegation as a dynamic mechanism for adaptive task reassignment.

4.2.3 Runtime Delegation and Dynamic Task Reassignment

Since concrete performers cannot always be determined during process modeling, WFMS require mechanisms that support reassignment during execution, for example in cases of staff absence, workload shifts, or unforeseen disruptions (Xue & Tang, 2009). Delegation may additionally be motivated by expertise considerations when alternative actors possess superior knowledge or qualifications for a task, as well as by organizational policies requiring temporary reassignment under specific execution conditions (Atluri et al., 2004; Gaaloul & Charoy, 2009). To enable such flexibility, delegation is often represented as an executable process embedded within workflow execution rather than as a one-time reassignment decision. Gaaloul et al. (2011), for example, describe delegation as an event-driven lifecycle consisting of delegation, acceptance, execution, validation, and revocation. This process-oriented perspective enables WFMS to monitor delegated activities and maintain control over reassignment decisions throughout execution.

Different forms of runtime delegation also vary in the degree of authority transfer. Beyond delegation scope, runtime approaches also differ regarding how authority is transferred between actors. In grant delegation, execution authority is shared between delegator and delegatee, allowing the original actor to retain supervisory control and revoke delegated permissions if necessary. In transfer delegation, by contrast, authority is temporarily removed from the delegator and fully assigned to the delegatee, making consistency with workflow constraints particularly important (Gaaloul et al., 2011; Crampton & Khambhammettu, 2008). Beyond user-initiated delegation, some approaches examine system-supported reassignment in which delegation decisions are partially automated. El Bakkali (2013), for example, proposes a method in which WFMS temporarily assign required permissions to alternative actors when no authorized performer is available. Such mechanisms aim to improve process resilience while preserving authorization requirements and execution continuity.

Runtime delegation therefore extends task reassignment beyond static allocation decisions. At the same time, its effectiveness depends on maintaining consistency between adaptive reassignment, authorization requirements, and process constraints.

4.3 Substitution Mechanisms in WFMS Literature

Compared to delegation, substitution receives considerably less explicit attention in workflow research and is often discussed indirectly through concepts such as resource reassignment, replacement performers, or dynamic allocation decisions. Existing contributions rarely examine substitution as a distinct workflow mechanism. Instead, research primarily focuses on how alternative performers can be identified when originally assigned actors become unavailable or unsuitable during process execution. Substitution is mainly discussed in relation to disruptions affecting resource availability or execution-related disruptions that require temporary or permanent reassignment of tasks or workflow cases (Hochmüller & Dobrovnik, 2005; Dulai & Werner-Stark, 2015). In contrast to static assignment approaches, substitution mechanisms aim to avoid workflow interruptions by enabling alternative actors to take over affected activities when regular performers are unavailable.

Approaches to substitution differ not only in how replacement actors are identified but also regarding their contextual dependency, activation logic, degree of automation, and organizational flexibility. Some approaches rely on predefined substitution rules and organizational relations, whereas others determine substitutes dynamically based on contextual information, resource availability, historical execution data, or predicted performance. This distinction structures the following analysis.

4.3.1 Predefined and Context-Dependent Substitution

Rule-based substitution mechanisms define in advance how alternative performers are selected when regular actors are unavailable. A key difference concerns how substitute actors are determined. Early approaches commonly rely on predefined substitution rules in which replacement performers are specified beforehand, either permanently or for specific situations. Such arrangements support different forms of actor unavailability and may additionally permit ad hoc substitution, allowing users to manually designate replacements when predefined arrangements are unavailable or insufficient (Hochmüller & Dobrovnik, 2005). Context also plays an important role in substitution decisions. Suitable replacements may depend not only on the unavailable actor but also on characteristics of the affected process, task, or workflow instance. Bauer (2009) therefore argues that substitution should not rely exclusively on predefined user-to-user mappings but instead incorporate contextual conditions that determine appropriate substitutes for specific execution situations. Relevant factors may include process type, task characteristics, or case-specific information.

Beyond runtime behavior, research also examines how substitution rules can be modeled efficiently and reused across organizational contexts. Bauer (2021) argues that substitution rules often depend on contextual factors such as the affected activity, the original actor, or organizational characteristics. To address this issue, Bauer proposes Target Person Rules (TPR), which define substitute performers through reusable references to the regular actor assignment. Instead of specifying fixed substitute users, replacement actors may be determined through organizational characteristics such as roles, groups, organizational units, or competencies associated with the originally assigned performer. This increases flexibility while reducing the need for activity-specific substitute definitions. To reduce the risk of workflow interruption, substitution mechanisms may also operate across multiple levels. If an initially designated substitute is unavailable, WFMS may identify additional replacement actors through chained substitution rules. Such multi-stage arrangements increase organizational robustness in situations involving multiple absences and reduce the risk of workflow interruption (Bauer, 2009).

Another important distinction concerns substitution activation and revocation. Bauer (2009) emphasizes that substitution mechanisms may differ regarding when substitute assignments become active and how they are revoked. Some approaches activate substitution immediately when individual actors become unavailable, whereas others only trigger substitution when no regular performer remains available. Likewise, substitute assignments may be revoked differently once the original actor returns, depending on organizational requirements and substitution strategy.

4.3.2 Dynamic Runtime Substitution

While predefined substitution mechanisms provide predictability and organizational control, they may become insufficient in situations where suitable substitutes cannot be fully anticipated at design time. This limitation has motivated research on dynamic runtime substitution approaches. Bellaaj et al. (2017), for example, describe substitution as a corrective intervention responding to unavailable or unsuitable resources. Their approach extends conventional allocation logic by considering not only the functional suitability of resources but also the likelihood that a resource may negatively affect process execution with regard to time, cost, or quality.

Substitution is also increasingly treated as a performance-sensitive allocation problem. Wibisono et al. (2016) propose a human-centric allocation mechanism that predicts expected resource performance based on contextual variables such as workload, queue length, activity type, and time of day. By continuously incorporating historical execution data, WFMS can redirect tasks to actors with higher expected performance when circumstances change.

Lee et al. (2019) place stronger emphasis on similarities in work experience when identifying suitable substitutes. Their approach assumes that replacement performers should possess comparable task experience to the originally assigned actor. Using historical execution logs, substitute suitability is assessed through similarities in task execution behavior and work transfer patterns. To formalize this assessment, the authors introduce a Degree of Substitution, which quantifies the overlap between performers' work profiles.

Other approaches place greater emphasis on organizational and contextual information. Lawall et al. (2013), for example, describe a declarative selection procedure in which actor assignment is derived from organizational queries and contextual constraints. When no suitable actor is available, substitute performers are identified through predefined deputyship relations, enabling context-sensitive replacement decisions instead of relying solely on fixed substitute mappings. Not all approaches assume that substitution should occur immediately. Kunkler and Rinderle-Ma (2024) argue that, under uncertain resource availability, temporarily postponing task allocation may be preferable to assigning work to a poorly suited substitute. Their concept of dummy resources represents the deliberate temporary non-allocation of tasks until a more suitable performer becomes available.

Despite growing research interest in context-aware substitution, practical implementation remains limited. Bauer and Laue (2021) emphasize that many commercial Process Management Systems (PMS) still provide only limited support for sophisticated substitution requirements and often rely on simplified approaches such as fixed substitute assignments or manual reassignment by supervisors.

Dynamic runtime substitution expands the ability of WFMS to respond to changing execution conditions by enabling context-sensitive and performance-aware reassignment decisions. At the same time, effective substitution requires balancing availability, qualifications, contextual suitability, and process performance.

4.4 Escalation Mechanisms in WFMS Literature

In workflow research, escalation is primarily examined as a mechanism for addressing situations in which process execution becomes endangered. In contrast to unforeseen workflow disruptions, escalation is commonly conceptualized as a form of anticipated exception handling. Rather than responding to entirely unexpected events, escalation mechanisms address situations that are sufficiently predictable to allow predefined intervention strategies to be embedded into workflow execution logic (van der Aalst et al., 2007).

Escalation strategies are commonly differentiated according to the aspect of workflow execution being adapted. Existing approaches primarily target resource allocation, process execution, or information requirements, depending on the nature of the disruption and the available intervention options (van der Aalst et al., 2007). Some contributions additionally examine adaptive escalation behavior that incorporates monitoring information or historical execution data to improve intervention decisions (Winkler et al., 2008).

4.4.1 Escalation Triggers and Intervention Logic

Workflow research predominantly examines escalation in relation to temporal constraints, particularly situations in which activities are expected to miss predefined deadlines. Many approaches aim to intervene proactively before deadline violations occur (van der Aalst et al., 2007). Beyond temporal constraints, the literature additionally discusses resource- and performance-related escalation triggers. Temporal escalation commonly addresses missed or endangered deadlines, service-level violations, or delayed completion times. Other approaches emphasize resource-driven escalation situations resulting from excessive workloads, unavailable personnel, insufficient expertise, or technical disruptions such as machine failures (Bellaaj et al., 2017). Escalation may also emerge from performance-related execution problems, including inefficient runtime task assignment, workload imbalances, or process bottlenecks that increase the likelihood of deadline violations (Zeng & Zhao, 2005).

Chan et al. (2009), for example, describe early escalation as a proactive intervention approach in which WFMS estimate the likelihood of future deadline violations based on indicators such as queue length, processing times, or historical execution behavior. Corrective actions can therefore be initiated before workflow performance deteriorates. Escalation is typically embedded in monitoring and decision procedures that continuously evaluate workflow progression and determine appropriate responses when execution risks emerge. Van der Aalst et al. (2007) describe escalation management as a sequence of detecting emerging problems, selecting a suitable intervention, and implementing corrective measures.

Some approaches formalise escalation through dedicated escalation subprocesses. Si et al. (2014) propose triggering specialised subprocesses when workflow instances are expected to violate deadlines or have already exceeded them. Depending on the severity of disruption, such interventions may include renegotiating deadlines, compensating affected stakeholders, or terminating workflow instances when successful completion is no longer feasible.

4.4.2 Escalation Strategies Across Workflow Perspectives

Escalation strategies differ not only regarding what aspect of workflow execution is adapted but also in terms of the intervention form employed. Some approaches rely primarily on informational escalation, such as notifications or reminders directed at responsible actors. Others involve authority-based escalation mechanisms in which supervisors or higher

organizational levels intervene once normal task execution becomes endangered (Winkler et al., 2008).
A substantial part of the literature examines escalation through modifications to resource allocation. Such interventions seek to increase execution capacity when delays threaten process performance. Common approaches include assigning additional personnel, redeploying employees across organizational units, or broadening the set of actors eligible to execute delayed tasks (van der Aalst et al., 2007; Chan et al., 2009). Kumar et al. (2002), for example, argue that increasing task urgency may justify gradually relaxing assignment restrictions, allowing less specialised actors to perform activities when deadlines become critical.
Other escalation strategies intervene directly in process execution. Instead of increasing resource capacity, WFMS may adapt control-flow structures to reduce delays. Discussed approaches include selecting faster execution paths, skipping non-essential activities, or splitting tasks into smaller units that can be executed in parallel (Chan et al., 2009; Si et al., 2014). Prioritization represents another common intervention, whereby workflow instances at risk of missing deadlines are processed before lower-priority cases, although this may shift delays to other parts of the process (Si et al., 2014).
Escalation may also involve temporary adjustments to information requirements. When obtaining complete information would further delay processing, WFMS may relax data requirements or postpone data collection activities. Van der Aalst et al. (2007) describe such interventions as modifications to the data perspective, while Chan et al. (2009) and Si et al. (2014) discuss related mechanisms such as data degradation and deferred data gathering, both intended to accelerate early process stages.
Beyond escalation actions themselves, the literature also differs regarding how escalation targets are determined. Depending on the affected activity, original task performer, and process context, different actors may become responsible for intervention. Escalation may therefore involve supervisors, substitute performers, organizational units, or role-based escalation paths. Bauer and Laue (2021) emphasize the need for flexible and context-sensitive escalation rules, while Bauer (2021) argues that escalation logic should remain reusable across activities and organizational settings without losing contextual specificity. Escalation may additionally involve collaborative problem-solving, particularly when disruptions require coordination across organizational actors or hierarchical levels (Antunes & Murão, 2011).
The effectiveness of escalation often depends on context-sensitive intervention decisions. Winkler et al. (2008) argue that historical execution data can support escalation management

by enabling WFMS to estimate execution risks and adjust intervention thresholds based on previous performance patterns. Since escalation interventions often require additional organizational effort and may increase operational costs, the literature frequently treats escalation as a corrective mechanism that should ideally be minimized. Research therefore increasingly emphasizes proactive monitoring, predictive intervention, and adaptive runtime control in order to reduce escalation frequency and severity (Panagos & Rabinovich, 1996). Escalation therefore increasingly evolves toward context-sensitive intervention mechanisms aimed at maintaining workflow continuity while minimizing execution risks and process disruption.

4.5 Identified Limitations of Current WFMS Approaches

The preceding analysis reviewed different mechanisms through which WFMS allocate, reassign, and adapt task execution under changing conditions. Despite these developments, the literature also points to several limitations that affect the practical applicability and flexibility of current approaches.

4.5.1 Limitations of Static and Role-Based Resource Allocation

Across the reviewed literature, a central limitation of existing WFMS lies in their strong reliance on static and role-based resource allocation mechanisms. Task assignment is commonly determined through predefined organizational descriptors such as roles, positions, or departments, while differences between actors within the same role and changing execution conditions remain insufficiently considered (Vanderfeesten & Grefen, 2015; Cabanillas, 2016).

A recurring criticism concerns the limited consideration of actor-specific and contextual characteristics in allocation decisions. Existing approaches frequently neglect differences in expertise, experience, workload, previous performance, or developmental objectives among actors sharing the same formal role (Vanderfeesten & Grefen, 2015; Erasmus et al., 2018). Likewise, task allocation is often performed independently of case-specific characteristics or higher-level process objectives, reducing the ability of WFMS to adapt assignments to concrete execution situations (Ariaans, 2014). Resource allocation is therefore often conceptualized as a matching problem between task requirements and eligible resources, while broader process-sensitive considerations remain comparatively underrepresented.

The literature further shows that many WFMS insufficiently account for the dynamic nature of resource availability. Traditional assignment approaches commonly assume that authorized resources remain continuously available and suitable for execution. In practice, however, absence, workload fluctuations, turnover, or technical failures may substantially affect execution performance (Bellaaj et al., 2017). Static or group-based assignment mechanisms may therefore create bottlenecks, waiting times, or unclear responsibility structures, particularly when tasks remain unclaimed or multiple process instances compete for the same limited resources (Illibauer et al., 2016; Durán et al., 2019).

Existing WFMS also rely on a comparatively narrow set of allocation criteria. While current mechanisms primarily consider role-related and workload-based factors, the literature increasingly points to additional influences on allocation quality, including prior execution performance, collaboration history, social compatibility, and individual preferences (Arias et al., 2016; Abdulhameed et al., 2016; Bidar et al., 2019). Despite evidence of their relevance for execution efficiency and resource utilization, these factors remain only partially reflected in operational workflow systems.

In addition to these structural limitations, the literature also identifies technical shortcomings in current modeling and assignment mechanisms. BPMN-based resource allocation approaches, for example, provide only limited integration between process models and organizational structures, making complex assignment conditions difficult to express and maintain (Cabanillas et al., 2012). Likewise, workflow systems commonly distinguish between design-time assignment and runtime allocation without systematically prioritising eligible performers according to criteria such as workload, expertise, or cost (Cabanillas et al., 2013). Empirical findings indicate that performance-aware allocation strategies can substantially improve process performance compared to static assignment rules (Wibisono et al., 2016).

Taken together, the literature indicates that current workflow resource allocation approaches remain insufficiently equipped for dynamic and context-dependent execution environments. Although more sophisticated allocation approaches have been proposed, their integration into operational WFMS remains limited (Arias et al., 2018; Cabanillas, 2016). Beyond these structural limitations, adaptive workflow execution also creates tensions between organizational flexibility and the need to preserve control and security requirements.

4.5.2 Tensions Between Flexibility, Control and Security

The reviewed literature highlights a fundamental tension between the need for adaptive workflow execution and the requirement to maintain control, authorization consistency, and process integrity. Mechanisms such as delegation, substitution, and escalation are intended to increase flexibility under changing execution conditions; however, greater adaptability may simultaneously create risks for workflow security and execution correctness.

This tension becomes particularly visible in delegation mechanisms. Since delegation modifies existing authorization relations, reassignment decisions may conflict with workflow constraints or unintentionally expand access rights. Hasebe and Mabuchi (2010) note that capability-based delegation increases organizational flexibility by enabling dynamic permission transfer without direct administrative intervention. At the same time, insufficiently restricted capability propagation may create security risks if delegated permissions spread beyond intended boundaries. To mitigate such risks, delegation mechanisms often require additional safeguards, including temporal restrictions or limitations on delegation scope.

A related challenge concerns the balance between workflow flexibility and authorization enforcement. Strict access control policies may prevent workflow completion when no authorized performer is available, whereas overly permissive delegation mechanisms may compromise security requirements and enable inappropriate task execution (El Bakkali, 2013). Maintaining workflow continuity therefore requires mechanisms that preserve both process executability and compliance with organizational security policies. Comparable tensions may also emerge in substitution and escalation mechanisms. Context-sensitive substitution may improve workflow continuity but can complicate authorization enforcement and organizational accountability when substitute performers differ substantially from originally assigned actors. Likewise, escalation mechanisms may increase process responsiveness while simultaneously introducing additional organizational effort, intervention costs, or execution complexity.

Beyond authorization requirements, adaptive reassignment must remain compatible with organizational responsibilities and workflow constraints. While increased flexibility may improve workflow continuity, it also makes the consistent enforcement of access control and execution rules more challenging.

The literature therefore indicates that flexibility alone does not guarantee improved workflow performance. Rather, the effectiveness of adaptive mechanisms depends on balancing flexibility with authorization consistency and process control.

4.5.3 Limited Support for Adaptive Runtime Intervention

Despite growing research on runtime reassignment and escalation, practical WFMS still struggle to support flexible runtime intervention. Existing systems continue to rely predominantly on predefined assignment structures and manual adjustments, reducing their ability to respond effectively to changing execution conditions (Cabanillas, 2016).

One limitation concerns the restricted ability of WFMS to adapt task allocation once execution has already begun. Many resource allocation approaches focus on identifying suitable performers at the moment of assignment but provide only limited support for modifying assignments when disruptions occur (Arias et al., 2018). This limitation becomes particularly visible in substitution mechanisms, which are frequently embedded in broader discussions of reassignment and resource allocation, resulting in comparatively limited formalization. As a result, substitution remains less formally specified than delegation or escalation and is often handled through predefined replacement rules or ad hoc reassignment instead of integrated runtime support. Consequently, workflow systems often struggle to respond adequately to staff absence, workload fluctuations, resource failures, or changing process conditions. Bellaaj et al. (2017) further argue that many allocation approaches insufficiently consider situations in which resources themselves become sources of disruption, including sickness, turnover, or technical breakdowns.

Historical execution data and predictive allocation models may improve runtime decisions by identifying resources with favourable performance characteristics regarding execution time, cost, or reliability. Empirical findings indicate that such approaches can reduce waiting times and overall process duration compared to static allocation rules (Dulai & Werner-Stark, 2015; Wibisono et al., 2016). Despite these benefits, predictive and performance-aware approaches remain insufficiently integrated into operational workflow environments.

Limitations are particularly visible in escalation management. Contemporary WFMS often provide only basic escalation support, requiring organizations to implement escalation procedures through modeling workarounds that increase process complexity and reduce maintainability (van der Aalst et al., 2007). In many practical settings, escalation remains limited to notification mechanisms that inform actors about emerging problems without automatically adapting workflow execution or initiating corrective interventions (Winkler et al., 2008). Adaptive interventions additionally involve difficult trade-offs that complicate implementation. Escalation strategies intended to reduce delays may simultaneously increase operational costs, resource consumption, or execution complexity. Their effectiveness further depends on process context, resource utilization, and the location of bottlenecks within the workflow (Chan et al., 2009; Si et al., 2014). In highly unpredictable situations, predefined exception handling may become insufficient altogether, requiring human intervention beyond formal workflow constraints (Antunes & Murão, 2011).

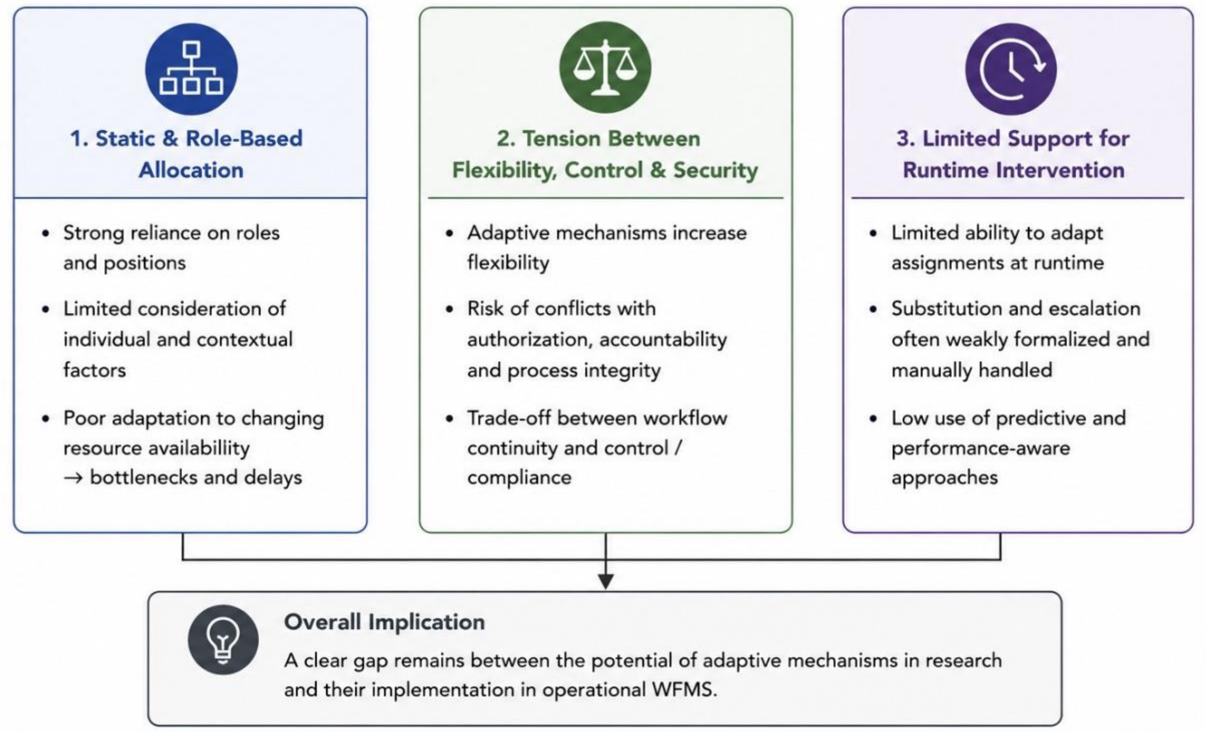

Figure 8: Main limitations of current WFMS approaches for adaptive workflow execution

Source: Own illustration based on Cabanillas (2016), Vanderfeesten & Grefen (2015), Arias et al. (2018), Bellaaj et al. (2017), Hasebe & Mabuchi (2010), El Bakkali (2013), van der Aalst et al. (2007), Winkler et al. (2008), Chan et al. (2009), and Si et al. (2014).

Figure 8 summarizes the main limitations identified in the literature regarding adaptive workflow execution in WFMS. The identified limitations concern structural constraints in resource allocation, tensions between flexibility and control, as well as limited support for adaptive runtime intervention. A persistent gap therefore remains between adaptive mechanisms proposed in research and their practical implementation in operational WFMS. While research increasingly emphasizes predictive and context-sensitive runtime intervention, operational systems continue to rely predominantly on predefined assignment structures and manual adaptation.

5 Synthesis of Findings and Research Gaps

This chapter synthesizes the findings of the literature analysis by identifying broader developments, recurring tensions, and unresolved challenges in workflow resource management. Rather than discussing delegation, substitution, and escalation separately, the focus lies on how these mechanisms collectively reflect changing approaches to runtime adaptation in WFMS.

5.1 From Static Allocation to Runtime Adaptation

A central development across the literature concerns the changing understanding of resource allocation in WFMS. Earlier approaches predominantly treat task assignment as a static design-time activity based on predefined organizational structures such as roles, positions, or organizational units. Under this logic, actor assignment primarily serves authorization and process consistency, assuming that execution conditions remain sufficiently stable over time (Russell et al., 2004). The literature has begun to question static role-based assignment and advocates more adaptive allocation approaches under changing execution conditions. As a result, delegation, substitution, and escalation increasingly function as mechanisms for maintaining process continuity under uncertainty rather than merely correcting isolated execution failures (Durán et al., 2019; Illibauer et al., 2016).

When considered together, these developments indicate a broader transition from predefined actor assignment towards runtime-sensitive resource management. Resource allocation is progressively treated as a continuous decision process in which task assignment must adapt to contextual information, including workload, urgency, historical execution performance, and case-specific characteristics (Vanderfeesten, 2024; Wibisono et al., 2016). This shift reflects a broader understanding of workflow execution as a dynamic coordination problem rather than a purely organizational assignment task.

5.2 Increasing Contextualization of Allocation Decisions

A second recurring development concerns the increasing complexity of allocation criteria used to determine suitable performers. Traditional approaches primarily rely on formal authorization structures, assuming that actors within the same role can be treated as largely interchangeable. Across the literature, however, this assumption is progressively questioned.

More recent contributions consistently expand allocation decisions beyond role membership by incorporating contextual, behavioral, and performance-related information. Factors such as workload, task urgency, qualifications, collaboration history, prior execution performance, and case-specific conditions are increasingly considered relevant for determining suitable task performers (Shen et al., 2003; Abdulhameed et al., 2016; Vanderfeesten, 2024).

At the same time, workflow execution data increasingly serves as a basis for improving allocation quality. Historical event logs and behavioral patterns are no longer used solely for process monitoring but increasingly support predictive and performance-aware assignment decisions (Ly et al., 2005; Huang et al., 2011; Bidar et al., 2019). Considered collectively, these developments suggest a gradual shift from authorization-oriented allocation towards context-sensitive and data-informed decision-making in WFMS.

5.3 Persistent Tensions in Adaptive Workflow Execution

Although the literature increasingly advocates adaptive runtime intervention, flexibility consistently emerges as a source of competing demands rather than as an unconditional improvement. Across delegation, substitution, and escalation, adaptive reassignment repeatedly involves trade-offs between competing process objectives.

One recurring tension concerns flexibility and organizational control. While runtime reassignment may improve workflow continuity and responsiveness to disruptions, increased adaptability simultaneously complicates authorization enforcement, workflow satisfiability, and compliance with organizational constraints (Crampton & Khambhammettu, 2008; Hasebe & Mabuchi, 2010; El Bakkali, 2013). A similar tension emerges between process performance and execution quality. Faster task completion may be achieved through broader assignment flexibility or escalation measures, yet this may also increase costs or reduce execution quality if less suitable actors perform tasks under time pressure (Kumar et al., 2002; Si et al., 2014).

Finally, the literature repeatedly points to limits of formalized adaptation. Although WFMS increasingly support runtime intervention, highly uncertain execution environments continue to require human judgement, particularly when disruptions exceed predefined workflow logic or cannot be anticipated through modeling alone (Antunes & Murão, 2011). Adaptive workflow management therefore appears to depend not only on technical flexibility but also on the ability to balance competing objectives under uncertainty.

5.4 Remaining Research Gaps

The findings additionally reveal several remaining research gaps concerning the conceptualization, integration, and operationalization of adaptive workflow mechanisms in WFMS. First, substitution remains comparatively underdeveloped as an independent workflow mechanism. While delegation and escalation are often supported through formal modeling constructs and explicit runtime procedures, substitution is frequently embedded in broader discussions of resource allocation or exception handling (Bauer, 2009; Lee et al., 2019). As a result, the literature provides only limited systematic guidance regarding how substitute actors should be identified dynamically under changing execution conditions.

Second, the literature predominantly examines delegation, substitution, and escalation as isolated mechanisms despite their shared objective of maintaining workflow continuity under runtime disruptions. Although these mechanisms frequently rely on overlapping contextual information such as workload, urgency, or resource availability, limited research addresses how they should interact or be prioritised during workflow execution. Consequently, integrated runtime adaptation strategies remain insufficiently developed.

Third, a gap persists between conceptual advances in adaptive workflow management and their systematic operationalization in WFMS. Although research increasingly proposes predictive allocation, context-sensitive reassignment, and performance-aware intervention strategies, limited evidence exists regarding how these approaches can be implemented and maintained within operational workflow environments characterised by organizational complexity and continuously changing execution conditions.

Taken together, the findings indicate a growing shift towards adaptive workflow management while simultaneously revealing unresolved challenges concerning conceptual maturity, mechanism integration, and practical operationalization. These observations provide the basis for the discussion of the research questions in the following chapter.

6 Discussion

Building on the synthesis presented in Chapter 5, this chapter interprets the findings in relation to the research questions and examines what they imply for the modeling and implementation of adaptive workflow mechanisms in WFMS. Rather than suggesting a single universally applicable approach, the findings indicate that effective workflow adaptation depends on balancing organizational control with runtime flexibility. The following sections discuss how this balance affects substitute identification, delegation, substitution, escalation, and the limitations of current WFMS approaches.

6.1 Substitute Identification in WFMS

The first research question asked: *How can substitute actors be identified and selected at design time and runtime?* Substitute identification can be distinguished between predefined and runtime-sensitive approaches. At design time, substitution is primarily defined through organizational structures, predefined substitute relations, and formal replacement rules that determine alternative performers before workflow execution begins (Hochmüller & Dobrovnik, 2005; Bauer, 2009). Such approaches provide organizational clarity and support predictable workflow execution, particularly in situations involving planned absences or recurring organizational conditions. However, the literature consistently indicates that predefined substitute mappings alone provide only limited flexibility under dynamic execution conditions. Fixed replacement relations may become ineffective when substitute actors themselves are unavailable, overloaded, or insufficiently qualified for specific workflow situations (Bauer, 2009).

As a result, more recent research increasingly advocates runtime-sensitive substitute selection. Instead of relying exclusively on predefined substitute lists, WFMS increasingly incorporate contextual and actor-specific information to determine suitable replacements during execution. Across the reviewed literature, runtime substitute identification is frequently based on capability matching and case matching, in which actors are selected according to task requirements, process context, expertise, workload, or historical execution performance (Vanderfeesten & Grefen, 2015; Bellaaj et al., 2017).

Historical execution data additionally plays an increasingly important role in substitute selection. Lee et al. (2019), for example, identify substitute actors based on similarities in work experience and execution behavior derived from event logs. Their findings suggest that substitute actors with comparable work profiles to unavailable performers may improve workflow continuity and reduce execution delays compared to static role-based assignment.

Substitute identification should therefore not rely exclusively on either predefined organizational substitution or purely dynamic runtime allocation. Instead, effective WFMS appear to require hybrid approaches that combine organizational transparency with runtime adaptability. Purely predefined substitution mechanisms improve predictability and governance but frequently fail under changing runtime conditions, whereas fully dynamic reassignment risks reducing transparency and organizational control.

6.2 Modeling Constructs for Delegation and Substitution

The second research question asked: *What modeling constructs and allocation rules can be applied to define delegation and substitution?* The findings show that delegation and substitution require modeling approaches that extend beyond static role-based assignment because runtime reassignment directly affects organizational responsibilities, workflow execution, and authorization structures.

For delegation, the literature primarily emphasises authorization-sensitive modeling constructs. Since delegation modifies existing task assignment relations, delegation rules must remain compatible with workflow authorization requirements, including separation of duties (SoD), binding of duties (BoD), and workflow satisfiability constraints (Atluri et al., 2004; Crampton & Khambhammettu, 2008). Delegation therefore cannot be treated as unrestricted reassignment but instead requires formally defined rules governing permission transfer, delegation scope, and revocation rights (Gaaloul et al., 2011).

For substitution, the literature increasingly points to limitations of conventional role-based resource models. Vanderfeesten and Grefen (2015) argue that traditional two-dimensional role models insufficiently represent relevant actor differences and contextual conditions required for selecting suitable substitutes. To address these limitations, the authors propose extended resource models that incorporate capability matching and case matching, thereby allowing WFMS to evaluate substitute suitability based on actor characteristics, process requirements, and execution context. Rule-oriented approaches additionally propose reusable substitution and escalation rules. Bauer (2021) introduces Target Person Rules (TPR), which allow replacement actors to be identified by referencing existing actor assignments and organizational relations rather than relying solely on predefined substitute lists. Taken together, the literature suggests that effective delegation and substitution require modeling approaches that combine formal authorization structures with context-sensitive allocation rules. Runtime reassignment therefore depends on integrating organizational constraints with actor-specific and execution-sensitive information.

6.3 Escalation Responses to Deadline Violations and Resource Unavailability

The third research question asked: *How should workflow systems react to deadline violations and resource unavailability through escalation mechanisms?* Across the literature, escalation is primarily treated as an adaptive runtime intervention intended to preserve workflow continuity and prevent process deterioration under emerging execution risks. Rather than responding only after disruptions materialise, some approaches propose predictive escalation strategies capable of identifying workflow risks before deadlines are violated (van der Aalst et al., 2007; Chan et al., 2009). Historical execution patterns, workload indicators, queue lengths, and resource utilization are increasingly used to anticipate workflow disruptions and initiate interventions proactively (Winkler et al., 2008).

Across the reviewed literature, escalation mechanisms primarily involve two forms of intervention. First, WFMS may adapt resource allocation by assigning additional personnel, broadening the set of eligible performers, or temporarily relaxing assignment restrictions when execution delays become critical (Kumar et al., 2002; Chan et al., 2009). Second, escalation may involve adapting workflow execution itself through modified control-flow structures, task parallelisation, or skipping non-essential process activities to reduce delays (Si et al., 2014). At the same time, the findings of this review suggest important trade-offs associated with escalation strategies. While escalation may improve execution speed and workflow continuity, interventions frequently involve increased resource demand, operational costs, or execution complexity (Chan et al., 2009; Si et al., 2014). Effective escalation therefore appears to depend not on fixed intervention rules but on context-sensitive decision-making that accounts for workflow criticality, resource constraints, and process priorities. Taken together, the literature suggests that WFMS should not rely solely on reactive notifications or fixed escalation paths. Instead, escalation should combine monitoring, reassignment, and context-sensitive intervention logic, especially when deadline violations or resource unavailability threaten process continuity.

6.4 Implications of Current WFMS Approaches

The fourth research question asked: *What limitations exist in current approaches?* The analysis indicates that current WFMS are constrained primarily by limited adaptability rather than by the absence of delegation, substitution, or escalation mechanisms. Although workflow systems increasingly support reassignment and intervention procedures, these mechanisms remain strongly dependent on predefined organizational structures and manually specified rules.

A key limitation concerns the dominance of role-based allocation logic. Existing WFMS largely prioritise authorization consistency and organizational stability, while contextual execution conditions remain secondary (Vanderfeesten & Grefen, 2015; Cabanillas, 2016). As a consequence, allocation decisions often fail to account sufficiently for changing workload conditions, resource availability, or case-specific requirements. This reduces the suitability of static assignment structures in dynamic execution environments.

A second limitation concerns the fragmented treatment of adaptive mechanisms. Delegation, substitution, and escalation are predominantly modeled independently despite responding to closely related execution problems. Resource shortages, deadline risks, and workload imbalances frequently require combined intervention strategies rather than isolated reassignment procedures. However, the reviewed literature provides only limited guidance regarding the interaction and prioritization of these mechanisms during workflow execution.

Finally, the findings reveal a discrepancy between conceptual advances and practical implementation. Research increasingly proposes predictive allocation models, context-sensitive reassignment, and adaptive escalation strategies, whereas operational WFMS continue to rely largely on predefined substitute relations, manual intervention, or notification-based escalation procedures (van der Aalst et al., 2007; Winkler et al., 2008; Bauer, 2009). Adaptive runtime intervention therefore remains only partially operationalised in practice. Current WFMS are therefore limited less by conceptual deficits than by restricted runtime sensitivity and insufficient integration of adaptive workflow mechanisms.

6.5 Practical Reflection on Current WFMS

Practical WFMS already provide mechanisms that correspond to delegation, substitution, and escalation, but these mechanisms are mainly implemented through predefined assignments, configurable rules, task lifecycle functions, substitute profiles, timers, and notifications. Camunda supports user task assignment through assignees, candidate users, candidate groups, task reassignment, delegation, due dates, and timer-based escalation paths (Camunda Documentation, n.d.). SAP Business Workflow provides substitution through personal or position-related substitutes, active and passive substitution, validity periods, and substitute profiles (SAP Help Portal, n.d.). These examples show that current WFMS provide important building blocks for workflow adaptation, but they do not appear to provide runtime-sensitive and context-aware decision logic as a standard capability.

The practical gap is therefore not the absence of delegation, substitution, or escalation mechanisms, but their limited integration into a unified adaptive decision logic. In both examples, adaptation remains largely rule-based, administratively configured, or dependent on explicit modeling decisions. This supports the finding that future WFMS should combine predefined organizational structures with runtime information such as resource availability, workload, task urgency, and capability profiles.

6.6 Answer to the Main Research Question

The main research question asked: *How should workflow management systems model and implement delegation and substitution mechanisms, and how should escalation scenarios arising from deadline violations or resource unavailability be handled?* Drawing on the findings across the four research questions, the results indicate that delegation and substitution mechanisms should be modeled and implemented in WfMS through a combination of predefined organizational structures and runtime-sensitive allocation logic. Design-time rules, substitute profiles, authorization constraints, and organizational roles provide transparency and control, while runtime information such as availability, workload, capabilities, task urgency, and case context is necessary to support adaptive reassignment. Escalation scenarios arising from deadline violations or resource unavailability should not be handled solely through notifications, reminder mechanisms, or fixed escalation paths, but through context-sensitive intervention logic that combines monitoring, reassignment, and process-specific escalation strategies. The findings therefore indicate that effective WFMS require integrated adaptation logic rather than isolated delegation, substitution, and escalation functions.

7 Conclusion and Outlook

This thesis investigated how delegation, substitution, and escalation mechanisms can support adaptive workflow execution in workflow management systems. The analysis showed that these mechanisms are not isolated technical functions, but part of a broader challenge in workflow resource management: maintaining process continuity when predefined actor assignments no longer fit changing runtime conditions.

The main finding is that effective WFMS cannot rely solely on static role-based assignment rules. While predefined organizational structures, roles, authorization constraints, and substitute profiles remain important for transparency and control, they are insufficient when actors become unavailable, workloads change, or deadlines are at risk. Under such conditions, workflow systems require runtime-sensitive mechanisms that consider availability, workload, capabilities, task context, and process urgency. Delegation requires particular attention to authorization and workflow constraints, since reassignment may affect permissions, SoD, and process executability. Substitution requires context-sensitive identification of replacement actors rather than simple fixed substitute lists. Escalation mechanisms should move beyond reminders or notifications and support corrective interventions such as reassignment, prioritization, or process adaptation when workflow continuity is endangered. The thesis also showed that current approaches remain limited by fragmentation. Delegation, substitution, and escalation are often discussed and implemented separately, although they respond to related runtime problems. In practice, systems such as Camunda and SAP Business Workflow provide relevant building blocks, but these mechanisms remain largely rule-based, manually configured, or dependent on explicit modeling decisions. The key challenge is therefore not the complete absence of such mechanisms, but their limited integration into a unified adaptive decision logic.

This thesis is limited by its literature-based and conceptual design. It does not evaluate a specific WFMS implementation empirically and does not test the proposed interpretation in an organizational setting. In addition, the included literature uses inconsistent terminology, especially regarding substitution, which required a concept-oriented synthesis across related terms such as reassignment, replacement, delegation, and resource allocation.

Future research should therefore examine how integrated adaptation logic can be implemented and evaluated in operational WFMS. Further studies could compare different WFMS platforms, analyse real workflow event logs, or investigate how runtime data such as workload, availability, and performance history can be used to support substitution and escalation decisions. Particular attention should be given to the usability and maintainability of such mechanisms, since adaptive workflow support must remain understandable for process designers and controllable for organizations.

Overall, the thesis concludes that delegation, substitution, and escalation should be understood as complementary mechanisms for adaptive workflow execution. Their effectiveness depends on combining organizational control with runtime flexibility, enabling WFMS to preserve process continuity without losing transparency, authorization consistency, or operational maintainability.

References

Aalst, W. van der, & Hee, K. M. van. (2004). *Workflow Management: Models, Methods, and Systems*. MIT Press.

Abdulhameed, N. M., M., I., Awad, A., & Ezat, E. (2018). A Resource Recommendation Approach based on Co-Working History. *International Journal of Advanced Computer Science and Applications*, *9*(7). https://doi.org/10.14569/IJACSA.2018.090734

Antunes, P., & Mourão, H. (2011). Resilient Business Process Management: Framework and services. *Expert Systems with Applications*, *38*(2), 1241–1254. https://doi.org/10.1016/j.eswa.2010.05.017

Ariaans, B. (2014.). *Task allocation within teams*

Arias, M., Gama, J. M., Sepúlveda, M., & Miranda, J. C. (2018). Human resource allocation or recommendation based on multi-factor criteria in on-demand and batch scenarios. *European J. of Industrial Engineering*, *12*(3), 364. https://doi.org/10.1504/EJIE.2018.092009

Arias, M., Munoz-Gama, J., & Sepúlveda, M. (2018). Towards a Taxonomy of Human Resource Allocation Criteria. In E. Teniente & M. Weidlich (Eds.), *Business Process Management Workshops* (Vol. 308, pp. 475–483). Springer International Publishing. https://doi.org/10.1007/978-3-319-74030-0_37

Atluri, V., Bertino, E., Ferrari, E., & Mazzoleni, P. (2004). Supporting Delegation in Secure Workflow Management Systems. In S. De Capitani Di Vimercati, I. Ray, & I. Ray (Eds.), *Data and Applications Security XVII* (Vol. 142, pp. 190–202). Springer US. https://doi.org/10.1007/1-4020-8070-0_14

Atluri, V., & Warner, J. (2005). *Supporting conditional delegation in secure workflow management systems*. 49–58. https://doi.org/10.1145/1063979.1063990

Bakkali, H. E. (n.d.). *Enhancing Workflow Systems Resiliency by Using Delegation and Priority Concepts*.

Bauer, T. (2021.). *Effiziente Modellierung von Eskalationen und Stellvertretungen in Geschäftsprozessen*.

Bauer, T. (2009). Stellvertreterregelung für Taskbearbeiter in prozessorientierten Applikationen. *Datenbank-Spektrum*, *9*(31), 40–51.

Bauer, T., & Laue, R. (2021). Assignment of Actors to Activities at Process-Oriented Applications: A Research Agenda. In E. Serral, J. Stirna, J. Ralyté, & J. Grabis (Eds.), *The Practice of Enterprise Modeling* (Vol. 432, pp. 281–291). Springer International Publishing. https://doi.org/10.1007/978-3-030-91279-6_20

Bellaaj, F., Sellami, M., Bhiri, S., & Maamar, Z. (2017). Obstacle-Aware Resource Allocation in Business Processes. In W. Abramowicz (Ed.), *Business Information Systems* (Vol. 288, pp. 207–219). Springer International Publishing. https://doi.org/10.1007/978-3-319-59336-4_15

Bessai, K., & Charoy, F. (2016). Business Process Tasks-Assignment and Resource Allocation in Crowdsourcing Context. *2016 IEEE 2nd International Conference on Collaboration and Internet Computing (CIC)*, 11–18. https://doi.org/10.1109/CIC.2016.016

Bidar, R., Ter Hofstede, A., Sindhgatta, R., & Ouyang, C. (2019). Preference-Based Resource and Task Allocation in Business Process Automation. In H. Panetto, C. Debruyne, M. Hepp, D. Lewis, C. A. Ardagna, & R. Meersman (Eds.), *On the Move to Meaningful Internet Systems: OTM 2019 Conferences* (Vol. 11877, pp. 404–421). Springer International Publishing. https://doi.org/10.1007/978-3-030-33246-4_26

Bollen, P., & Illibauer, C. (2016). Business Process Model Semantics in BPMN. In M. Felderer, F. Piazolo, W. Ortner, L. Brehm, & H.-J. Hof (Eds.), *Innovations in Enterprise*

Information Systems Management and Engineering (Vol. 245, pp. 31–45). Springer International Publishing. https://doi.org/10.1007/978-3-319-32799-0_3

Cabanillas, C. (2016). *Process- and Resource-Aware Information Systems*. 1–10. https://doi.org/10.1109/EDOC.2016.7579383

Cabanillas, C., García, J. M., Resinas, M., Ruiz, D., Mendling, J., & Ruiz-Cortés, A. (2013). Priority-Based Human Resource Allocation in Business Processes. In P. P. Maglio, M. Weske, J. Yang, & M. Fantinato (Eds.), *Service-Oriented Computing* (Vol. 6470, pp. 374–388). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-45005-1_26

Cabanillas, C., Resinas, M., & Ruiz-Cortés, A. (2012). RAL: A High-Level User-Oriented Resource Assignment Language for Business Processes. In F. Daniel, K. Barkaoui, & S. Dustdar (Eds.), *Business Process Management Workshops* (Vol. 99, pp. 50–61). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-28108-2_5

Chan, K.-L., Si, Y.-W., & Dumas, M. (2009). *Simulation-Based Evaluation of Workflow Escalation Strategies*. 75–82. https://doi.org/10.1109/ICEBE.2009.20

Camunda Services GmbH. (n.d.). User task assignment. Camunda Documentation. Retrieved May 20, 2026, from https://docs.camunda.org/manual/latest/reference/bpmn20/tasks/user-task/

Crampton, J., & Khambhammettu, H. (2008). *Delegation and satisfiability in workflow systems*. 31–40. https://doi.org/10.1145/1377836.1377842

Delias, P., Doulamis, A., Doulamis, N., & Matsatsinis, N. (2011). Optimizing Resource Conflicts in Workflow Management Systems. *IEEE Transactions on Knowledge and Data Engineering*, *23*(3), 417–432. https://doi.org/10.1109/TKDE.2010.113

Djedovic, A., Karabegovic, A., Avdagic, Z., & Omanovic, S. (2018). Innovative Approach in Modeling Business Processes with a Focus on Improving the Allocation of Human

Resources. *Mathematical Problems in Engineering*, *2018*, 1–14. https://doi.org/10.1155/2018/9838560

Djedovic, A., Zunic, E., Avdagic, Z., & Karabegovic, A. (2016). *Optimization of business processes by automatic reallocation of resources using the genetic algorithm*. 1–7. https://doi.org/10.1109/BIHTEL.2016.7775724

Domagala-Schmidt, Z., van der Vleuten, J., Traganos, K., Vanderfeesten, I., & Grefen, P. (2022). MPDesign: 20th International Conference on Business Process Management, BPM 2022. *BPM 2022 Best Dissertation Award, Doctoral Consortium, and Demonstration & Resources Track, CEUR Workshop Proceedings*, 142–146. https://www.scopus.com/pages/publications/85139422063

Dulai, T., & Werner-Stark, Á. (2015). *A Database-oriented Workflow Scheduler with Historical Data and Resource Substitution Possibilities:* 325–330. https://doi.org/10.5220/0005204803250330

Dumas, M., La Rosa, M., Mendling, J., & Reijers, H. A. (2018). *Fundamentals of Business Process Management*. Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-662-56509-4

Durach, C. F., Kembro, J., & Wieland, A. (2017). A New Paradigm for Systematic Literature Reviews in Supply Chain Management. *Journal of Supply Chain Management*, *53*(4), 67–85. https://doi.org/10.1111/jscm.12145

Durán, F., Rocha, C., & Salaün, G. (2019a). A rewriting logic approach to resource allocation analysis in business process models. *Science of Computer Programming*, *183*, 102303. https://doi.org/10.1016/j.scico.2019.102303

Durán, F., Rocha, C., & Salaün, G. (2019b). A rewriting logic approach to resource allocation analysis in business process models. *Science of Computer Programming*, *183*, 102303. https://doi.org/10.1016/j.scico.2019.102303

Durán, F., Rocha, C., & Salaün, G. (2021a). Resource provisioning strategies for BPMN processes: Specification and analysis using Maude. *Journal of Logical and Algebraic Methods in Programming*, *123*, 100711. https://doi.org/10.1016/j.jlamp.2021.100711

Durán, F., Rocha, C., & Salaün, G. (2021b). Resource provisioning strategies for BPMN processes: Specification and analysis using Maude. *Journal of Logical and Algebraic Methods in Programming*, *123*, 100711. https://doi.org/10.1016/j.jlamp.2021.100711

Erasmus, J., Vanderfeesten, I., Traganos, K., Jie-A-Looi, X., Kleingeld, A., & Grefen, P. (2018). A Method to Enable Ability-Based Human Resource Allocation in Business Process Management Systems. In R. A. Buchmann, D. Karagiannis, & M. Kirikova (Eds.), *The Practice of Enterprise Modeling* (Vol. 335, pp. 37–52). Springer International Publishing. https://doi.org/10.1007/978-3-030-02302-7_3

Gaaloul, K. (n.d.). *A Secure Framework for Dynamic Task Delegation in Workflow Management Systems*.

Gaaloul, K., & Charoy, F. (2009). Task Delegation Based Access Control Models for Workflow Systems. In C. Godart, N. Gronau, S. Sharma, & G. Canals (Eds.), *Software Services for e-Business and e-Society* (Vol. 305, pp. 400–414). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-04280-5_31

Gaaloul, K., Proper, H. A., Zahoor, E., Charoy, F., & Godart, C. (2011). A logical framework for reasoning about delegation policies in workflow management systems. *International Journal of Information and Computer Security*, *4*(4), 365. https://doi.org/10.1504/IJICS.2011.044825

Goel, K., Fehrer, T., Röglinger, M., & Wynn, M. T. (2023). Not Here, But There: Human Resource Allocation Patterns. In C. Di Francescomarino, A. Burattin, C. Janiesch, & S. Sadiq (Eds.), *Business Process Management* (Vol. 14159, pp. 377–394). Springer Nature Switzerland. https://doi.org/10.1007/978-3-031-41620-0_22

Grefen, P., & Vanderfeesten, I. (Eds.). (2024). *Handbook on Business Process Management and Digital Transformation*. Edward Elgar Publishing. https://doi.org/10.4337/9781802206098

Hasebe, K., & Mabuchi, M. (2010). *Capability-Role-Based Delegation in Workflow Systems*. 711–717. https://doi.org/10.1109/EUC.2010.112

Havur, G., Cabanillas, C., Mendling, J., & Polleres, A. (2016). Resource Allocation with Dependencies in Business Process Management Systems. In M. La Rosa, P. Loos, & O. Pastor (Eds.), *Business Process Management Forum* (Vol. 260, pp. 3–19). Springer International Publishing. https://doi.org/10.1007/978-3-319-45468-9_1

Hochmüller, E., & Dobrovnik, M. (n.d.). *Flexibility Issues in Workflow Management Systems*.

Hsu, H.-J., & Wang, F.-J. (2010). Delegation Pattern in Workflow System. *2010 IEEE 34th Annual Computer Software and Applications Conference Workshops*, 189–195. https://doi.org/10.1109/COMPSACW.2010.75

Huang, Z., Lu, X., & Duan, H. (2011). Mining association rules to support resource allocation in business process management. *Expert Systems with Applications*, *38*(8), 9483–9490. https://doi.org/10.1016/j.eswa.2011.01.146

Ihde, S., Pufahl, L., Völker, M., Goel, A., & Weske, M. (2022). A framework for modeling and executing task-Specific resource allocations in business processes. *Computing*, *104*(11), 2405–2429. https://doi.org/10.1007/s00607-022-01093-2

Kassner, L., Hirmer, P., Wieland, M., Steimle, F., Königsberger, J., & Mitschang, B. (2017). *The Social Factory: Connecting People, Machines and Data in Manufacturing for Context-Aware Exception Escalation*. Hawaii International Conference on System Sciences. https://doi.org/10.24251/HICSS.2017.202

Kumar, A., Dijkman, R., & Song, M. (2013). Optimal Resource Assignment in Workflows for Maximizing Cooperation. In F. Daniel, J. Wang, & B. Weber (Eds.), *Business Process*

Management (Vol. 8094, pp. 235–250). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-40176-3_20

Kumar, A., Van Der Aalst, W. M. P., & Verbeek, E. M. W. (2002). Dynamic Work Distribution in Workflow Management Systems: How to Balance Quality and Performance. *Journal of Management Information Systems*, *18*(3), 157–193. https://doi.org/10.1080/07421222.2002.11045693

Kunkler, M., & Rinderle-Ma, S. (2024). *Online Resource Allocation to Process Tasks Under Uncertain Resource Availabilities*. 137–144. https://doi.org/10.1109/ICPM63005.2024.10723280

Lawall, A., Schaller, T., & Reichelt, D. (2013). Integration of Dynamic Role Resolution within the S-BPM Approach. In H. Fischer & J. Schneeberger (Eds.), *S-BPM ONE - Running Processes* (Vol. 360, pp. 21–33). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-36754-0_2

Lawall, A., Schaller, T., & Reichelt, D. (2015). Role and Rights Management. In A. Fleischmann, W. Schmidt, & C. Stary (Eds.), *S-BPM in the Wild* (pp. 171–185). Springer International Publishing. https://doi.org/10.1007/978-3-319-17542-3_10

Lee, J., Lee, S., Kim, J., & Choi, I. (2019). Dynamic human resource selection for business process exceptions. *Knowledge and Process Management*, *26*(1), 23–31. https://doi.org/10.1002/kpm.1591

Levy, Y., & J. Ellis, T. (2006). A Systems Approach to Conduct an Effective Literature Review in Support of Information Systems Research. *Informing Science: The International Journal of an Emerging Transdiscipline*, *9*, 181–212. https://doi.org/10.28945/479

Liu, D.-R., Wu, M.-Y., & Lee, S.-T. (2004). Role-based authorizations for workflow systems in support of task-based separation of duty. *Journal of Systems and Software*, *73*(3), 375–387. https://doi.org/10.1016/S0164-1212(03)00175-4

Ly, L. T., Rinderle, S., Dadam, P., & Reichert, M. (2006). Mining Staff Assignment Rules from Event-Based Data. In C. J. Bussler & A. Haller (Eds.), *Business Process Management Workshops* (Vol. 3812, pp. 177–190). Springer Berlin Heidelberg. https://doi.org/10.1007/11678564_16

Oberdorf, F., Stein, N., & Flath, C. M. (2021). Analytics-enabled escalation management: System development and business value assessment. *Computers in Industry*, *131*, 103481. https://doi.org/10.1016/j.compind.2021.103481

Panagos, E., & Rabinovich, M. (1997). Escalations in workflow management systems. *Proceedings of the Workshop on on Databases Active and Real-Time - CIKM '96*, 25–28. https://doi.org/10.1145/352302.352309

Poss, L., & Schönig, S. (2025). Location-aware business process modeling and execution. *Software and Systems Modeling*, *24*(1), 37–67. https://doi.org/10.1007/s10270-024-01224-7

Pufahl, L., Ihde, S., Stiehle, F., Weske, M., & Weber, I. (2024). *Automatic Resource Allocation in Business Processes: A Systematic Literature Survey* (arXiv:2107.07264). arXiv. https://doi.org/10.48550/arXiv.2107.07264

Roy, A., Sural, S., Majumdar, A. K., Vaidya, J., & Atluri, V. (2017). On Optimal Employee Assignment in Constrained Role-Based Access Control Systems. *ACM Transactions on Management Information Systems*, *7*(4), 1–24. https://doi.org/10.1145/2996470

Russell, N. (2005). *WORKFLOW RESOURCE PATTERNS*.

SAP SE. Substitution in SAP Business Workflow. SAP Help Portal. Retrieved May 20, 2026, from https://help.sap.com/docs/SAP_FIORI/d2c296c4f32d4f2a9e3752f58d5ef222/e4fa99f64eae4c5a9b189d035ec8adc9.html

Santos, D. R. dos, & Ranise, S. (2017). *A Survey on Workflow Satisfiability, Resiliency, and Related Problems* (arXiv:1706.07205). https://doi.org/10.48550/arXiv.1706.07205

Schaller, T., & Fleischmann, A. (2026). Mapping of Agents to Subjects in S-BPM. In M. Elstermann & M. Lederer (Eds.), *Subject-Oriented Business Process Management. Process Customer Satisfaction and Process Excellence* (Vol. 2630, pp. 19–34). Springer Nature Switzerland. https://doi.org/10.1007/978-3-032-04944-5_2

Schefer-Wenzl, S., & Strembeck, M. (2014). Modeling Support for Role-Based Delegation in Process-Aware Information Systems. *Business & Information Systems Engineering*, *6*(4), 215–237. https://doi.org/10.1007/s12599-014-0343-3

Shen, M., Tzeng, G.-H., & Liu, D.-R. (2003). *Multi-criteria task assignment in workflow management systems*. https://ieeexplore.ieee.org/abstract/document/1174458

Si, Y.-W., Dumas, M., & Chan, K.-L. (2014). Evaluation of trade-offs between workflow escalation strategies. *Concurrent Engineering*, *22*(1), 77–88. https://doi.org/10.1177/1063293X13516965

Strembeck, M., & Mendling, J. (2011). Modeling process-related RBAC models with extended UML activity models. *Information and Software Technology*, *53*(5), 456–483. https://doi.org/10.1016/j.infsof.2010.11.015

Stroppi, L. J. R., Chiotti, O., & Villarreal, P. D. (2015). Defining the resource perspective in the development of processes-aware information systems. *Information and Software Technology*, *59*, 86–108. https://doi.org/10.1016/j.infsof.2014.10.006

Van Der Aalst, W. M. P., Rosemann, M., & Dumas, M. (2007). Deadline-based escalation in process-aware information systems. *Decision Support Systems*, *43*(2), 492–511. https://doi.org/10.1016/j.dss.2006.11.005

Vanderfeesten, I., & Grefen, P. (2015). Advanced Dynamic Role Resolution in Business Processes. In A. Persson & J. Stirna (Eds.), *Advanced Information Systems Engineering Workshops* (Vol. 215, pp. 87–93). Springer International Publishing. https://doi.org/10.1007/978-3-319-19243-7_8

Venter, K. (n.d.). *A MODEL FOR THE DYNAMIC DELEGATION OF AUTHORIZATION RIGHTS IN A SECURE WORKFLOW MANAGEMENT SYSTEM*.

Vom Brocke, J., Simons, A., Riemer, K., Niehaves, B., Plattfaut, R., & Cleven, A. (2015). Standing on the Shoulders of Giants: Challenges and Recommendations of Literature Search in Information Systems Research. *Communications of the Association for Information Systems*, *37*. https://doi.org/10.17705/1CAIS.03709

Wang, J.-W., Chang, C.-H., & Wang, F.-J. (n.d.). *An Analysis of Delegation Mechanism in Workflow Management System*.

Wang, Q., & Li, N. (2007). Satisfiability and Resiliency in Workflow Systems. In J. Biskup & J. López (Eds.), *Computer Security – ESORICS 2007* (Vol. 4734, pp. 90–105). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-540-74835-9_7

Webster, J., & Watson, R. T. (2002). Analyzing the Past to Prepare for the Future: Writing a Literature Review. *MIS Quarterly*, *26*(2), xiii–xxiii. https://doi.org/10.2307/4132319

Wibisono, A., Nisafani, A. S., Bae, H., & Park, Y.-J. (n.d.). *A DYNAMIC AND HUMAN-CENTRIC RESOURCE ALLOCATION FOR MANAGING BUSINESS PROCESS EXECUTION*.

Winkler, S., Sonnberger, H., & Bodendorf, F. (2008). Escalation Management for the Monitoring of Critical Process and Project Activities. *2008 International Conference on Computational Intelligence for Modeling Control & Automation*, 889–894. https://doi.org/10.1109/CIMCA.2008.141

Wolter, C., & Schaad, A. (2007). Modeling of Task-Based Authorization Constraints in BPMN. In G. Alonso, P. Dadam, & M. Rosemann (Eds.), *Business Process Management* (Vol. 4714, pp. 64–79). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-540-75183-0_5

Xue, S., & Tang, X. (2009). Research on Dynamic Workflow Based on Delegation Authorization Mechanism. *2009 International Conference on Computational Intelligence and Software Engineering*, 1–5. https://doi.org/10.1109/CISE.2009.5364907

Zavatteri, M., & Viganò, L. (2019). Last man standing: Static, decremental and dynamic resiliency via controller synthesis. *Journal of Computer Security*, *27*(3), 343–373. https://doi.org/10.3233/JCS-181244

Zeng, D. D., & Zhao, J. L. (2005). Effective Role Resolution in Workflow Management. *INFORMS Journal on Computing*, *17*(3), 374–387. https://doi.org/10.1287/ijoc.1040.0067