Journal of Electronic Research and Application, 2025, Volume 9, Issue 5
http://ojs.bbwpublisher.com/index.php/JERA
ISSN Online: 2208-3510
ISSN Print: 2208-3502

A Collaborative Approach to Distributed Heterogeneous Process Engines for Cross-Organizations

Xuehu Zuo[1,2]*, Xin Shan[1,2], Zhongguo Yang[1,2]

[1]Beijing Key Laboratory on Integration and Analysis of Large-Scale Stream Data, Beijing 100144, China
[2]School of Information Science and Technology, North China University of Technology, Beijing 100144, China

**Author to whom correspondence should be addressed.*

Abstract: In today's complex and rapidly changing business environment, the traditional single-organization service model can no longer meet the needs of multi-organization collaborative processing. Based on existing business process engine technologies, this paper proposes a distributed heterogeneous process engine collaboration method for cross-organizational scenarios. The core of this method lies in achieving unified access and management of heterogeneous engines through a business process model adapter and a common operation interface. The key technologies include: Meta-Process Control Architecture, where the central engine (meta-process scheduler) decomposes the original process into fine-grained sub-processes and schedules their execution in a unified order, ensuring consistency with the original process logic; Process Model Adapter, which addresses the BPMN2.0 model differences among heterogeneous engines such as Flowable and Activiti through a matching-and-replacement mechanism, providing a unified process model standard for different engines; Common Operation Interface, which encapsulates the REST APIs of heterogeneous engines and offers a single, standardized interface for process deployment, instance management, and status synchronization. This method integrates multiple techniques to address API differences, process model incompatibilities, and execution order consistency issues among heterogeneous engines, delivering a unified, flexible, and scalable solution for cross-organizational process collaboration.

Keywords: Distributed; Collaboration; Meta-process; Cross-organization; Business process; Process engine

Online publication: October 17, 2025

1. Introduction

Distributed systems, with their cross-domain, intelligent, and collaborative characteristics, provide a new perspective for organizations to examine and solve cross-domain and cross-network business challenges [1]. Workflow, on the other hand, has revolutionized the enterprise through automation and standardized process management. However, with the expansion of service scope and the diversification of customer needs, the traditional single-organization service model has been difficult to keep pace with the rapid development of the market [2,3].

Due to the technical differences between different organizations, such as differences in data formats and system interfaces, cross-organizational cooperation often faces many challenges. One of the most prominent issues is how to realize collaboration effectively [4]. In previous studies, the concept and practice of collaboration have been widely explored and applied. For example, a reference architecture, eSourcing Reference Architecture (eSRA), was designed by Pena *et al.* [5], which realizes dynamic inter-enterprise business process collaboration by supporting the modeling, mapping, sharing, and execution of different types of enterprise business processes through multi-layer frameworks and components. Shan *et al.* [6] investigated the mining of collaborative business processes across organizations and proposed a new approach that discovers Business Process Model and Notation (BPMN) 2.0 collaborative process models and their corresponding choreographies, and is able to handle business processes from different organizations. All these research results are good proof of the wide application and importance of collaboration today.

We contribute by designing a novel distributed architecture that is capable of managing and executing cross-organizational heterogeneous processes, so that many subtasks of a business process run separately on heterogeneous process engines, while ensuring consistency with the execution order of the original process subtasks. Standardized APIs and data formats through common operating interfaces are used to improve interoperability between systems of different organizations, thus realizing information exchange and resource sharing. A process model adapter is designed, which can automatically identify and convert the differences in process models between different organizations so that the process models can be deployed for execution in a collaborative environment.

2. Related work

2.1. Distributed system

With the rapid development of information technology, distributed systems have become an important part of the computer science field. In order to improve efficiency, reliability, and scalability, it has become a common solution to decentralize computational tasks to multiple computers [7,8].

In this context, Fang *et al.* [9] proposed a distributed process engine based on ZooKeeper implementation, which realizes the task compensation of the failed node by creating a process processing model and transferring the process flow logic being processed to other processing nodes to continue execution. In the article by Domingos *et al.* [10], a distributed workflow scheduling method based on "meta-processes" was proposed, which aims to solve the security problems and performance bottlenecks of traditional centralized workflow management systems, while supporting the dynamic adjustment of workflow models. The method realizes inter-node communication by constructing special meta-processes and through process variables.

In this paper, we improve the sub-process generation algorithm based on the article by Domingos *et al.* [10]. Compared to the graph-based breadth traversal method in the article by Domingos *et al.* [10], this paper adopts a simplified single-task segmentation strategy. The original method needs to traverse and analyze the whole flowchart, which is a cumbersome process. In contrast, the single-task segmentation method proposed in this paper treats each service task as an independent sub-process and contains the start event, service task, and end event. The advantage of this method is that it simplifies the process of creating sub-processes and directly makes each task independent. At the same time, the meta-process ensures the consistency between the sub-process and the original process in the execution order. The specific realization process will be mentioned in subsection 4.1.

2.2. Collaboration of heterogeneous process engines

In terms of distributed workflow management, extensive research has been conducted in academia and industry, and various solutions have been proposed. Klinger and Bodendorf [11] proposed a distributed workflow management system framework for cross-organizational workflow management. By analyzing the dependencies between tasks, the workflow is divided into multiple sub-workflows and assigned to the corresponding workflow management system for execution and management. Distributed collaborative management of cross-organizational workflows is achieved.

3. Collaborative approaches for distributed heterogeneous process engines

In order to solve the cross-organizational problems, such as data transfer, collaborative scheduling, state synchronization, etc., in the collaborative work of heterogeneous process engines, this paper proposes a distributed heterogeneous engine collaboration architecture based on the distributed workflow scheduling method based on the "meta-process" of Domingos *et al.* [10]. The architecture mainly consists of six parts: front-end interaction interface, business process preprocessing, scheduler, business process model adapter, common operation interface, and distributed heterogeneous nodes, as shown in **Figure 1**.

(1) Front-end interaction interface: Users can input business process requirements through the front-end interaction interface, call the API of the process engine to start process instances, query process status, submit user tasks, and receive process execution results.

(2) Business process preprocessing: Generate a collection of meta-processes and sub-processes by decomposing the original business process and add a listener for each sub-process. The listener is responsible for listening to the completion status of the sub-process and notifying the meta-process of the next step. Sub-process information is stored in the meta-process variables for meta-process scheduling and management.

(3) Scheduler: as the meta-process controlling the process, it is responsible for scheduling the execution order of the sub-processes, ensuring that the execution order of the sub-processes is consistent with the original business process, and coordinating the order between different nodes.

(4) Business process model adapter defines adaptation rules by analyzing the process model differences of different process engines. Construct regular expressions to find the keywords of differences in the process model files and replace them with the keywords of the target process engine.

(5) Generic operation interface: encapsulate the APIs of different process engines, provide a unified interface to manage process instances, tasks, process variables, etc. Through the mapping mechanism, map the invocation request of the generic interface to the API of the target process engine.

(6) Distributed heterogeneous nodes: Multiple process engine nodes of different types can deploy sub-processes in a distributed manner. The nodes communicate with each other through meta-process variables to coordinate the execution order.

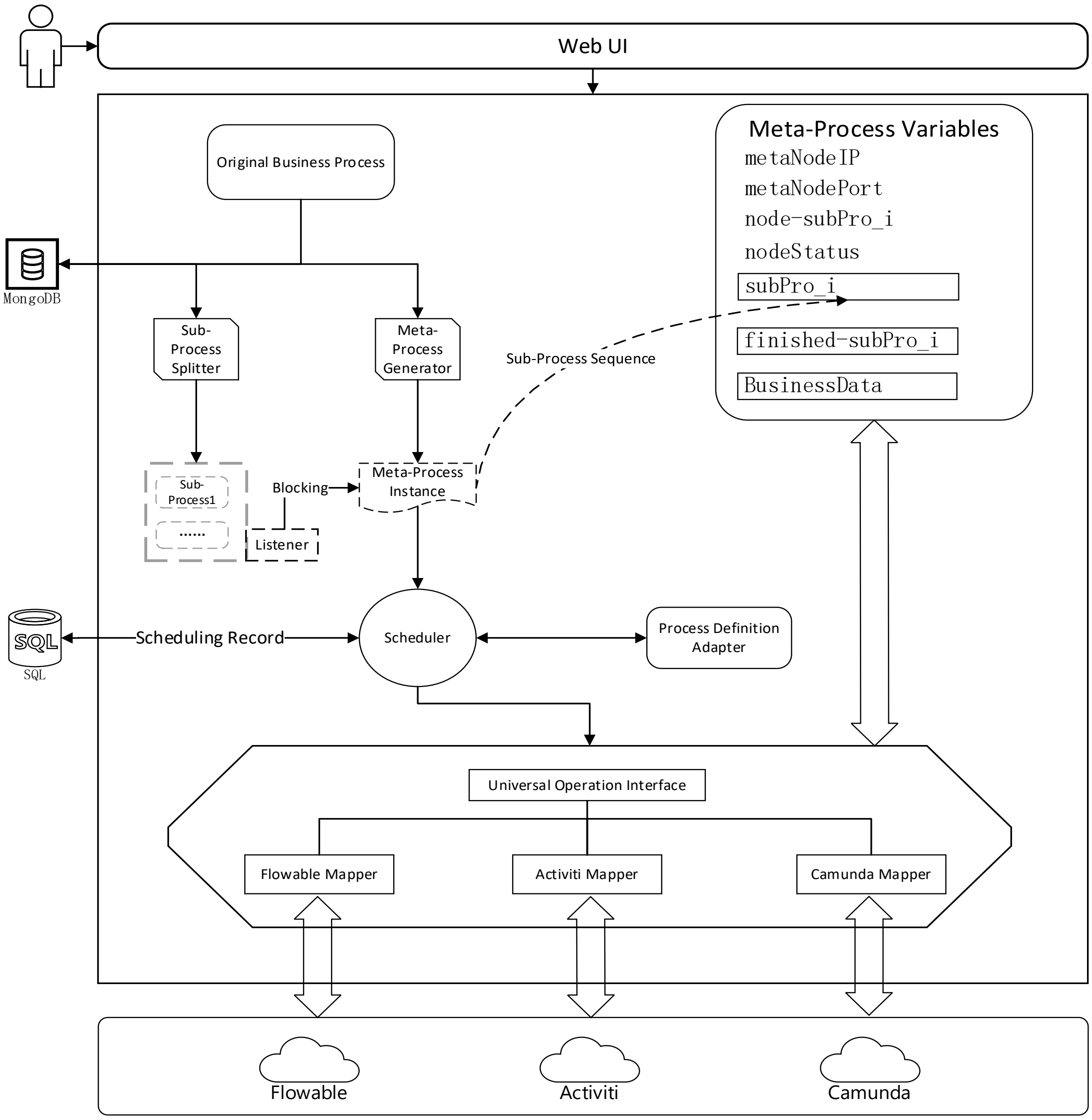

Figure 1. System architecture diagram for heterogeneous engine collaboration

3.1. Related definitions

In order to support distributed workflow scheduling, this paper is based on the “meta-process” method, which splits the original process into fine grains, each service task is split into independent sub-processes, and a listener is added to each sub-process to listen to its completion status. The generated sub-processes are stored as XML strings in meta-process variables for meta-process scheduling. Next, we will introduce the three key concepts of sub-process generation, meta-processes, and meta-process variables in detail.

Definition 1. Subprocess: the subprocess splitter (see **Figure 2**) represents the original process as a directed graph *G(V,E)*, where *V* is the set of nodes and *E* is the set of edges. For each service task node *v_task* in *G*, it is fine-grained partitioned into a set of sub-processes, sub-process *Sub_i* is represented as a directed graph $G\_task\left(V\_task, E\_task\right)$:

$$V\_task = \left\{v\_start, v\_task, v\_end\right\},\ E\_task = \left\{e\_start\_task, e\_task\_end\right\}$$

A listener node *v_listener* is inserted into *G_task*, which is connected to *v_task* and *v_end* to form an extension of *E_task*. The generated collection of sub-processes is stored as XML strings in the original process variables. The model ensures sequential process execution and data consistency.

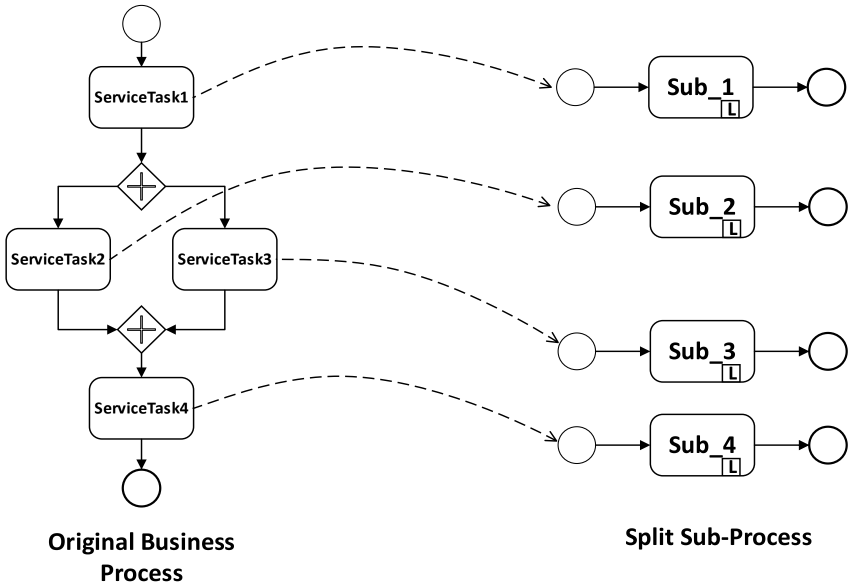

Figure 2. Sub-process splitter

Definition 2. Meta-process: meta-process *M*, as a control process, consists of a set of service tasks $C = \{C_1, C_2, ..., C_n\}$, which is responsible for coordinating and controlling the execution order of sub-process. *M* coordinates the execution order of *Sub_i* by transferring information through process variables. Each service task *Ci* corresponds to a sub-process *Sub_i.* When *M* is started, *Ci* will obtain the relevant information of *Sub_i* from the process variables of the meta-process, including its process model and deployment node information. After the successful deployment of sub-process *Sub_i*, *M* will block at the corresponding user task *Ui* and wait for the completion of *Sub_i.* When the execution of *Sub_i* is completed, the listener will be triggered, and the role of the listener is to complete the user task *Ui*, thus unblocking the meta-process.

In this way, the meta-process can ensure that the execution order of the sub-processes is consistent with the original process. In addition, the meta-process realizes the state synchronization and data transfer with the sub-processes through the transfer of process variables, which ensures the correctness and completeness of the execution of the whole workflow.

Definition 3. Meta-process variables: meta-process variables, including control data and business data, are the medium of communication between engine nodes, and are used to transfer information and state in different parts of the workflow, coordinate and control the execution of the workflow, and accompany the whole life cycle of the workflow. The data structure of meta-process variables is shown in **Figure 3**.

After the meta-process is started, the control task *Ci* reads the process definition of the process of the sub-process *Sub_i* and the deployment node information of the sub-process in the process variable of the meta-process instance. ci carries the information obtained above and deploys it to the engine *node_i* through the common operation interface. At this point the control task *Ci* of the meta-process is completed and blocked at the user task *Ui*. When the sub-process *Sub_i* is completed a completion marker is inserted into the meta-process variable and triggers the listener *L*. The listener completes the user task *Ui* through the generic operation interface and unblocks the meta-process. Subsequently the sub-process *Sub_i+1* is deployed until the meta-process completes all control tasks. The process variable design is shown in **Figure 3**.

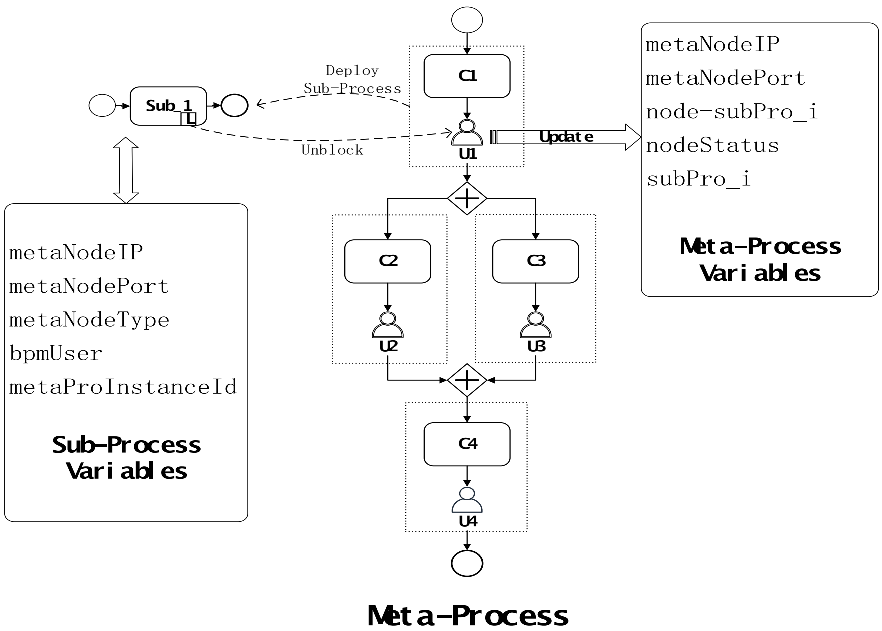

Figure 3. Process Variables design

4. Experimental analysis

4.1. Preparation

Consisting of two Flowable nodes and two Activiti nodes, the heterogeneous environment in the cross-organizational case is simulated, 300 business processes are randomly generated, and one sample is taken as a case sample (see **Figure 4**) for detailed analysis:

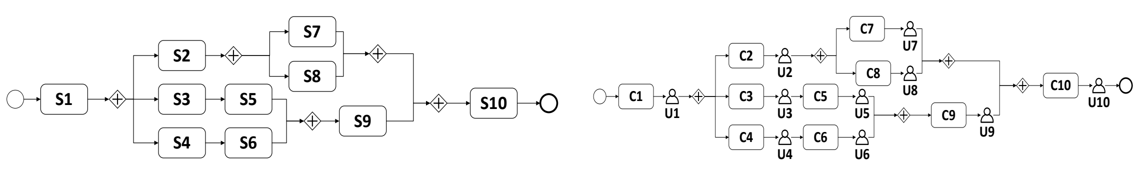

Figure 4. Experimental sample & sample meta-process

4.2. Experimental processes

The 300 business processes are deployed sequentially, and their execution status is observed. The deployment status and execution progress of each business process can be observed in the user portal platform. Take the case sample as an example, it is partitioned into 10 sub-tasks, and the control task in the meta-process will deploy these sub-processes according to the state of the engine node. The deployment of sub-processes is shown in **Table 1**.

Table 1. Results of sub-process deployment

Sub-task ID	Node	Node engine type
subPro_1	Wuhan:10.5.83.189:7001	activiti
subPro_2	Beijing:10.5.83.189:7000	activiti
subPro_3	Wuhan:10.5.83.189:7001	activiti
subPro_4	Wuhan:10.5.83.189:7001	activiti
subPro_5	Wuhan:10.5.83.189:7001	activiti
subPro_6	HangZhou:10.5.83.189:8282	flowable
subPro_7	Wuhan:10.5.83.189:7001	activiti
subPro_8	Beijing:10.5.83.189:7000	activiti
subPro_9	Wuhan:10.5.83.189:7001	activiti
subPro_10	Shanghai:10.5.83.189:8181	flowable

4.3. Experimental results and analysis

The experimental results show that 300 experimental samples were deployed to the four heterogeneous nodes, see **Figure 5**. The experiments demonstrate that the method can effectively improve inter-organizational process execution efficiency, ensure a consistent process execution order, and achieve the collaboration of heterogeneous process engines when dealing with cross-organizational business processes.

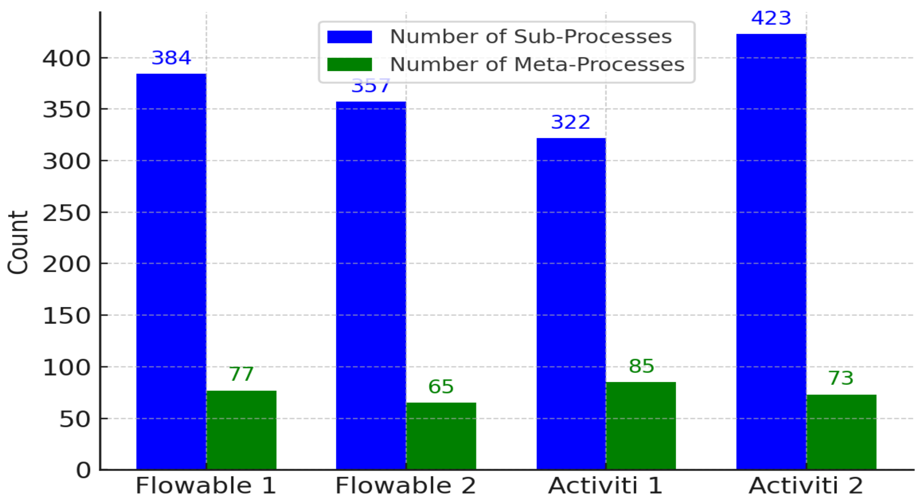

Figure 5. Deployment details of sub-processes and meta-processes for 300 business processes

5. Conclusion

In this paper, we have proposed a collaborative approach for a distributed heterogeneous process engine for cross-organizational purposes. The approach uses the meta-process as a control process and splits the original process into sub-processes. The proposed process model adapter and common operation interface achieve efficient collaboration between different process engines. In the future, we will further optimize the collaboration method of distributed heterogeneous process engines and explore more intelligent technical means, such as artificial intelligence, to achieve more intelligent and efficient process management.

Disclosure statement

The authors declare no conflict of interest.

References

[1] Xu X, Sheng QZ, Zhang LJ, et al., 2015, From Big Data to Big Service. Computer, 48(7): 80–83.

[2] Aouachria M, Leshob A, Ghomari AR, et al., 2024, A Process Mining Method for Inter-Organizational Business Process Integration. ACM Transactions on Management Information Systems, 15(1): 1–29.

[3] Gonzalez L, Delgado A, 2021, Compliance Requirements Model for Collaborative Business Process and Evaluation with Process Mining, 2021 XLVII Latin American Computing Conference (CLEI), 1–10.

[4] Norta A, Grefen P, Narendra NC, 2014, A Reference Architecture for Managing Dynamic Inter-Organizational Business Processes. Data & Knowledge Engineering, 91: 52–89.

[5] Pena L, Andrade D, Delgado A, et al., 2023, An Approach for Discovering Inter-Organizational Collaborative Business Processes in BPMN 2.0, International Conference on Process Mining, Oct 23, 2023, 487–498.

[6] Shan Z, Chiu DK, Li Q, 2005, Systematic Interaction Management in a Workflow View-Based Business-to-Business Process Engine, Proceedings of the 38th Annual Hawaii International Conference on System Sciences, Jan 6, 2005, 162.

[7] Verginadis Y, Mentzas G, 2008, Agents and Workflow Engines for Inter-Organizational Workflows in E-Government Cases. Business Process Management Journal, 14(2): 188–203.

[8] Osorio AL, Camarinha-Matos LM, 2008, Distributed Process Execution in Collaborative Networks. Robotics and Computer-Integrated Manufacturing, 24(5): 647–655.

[9] Fang L, Yang ZG, Qin S, et al., 2021, Meta-Process: A Novel Approach for Decentralized Execution of Process, 2021 International Conference on Service Science (ICSS), 38–44.

[10] Domingos D, Martinho R, Candido C, 2013, Flexibility in Cross-Organizational WS-BPEL Business Processes. Procedia Technology, 9: 584–595.

[11] Klinger P, Bodendorf F, 2020, Blockchain-Based Cross-Organizational Execution Framework for Dynamic Integration of Process Collaborations, Wirtschaftsinformatik (Zentrale Tracks) Mar 8, 2020, 1802–1817.