

Article

Dynamic Adaptation Method of Business Process Based on Hierarchical Feature Model

Le Zhang ^{1,2,3} , Qi Gao ^{1,*} and Tingyu Li ^{1,2,3}

¹ School of Mechanical Engineering, Shandong University, Jinan 250061, China; zhangle2341@mail.sdu.edu.cn (L.Z.); 202034413@mail.sdu.edu.cn (T.L.)

² Key Laboratory of High-Efficiency and Clean Mechanical Manufacture of Ministry of Education, Shandong University, Jinan 250061, China

³ National Demonstration Center for Experimental Mechanical Engineering Education, Shandong University, Jinan 250061, China

* Correspondence: gaoqi@sdu.edu.cn

Abstract: With the continuous complexity and frequent changes in business application scenarios, companies urgently need to establish a flexible business process management mechanism that includes dynamic rules, in which dynamic adaptation methods of business processes play a vital role. Aiming at the problem that the current methods only use the preset process template and the decision-making database, it cannot respond quickly to business changes and reconfigure the business process. This research proposes a dynamic adaptation method of business process based on the hierarchical feature model, builds a hierarchical feature model of complex processes, then establishes a hierarchical business policy set to achieve an agile response to business emergencies. By constructing a mapping model, the feature model is associated with the BPMN model to realize the rapid execution of the reconfiguration process model. The feasibility and effectiveness of the proposed method are verified by process examples and the developed business process dynamic adaptation tool.

Keywords: dynamic adaptation method; business process modeling; hierarchical management; feature model; business process modeling notation (BPMN)



Citation: Zhang, L.; Gao, Q.; Li, T.

Dynamic Adaptation Method of Business Process Based on Hierarchical Feature Model.

Information **2021**, *12*, 362. <https://doi.org/10.3390/info12090362>

Academic Editor: Kostas Vergidis

Received: 26 July 2021

Accepted: 3 September 2021

Published: 5 September 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Business process management (BPM), as the main business management method of an enterprise, has become one of the mainstream development trends of information technology [1]. Traditional BPM mainly considers static business processes, but today's business processes are dynamic and affected by the changing business environment. Structured business processes with strict constraints are gradually unable to cope with complex application scenarios that require frequent changes during execution [2,3], such as rapid changes in customer needs, sudden changes in markets and policies, etc. Efficiently establishing a flexible BPM mechanism [4] to respond quickly to changes in business needs is the guarantee for the success of modern enterprises. For this reason, given the business process model that has been built in the enterprise, business managers have begun to explore methods to change and reconfigure the process in response to changes during the execution of the process model. However, it is a very time-consuming and complicated task to make adaptive changes to business processes manually, since this work not only requires business personnel to be familiar with the internal structure of each business process model and the interrelationships between activities but also requires them to have professional domain knowledge and the ability to deal with various abnormal situations [5]. As the key to achieving flexible BPM, the dynamic adaptation method of business process can assist business personnel to quickly make correct and appropriate changes to the corresponding business process in response to emergencies in process execution, which

is of great significance to ensure the achievement of business goals. Meanwhile, as the focus of corporate BPM is no longer limited to offline analysis and configuration of process execution, more attention is paid to providing online support for the execution of business processes [3], which puts forward higher real-time requirements, including agile response and rapid deployment, for the dynamic adaptation of business processes. That is, it is not only necessary to quickly locate the exception to the specific business stage and take corrective measures but also to quickly deploy and execute the reconfigured business process. Therefore, it is necessary to study a business process dynamic adaptation method that can provide online support for the execution of complex process models in the enterprise and design tools that can support and run this method to meet the needs.

The current research on business process dynamic adaptation methods can be divided into data-driven methods and model-driven methods according to whether they depend on business process models [6]. The data-driven method mainly uses deep learning or reinforcement learning algorithms from the field of machine learning to train a data model based on the existing process log data and then automatically judge the execution status and output suggestions according to the input process information of the data model [7]. Since data-driven methods need to train and fit complex data models to prevent underfitting, the quality of process log data is required to be high. This requires the enterprise process log to have as many records as possible, and the process attributes covered by each record must be as comprehensive as possible. The enterprise is required to have enough big data accumulation and hardware computing resources, which does not meet the leanness emphasized by flexible BPM, that is, the use of simple and economical tools and techniques [8]. Some machine learning algorithms, such as tree-based algorithms, do not require a lot of data or computation, as is maybe the case when using a deep neural network algorithm. However, the data model can only be released and applied after all training and verification procedures are completed in an offline environment, which has a long period and cannot reflect better real-time performance [9].

This research takes into account that existing organizations, especially large enterprises, have achieved better process consolidation effects in their core processes, and the business process management system (BPMS) in the enterprise will store the corresponding business process model [10]. In this case, a model-driven approach based on business process models can make full use of the company's existing resources. However, when the current model-driven approach responds to business changes, the business stage to be executed is determined mainly based on the decision points preset in the business process model, and the process is generated by recutting according to the process template [11]. On the one hand, when encountering a business process model with a complex internal structure to find the corresponding decision point according to specific business changes, the current research will traverse the entire strategy database, which is costly and time-consuming and cannot meet the rapid response requirements of the online support. On the other hand, while current research completes the reconfiguration of the business process, research has not further explored converting the obtained model into the model standard supported by the workflow engine, such as the Business Process Modeling Notation (BPMN) standard, and lacks an effective automatic deployment mechanism to quickly execute the reconfigured process model.

This research proposes a dynamic adaptation method of business process based on the hierarchical feature model. According to the existing complex business process model of the enterprise, an extended feature model with hierarchical attributes is established to achieve hierarchical management of business stages with different granularities, analyze the internal structure of the process, and then establish a hierarchical business policy set to achieve efficient and automatic positioning of business emergencies and support dynamic adaptation of processes. In addition, to achieve rapid execution of the reconfigured process model, this research associates the feature model with the BPMN model by constructing a mapping model so that the changed model can be directly deployed and executed by the workflow engine through the mapping. Finally, combined with process examples

and the development of a dynamic business process adaptation tool, the feasibility and effectiveness of the proposed method are verified.

This paper is structured as follows: Section 2 is a summary of the research related to the dynamic adaptation method of business processes. Section 3 introduces the method proposed in this research in detail. Section 4 gives the management mechanism of the whole method. Section 5 describes the case study and the simulation results. Section 6 discusses the research performed and explains the limitations. The last section is the conclusion and outlook for future work.

2. Related Literature

BPM is a method and technology that can provide support for process design, implementation, execution, and analysis. The purpose of BPM is to establish an efficient operating model for the organization and create value for customers [12]. As an important response strategy for improving enterprise production capacity, order response agility, and risk management, BPM has received continuous attention and research. Previous BPM research used static definitions when designing business processes, which only provided limited forms of adaptability or insufficiently precise rules to handle anomalies, and lacked security constraints and inspection strategies. Such inadequate and incomplete business processes may lead to ambiguities and misunderstandings in implementation, resulting in failure with serious consequences [13]. This will also affect the implementation of the BPMS developed under the guidance of BPM, since organizational requirements are always changing, and business processes are often unable to be completed in a predetermined time, so BPMS will not be able to guarantee the automatic execution of business processes and achieve expectations [14]. Despite such efforts to help companies succeed in BPM adoption, there is an increasing need for rapid, flexible, and lean approaches to BPM that will help companies respond quickly and efficiently to changing environments [15]. The focus of BPM is gradually shifting to the research on the dynamics and adaptability of business processes [16], and the adaptability and flexibility of the process are also required by the European Industry 4.0 Initiative [17], which was initiated by the German government and described the digital and intelligent production of the factory in the future.

Recent research emphasizes the necessity of enhancing the flexibility and dynamics of BPM. If an organization is in a highly dynamic business environment, its business process must be able to respond to external changes [18]. Martins defines adaptive and flexible BPM as a business management technology that can discover changes and can quickly respond to changes [19]. Peyman added that flexible BPM needs to be lean, that is, to achieve continuous evaluation of process execution through simple and economical tools and technologies [15]. Different from the implementation of flexible BPM at the company level, Raschke proposes a dynamic adaptation method for business processes from the process level to realize the ability to redesign and reconfigure individual business process components and combine individual tasks and resource capabilities to respond to environmental changes [20]. The current research on business process dynamic adaptation methods can be divided into data-driven methods and model-driven methods according to whether they depend on business process models [6]. The data-driven method mainly uses machine-learning-related algorithms to train a data model based on the existing process log data and then automatically judge the execution status and output suggestions according to the input process information of the data model [7]. For example, Paschek uses artificial intelligence and machine learning methods to build core components that can detect changes, set configurations, and check during process execution based on big data to support the automation of business process dynamic adaptation method [21]. Pika applies machine learning to extract information from process execution data to form experience knowledge, which can help employees achieve optimal business resource allocation under different conditions [22]. From the perspective of effectively discovering business problems that hinder process execution, Ahn proposed a Metis+ framework that can identify potential business problems. This framework is based on big data to explore and verify problem

assumptions [23]. Palm uses reinforcement learning algorithms to simulate the process of employees dealing with changes and making decisions and explores the implementation of adaptive systems [9]. This method also requires sufficient data to train a good performance data model.

The dynamic adaptation method of the business process based on the process model, that is, the model-driven approach, uses a variety of methods to set variability rules based on the process model to meet the adaptability requirements for business changes during process execution [24]. The current research on model-driven methods is mainly oriented to three main aspects: predefined decision points, configurable templates, and bottom-up mechanisms. Regarding the method of predefined decision points in the process model, Pucher proposes an agile process construction method by predefined decision points in the modeling phase, where business users are allowed to make changes to the process during process execution, query the set of decision points, and choose to execute different subprocesses [25]. Agostini further considers a variety of ways to enhance the flexibility of decision making to adapt to various changes [26]. Bizagi introduces a flexible ad hoc business process, which does not have a complete definition of the underlying business process, and consists of a set of activities with decision points. When the ad hoc process is executed, the user decides which activities should be executed and when to execute according to business needs [27]. However, since future changes cannot be fully predicted and enumerated, the method of predefined decision points has great limitations in practical applications. From the perspective of constructing a configurable template, Hallerbach provides an optional template for the processing of different business situations, which can change the sequence of activities in the process when conditions change and achieve dynamic adaptation based on the template [28]. To establish a business process reference model, Rose proposes two ways to expand the process modeling language by setting configurable nodes by adding primitively hidden and blocking switch attributes to nodes [29]. Han directly extends the configuration of BPMN and establishes a configurable BPMN language based on the extension of control flow primitives to construct a configurable template for the process [30] but does not consider the achievability of the complex process template. Regarding the method of applying a bottom-up mechanism to construct a variable process model, Yan uses a bottom-up mechanism to extract the necessary atomic-level business process static components, combined with descriptive policy sets to build a derived process model, and the process can be constructed flexibly when the business requirements are known [31]. Xiao uses a large number of reusable process fragments to construct the process, but the new business process generated according to the business conditions is only suitable for the execution of a specific instance [32]. Since the bottom-up mechanism needs to comply with the descriptive specifications and constraints of Dec Ser Flow [33] and the method of minimizing conditions [34], the dynamics it can achieve are quite limited.

From the current research on business process dynamic adaptation methods, it can be concluded that although data-driven methods and model-driven methods have different starting points and technologies, both essentially rely on models to perceive business changes and deal with them. The model on which data-driven methods rely is a data model trained based on process big data, and the model on which model-driven methods rely is a new process model formed by adding decision points or an adaptive model directly derived. The main difference between the two methods lies in the preconditions of their application, namely the accumulation of big data and the business process model. Although some companies are advancing the establishment of big data systems, considering the current business process models established by expert knowledge in the current companies, the model-driven approach still has high application value because it can make full use of the company's existing resources and meet the lean requirements of flexible BPM. Existing research on model-driven methods mostly adopts the method of constructing an overall decision-making library or preset process templates so as to find and implement corresponding solutions according to business emergencies or needs. However, in the

actual business process of an enterprise, due to the complex nesting relationship between subprocesses and activities, and because various business stages involve changes, the use of templates and decision libraries cannot take into account the needs of all business stages. It is difficult to meet the rapid response and flexible execution of process online support. The hierarchical approach has unique advantages when solving problems with many management objects and complex relationships. By combining hierarchical management ideas, dividing different subprocesses or activity levels, and establishing hierarchical business strategies, the business level can be quickly located where measures need to be taken according to business change factors, achieving agile response under the complex process model. Meanwhile, after the completion of process changes or dynamic adaptation, the rapid deployment of processes without further exploration is also a problem in current research.

3. Dynamic Adaptation Method Based on Hierarchical Feature Model

This research proposes a dynamic adaptation method of business process based on the hierarchical feature model. By establishing a feature model with hierarchical attributes, mapping the feature model, formulating a hierarchical business policy set, and other related methods to obtain a hierarchical BPMN process model, a dynamic management method is realized that can flexibly and efficiently configure the process model during business execution. The overall framework of the proposed method is shown in Figure 1, which mainly includes four phases: create hierarchical feature model, correlation feature model and BPMN model, formulate hierarchical business policy, and execution phase. Each phase is described in detail below.

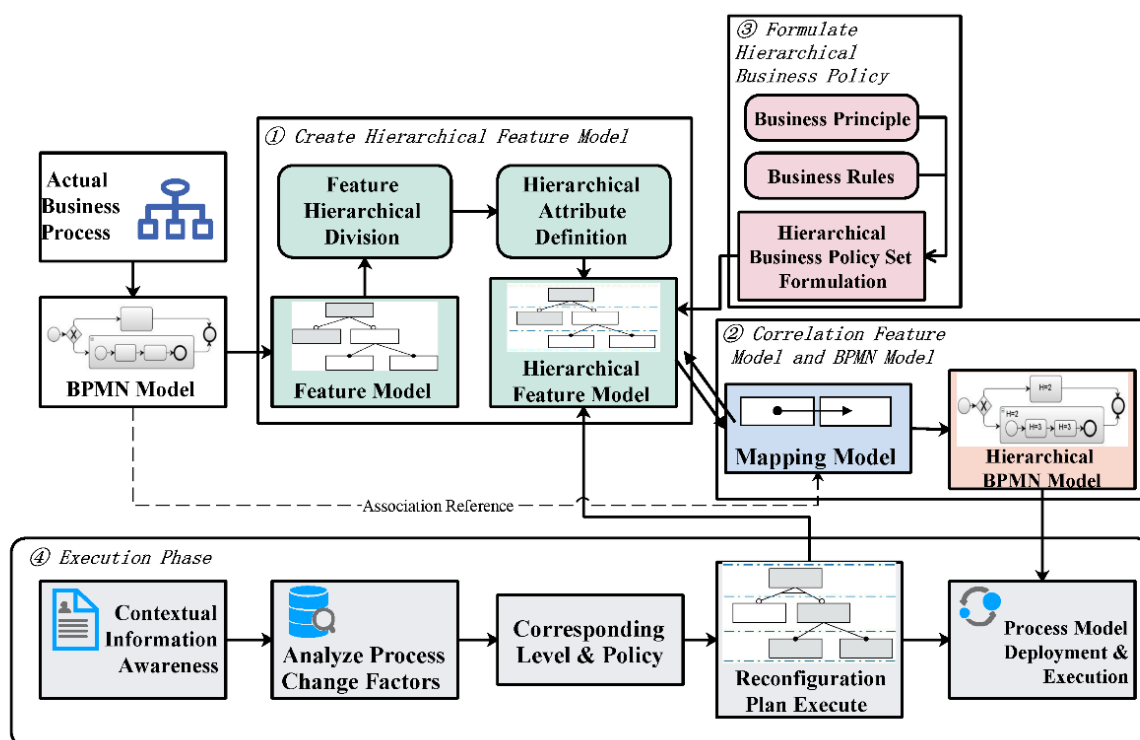


Figure 1. Overall framework of the HPMF method.

3.1. Create Hierarchical Feature Model

The purpose of creating a hierarchical feature model is to establish a variability model that can fully display the internal hierarchy of the process according to the actual business process of the enterprise, in which the setting of variants is mainly for the flexible configuration of the process during business execution. The current methods of using

variability modeling technology to implement variability models include feature modeling [35], Common Variability Language (CVL) [36], and languages in any specific field. The method of feature modeling is adopted in this research because it can provide hierarchical management of features, reflect the hierarchical relationship between different business stages in the process model, and has good tool support for variability reasoning [37]. In the feature model, features are hierarchically linked in a tree structure through variability relationships such as Optional and Mandatory. There is only one root node in the feature model, and all the features that depend on the root feature are leaf features. In this research, the root feature represents the entire business process, and the leaf feature represents the different subprocesses and activities contained in the process, where activities are the lowest-level features. According to an example of a company's air-conditioning service support process, the corresponding feature model can be established. Considering the display effect, some feature subnodes have been simplified, as shown in Figure 2.

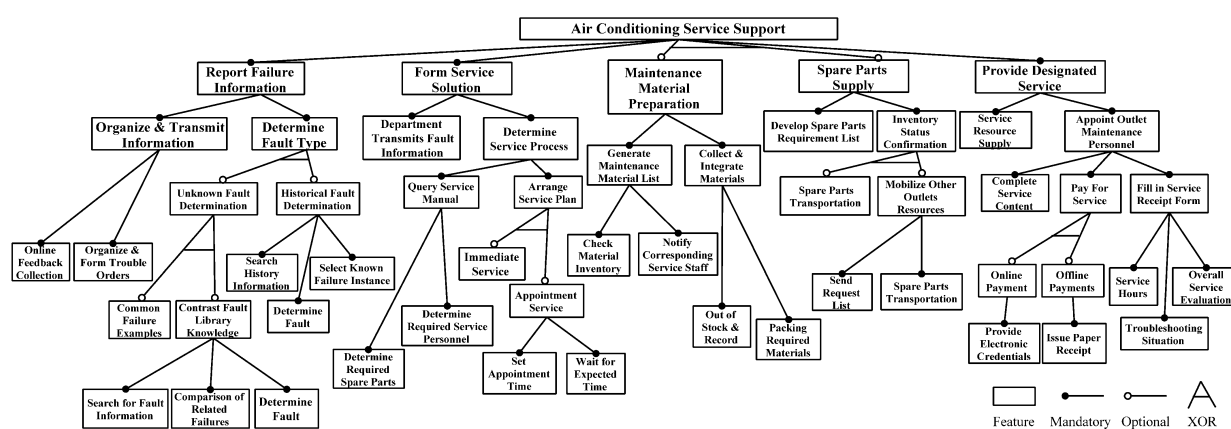


Figure 2. Air-conditioning service support feature model.

Generally, in order to divide the management authority of different business stages and refine the allocation of business resources according to the organizational structure, the enterprise will further divide the process into different subprocesses or activities to form grid-based local management and effectively supervise the execution of the business. The fundamental difference between different processes of an enterprise lies in its object-oriented and different resource calls. On the one hand, different business processes involve different functional scopes, from cross-organizational processes and cross-departmental processes across functional areas to specific functional areas such as departmental internal processes; on the other hand, the resources called by different business processes are different, and the scope of resource scheduling affected when the process is changed is also different, that is, the management authority of different granular resources is quite different, which is specifically reflected in the corporate organizational structure. The resource allocation permissions set for different personnel are different. Combining the above two points, to have a clear basis when dividing the feature model hierarchy and avoid the blurring of hierarchy boundaries and possible confusion between different levels, this research proposes a feature model hierarchy division method based on a decision matrix. The decision matrix is shown in Figure 3, which is vertically based on the scope of business involved and horizontally combined with the level of resource refinement under the same functional level for further division. If the scope of functions involved in the business stage is larger, the level of division is higher. If the longitudinal indicators of multiple business stages are the same, the horizontal indicators are combined for further differentiation, and it stipulates that the vertical index has a greater influence on the weight of the level division than the horizontal index. Taking Area ① and Area ② in Figure 3 as an example, in order to judge the hierarchical relationship, Area ③ can be used for comparison. The vertical index of Area ① is higher than that of Area ③, and the vertical

index of Area ② and Area ③ is the same. Indexes are more weighted in the hierarchical division, so the level of Area ① is higher than that of Area ②, and the horizontal index of Area ② is higher than that of Area ③. Therefore, the hierarchical relationship of the three areas in the figure is: ① > ② > ③.

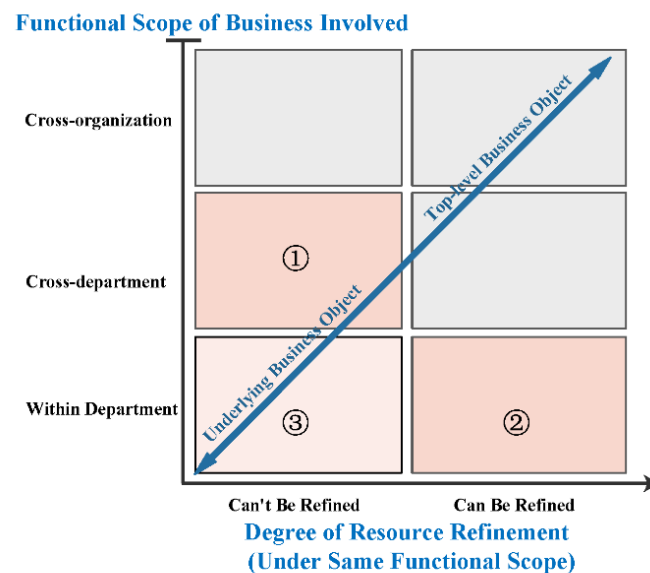


Figure 3. Hierarchical division decision matrix.

After completing the hierarchical division of features, in order to effectively establish the relationship between feature levels and features, an extended feature model, that is, a feature model with attributes [38], is used in this research, which adds an attribute H representing the feature level to all the features in the model.

Definition 1. Hierarchical attributes H . H represents the level of each feature in the feature model, $H: \{a\}$, where a is the level of the feature.

Definition 2. Hierarchical Feature Model FM^H . FM^H is obtained by expanding the hierarchical attributes of the features in feature models. The feature model is built by the feature modeling method. FM^H can provide a clear view of the hierarchical and granular relationship between different business stages.

Combining the hierarchical division decision matrix to hierarchically divide the air-conditioning service support feature model and define the feature attributes, the obtained air-conditioning service support FM^H is shown in Figure 4. The functional scopes of $H = 1$ and $H = 2$ are cross-organization and cross-department, and business resources cannot be further refined under the same functional level. $H = 3$ and $H = 4$ belong to the same functional level within the department, but the resources involved in the business objects in $H = 3$ need to be further refined and allocated, while the resources in $H = 4$ cannot be further refined, thus forming a further hierarchical division under the same functional scope.

3.2. Correlation Feature Model and BPMN Model

FM^H can provide a hierarchical structure view of different features. Through the dynamic activation or deactivation of specific features in the feature model, the business can be reconfigured when encountering emergencies. For the final configuration result to be accepted by the workflow engine that supports the BPMN specification, it is necessary to associate the features of each level of the feature model with the subprocesses and activities in the BPMN model.

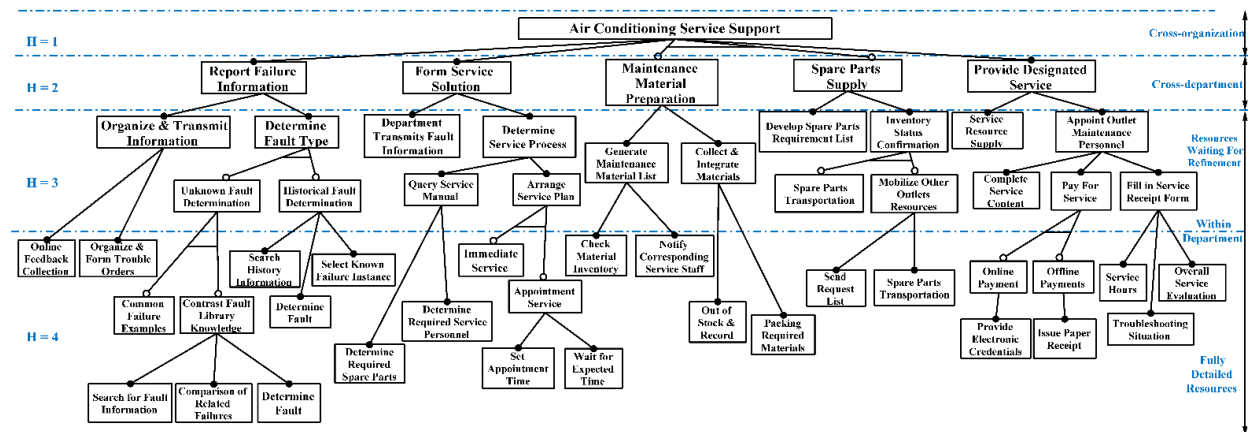


Figure 4. Air-conditioning service support FM^H .

To realize the association between the feature model and the BPMN model, the method of establishing a mapping model [39] is adopted in this research because it can be easily implemented by the ATLAS Model Weaver tool [40,41]. The mapping relationship between the problem space model (i.e., feature model) and the problem-solving space model (i.e., BPMN model) can be defined in the mapping model, where each link has two endpoints: the first endpoint represents each feature in the feature model; the second endpoint represents a subprocess or activity in the BPMN model. The mapping model is used as a bridge to connect the two endpoints, and the hierarchical attributes are also reflected and transmitted in the mapping process. The mapping relationship established based on the air-conditioning service support FM^H is shown in Figures 5 and 6.

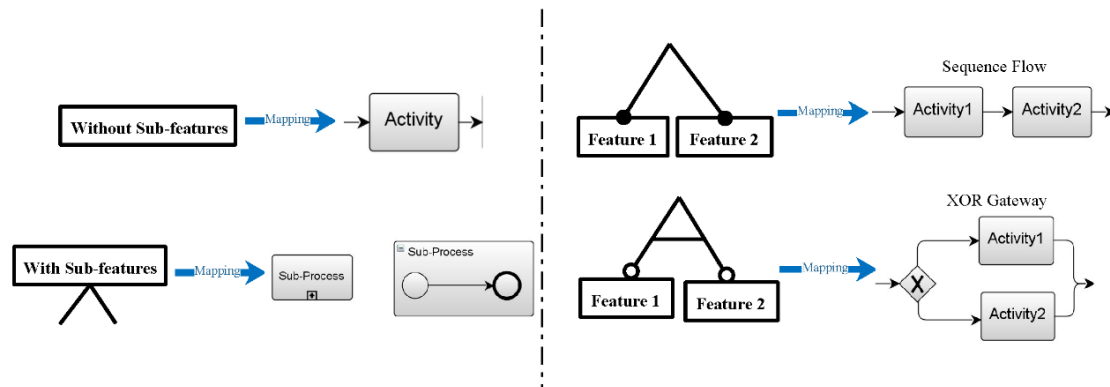


Figure 5. Mapping example.

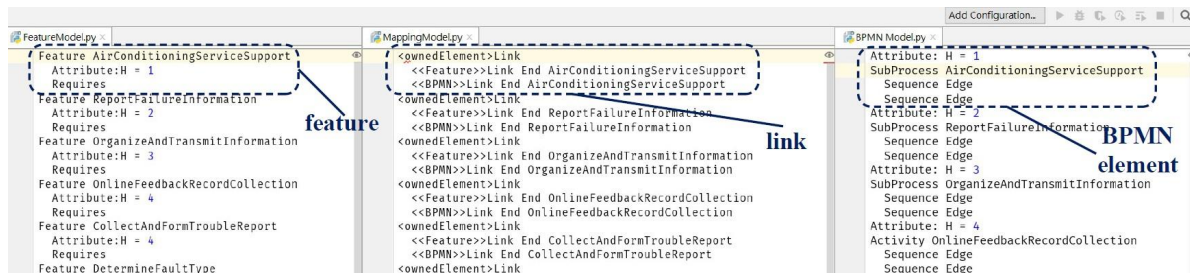


Figure 6. Mapping model fragments.

Definition 3. Hierarchical BPMN Model BM^H . The hierarchical feature model with hierarchical attributes generates a corresponding hierarchical BPMN model through the mapping model. The

hierarchical structure is represented by the subprocesses defined in the BPMN specification, and the lowest level is a single business activity.

The corresponding BM^H generated by the air-conditioning service support FM^H through the mapping model is shown in Figure 7.

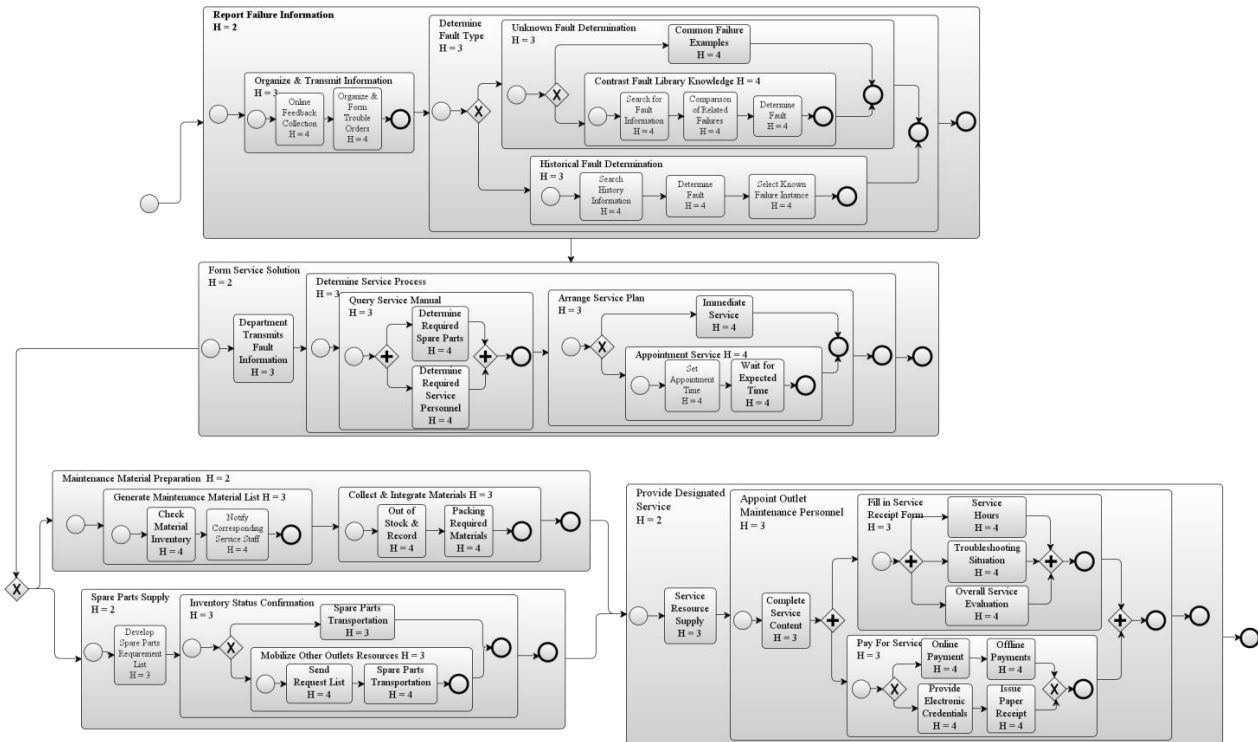


Figure 7. Air-conditioning service support BM^H .

3.3. Formulate Hierarchical Business Policy

To solve the management drawbacks in the current business process dynamic adaptation research, that is, the difficulty of maintaining business policy and the inefficiency of finding specific business strategies when responding to emergencies, it is necessary to formulate the same hierarchical business policy in conjunction with the created FM^H .

Definition 4. Hierarchical Business Policy P^H . Given a hierarchical feature model FM^H , the corresponding hierarchical policy is the set $P^H = (Pr, RI)$, where Pr is a set of business principles oriented to all the features of a single level, representing the fundamental difference and basis of each level in dealing with actual business needs. All the features of the corresponding level need to be scheduled and reconfigured under the restrictions of Pr . RI is a set of business rules for a single feature, which defines what state the variable feature should be in under different contexts, that is, activated or deactivated.

Definition 5. Business Principle Pr . Given a hierarchical feature model FM^H , all the features contained in each level need to meet the business principles p of the corresponding level during configuration or reconfiguration, $p \in Pr$, which is defined as the following form: configure/reconfigure REQUIRE principle ψ . ψ is a logical expression related to the characteristic level attribute H .

Since business processes in the enterprise face different objects and resources called, there will be big differences in handling actual problems in various business stages. Specifically, from the object-oriented perspective:

- For business stages that span multiple enterprises or organizations, because they involve a wide range of functions, it is equivalent to a global scheduler, coordinating the task allocation and execution policies between different partners, so when configuring the features of the business stages of the cross-enterprise organization, the scope of its responsibilities should be defined, and the scope of powers that can be assigned and authorized to control should be clearly defined;
- For business stages that span multiple departments within an enterprise, because the products and services of different business departments are quite different, the strategies adopted by each department in the process of participating in the business are also very different. If the business arrangement exceeds the capability level of a certain department, it will result in a reduction in business execution efficiency and a waste of resources. Therefore, when configuring the features of the cross-departmental business stages in the enterprise, within the scheduling scope, it should also be checked whether it meets the corresponding department-level guidance strategy to ensure that the actual business can be effectively carried out;
- For business stages in a specific department of an enterprise, because it is directly oriented to business personnel, the company mostly adopts ability-based selective management. After personnel meeting business ability requirements are transferred, they must implement and complete the goals, so that the upper-level business can be specifically implemented. When configuring the corresponding features of the business stages in an enterprise department, it is necessary to ensure that the business personnel's capabilities and permissions can meet the business requirements.

From the perspective of process call resources, due to the different granularities of process call resources, there are also big differences in the resource scheduling authority involved when reconfiguring the process:

- For business stages that call refined professional business resources, that is, resources of a specific category, because the resource scheduling does not affect other stages in the same functional layer, additional resource management rights are not required when configuring;
- For business stages that call integrated business resources, that is, the combination of coarse-grained resources, because the resource scheduling will affect other business stages at the same level, additional resource management rights are required. Take the air-conditioning service support process as an example, when business reconfiguration involves integrated resource scheduling, if it occurs within a department, the department manager-level resource management authority is required.

Combining the above two aspects, for the air-conditioning service support examples, the hierarchical business principles of different business stages can be summarized as shown in Table 1.

Table 1. Business principle at all levels.

| Object Oriented | Owned Level | Business Principle (<i>p</i>) |
|--|-------------|--|
| Cross-organization | $H = 1$ | Assignment and authorization control rights limited to the scope of the business object's responsibility |
| Cross-department Within enterprise | $H = 2$ | Business configuration conforms to the corresponding department-level guidance strategy |
| Within a specific department | $H = 3, 4$ | Business configuration is within the limits of the executive's own capabilities and authority |
| Resources Called | Owned Level | Business Principle (<i>p</i>) |
| Comprehensive Business resource set | $H = 3$ | Demand department manager-level resource management authority |
| Professional business resources | $H = 4$ | No additional business management authority requirements |

In this research, the air-conditioning service support business example involves variable features at the second, third, and fourth levels. The corresponding business principles can be known from the above table. The business principles corresponding to these three levels can be described as:

$$\begin{aligned}
 p_{h=2} &: R_f(\{f \mid f \in [FM^H] \wedge \{H \mid H = 2\}\}) \subseteq \{L \mid L \in \{limits_Departmental\}\} \\
 p_{h=3} &: R_f(\{f \mid f \in [FM^H] \wedge \{H \mid H = 3\}\}) \subseteq \{L \mid L \in \{limits_BusinessPersonnel\}\} \cup \{A \mid A \in \{authority_DepartmentManager\}\} \\
 p_{h=4} &: R_f(\{f \mid f \in [FM^H] \wedge \{H \mid H = 4\}\}) \subseteq \{L \mid L \in \{limits_BusinessPersonnel\}\}
 \end{aligned}$$

where R_f is the configuration for feature f , L is the limitation factor, and A is the authority level of resource management.

Definition 6. Business rules RI . Given a hierarchical feature model FM^H , the business rule $r \in RI$ corresponding to each variable feature on it can be expressed in the following two forms:

IF ψ THEN activate feature f
 IF ψ THEN change feature f state to $f.s$
 where ψ is a logical expression related to the actual context, f is a feature in the hierarchical feature model, s is one of the states that the feature can be given, and the business rule and the corresponding variable feature have the same hierarchical attribute H .

A variable feature that can be activated or deactivated can be represented by a triple:

$$f = \{(feature_name, H, S) \mid H = \{a\}, S \in \{Active, Inactive\}\}$$

Taking an example of the hierarchical air-conditioning service support process, partial configuration R for the $H = 3$ subprocess service department determines the type of failure can be expressed as the following form:

$$\begin{aligned}
 R_{DetermineFaultType} &= \{\{UnknownFaultDetermination, H = 3, Active\} \vee \{HistoricalFaultDetermination, H = 3, Inactive\}\} \\
 &\wedge \{\{CommonFailureExamples, H = 4, Active\} \vee \{ContrastFaultLibraryKnowledge, H = 4, Inactive\}\}
 \end{aligned}$$

3.4. Execution Phase

In the execution phase, this research mainly uses context awareness technology to define contextual conditions to support effective reasoning about real-time status and possible situations in the process execution to obtain process change factors. Since the focus of the process operation instance in this research is mainly whether the actual business state of each stage meets the preset process model, after obtaining the context information, it is necessary to determine whether the current context state meets the business stage being executed through rules. The rules can be expressed in the following form:

$$context_state:: Hierarchy: feature_name$$

where $context_state$ represents the context state. When it is a specific condition, the specific features at a certain level can be directly specified. On the one hand, the corresponding level is specified, and the specific object is specified on the other hand. *Hierarchy* is the hierarchical attribute H . *feature_name* corresponds to the specific single variable feature F in the feature model. The $context_state$ can be set to empty, and the implicit condition is specified at this time, that is, the corresponding feature is always active.

The actual context state depends on the relevant attributes in the business environment, including conditional judgments during business execution and related information as business inputs. Before determining the features that need to be reconfigured and starting model reconfiguration, context analysis rules can be used to determine whether there is a situation that exceeds the original preset of the process model, that is, whether the feature

that should be activated according to the business environment is in the activated state. For the air-conditioning service support process instance, the following rules can be defined:

TimeSetting = 0: *H* = 4:: *ImmediateService*
TimeSetting > 0: *H* = 4:: *AppointmentService*
H = 3:: *QueryServiceManual*, *ScheduleMaintenancePlan*
Service = "maintenance": *H* = 2:: *MaintenanceMaterialPreparation*
Service = "repair": *H* = 2:: *SparePartsSupply*

Through rule-based reasoning and comparative judgment on the context state, it is possible to confirm the specific features and levels that need to be reconfigured when the process changes, and then the model reconfiguration module generates a process model that meets the requirements of the new business based on the hierarchical feature model configuration of the original process during the process execution.

The model reconfiguration process is shown in Algorithm 1. By inputting the context state *C*, including the change in business conditions and outputting the reconfigured FM^H . The algorithm first locates the configuration range of the feature set $h(C(H))$ of the corresponding level through the hierarchical attribute $C(H)$ of the context state *C* and accurately activates the feature through the feature name $feature_name(C)$ linked by the context state, and at the same time, the same level feature for the preset context state in the model is deactivated. Then, it checks whether the new configuration of the level is under the constraint of the corresponding level *pr*, and if it exceeds the constraint range, the algorithm will issue a warning to the business manager and return. After that, the partial configuration code *l* is generated through the function $lccGenerate(l)$, which is composed of the configured level $C(H)$ and the number of frequencies *i* are reconfigured for the model. Finally, the partial configuration code *l* is integrated into $reconfigure(FM^H)$ and returns the result.

Algorithm 1. Model reconfiguration.

Input: Context state *C* with changed elements

Output: Reconfigured hierarchical feature model $reconfigure(FM^H)$

h: The set of features with Hierarchy *H*;

$preset(FM^H)$: The set of preset hierarchical feature model;

$reconfigure(FM^H)$: The set of reconfigured hierarchical feature model and $reconfigure(FM^H) = \emptyset$;

Let *l* be the local configuration code and $l \in \mathbb{R}$;

Let *i* be the global variable and $i = 0$;

01: **for all** $f \in h(C(H))$ **do**

02: **if** $feature_name(f) = feature_name(C)$ **then**

03: set $feature_state(f)$ to {Active}

04: **end**

05: **else if** $feature_name(f) = feature_name(preset(C))$ **then**

06: set $feature_state(f)$ to {Inactive}

07: **end**

08: **end for**

09: **while** ($h = h(C(H))$) **do**

10: **if** $C \notin pr(C(H))$ **then**

11: warning configuration not within *pr* and return

12: **end**

13: **end while**

14: $lccGenerate(l)$

15: $reconfigure(FM^H) \leftarrow preset(FM^H)$

16: wrap *l* in $reconfigure(FM^H)$

17: return $reconfigure(FM^H)$

18: **end**

19: **function** $lccGenerate(l)$

20: $l = C(H) * 100$

21: $i += 1$

4. Hierarchical Process Model Management Mechanism

The dynamic adaptation method based on the hierarchical feature model uses the hierarchical feature model to divide different business stages to meet the rapid positioning of complex business processes in the face of emergencies and associates with the BPMN model through the mapping model so that the generated configuration can be accepted and executed by the workflow engine. The main task of process management during business execution is to select the appropriate business policy to configure the hierarchical feature model locally or globally according to changes in the external environment to achieve flexible response to emergencies. Combining the previous relevant definitions, a complete mechanism of HPMF can be proposed as shown in Figure 8.

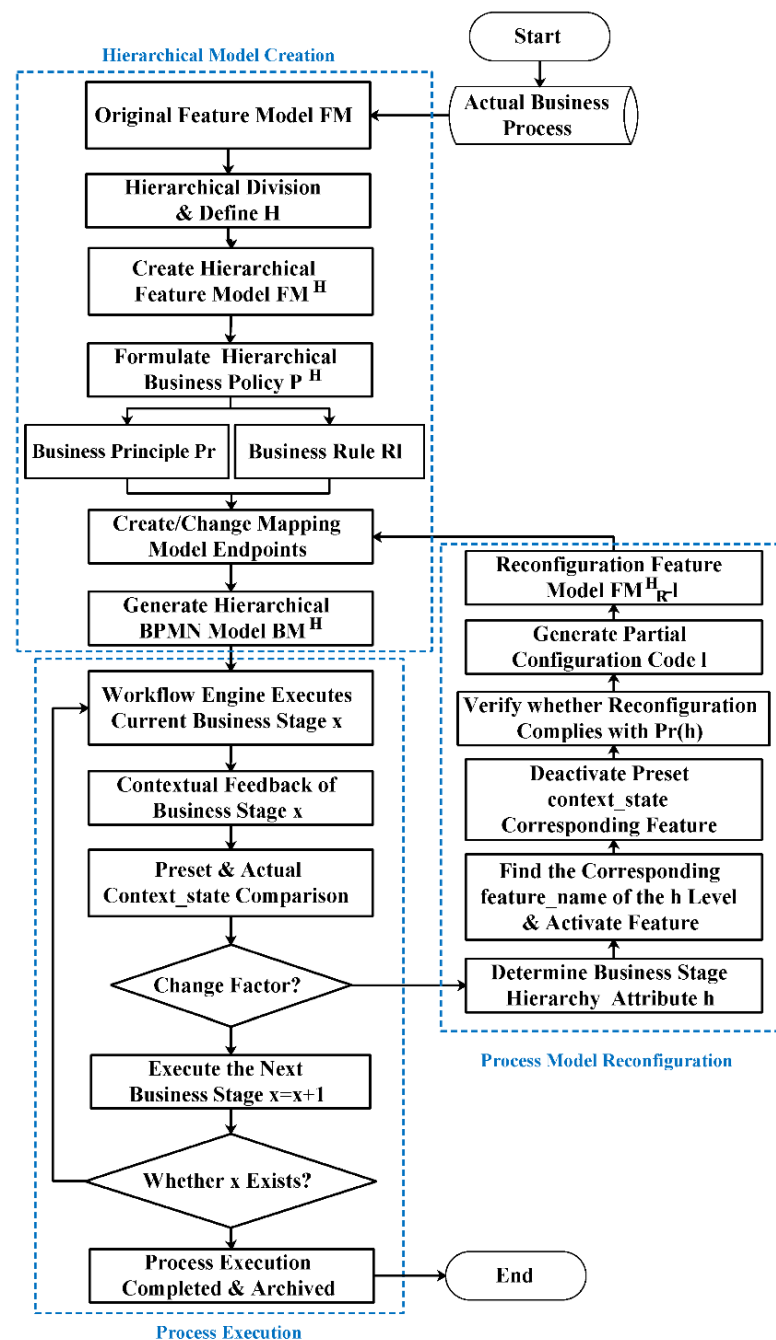


Figure 8. Hierarchical process model management mechanism.

5. Case Study

In order to verify that the proposed method can effectively formulate business strategies for different business stages in the application, efficiently reconfigure the process model in an execution environment with many emergencies, and solve the problems of difficult management and maintenance of corporate business strategy and long response time to model changes, this research conducted a case study based on corporate process examples. Because this research is aimed at the hierarchical dynamic adaptation method of the complex process model of the enterprise, the two dimensions considered by the hierarchical division method are object oriented and resource call. Therefore, the selected business process instance needs to involve as much as possible the functional scope of all possible business objects, including cross-organization, cross-department, and internal department, and reflect the different degrees of refinement of resources under the same functional scope. The selected process instance should also be represented in the enterprise to demonstrate the universality of this method. As the value created by product services accounts for an increasing proportion of the company's overall revenue, more and more companies now regard the product service process as their core process, hoping to create greater added value [42]. The selection of product service-related processes is more in line with the company's current application requirements for BPM. Based on the above considerations, this study chose the air-conditioning service support process as an instance for method verification, since it has a wide range of functional scopes, many nested sub-processes and activities in the process model, and the degree of internal resource refinement is different.

Based on the selected business process instance, this study adopts the following methods to explore the results: (1) Develop a business process dynamic adaptation tool and demonstrate the various stages of the method in combination with the process instance to verify the feasibility of the method at the application level. (2) Based on the business process dynamic adaptation tool, the simulation of the model reconfiguration process is carried out to quantitatively demonstrate the time response performance of the method and verify the effectiveness of the method in the implementation. The rest of this section will introduce these two aspects.

To verify the feasibility of the proposed method at the application level, this research develops the business process dynamic adaptation tool through C# and NET frameworks and develops the user interface with the Windows Presentation Foundation component. The actual application situation is shown in Figure 9. First, the air-conditioning service support process or other process examples through the BPMN-Editor [43] in the enterprise are obtained, and a hierarchical feature model is created through the business process dynamic adaptation tool and the hierarchical division decision matrix. The features that are preset to be activated are marked as green. The features that are preset to be deactivated are marked in gray to visually display the process preset conditions on the interface. Then, the endpoints of the mapping model are defined to convert the hierarchical feature model to the hierarchical BPMN model, which is deployed and executed by the workflow engine. During the process of operation, the context information is collected, and the business rules are compared to determine whether the current business stage conforms to the model preset, if any. If the situation exceeds the expected situation, the corresponding context information and hierarchical attribute information will be returned to the management tool. Finally, after receiving the reconfiguration instruction information, the business process dynamic adaptation tool directly performs partial reconfiguration of the model under the constraints of business principles on the features at the specified level, deactivates the corresponding features, and activates the features that meet the business needs. It then changes mapping model endpoints accordingly, completes the modification of the BPMN model, and deploys and executes the reconfiguration business process to respond to business changes.

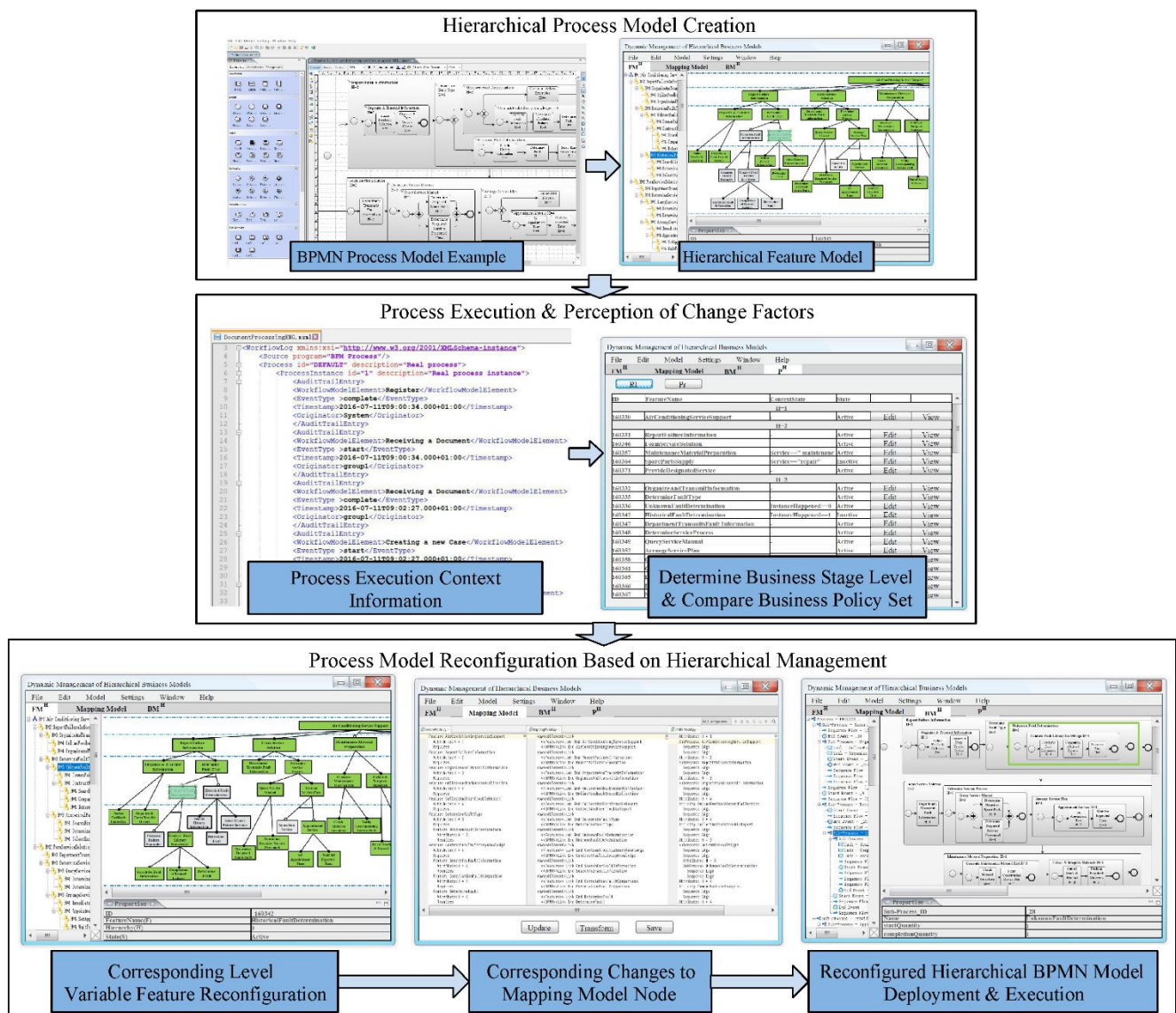


Figure 9. Application examples of business process dynamic adaptation tool.

To verify the effectiveness of the proposed method at the implementation level, this study draws on the method of Xiao [32] to evaluate the time response performance of dynamic adaptation cases. A feature model containing 1000 elements is randomly generated, 25 to 600 variable features are set in sequence, the number of variable features between each group is 25, and a total of 24 sets of simulations are performed. Each group is simulated for the two cases of direct traversal and hierarchical method to find the features that need to be reconfigured, and the time response results are compared. The feature that needs to be reconfigured is always placed in the middle position of all variable features. The simulations are performed on a machine running Windows 7 with a 3.0 GH Intel dual-core CPU and 4 GB of RAM. The response time of the hierarchical method and direct traversal method can be obtained, as shown in Figure 10. The data in the figure are the mean values calculated based on the results of 30 simulations. The coefficient of variation that can represent the fluctuation of the data is calculated according to the simulation results of each group, and values are between 3% and 4%. The time response fluctuation is small. It can be analyzed from the simulation results. Although the operating cost of the hierarchical method is not much different from that of the direct traversal method when the number of variable features is close to 60% of the total number, most of the stages in the actual process

example are mandatory. In the scenario where there are fewer variable business stages in the complex model that need to be reconfigured, the running response time cost of this method can be reduced by up to 50%, which also reflects the good feasibility of the method in the actual business scenarios of the enterprise.

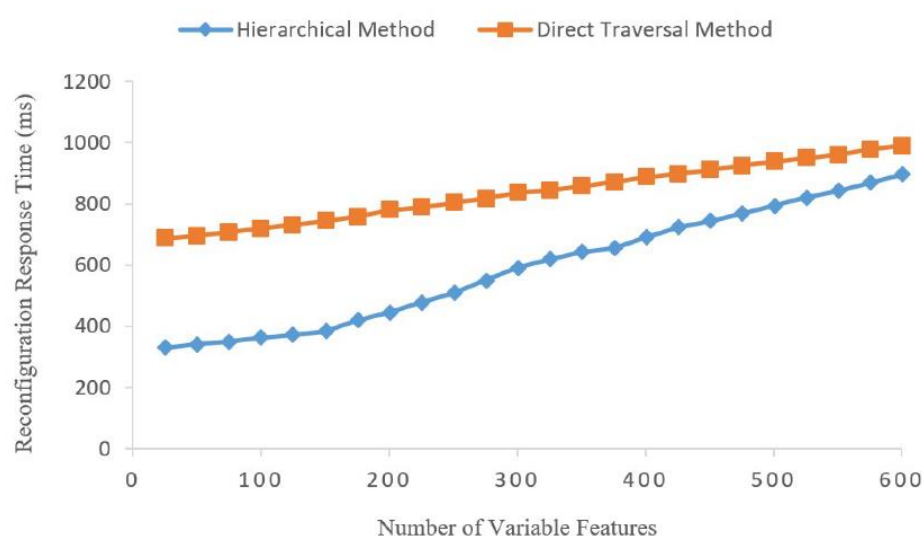


Figure 10. Response time for different methods.

6. Discussion

This research proposes a model-driven dynamic adaptation method for business processes. First, the feature model is established to reflect the hierarchical structure, and different business stages are divided through the decision matrix to expand the hierarchical attributes. As a result, the internal structure of the complex process model is presented to achieve rapid positioning of business changes. Then, the hierarchical business policy is defined to make decisions on the changes of the corresponding level, realizing the automatic reconfiguration of the process model and verifying the correctness of the reconfigured model. Finally, the rapid deployment and execution of the reconfigured model are realized through the mapping model. The main advantages and uniqueness of this research include:

- (1) Improving the overall performance of the dynamic adaptation method based on the hierarchical management idea.

Through the intervention of the hierarchical management idea, this research effectively divides the levels of many subprocesses and activities in the complex process model. On this basis, the business policy for different levels of business stages is formulated to realize the perception and positioning of business changes and the automatic reconfiguration of the corresponding business stages. Compared with the existing business process dynamic adaptation methods, such as related research on data-driven methods, because data-driven methods mostly focus on specific algorithms for training models, such as business anomaly prediction algorithms, simulation decision-making algorithms, etc., they are geared to specific learning targets and may be restricted to the underlying data, and therefore, a dynamic process adaption is not possible in a holistic and more abstract way. This research has established a complete framework for dynamic adaptation of business processes, with hierarchical thinking as the mainline, and the entire dynamic adaptation mechanism of the business process has been improved. Compared with the data-driven methods, the proposed method has better continuity and integrity. Compared with existing model-driven methods, such as predefining decision points and establishing process templates, because the hierarchical feature model derived in this research can effectively display the internal structure of the original process, the problem of slow model reconfiguration response and difficult maintenance of the decision database can be avoided with the aid of hierarchical policy.

- (2) Realizing the automatic deployment and execution of the reconfigured model through the mapping model.

This research uses the method of constructing the mapping model to connect business process models based on a specific standard (such as a BPMN model) and derived process models and realizes the direct execution of the reconfigured model on the workflow engine through the automatic conversion of endpoints. This mechanism can bring enlightenment to the current research on business process dynamic adaptation methods, especially the research related to model-driven methods. Existing model-driven methods for constructing derived process models, such as establishing business process templates, restructuring process models based on bottom-up mechanisms, etc., lack an automatic conversion mechanism to restore derived process models to original process models. This deficiency leads to the fact that even if business changes are sensed and model reconfiguration is performed, the workflow engine cannot directly execute the reconfigured process. The intervention of the mapping model and automated mapping mechanism can better meet the requirements for agile response and rapid deployment of online support during process execution.

The method proposed also has some limitations. This research is based on the existing business process model to explore the proposed method. Since the establishment of a business process model requires a certain amount of expert knowledge, for business processes that have not been modeled through the accumulation of experience and knowledge, this application scenario has not been considered in this research. Secondly, the process model discussed in this research adopts the current mainstream BPMN workflow standard. Other workflow standards such as Business Process Modeling Language (BPML), Business Process Execution Language (BPEL), etc., have not been further verified in this research.

7. Conclusions

This research builds a complete implementation framework for the dynamic adaptation method of business processes. By combining hierarchical management ideas with business process execution, hierarchical decision matrices and hierarchical business policies are designed and proposed. Since the internal structure of the complex process model is clearly presented, rapid response to business changes and adaptive reconfiguration of the process model can be realized. The automatic deployment and execution of the reconfigured model on the workflow engine are realized through the mapping model and endpoint conversion mechanism. Meanwhile, the business process dynamic adaptation tool prototype is developed, combined with the time response simulation results, and the feasibility and effectiveness of the proposed method are verified.

The method proposed analyzes the context state of business processes during the execution phase, mainly using artificially defined rules to judge possible changes in the business environment, but the actual changes may be more complicated and exceed the scope of traditional empirical judgments. In future work, the researchers will combine data-driven related methods, such as business process anomaly detection technology, to achieve more comprehensive coverage of the process change identification link and build a business process dynamic management mechanism that combines model-driven and data-driven methods.

Author Contributions: Conceptualization, L.Z.; methodology, L.Z.; software, L.Z.; investigation, L.Z. and Q.G.; Writing—original draft preparation, L.Z.; writing—review and editing, L.Z., Q.G. and T.L.; supervision, Q.G. and T.L. All authors have read and agreed to the published version of the manuscript.

Funding: The research was funded by the National Key Research and Development Program of China (No. 2018YFB1702601); and the Major Scientific and Technological Innovation Projects in Shandong Province (No. 2019JZZY010442).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

- Mendling, J.; Baesens, B.; Bernstein, A.; Fellmann, M. Challenges of smart business process management: An introduction to the special issue. *Decis. Support Syst.* **2017**, *100*, 1–5. [CrossRef]
- Vasilecas, O.; Kalibatiene, D.; Lavbič, D. Rule and context based dynamic business process modelling and simulation. *J. Syst. Softw.* **2016**, *122*, 1–15. [CrossRef]
- Sun, X.; Hou, W.; Shen, H.; Ying, Y.; Yu, D. Multi-Perspective Online Anomaly Detection Method of Business Processes Based on Context Awareness. Available online: <https://kns.cnki.net/kcms/detail/11.5946.TP.20210329.1220.006.html> (accessed on 19 August 2021).
- Kim, G.Y.; Lee, J.Y.; Park, Y.H.; Noh, S.D. Product life cycle information and process analysis methodology: Integrated information and process analysis for product life cycle management. *Concurr. Eng.* **2012**, *20*, 257–274. [CrossRef]
- Klammer, A.; Gueldenberg, S. Unlearning and forgetting in organizations: A systematic review of literature. *J. Knowl. Manag.* **2018**, *23*, 860–888. [CrossRef]
- Gui, S.C.; Wang, J.X.; Hong, F.; Cao, B. Behavior-based automated process modeling method using recommendation. *Comput. Integr. Manuf. Syst.* **2020**, *26*, 1500–1509. [CrossRef]
- Teinemaa, I.; Dumas, M.; la Rosa, M. Outcome-Oriented Predictive Process Monitoring: Review and Benchmark. Available online: <https://arxiv.org/pdf/1707.06766.pdf> (accessed on 19 August 2021).
- Senivongse, C.; Bennet, A.; Mariano, S. Utilizing a systematic literature review to develop an integrated framework for information and knowledge management systems. *VINE J. Inf. Knowl. Manag. Syst.* **2017**, *47*, 250–264. [CrossRef]
- Palm, A.; Metzger, A.; Pohl, K. Online reinforcement learning for self-adaptive information systems. In Proceedings of the 32nd International Conference of Advanced Information Systems Engineering, LNCS, Grenoble, France, 8–12 June 2020; pp. 169–184.
- Klun, M.; Trkman, P. Business process management—At the crossroads. *Bus. Process. Manag. J.* **2018**, *24*, 786–813. [CrossRef]
- Ubaid, A.M.; Dweiri, F.T. Business process management (BPM): Terminologies and methodologies unified. *Int. J. Syst. Assur. Eng. Manag.* **2020**, *11*, 1046–1064. [CrossRef]
- Malinova, M.; Mendling, J. Identifying do's and don'ts using the integrated business process management framework. *Bus. Process. Manag. J.* **2018**, *24*, 882–899. [CrossRef]
- Letia, I.A.; Goron, A. Model checking as support for inspecting compliance to rules in flexible processes. *J. Vis. Lang. Comput.* **2015**, *28*, 100–121. [CrossRef]
- Bider, I.; Jalali, A. Agile business process development: Why, how and when—applying Nonaka's theory of knowledge transformation to business process development. *Inf. Syst. e-Bus. Manag.* **2016**, *14*, 693–731. [CrossRef]
- Peyman, B.; Kieran, C.; Thomas, C. Agile business process management: A systematic literature review and an integrated framework. *Bus. Process. Manag. J.* **2019**, *26*, 1505–1523. [CrossRef]
- Zacarias, M.; Martins, P.V.; Gonçalves, A. An agile business process and practice meta-model. *Procedia Comput. Sci.* **2017**, *121*, 170–177. [CrossRef]
- Hofstede, H.M.; van der Aalst, W.M.P.; Adams, M.; Russell, N. *Modern Business Process. Automation: YAWL and Its Support Environment*; Springer: Berlin/Heidelberg, Germany, 2010; pp. 103–119.
- von Rosing, M.; von Scheel, J.; Gill, A.Q. Applying agile principles to BPM. *Compleat. Bus. Process. Handb.* **2015**, *01*, 557–581. [CrossRef]
- Martins, P.V.; Zacarias, M. An agile business process improvement methodology. *Procedia Computer Science.* **2017**, *121*, 129–136. [CrossRef]
- Raschke, R.L. Process-based view of agility: The value contribution of IT and the effects on process outcomes. *Int. J. Account. Inf. Syst.* **2010**, *11*, 297–313. [CrossRef]
- Paschek, D.; Luminosu, C.T.; Draghici, A. Automated business process management—In times of digital transformation using machine learning or artificial intelligence. *MATEC Web Conf.* **2017**, *121*, 04007. [CrossRef]
- Pika, A.; Wynn, M.T. A machine learning based approach for recommending unfamiliar process activities. *IEEE Access.* **2021**, *9*, 69–79. [CrossRef]
- Ahn, R.; Goncalves, R.; Hill, T.; Chung, L.; Supakkul, S.; Zhao, L.P. Discovering business problems using problem hypotheses: A goal-oriented and machine learning-based approach. In Proceedings of the IEEE International Conference on Big Data and Smart Computing, Jeju Island, Korea, 17–20 January 2021; pp. 137–140.
- The State of Business Process Management 2014. Available online: <https://www.bptrends.com/bpt/wp-content/uploads/BPTrends-State-of-BPM-Survey-Report.pdf> (accessed on 19 August 2021).
- Pucher, M. Agile-, Adhoc-, Dynamic-, Social-, or Adaptive BPM. Available online: <http://isismjpucher.wordpress.com/2010/03/30/dynamic-vs-adaptive-bpm/> (accessed on 19 August 2021).
- Agostini, A.; De Michelis, G. Improving flexibility of workflow management systems. *Bus. Process. Manag.* **2000**, *32*, 218–234. [CrossRef]
- Understanding Ad Hoc Processes. Available online: https://help.bizagi.com/bpm-suite/en/index.html?understanding_ad_hoc_processes.htm (accessed on 19 August 2021).

-
28. Hallerbach, A.; Bauer, T.; Reichert, M. Issues in modeling process variants with provop. In *Business Process Management Workshops*; Springer: Berlin/Heidelberg, Germany, 2008; pp. 56–67.
 29. La Rosa, M.; Dumas, M. Modelling Business Process Variability. Available online: <http://eprints.qut.edu.au/13358/1/13358.pdf> (accessed on 19 August 2021).
 30. Han, W.L.; Zhang, H.Y. Configurable process modeling techniques for BPMN. *Comput. Integr. Manuf. Syst.* **2013**, *19*, 1928–1934. [[CrossRef](#)]
 31. Yan, Z.; Xu, W. Bottom-up workflow modeling approach for business changes. *Comput. Integr. Manuf. Syst.* **2011**, *17*, 1595–1602. [[CrossRef](#)]
 32. Xiao, Z.; Cao, D.; You, C.; Mei, H. Towards a constraint-based framework for dynamic business process adaptation. In *Proceedings of the 2011 IEEE International Conference on Services Computing*, IEEE, Washington, DC, USA, 4–9 July 2011; pp. 685–692.
 33. Tan, W.; Fan, Y.S. Architecture and key technology for business process management. *Comput. Integr. Manuf. Syst.* **2004**, *10*, 737–743. [[CrossRef](#)]
 34. Madhusudan, T.; Zhao, J.L.; Marshall, B. A case-based reasoning framework for workflow model management. *Data Knowl. Eng.* **2004**, *50*, 87–115. [[CrossRef](#)]
 35. Benavides, D.; Segura, S.; Ruiz-Cortés, A. Automated analysis of feature models 20 years later: A literature review. *Inf. Syst.* **2010**, *35*, 615–636. [[CrossRef](#)]
 36. Haugen, O.; Møller-Pedersen, B.; Oldevik, J.; Olsen, G.K.; Svendsen, A. Adding standardized variability to domain specific languages. In *Proceedings of the 12th International Software Product Line Conference, SPLC, Limerick, Ireland, 8–12 September 2008*; pp. 139–148.
 37. Mazo, R.; Salinesi, C.; Diaz, D.; Lora-Michiels, A. Transforming attribute and clone-enabled feature models into constraint programs over finite domains. In *Proceedings of the 6th International Conference on Evaluation of Novel Approaches to Software Engineering, ENASE, Beijing, China, 8–11 June 2011*; pp. 188–199.
 38. Zhang, W.; Mei, H. Feature-oriented software reuse technology: Development and status quo. *Chin. Sci. Bull.* **2014**, *59*, 21–42.
 39. Alférez, G.H.; Pelechano, V. Context-aware autonomous web services in software product lines. In *Proceedings of the 15th International Software Product Line Conference, SPLC, Munich, Germany, 21–26 August 2011*; pp. 100–109.
 40. Van der Aalst, W.M.P. Business process management: A comprehensive survey. *ISRN Softw. Eng.* **2013**, *2013*, 1–37. [[CrossRef](#)]
 41. Inria, L. The ATLAS Model Weaver (AMW) Project. Available online: https://www.inf.ufpr.br/didonet/amw/AMW_Poster_web.pdf (accessed on 19 August 2021).
 42. Bustinza, O.F.; Gomes, E.; Vendrell-Herrero, F.; Baines, T. Product-service innovation and performance: The role of collaborative partnerships and R&D intensity. *RD Manag.* **2017**, *13*, 1–13. [[CrossRef](#)]
 43. Yaoqiang BPMN Editor Version 5.5.1. Available online: <https://sourceforge.net/projects/bpmn/> (accessed on 19 August 2021).