

# Method for defining multiple homogeneous activities in distributed workflow management system

**Yinzhou Zhu\*, Hui Yang, Baolin Yin**

National Laboratory Software Development Environment, Beihang University, Beijing 100191, China

Received 1 January 2014, www.tsi.lv

**Abstract**

In current process-oriented software systems, most of the processes have large number of parallel activities, which are homogeneous. These parallel activities are often used in the split-merge workflow structure and make the workflow model too complex to manage, as in the traditional workflow management systems each activity has to be defined respectively and bind to one resource. In this paper, we explore a novel method to define the distributed workflow model, which replaces the multiple homogeneous parallel activities with a batch-activity node to simplify the workflow model. An architecture is designed based on this method, which involves the model of organization structure, resource allocation and the sub-workflow. This architecture allows one batch-activity node bind to multiple resources, which are distributed, over a wide geographic area. Real-world scenarios, which are built and implemented based on this architecture, are shown to prove the effectiveness and usefulness of the method.

*Keywords:* workflow management, business process, distributed systems, resource allocation, multiple-instances pattern

**1 Introduction**

In the past decade, workflow management technology has played an important role in the fields of business process management. More and more enterprises consolidate their project implementation into a workflow management system [1, 2]. In the traditional workflow management systems, the tasks on one activity are often allocated to one resource to execute. This model meets the challenge, as the modern business processes have lots of parallel activities which are homogeneous and are distributed

over a wide geographic area. These parallel activities make the workflow model very complex, and increase the difficulty of business process management, especially when the process is deployed on a distributed mobile network. In previous studies, this problem is generally attributed to the workflow patterns involving multiple instances [3]. However, the existing multiple-instances patterns are focus on the run-time mechanism, and they can only support the simple split-merge workflow structures. However, the workflow structures in the real world are far more complicated.

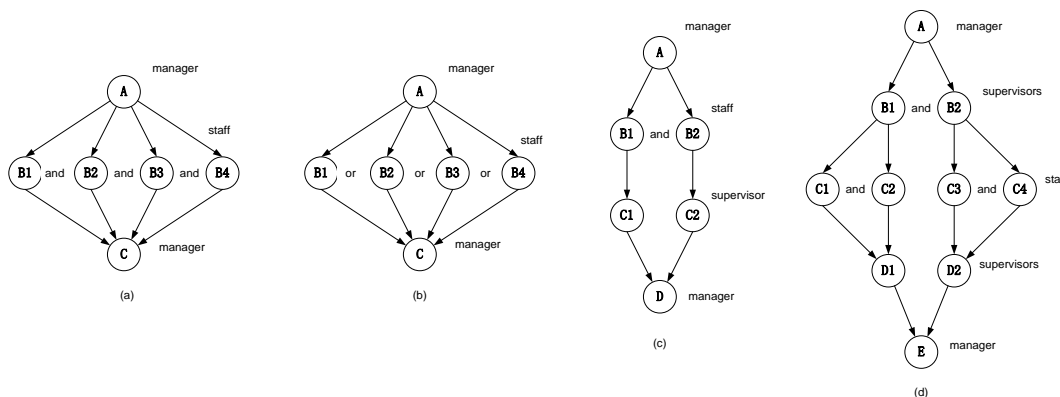


FIGURE 1 The diagram of the split-merge workflow structure.

The split-merge workflow structure often contains a lot of homogeneous activities, which have the similar tasks and are allocated to the resources that belong to the same role. Some typical split-merge workflow structures

are shown in Figure 1. The structures (a) and (b) contain the classical workflow patterns of AND-split and XOR-split respectively. In above structures, the activities with the B prefix are homogeneous activities. The structures (c)

\* Corresponding author.  
E-mail: zhuyinzhou@nlsde.buaa.edu.cn

and (d) contain more complex workflow patterns, which are extended from above basic workflow patterns. These structures are more common in the real-world workflows, and we define them as the multi-level split-merge workflow structures. In these structures, the activities with the B prefix are the first-level homogeneous activities; the C and D activities are the homogeneous activities in the deeper level. In real-world workflows, the number of homogeneous activities can be quite large. Therefore, a method that can simplify the description of the multiple homogeneous activities in distributed environment is necessary.

Along the lines presented in this paper, we propose a description model of multiple homogeneous activities (MHA) that can simplify the description of the complicated multi-level split-merge workflow structure. This model is based on a multiple resource allocation model, which use one "MHA" node to represent multiple activities, and a sub-workflow model, which can extend a "MHA" node to a deeper level workflow. An architecture based on this model is designed and implemented in the EasyWork system. The EasyWork system is a distributed workflow management system that developed by our group [4, 5]. It is based on the persistent messaging mechanism, decentralized distributed workflow model, and the universal data bus mechanism.

This paper focuses on the description model and the runtime support mechanism of the multiple homogeneous activities. And many real-world workflow scenarios developed based on EasyWork system is discussed to prove the effectiveness and usefulness of the architecture.

## 2 Related works

The distributed workflow management technology is mainly applied in scientific computation environments [6]. With the advent of rapid evolution of the business process in large enterprises, however, distributed workflow management is attracting much attention in business process management field these days [1].

The problem of the defining multiple homogeneous activities in the workflow model is generally attributed to the workflow patterns involving multiple instances in previous studies. However, as these multiple-instances patterns covered a lot of ground in a short space, they were only discussed and implemented at a junior level. The workflow management systems that support these patterns were very rare, let alone the distributed workflow management systems [3].

In recent works on distributed workflow management systems, Muthusamy et al developed a flexible and distributed platform to develop, execute, and monitor business process [7, 8]. This platform supports service discovery and composition among multiple resources that offer the same functionality, but cannot simplify the description of the multiple homogeneous activities.

Khalaf and Leymann present a BPEL fragmentation covering data and explicit control dependencies, and an approach to handle fragmenting loops and scopes [9]. Hamann et al present a migration data meta-model for business processes with the ability for runtime migration, which enhance. The flexibility of the distribution of the ad-hoc workflow [10]. The workflow description models in these researches are extended from the BPEL, and cannot describe the multiple homogeneous activities in simple forms.

Besides, most of common business process management systems, such as Staffware, WebSphere, FLOWer, and COSA, are able to support some multiple-instances patterns though the extend mechanisms, such as "bundle model", or "dynamic parallel process management table" [3, 11, 12]. However, their description method is too complex for normal people, and can not support the complicated workflow model that contain multi-level split-merge workflow structures, and most of them can only deployed in a centralized environment.

Our work distinguishes itself from these other approaches by concentrating on a small part of the multiple-instances patterns - multiple homogeneous activities. This help to reduce the model's complexity, make the workflow description simpler and easier to study, and support far more complicated workflow structure than previous patterns.

## 3 MHA model

The description model of MHA is composed of three fundamental models: organization structure model, resource allocation model, and sub-workflow model.

### 3.1 ORGANIZATION STRUCTURE MODEL

The organization structure model, on which the MHA model is based, is derived from EasyWork System. The model consists of four main elements: Department, Workgroup, role and resource. Department and Workgroup are the basic units of the organization structure. They are organized into tree-like hierarchical structures. A department can be only attached to a department, and a workgroup can be only attached to a workgroup. Each department and workgroup can have many roles and resources. The difference between Department and Workgroup is that one resource can and must attached to only one department while it could attached to many workgroups. The role is used to classify the resources by their position or job, such as Manager, Staff, and so on. Each resource could play one or more roles. For example, staff A is the manager of the department X, while at the same time doubled as the manager of the department Y. The diagram of the organization structure model can be seen in Figure 2.

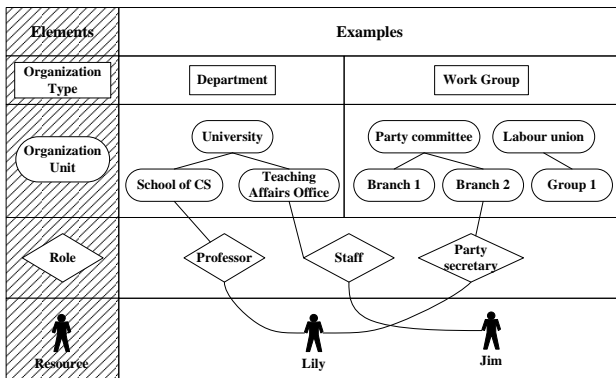


FIGURE 2 The diagram of the organization structure model in EasyWork System

3.2 RESOURCE ALLOCATION MODEL

The resource allocation model is the main part of the MHA model, because it explains the MHA node. A MHA node is an enhanced activity node, which can represent one or more homogeneous activities in order to simply the complexity of the workflow model. The resource allocation description on a MHA node is based on the Organization Unit, which includes departments and workgroups in the organization structure model. The format of the resource allocation description is like this: **quantifier** (*spec\_role*) **in/of** (*org\_unit*)

In this line, **quantifier** is a word that can be chose from ALL, ANY and THE. “ALL” means the tasks which under the *org\_unit* and play the *spec\_role*; “ANY” means the tasks should be allocated to any of the given resources; “The” means the tasks should be allocated to the only one resource that is given. The *spec\_role* represents the role of the resources to which the tasks should be allocated. Besides the user-defined roles, such as “manager”, “staff”, there are three built-in roles: Dept, Group and Node. The Dept represents the departments, the Group represents the workgroups, and the Node represents all the resources under the *org\_unit*. The *org\_unit* represents the Organization Unit that the resources should be attached to. Besides the user-defined organization unit, there are two built-in unit

“this\_domain” or “this\_node”, which are often used in sub-workflow and represents the default organization unit in the current workflow.

TABLE 1. Description examples of the resource allocation

Examples
<b>all</b> ( student ) <b>in</b> ( classA )
<b>any</b> ( teacher ) <b>in</b> ( schoolB )
<b>the</b> ( headmaster ) <b>of</b> ( schoolC )
<b>all</b> ( Node ) <b>in</b> ( this_domain )

The Table 1 gives some description examples of the resource allocation on MHA nodes: It shows the understandability of the description model.

3.3 SUB-WORKFLOW MODEL

The sub-workflow here means the workflow that replaces the functions of an activity in the upper level workflow model. In the MHA model, sub-workflows are used to extend the functionality of a MHA node. It helps the MHA model support the multi-level split-merge workflow structures. The model of the sub-workflow is similar as the common workflow model, which is represented as a directed graph that consists of many activity nodes and transition paths. The difference from the common workflow model is that the start activity node and the finish activity node must be allocated to the same resource as the activity from which this sub-workflow extended.

In traditional workflow management systems, the sub-workflows are often used to increase the reusability of the process fragment. More than that, the sub-workflows in the MHA model are focus on simplifying the description of the multi-level split-merge workflow structure. For example, the workflow structures (c) and (d) in Figure 1 are too complex to define only based on MHA nodes. To resolve this problem, we extend the MHA node though a sub-workflow, which can be seen in Figure 3. The diagrams (c') and (d') in this figure show the description models of the workflow structures (c) and (d) in Figure 1 respectively.

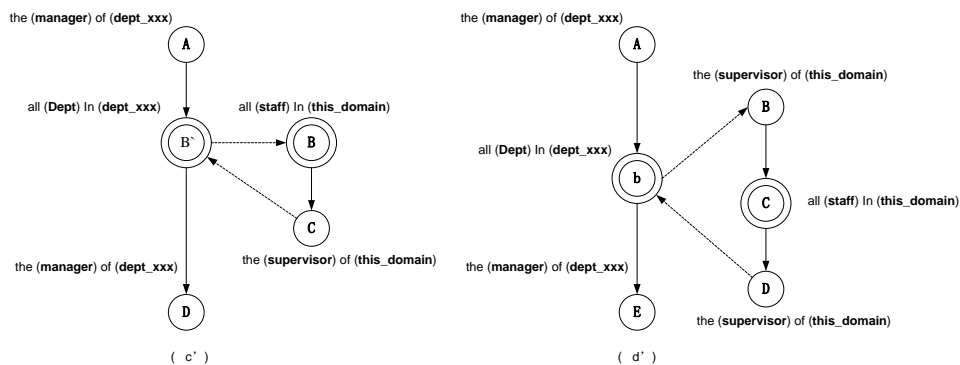


FIGURE 3 The diagram of the sub-workflow model

In the sub-workflow model, which is, extend from a MHA node, the tasks of activities can be allocated to the resources that play the role under the default organization. The default organization is not an absolute organization, but the each organization defined in the MHA node. This method opens the possibility of defining the workflow model through a recursive process, so that simplifies the description of the multi-level split-merge workflow structure.

#### 4 Execution Mechanism

The basic execution mechanism is provided by EasyWork distributed workflow management system. Besides, we extend the mechanisms to support the XOR-split workflow pattern and the data convergence in MHA synchronizing merge workflow pattern.

##### 4.1 ARCHITECTURE OF EASYWORK SYSTEM

The architecture of the EasyWork system, which is shown in Figure 4, aims to define, execute and monitor the workflows for the cross-regional enterprises, which often have several highly autonomous subsidiaries. The network architecture of the EasyWork system is a hybrid structure, which is based on the peer to peer network and the client/server framework. There are two kinds of nodes in the EasyWrok network: EasyWork Server node and the EasyWork Client node. The EasyWork Servers are the basic nodes, which are used to store and dispatch the distributed workflow instances. The relationships between EasyWork Servers are symmetrical, while the EasyWork Clients are client nodes of the EasyWork Server. The EasyWork Platform is installed on every EasyWork Server, and offer the access interface that allows users to get and do their job in remote EasyWork Clients though a standard web browser.

EasyWork Process Definition Server is a kind of EasyWork Server on which the workflow definition tool is installed. The workflow definition tool is used by workflow administrators to define, compile and deploy the distributed workflow. After a workflow model has been designed in the Process Definition Server, it will be split into several segments by activities, and be compiled to configuration files. These configuration files will then be distributed to EasyWork Servers, which are defined as the computing resources of the activities.

The EasyWork Platform on EasyWork Server is composed of three main parts: the workflow engine, the user task manager and the application framework. The workflow engine is used to receive, store, and dispatch the workflow instances, and invoke the application to process the tasks of the activities. The user task manager offers the access interface to the workflow system, shows the task-lists, and communicates with the workflow engine to process user commands. The application framework is a set of applications, which are created, based on the components with application-level

granularity [4, 13, 14]. These applications, which are invoked by the workflow engine to process the tasks of the activities, are executable programs, such as the executable binary files of the operational systems of Microsoft Windows or UNIX, or the web pages that could be interpreted by http server or client browser.

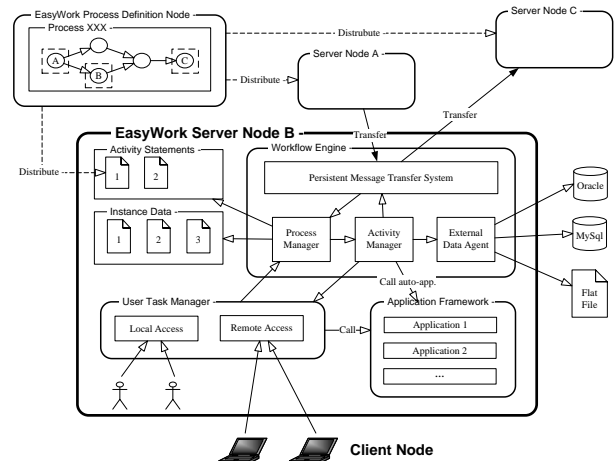


FIGURE 4 The architecture diagram of the EasyWork system

##### 4.2 MECHANISM FOR XOR-SPLIT PATTERN

The workflow models that include the MHA nodes or the sub-workflows is converted to the normal workflow model which only contains the simple activities before the deployment. Then the tasks of the activities are distributed to the resources that would execute them. To support AND-split pattern in the EasyWork system is easy and direct, but to support XOR-split pattern is difficult, because the latter involves the dynamic work allocation in the runtime phase in the distributed environment.

To resolve this problem, a program named "XS" is developed and is added to the end of task queue of every XOR-split activity. The XS knows the number of the activities that can be select as the successor by the XOR-split activity, and the identifier and the weight of each of these activities. When XS is executed, it select the proper activity as the successor based on the serial number of the current workflow instance, and then route the workflow instance. The successor selection can be made based on round-robin, weight-based or random algorithm in current system, and can be extended easily by modifying the XS program.

##### 4.3 MECHANISM FOR DATA CONVERGENCE

Data convergence is a difficult problem when merging multiple parallel workflow instances in AND-split pattern. In the traditional workflow management systems, the workflow data of the parallel workflow instances are often imported into the database individually, and then are fetched together by the synchronization activity. However, this approach increases the coupling between

the workflow applications and the database, and is inappropriate in the distributed environment especially.

To solve this problem, our system adopts a specialized model to describe the data convergence in the synchronization activity, and provide a specialized data type – “collect data”. When the workflow instance go through the MHA node which split the instance into multiple parallel workflow instances and merge them after finish the tasks, EasyWork system would gather all collect data and combine them into an array.

The format of the declaration of the collect data is like this: **Define\_collect\_var** ( collect\_array )

In this line, *collect\_array* is the workflow data which would be used to store the combined data of the multiple parallel workflow instances. The data model of the *collect\_array* after the data convergence is a relative table, which is shown in following table.

TABLE 2. The model of the collect data

Collect Array
{node1_id, data1, ...},
{node2_id, data2, ...},
{ ... }
{nodeN_id, dataN, ...}

In Table 2, the variables from “node1\_id” to “nodeN\_id” store the identifier of the multiple activities represented by MHA; the variables with the “data” prefix store the values of the collect data of multiple parallel workflow instances. The data model of the variables that are combined can be simple data, list or relative table. A diagram of the data convergence model when using MHA nodes can be seen in Figure 5. The model at the left of the figure is a normal AND-split workflow, and the model at the right side is the same model but adopts the MHA node.

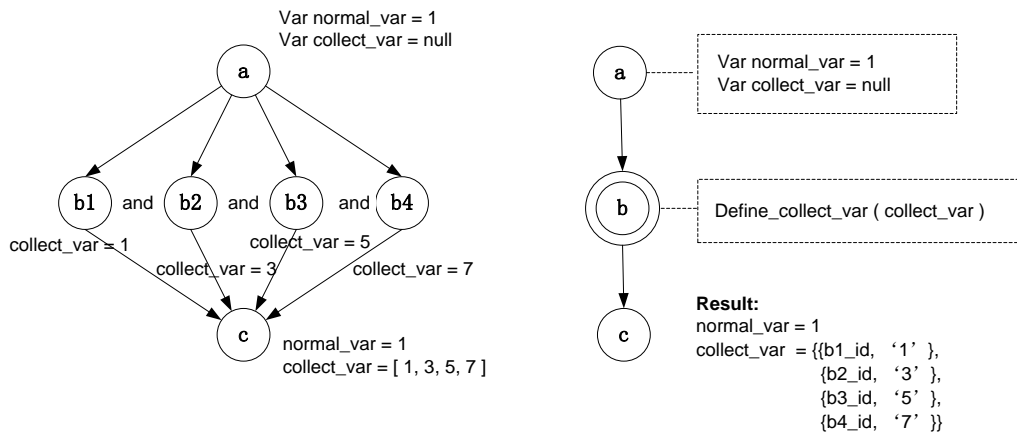


FIGURE 5 The diagram of the data convergence model when using MHA nodes

### 5 Scenarios

The real world scenarios, which implemented based on EasyWork system, are discussed in this section. As shown in Table 3, the scenarios are grouped into three classes:

(1) Office automation (OA) applications, which are the collaboration systems based on the workflow. The OA system is a type of classic workflow system, and plays an important role in paperless office. The OA systems are widely used almost in all fields without the limit of the industry or geographic region. The key element of the OA is the form, which often is easy to be customized based on the web components in the EasyWork system.

(2) Enterprise resource planning (ERP) system, which is a more integrated platform, compared to OA systems, used to manage various kinds of information in the enterprise. From the model layer, such as the data model and business logic, to the view layer, such

as the view of the decision support, all kinds of the information in the enterprise are integrated together into a strict architecture. The design of the data model and process in ERP system are more professional than other information management systems, and are more difficult to be implemented.

(3) Report generation systems, which are the process-oriented systems used to collect and summarize the information, which is distributed in different areas, and generate the Summary Report finally. This kind of application is very common in different areas, and is not easy to implement for the complex computation and distributed architecture. The real-world report generation scenario implemented based on EasyWork system is the Freshwater Quality Monitoring (FQM) which is a process required by the National Oceanic Administration of China to investigate the freshwater quality in the coastal areas and provide a summary report every year.



TABLE 3. Real world workflows implemented by EasyWork system

	Class	Total number	AS5	XS5	AS20	XS20	Depth
OA	Document Circular	3	2	3			
	Administrative Approval	6	2	4			3
	Personnel Management	9	6	3	1		1
	Financial Management	3	2	2			2
	Product Management	6	1	6			
	Customer Service	5	1			3	
ERP	Data Maintenance	15	6	10	3	4	7
	Purchase Processes	6	2	4		1	2
	Sale Processes	7	2	4	2	1	3
	Bank Processes	4	2	3			
	Inventory Management	4	2	4			
	Production Processes	5	1	4			
	Report Query	18	3	12	1	1	
	Other Processes	4	1	3	3	1	1
<b>Summary</b>		95	33	62	10	11	19
		100%	34.74%	65.26%	10.53%	11.58%	20.00%

In the above scenarios, there are 95 workflows implemented by EasyWork system. The details of the workflows are shown in Table 1. The “AS5” and “AS20” represent the workflows that involves AND-split workflow pattern and the max number of the multiple heterogeneous activities is more than five and twenty respectively. In the same way, the “XS5” and “XS20” represent the workflows that involves XOR-split workflow pattern and the max number of the multiple heterogeneous activities is more than five and twenty respectively. The “depth” means the total number of levels of the multi-level split-merge workflow. The Table 1 shows that, nearly half of the business workflows that involve the AND-split workflow pattern are suggested to adopt the MHA model, and 20% of the them have to adopt the MHA model; nearly 4/5 of the business workflows that

involves the XOR-split workflow pattern are suggested to adopt the MHA model, and 15% of them have to adopt the MHA model. Although these statistics are only based on the experience gathered in our work history, they still explain the seriousness of the MHA problem to some extent.

5.1 CASE STUDY

To illustrate the effectiveness of the MHA description model, a real-world scenario which described by the EasyWork system is shown in this section. The workflow in this scenario is used by a state-owned enterprise to reimburse the project expenses spent during the last year. The diagram of the organization structure of the enterprise is shown in Figure 6.

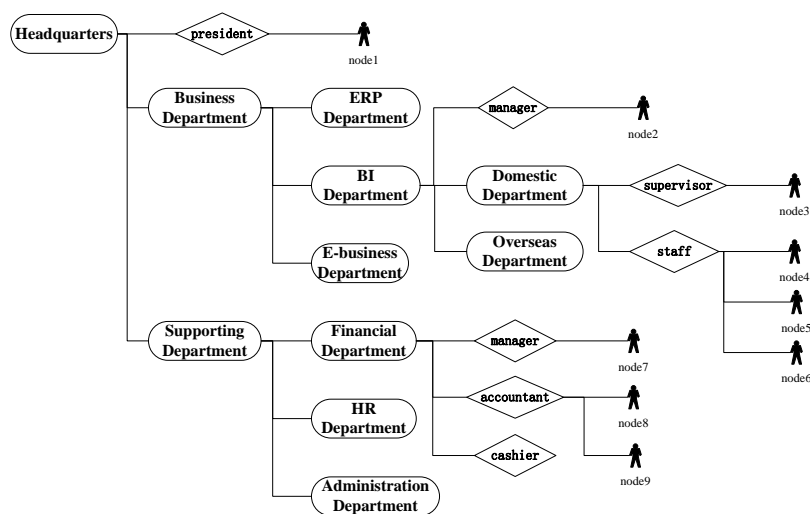


FIGURE 6 The diagram of the enterprise organization structure

The reimbursement workflow is started by the manager of the financial department, and then the reimbursement instructions are forwarded to every

business department. After every supervisor filled out the expenses claim form, the forms are grouped and submit to their managers to approve. If the application is not

approved, it would be sent back and be filled again. After the verification of the department manager, all expenses claim forms are gather together and transfer to the accountant of the financial department to do the second

check and create the expenses report. Then the report is checked by financial manager and is delivered to the cashier to do the banking business. The model of the workflow can be seen in Figure 7.

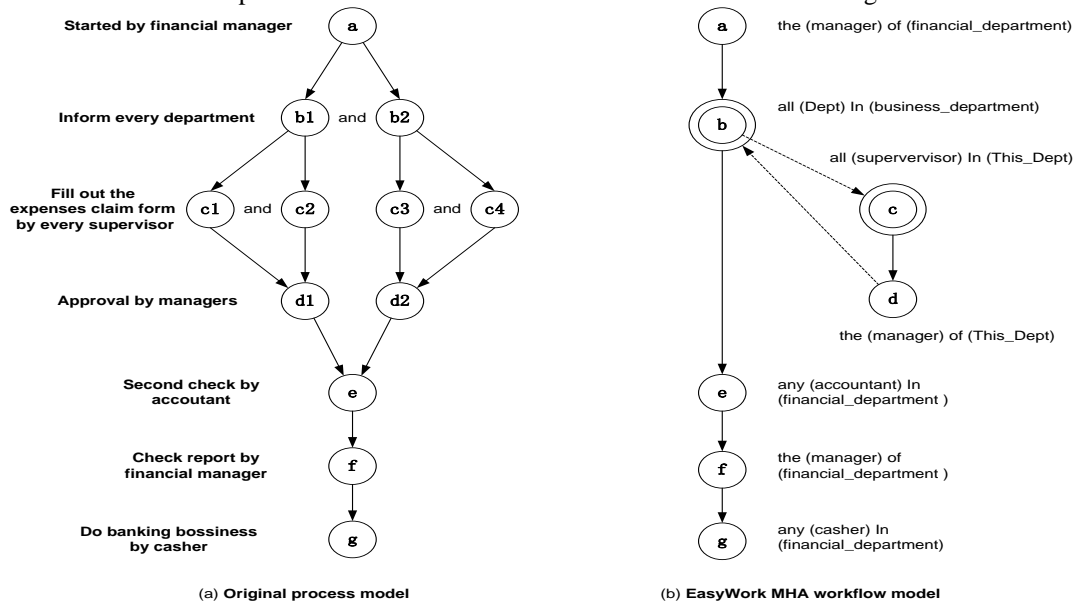


FIGURE 7 The model of the annual reimbursement workflow for project expenses

This reimbursement workflow is a complex workflow that has two-level split-merge workflow structure, but it is very common in the real-world scenarios. In the enterprise that was described above, the actual number of the departments attached to the business department is 16. Moreover, each of these departments has more than 10 sub-departments. This means in second level of the workflow model, there are about 200 heterogeneous activities. It is very difficult to manage this complex workflow model by traditional workflow management system. However, EasyWork system can play a maximum efficacy in this situation. Two MHA nodes and a simple sub-workflow are enough to describe this model. Besides, the descriptions of resource allocation are short

and easy to understand. It shows the high efficiency and the usability of the MHA description model.

5.2 SYSTEM COMPARISON

This section discusses the comparison between EasyWork system, which implements the MHA description model and the other mainstream workflow management systems. The workflow management systems to which we compared are Staffware, COSA, FLOWer, WebSphere MQ Workflow and SAP/R3. The comparisons can be seen in Table 4. Some part of the information in this table is derived from [3].

TABLE 4. Comparisons between EasyWork system and other WFMS

Systems	AND-split MHA	XOR-split MHA	Multi-level MHA	Supporting distributed workflow	MHA data convergence mechanism	MHA task auto distribution
Staffware 9	-	+	-	-	-	-
COSA 4.2	-	-	+ / -	+	-	+ / -
FLOWer 3	+	+	-	-	-	-
Meteor	-	-	-	+	-	-
WebSphere MQ Workflow 3.3.4	-	+	-	-	-	-
SAP R3	+ / -	+	-	-	-	-
EasyWork	+	+	+	+	+	+

The table shows that the Staffware and Websphere MQ Workflow are only support the basic XOR-split MHA description. The SAP/R3 and FLOWer can support basic AND-split MHA description but cannot support multi-level MHA description. The COSA support the multi-level workflow and task auto distribution, but it

cannot support MHA description well. In the last, all these traditional workflow management system do not support MHA data convergence. Compare to these system, EasyWork support all these functionalities. Besides, the description rules in EasyWork system are very simple and easy to learn and use.




## 6 Conclusions

In this research, we propose a workflow management framework for multiple heterogeneous activities. This framework supports the description and execution of the MHA nodes, and increases the efficiency of workflow model management. Compared to other mainstream workflow management systems, this framework offers much fuller support on MHA management. The MHA description model involves organization structure, resource allocation model, and sub-workflow description model, and offers two key mechanisms for the automatic task distribution in the XOR-split pattern and the data convergence in AND-split pattern. In the future, the relationship between resource allocation model and the sub-workflow description model will be enhanced in order to increase the flexibility on defining the resource allocation in multi-level workflows.

## References

- [1] van der Aalst W M P 2013 Business Process Management: A Comprehensive Survey *ISRN Software Engineering* **2013**  
 [2] Sinur J, Hill J B 2010 *Magic Quadrant for business process management suites* Gartner RAS Core research note 1-24

- [3] van Der Aalst W M P, Ter Hofstede A H, Kiepuszewski B, Barros A P 2003 *Distributed and parallel databases* **14**(1) 5-51  
 [4] Yinzhou Zhu, Baolin Yin 2013 *Journal of U- & E-Service, Science & Technology* **6**(4)  
 [5] Yinzhou Zhu, Baolin Yin 2013 *Journal of Computational Information Systems* **9**(20) 8297-306  
 [6] Deelman E, Gannon D, Shields M, Taylor I 2009 *Future Generation Computer Systems* **25**(5) 528-540  
 [7] Muthusamy V 2011 *Flexible Distributed Business Process Management* University of Toronto.  
 [8] Ganesan S, Yoon Y, Jacobsen H A 2011 NIÑOS take five: the management infrastructure for distributed event-driven workflows *In Proceedings of the 5th ACM international conference on Distributed event-based system, DEBS 11* 195-206  
 [9] Khalaf R, Leymann F 2012 *Information Systems* **37**(6) 593-610  
 [10] Hamann K, Zaplata S, Lamersdorf W 2012 Process instance migration: Flexible execution of distributed business processes *In Software Services and Systems Research-Results and Challenges (S-Cube) 2012 Workshop on European IEEE*.  
 [11] Russell N, van der Aalst W M P, ter Hofstede A H, Edmond D 2005 Workflow resource patterns: Identification, representation and tool support *Advanced Information Systems Engineering Heidelberg: Springer*.  
 [12] Russell N, ter Hofstede A H, Edmond D, van der Aalst W M P 2005 *Conceptual Modelling-ER* **3716** 353-68  
 [13] Guojie Jie, Baolin Yin, Qiyang Zhao 2011 *Journal of Software* **6**(3) 374-85  
 [14] Guojie Jin, Baolin Yin, Qiyang Zhao 2011 *Computer Integrated Manufacturing Systems* **17**(2) 425-32

Authors	
	<p><b>Yinzhou Zhu</b></p> <p><b>University studies:</b> M.S. in 2006 and currently is a Ph.D. candidate, both in Computer Science and Engineering Department at Beihang university (Beijing).  <b>Scientific interest:</b> workflow management, enterprise application integration and distributed systems.</p>
	<p><b>Hui Yang</b></p> <p><b>University studies:</b> Bachelor. in 2011 at Xidian University and currently is a M.S. candidate, in Computer Science and Engineering Department at Beihang university (Beijing).  <b>Scientific interest:</b> workflow management, Web component and database.</p>
	<p><b>Baolin Yin</b></p> <p><b>Current position, grades:</b> Professor in Computer Science and Engineering Department at Beihang university (Beijing), and is currently head of the State Key Laboratory of Advanced Software Development.  <b>University studies:</b> He received his Ph.D. in Artificial Intelligence in the University of Edinburgh (1984).  <b>Scientific interest:</b> His research activity is concerned with distributed systems, workflow management, image identification, and computer security.</p>