Journal of Southeast University (English Edition) Vol.19 No.4 Dec. 2003 ISSN 1003—7985

Research on workflow meta model supporting activity based costing

Zhou Xiaojun Cao Jian

(Computer Integration Technology Laboratory, Shanghai Jiaotong University, Shanghai 200030, China)

Abstract: Activity based costing (ABC) is a method which can solve many limitations of the traditional cost systems in manufacturing management. In this paper, we investigate how to integrate ABC with workflow technology, and build a workflow meta model supporting ABC. Firstly, the concept and concept model of activity based costing (ABC) are introduced. Next, the meta model of P_ PROCE (Process, Product, Resource, Organization, and Cost & Evaluation) is presented. Then the cost meta model is defined by adding ABC to P_ PROCE model. Object constraint language (OCL) is used to express meta model and constraints. Finally, we show an enterprise modeling and simulation tool based on the workflow meta model. We can systematically construct an enterprise model and easily and efficiently conduct simulation. Moreover it enables us to analyze and evaluate business processes and its costs.

Key words: activity based costing; workflow; meta model; object constraint language; enterprise model

With the changing manufacturing environment, many managers have started to believe that traditional, volume-based product-costing systems cannot accurately reflect product costs. The product-costing systems structured on single, volume-based cost drivers, such as direct labor or machine hours, often tend to overcost high-volume products and undercost low-volume or complex products. To address these problems, more and more firms have adopted an activity based costing system based on multiple cost drivers.

Cost reduction is an important target for adjusting the business process, so the integration of activity based costing (ABC) with enterprise modeling and simulation technology has interested many researchers. Ref.[1] introduced an approach to integrating the ABC technique within the framework of GRAI integrated methodology (GIM) to assist business process reengineering justification and evaluation. Ref.[2] introduced a simulation aided approach of economy for organizational changes based on ABC which was supported by the simulation package FEMOS. Ref.[3] applied discrete event simulation to the activity based costing of manufacturing systems.

Nevertheless these researches did not provide an integrated approach to modeling and simulation. Nor did they reflect the close relations between cost and product, process, resource and organization. Thus they cannot provide the information about business process accurately. This paper introduces the workflow meta model named P_ PROCE and defines its cost sub meta model. We build an enterprise modeling and simulation tool based on the workflow meta model.

The remainder of this paper is organized as follows. First we introduce activity based costing and workflow meta model. Section 2 presents the workflow meta model called P_PROCE. In section 3, we explore the cost sub meta model which adds activity based costing to P_ PROCE meta models, and OCL as a language to express workflow constraints. In section 4, we provide an enterprise modeling and simulation system based on the meta model. At last, some conclusions are drawn for this paper.

1 Activity Based Costing and Workflow Meta Model

Activity based costing was introduced by Kaplan and Cooper of Harvard Business School as an alternative to traditional accounting techniques in the 1980's[4]. The main idea behind ABC is to classify indirect (overhead) costs and to allocate them to customer-required products or services, based on the activities needed to produce these products. In this manner it is possible to cost out the products more accurately and completely[5]. It traces the overhead costs to the activities that consume the costs, and then to the cost objects which consume the activities. Fig.1 shows the ABC concept model which contains six layers.

Received 2003-05-20.
Foundation item: The National High Technology Research and Development Program of China (2001AA415310).
Biographies: Zhou Xiaojun (1975—), male, graduate; Cao Jian (corresponding author), male, associate professor, cao-jian@cs.sjtu.edu.cn.

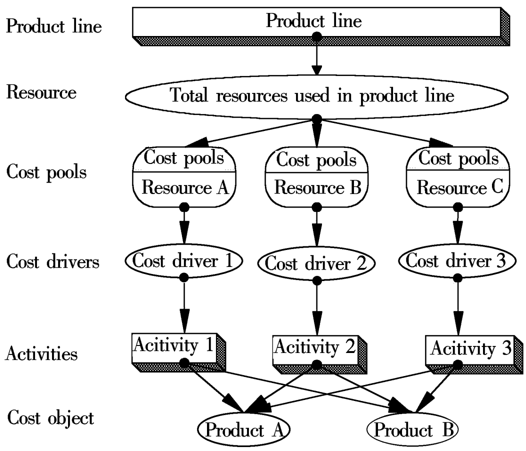

Fig.1 The ABC concept model

In a product line, many resources will be consumed while cost objects are produced. A cost object is the product or service being product. The first step of the model groups the total resources into cost pools and determines the cost of each cost pool. A cost pool is a collection of costs to be assigned to a set of cost objects. Every cost pool corresponds to a cost driver. Some examples of cost pools include rental expense, salaries, and machinery. Cost driver is a characteristic of an activity, or event that results in the incurrence of costs by that activity or event. Cost drivers measure the frequency and intensity of the demands placed on cost pools by cost objects, which have an important attribute named the pool rate. The pool rate is the cost per unit of the cost driver for a particular activity cost pool. Various tasks performed in the organization are then grouped into major, functionally homogeneous activity categories, such as R&D and quality management. Each activity consumes different amounts of one or more cost drivers. The cost of performing the activities is determined by allocating the costs from the cost pools to each activity according to the cost drivers consumed. Once the activity costs are determined, they are then allocated to the cost objects.

The purpose of workflow meta-models is to provide a formal language or representational system for specifying an interrelated family of workflow models. In order to establish an enterprise modeling and simulation tool that can not only support activity based costing but also reflect the close relations among cost, product, process, resource and organization, a workflow meta model supporting ABC should be built firstly. In the next two sections, a workflow meta model named P_ PROCE is presented, which supports activity based costing.

2 A Glance at Meta Model of P_ PROCE

The workflow meta model of P_ PROCE (Process, Product, Resource, Organization, and Cost & Evaluation)[6] is a reference model for enterprise which was developed by the Computer Integration Technology Laboratory, Shanghai Jiaotong University (SJTU-CIT Lab). It is a multi-view meta model which contains five sub-models, respectively called: process model, product model, resource model, organization model and cost & evaluation model. Fig.2 shows the UML class diagram representing the meta model of P_ PROCE. We will introduce the classes in the metal model of P_ PROCE abbreviated as follows:

Workflow In a business model, there is only one Workflow object representing the whole business process, which contains many other objects, such as Process object, Resource object, and Task object, etc.

Process The class Process is used to specify a business process, which contains a set of Tasks.

Resource Something that can be used for support or to help execute a Task. While the Task is executed, the Resources are used and consumed. The class Resource represents the material and the people, which are needed in the Task.

Organization The Organization in enterprise is a group of persons organized for a particular business purpose. There are two types of organization structure, respectively called, department oriented and team oriented.

Task The Task is the execution step of the process. There is an object decomposition hierarchy for a Task object. So there are two types of Task, respectively called, atomic Task and complex Task. A complex Task can be made up of an atomic Task and a complex Task. Tasks can be executed sequentially, iteratively, conditionally, or concurrently.

Product The class Product represents the production of the Task. A Product can be composed of aggregations of other Products. In terms of costing, values are added to a Product and Resources are used and consumed while an activity is performed.

Metric, ResMetric, OrgMetric, TaskMetric, ProMetric The Metric class is the basic part for the Evaluation sub model in P_ PROCE. ResMetric, OrgMetric, TaskMetric, and ProMetric are the subclasses of Metric and inherit all methods and properties from the Metric class.

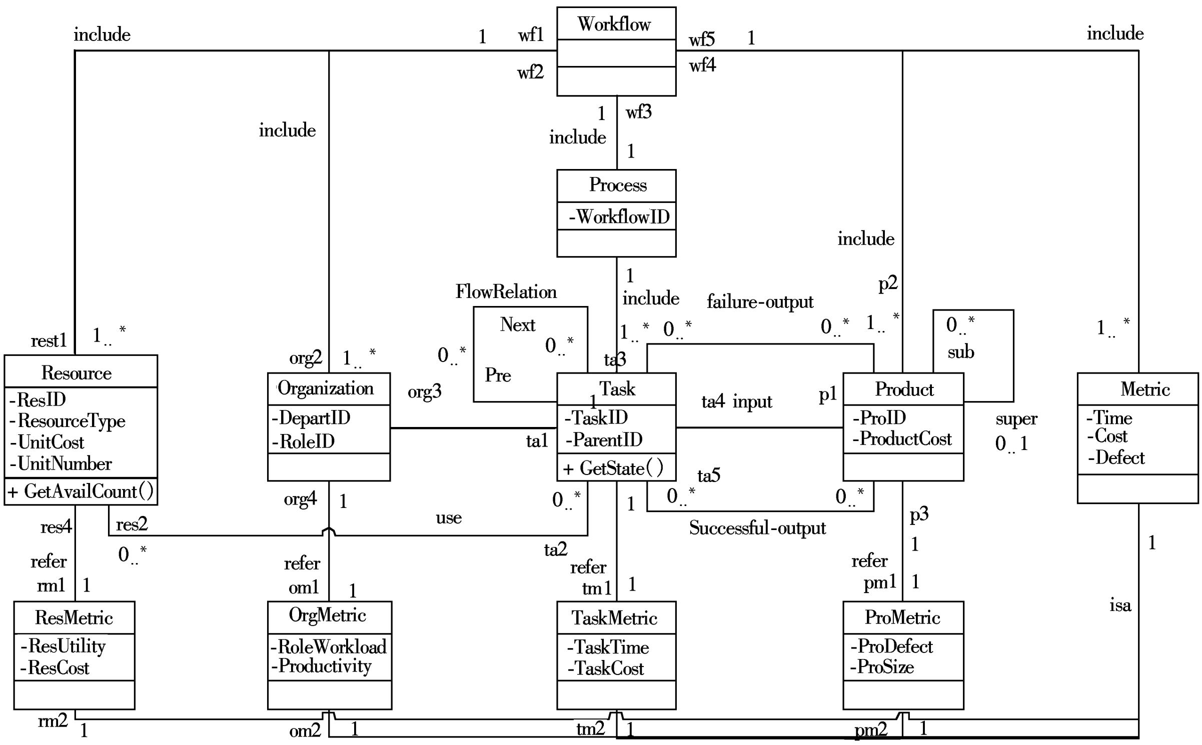

Fig.2 The meta model of P_ PROCE

In the meta model of P_ PROCE, we only define the primary classes and the relations between these classes. In order to describe a certain aspect of P_ PROCE in detail, a corresponding sub meta model should be defined. In the sub meta model, there are not only primary classes, but also more new classes. In the next section, the cost sub model will be defined.

3 Adding Activity Based Costing to P_ PROCE Models

In this section we present a cost sub meta model which extends meta model of P_ PROCE by concepts from ABC. We also show how OCL can be used to formally model constraints.

3.1 The classes in the cost sub metal model of P_ PROCE

Fig.3 shows the UML class diagram representing the cost sub meta model. The eight classes correspond to the concepts role, product, activity, task, resource, product line, cost pool and cost driver. Some classes are defined in the main metal model of P_PROCE. And some new classes are added to the cost sub metal model.

Role A Role represents skill sets and task assignment. The right of operating a Task is assigned to one or more Roles. Since the class Role represents skill sets and task assignment, it is used to specify the mapping of Tasks from concrete Resources.

Activity Activity is the instantiation of task for a given case, which is comprised of operation and object. A task can be instanced for many times.

ProductLine In a ProductLine, many of the same kinds Products can be produced.

CostPool The CostPool is a collection of costs to be assigned to a set of cost objects. Every CostPool corresponds to one CostDriver.

CostDriver CostDriver is a characteristic of an activity or event that results in the incurrence of costs by that activity or event. While the Activity is performed, Cost Drivers are consumed. And at the same time, value is added to Product.

Fig.3 also lists some representative attributes and methods. These attributes are illustrative and not exhaustive. For example, we draw attention to the cost of a product, so there is an attribute called ProductCost in the class Product.

3.2 The web relations in the metal model of P_ PROCE

There are associative links between business process classes, so the classes of P_ PROCE are not isolated. The web relations in the metal model of P_ PROCE identify processing orders or directions for message passing, etc.

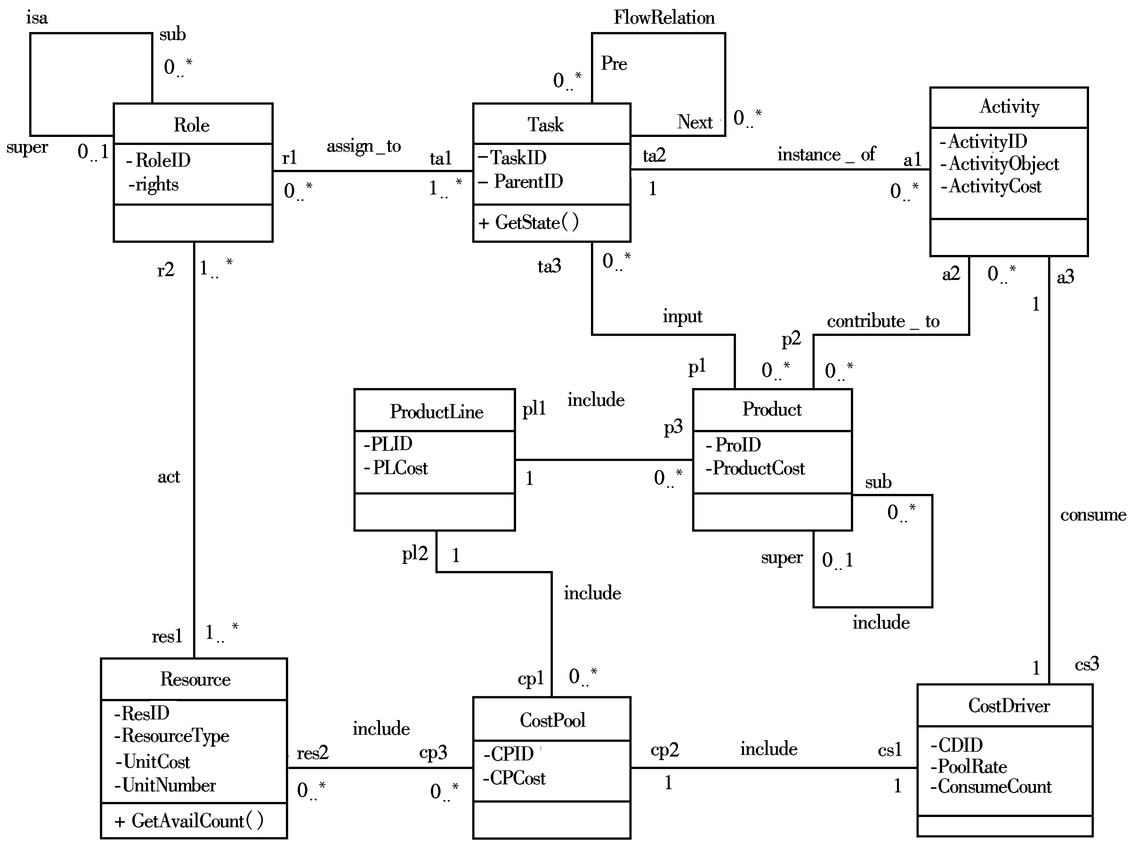

Fig.3 The cost sub meta model of P_ PROCE

Definition 1 Include relation $R_{\text{include}} = \{\langle x, y\rangle \mid x \in C, y \in C$, and y is a part of $x\}$. C is the set of all classes in the metal model of P_ PROCE. The Include relation pair forms the decomposition hierarchy for an object in a business process model. According to this definition and Fig.1, $R_{\text{include}} = \{\langle$Product, Product$\rangle$, $\langle$ProductLine, CostPool$\rangle$, $\langle$CostPool, CostDriver$\rangle$, $\langle$Resource, CostDriver$\rangle\}$.

Definition 2 Flow relation $R_{\text{Flow}} = \{\langle x, y\rangle \mid x \in C, y \in C$, and there is a sequence relation between x and $y\}$. The R_{Flow} forms a processing link among a set of objects of Task, which are used to model sequential, conditional, parallel, and iterative routing.

Definition 3 Input relation $R_{\text{input}} = \{\langle x, y\rangle \mid x \in C, y \in C$, and x is the condition for the execution of $y\}$. The R_{input} specifies product that are input to the Task. According to this definition and Fig.1, $R_{\text{input}} = \{\langle$Product, Task$\rangle\}$.

Definition 4 Assign relation $R_{\text{assign\_to}} = \{\langle x, y\rangle \mid x \in C, y \in C$, and assign x to y, y is respond to $x\}$. It specifies the role assigned to perform the Task.

Definition 5 Instantiation relation $R_{\text{instance\_of}} = \{\langle x, y\rangle \mid x \in C, y \in C$, and x is the instantiation of $y\}$.

For example, because an Activity is an instantiation of a Task for a given case, $\langle$Activity, Task$\rangle \in R_{\text{instance_of}}$.

Definition 6 Contribute to relation $R_{\text{contribute\_to}} = \{\langle x, y\rangle \mid x \in C, y \in C$, and x will adds value to $y\}$.

For example, because while an activity is performed, value will be added to product, $\langle$Activity, Product$\rangle \in R_{\text{contribute_to}}$.

Definition 7 Consume relation $R_{\text{consume}} = \{\langle x, y\rangle \mid x \in C, y \in C$, and y will be consumed by $x\}$

For example, because while an activity is performed, cost drivers will be used and consumed, $\langle$Activity, CostDriver$\rangle \in R_{\text{consume}}$.

Definition 8 Act relation $R_{\text{act}} = \{\langle x, y\rangle \mid x \in C, y \in C$, and x has the right and responsibility of $y\}$. $\langle$Resource, Role$\rangle \in R_{\text{act}}$ means that Resource has the right of Role and is responsible for the Role.

In the UML class diagram, a relation is represented by a line. At the two ends of the line, there is a number or number range called multiplicity, which indicates the number of objects in the relation. For example, because of $\langle$Activity, Task$\rangle \in R_{\text{instance_of}}$, the multiplicity of role ta2 of association include is 1 and the multiplicity of role a1 is 0⋯ * , therefore an Activity object can be instanced as a Task for zero or many times.

3.3 Modeling constraints for the cost meta model in OCL

In the object-oriented modeling of a graphical model, like a class model, there does not contain enough information for a precise and unambiguous specification. There is a need to describe additional constraints about the objects in the model. So OCL has been developed within the IBM Insurance division and allows for the definition of integrity constraints. It is a formal language that remains easy to read and write[7]. OCL is based on set theory and can be used to specify invariants on classes and the relationships among classes in UML class diagrams.

In order to define metal model precisely, OCL is used to express the cost meta model and the constraints on the cost sub meta model. In an expression of OCL, "self" represents an instance of a class. Through the navigation paths which are composed of symbol "." and the relationships between classes, we can access other objects and these attributes. If the multiplicity at the end of the navigation path is bigger than 1, then the expression represents a set of objects or attributes. Otherwise the expression represents an object or an attribute. For example, when "self" is a contextual instance of CostPool, thus self.cs1 is the CostDriver object that corresponds to the CostPool object, and self.res2 represents the set of Resource objects that belong to the CostPool object.

Property 1 ProductLine

$$\text{self.PLCost} = \sum_{i=1}^{n} \text{self.cp1.CPCost}$$

self.PLCost refers to the cost of product line which is consumed in a period of time. self.cp1 is the set of CostPool which belongs to the product line. $\sum_{i=1}^{n}$ self.cp1.CPCost represents the sum of all these cost pools' costs. The expression means that the cost of the product line is equal to the sum of costs of the cost pools belonging to the product line.

Property 2 Activity

$$\text{self.ActivityCost} = \text{self.cs3.PoolRate} * \text{self.cs3.ConsumeCount}$$

self.ActivityCost refers to the cost of an activity object and self.cs3 refers to the cost driver object for the activity object. The expression states that the cost of one activity is equal to the pool rate of cost driver multiplied by ConsumeCount, which is the number of cost drivers consumed by the activity.

Property 3 Product

$$\text{self.ProductCost} = \sum_{i=1}^{n} \text{self.a2.ActivityCost}$$

The expression states that the cost of a product is equal to the total cost of all the activities of the producing progress.

Property 4 CostPool

$$\text{self.CPCost} = \sum_{i=1}^{n} \text{self.res2.UnitCost} * \text{self.res2.UnitNumber}$$

The expression states that the cost of cost pool is equal to the sum cost of resources which are used and consumed in the cost pool.

Property 5 CostDriver

$$\text{self.PoolRate} = \text{self.cp2.CPCost} / \sum_{i=1}^{n} \text{self.cp2.pl2.p3.a2.cs1.ConsumeCount}$$

The expression represents that the pool rate of cost driver is equal to the cost of cost pool divided by the number of cost drivers consumed by the cost pool.

4 Realization and Business Process Simulation Modeling Lifecycle

4.1 The system architecture of enterprise modeling and simulation tool supporting ABC

A graphic enterprise modeling and simulation tool is developed to support the P_ PROCE model. Fig.4 shows its system architecture, which contains three layers: user service, business service and data service.

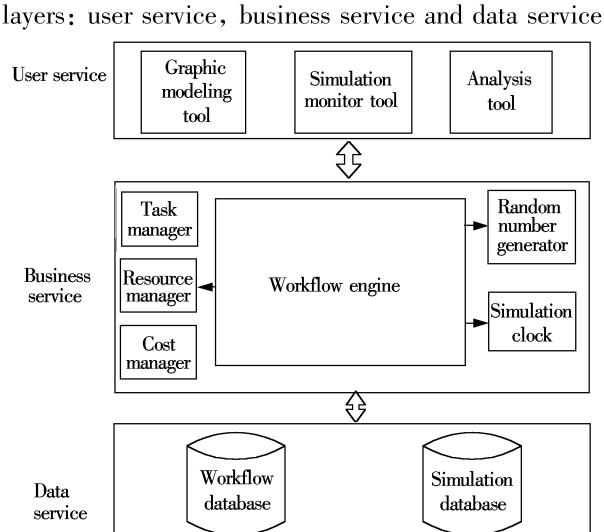

Fig.4 The system architecture of enterprise modeling and simulation tool

In the user service layer, there are three tools, respectively called, graphic modeling tool, simulation monitor tool and analysis tool. With the graphic modeling tool, a workflow model based on the meta model of P_PROCE can be modeled in a graphic way. Next the workflow model will be enacted by the workflow engine in a simulation model. Through the simulation monitor tool, we can watch what happens. The analysis tool helps us evaluate a business process.

In the business service layer, there are six components including resource manager, task manager, random number generator, simulation clock, cost manager, and workflow engine. The workflow engine enacts the workflow models in a simulation model, which work with the simulation clock and random number generator. Cost manager calculates the cost of every activity and every product. Task manger performs tasks in a simulation model. Resource manager allocates all kinds of resources.

In the data service layer, there are two databases, named workflow database and simulation database. The workflow models are saved in the workflow data base. And the simulation parameters and simulation results of the workflow are saved in the simulation database.

The enterprise modeling and simulation tool is developed with Visual C++.Net and Visual Basic.Net. Taking advantage of its GUI, users can construct an enterprise model and conduct simulation easily and efficiently.

4.2 The lifecycle of business simulation modeling supporting ABC

We have developed a characterization of the set of activities we perform in constructing and manipulating business process models. We refer to this set of activities as the business process simulation modeling lifecycle. The set of activities that entail the life cycle includes the following.

Process modeling Capturing informal descriptions of business process objects and their conversion into a formal business process model. The information of Process, Product, Resource, Organization, Cost and Evaluation is modeled in a workflow model based on the metal model of P_PROCE through the graphic modeling tool. Fig.5 shows the screen snapshot of graphic modeling tool.

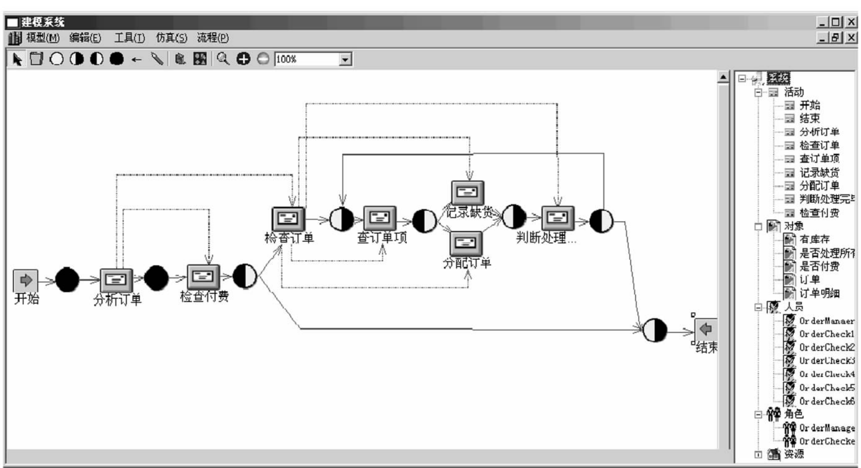

Fig.5 Screen snapshot of graphic modeling tool supporting ABC

Analysis Evaluating static and dynamic properties of a business process model, including its consistency, completeness, etc.

Simulation Symbolically enacting a business model in order to determine the path and flow of the intermediate state transitions in ways that can be dynamically analyzed, and reconfigured into multiple alternative scenarios. In a business process, many tasks needed to be performed. When the execution condition of a task is satisfied, an activity (as the instance of the task) will be allocated to one or more roles. While the activity is performed, the corresponding cost driver will be consumed. According to the property 2, the cost of an activity can be calculated. Before a cost object is produced, a set of activities will be performed. According to property 3, the cost of a product can be obtained. All simulation data will be saved in the simulation database for future evaluation.

Evaluating Collecting and measuring business process model data needed to improve subsequent process enactment iterations. Through the sensitive analyses, we can compare the different business processes. The result will display as a chart or grid.

Improving Incrementally and iteratively enhancing, restructuring, tuning, migrating, or reengineering a business process model or the preceding activities to more effectively meet emerging user requirements.

5 Conclusion

This paper presents a workflow meta model supporting activity based costing. A cost sub meta model is established by adding activity based costing to P_PROCE model. In order to define the cost sub metal model precisely, OCL is used to express cost meta model and constraints. A graphic enterprise modeling and simulation tool based on the workflow meta model is developed using Visual C++.Net and Visual Basic. Net to systematically construct enterprise model and conduct simulation easily and efficiently. Moreover it enables one to analyze and evaluate the business process and its costs.

References

[1] Tatsiopoulos I P, Panayiotou N. The integration of activity based costing and enterprise modeling for reengineering purposes [J]. *International Journal of Production Economics*, 2000, **66**(1):33-44.

[2] Gert Zulch, Bernd Brinkmeier. Simulation of activity costs for the reengineering of production systems [J]. *International Journal of Production Economics*, 1998, **56**, **57**(9): 711-722.

[3] Spedding T A, Sun G Q. Application of discrete event simulation to the activity based costing of manufacturing systems [J]. *International Journal of Production Economics*, 1999, **58**(1): 289-301.

[4] Cooper R, Kaplan R S. Measure costs right: make the right decisions [J]. *Harvard Business Review*, 1998, **2**(3): 97-98.

[5] Tzvi Raz, Dan Elnathan. Activity based costing for projects [J]. *International Journal of Project Management*, 1999, **17**(1): 61-67.

[6] Fu Qian. Research on workflow simulation for software process improvement [D]. Shanghai: Department of Computer Science and Engineering, Shanghai Jiaotong University, 2002. (in Chinese)

[7] Richters M, Gogolla M. On formalizing the UML object constraint language OCL [A]. In: Ling T W, Ram S, Lee M L, eds. *Proc of 17th Int Conf Conceptual Modeling*[C]. Berlin: Springer, 1998. 1507-1516.

支持作业成本法的工作流元模型研究

周晓俊　　曹　建

（上海交通大学计算机集成技术实验室，上海 200030）

摘　要　传统的会计方法已经难以满足现代制造模式的需要,而作业成本法可以克服传统成本系统的局限性.本文把作业成本法和工作流技术相结合,构建了一个支持作业成本法的工作流元模型.文中首先介绍了一个作业成本法的概念及其概念模型，其次介绍了一个工作流元模型 P_PROCE,通过为其添加作业成本法的思想,建立了一个支持作业成本法的成本子模型.最后,给出了一个基于该工作流元模型的企业过程仿真建模工具.通过以上工作,可以系统地对企业过程进行建模和仿真,并可以分析成本在内的各个方面信息.

关键词　作业成本法；工作流；元模型；对象约束语言；企业模型

中图分类号　TP391.9