SOCA (2016) 10:303–315
DOI 10.1007/s11761-015-0188-8

ORIGINAL RESEARCH PAPER

# A process-mining-based scenarios generation method for SOA application development

Lihong Jiang[1] · Jianyi Wang[1] · Nazaraf Shah[2] · Hongming Cai[1] · Chengxi Huang[1] · Ray Farmer[2]

Received: 27 March 2015 / Revised: 11 December 2015 / Accepted: 12 December 2015 / Published online: 23 December 2015

**Abstract** Business process models which are usually constructed by business designers from experience and analysis are the main guidelines for services composition in the service-oriented architecture (SOA) applications development. However, due to the complexity of business models, it is a challenging task for business process designers to optimize the process models dynamically in accordance with changes in business environments. In this paper, a process-mining-based method is proposed to support business process designers to monitor efficiency or capture the changes of a business process. Firstly, we define a scenario model to depict business elements and their relationships which are critical to business process design. Based on the proposed scenario model, process mining algorithms, including control flow mining, roles mining and data flow mining are carried out in a certain sequence synthetically to extract business scenarios from event logs recorded by SOA application systems. Finally, we implement a prototype using a logistic scenario to illustrate the feasibility of our method in SOA applications development

**Keywords** SOA · Business process model · Process mining · Scenario discovery

## 1 Introduction

Business process management (BPM) systems provide a broader scope of perspectives to describe, configure and improve business processes to manage information systems as opposed to workflow management (WFM) systems which focus mainly on processes automation for the organizations [1,2]. In the lifecycle of BPM, business process models play a significant role covering the stages of business processes design and system implementation. For business process management, two problems have attracted more attention of the researchers and practitioners. Firstly, in many cases the process models could not represent all the possible required tasks and related control flows in design stage. Secondly, the process models are not adaptive enough to react to the dynamic environments at runtime. Therefore, it is important to propose methods to refine the business process models by monitoring process execution dynamically and realizing process optimization continuously.

In the BPM life cycle, there are two kinds of business process model refinement methods: these are model based and data based [3]. Model-based methods often use simulation tools such as ARIS to verify the correctness or estimate the performance of the designed business processes before they are executed [4]. However, the accuracy of the simulations usually depends heavily on the simulation environment configurations, which sometimes have deviations from the real environment. Comparatively, data-based meth-

✉ Hongming Cai
hmcai@sjtu.edu.cn

Lihong Jiang
jianglh@sjtu.edu.cn

Jianyi Wang
wangjianyi@sjtu.edu.cn

Nazaraf Shah
aa0699@coventry.ac.uk

Chengxi Huang
hichens@sjtu.edu.cn

Ray Farmer
csx254@coventry.ac.uk

[1] School of Software, Shanghai Jiao Tong University, Shanghai, China

[2] Faculty of Engineering and Computing, Coventry University, Coventry, UK

ods attracted more research attention because they better reflect the dynamic changes of the business processes at runtime as the event logs are collected and analyzed [5]. In data-based methods, the α-series algorithms are designed to trace the business processes from event logs using Petri Nets, and the statistical algorithms are used to diagnose bottlenecks or resource contention during business process execution. Although the data-based methods analyze business processes from multiple viewpoints, such as control flows, data flows and user roles, the weakness of these process mining algorithms is that they are independent from each other and it is hard for users to analyze the performances or the changes of the whole business process synthetically.

This paper studies the methods of scenario generation from process logs perspective for the purpose of SOA system configuration, services governance and business process optimization. Instead of starting from the experiences of business experts of process modeling, we discover scenarios from the running environments (i.e., process instances, service and users) using process mining technology. We consider three types of business elements for scenario construction, which are the roles that carry out tasks in business processes, the tasks which are related to SOA service invocations and finally the business objects which are needed to fulfill the tasks requirements. In other words, we generate business scenarios from the executing environments considering not only the sequences of the tasks, but also the users and the business objects. Based on these elements, we propose a scenarios-based process mining method to construct comprehensive views of business scenarios to help improve the business processes. Furthermore, the scenarios from the executing business processes will help to understand the changes in the business processes more precisely.

The rest of the paper is organized as follows. In Sect. 2, related work is reviewed. The motivation for our work is presented in Sect. 3. In Sect. 4, the detailed approach for mining scenarios from event logs is given. In Sect. 5, we present a case study and a discussion to illustrate the scenario generation method in the area of logistic management. In Sect. 6, the conclusion and future work are described.

## 2 Related work

With the application of SOA technology in the development of complex enterprise information systems [6,7], the analysis and modeling of complicated business processes pose challenges to the system analysts and developers. Various efforts have been made to model the operational business processes (e.g., Petri nets, BPMN, UML and EPCs) [1]. In these methods, processes are described in terms of activities. However, due to the increasing complexity of the business models and the demands for the flexibility of enterprise information systems, additional factors, such as enterprise resources consumed by the business processes and the information delivering flow involved in the business processes, need to be considered in business process modeling and monitoring.

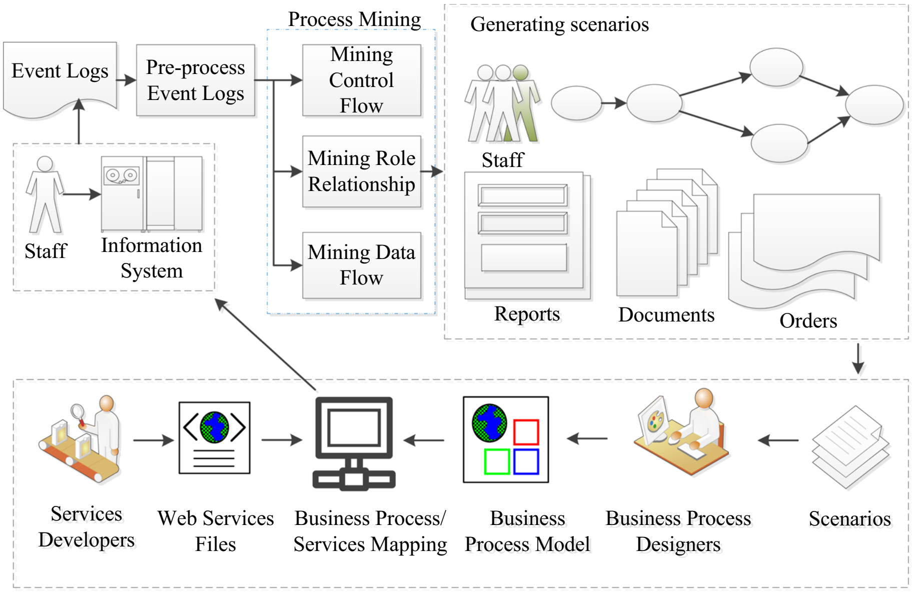
Therefore, in order to fully interpret the business processes for SOA system development, researchers have considered other factors to describe the ways business processes are executed and enterprise resources interact with each other (such as roles, business rules, and priorities of resources manipulation) to enrich the business process models. For instance, multi-view models describe a system from multiple perspectives for the different participants involved in the modeling phase to avoid confusion about the business logic, system architecture and implementation [8]. Multi-view methods provide more comprehensive understanding of business processes from different viewpoints, such as organizational structures, functions and data. Although the designers could gain an overall analysis using multi-model method, relationships among the different views have not been fully described in the multi-view models. As a result, the designers find it challenging to optimize enterprise resource utilities during business process modeling or business process running.

Business scenarios provide ways to characterize the uncertainty in decision making as well as in business process requirements analysis [9]. Researchers also use scenarios to reduce uncertainties involved in multi-objective decisions [10]. A scenario-driven approach is also used to capture the requirement of business processes, while UML sequence diagrams may be used to define the scenarios for business process modeling [11].

In order to extract business requirements more precisely, process mining methods are adopted to discover business scenarios from event logs.

Both clustering and abstracting approaches have been widely used in process mining based on event log analysis from different viewpoints to analyze business processes [12,13]. Researchers in [14] use a clustering approach to identify both outliers and groups of normal instances to obtain multi-perspective process model. Clustering methods could also be used to learn process variant adaptations during runtime [15]. Abstracting methods deal with the relations between events and activities to mine the control flows and data flows involved in business processes [16,17]. In [18], a process mining method is used to provide detailed clinical and hospital analysis of the delivery of health care. Some researchers use process mining methods to identify the difference between observed behaviors and pre-modeled behaviors [19]. It is also used to discover service behavior patterns during business process execution in enterprise system [20]. In SOA systems, activities carried out by services are recorded in event logs and business processes could be extracted from event logs. SOA execution logs are used to mine Web service

**Fig. 1** Research framework



composition patterns, and the gaps between the discovered model and the initial composite model are used to re-engineer the initial Web service composition model [21].

However, more research needs to be done using scenario-driven methods to improve the business process models to adapt to the changes in business environments.

In this paper, scenarios are built through mining event logs from multiple views including control flows, organization structure and data flows for business process optimization and service orchestration. Using the scenarios, we aim to depict business processes in a more detailed way so that the quality of services composition could be improved.

## 3 Scenario-based business process analysis and improvement in SOA

### 3.1 Research framework

In SOA-based information systems, Web services are often composed in a loosely coupled style to implement flexible IT infrastructures to support changing business models [22]. In order to improve the correctness and efficiency of services composition in SOA, services orchestration and choreography should not only consider the business functional requirements, but also consider the required resources such as users, data entities and involved control flows in business process execution.

In this paper, an extended scenario-based business process model is adopted to involve the business context in services composition. The business contexts would help the business designers to understand and optimize the business processes more precisely. Meanwhile, business context would provide hints to the services developers to adjust the parameters of the services to meet new business requirements. Therefore, the whole business process lifecycle would cover activities of business process modeling, services development, services composition, information systems deployment and business process optimization based on mining. The proposed framework is shown in Fig. 1.

In Fig. 1, the proposed framework includes three main components, which are process mining module, scenario generation module and services composition module. In the process mining module, the tools for control flow mining, role-relationship mining and data flow mining are selected to analyze event logs of SOA services invocations. In the scenario generation module, business control flows are synergized with data flow and entity resources to provide business processes designers and services developers with a whole view of business processes containing business contexts.

### 3.2 Business scenario model

In the scenario-based method of business process analysis and improvement in SOA shown in Fig. 1, scenarios involving run-time contexts are crucial references for improving the business processes. We mine scenarios from event logs recorded in the SOA systems, and then these scenarios are defined as a composition of control flows, related data entities and task performers.

Control flows determine the logical sequences of tasks triggered and accomplished by SOA services invocation. Data entities are abstractions of conceptual and physical business objects related to business processes, such as product orders, manufacturing plans and equipment states, represented by sets of properties [23]. Task performers are not limited to human beings such as customers, suppliers and employees in a company. If some tasks are performed by a software component automatically without the direct interaction of people, such as services in SOA, these services should be regarded as task performers as well [24].

The aim of the design of our proposed business scenario model is to help business process analysts consider more critical business elements during business processes modification and optimization by capturing run-time business process changes reflected in SOA services invoking logs.

The following business elements are involved in the business scenarios.

**Definition 1** *Event*

Events are the activities representing task execution. Events are defined as tuples as follow.

$$Event = \langle Case\_ID, Event\_ID, Timestamp, \\ Performer, \{BO\}\rangle$$

The *Case_ID* is the serial number of the business process instance to be performed. The *Event_ID* corresponds to the task or the SOA service name. *Timestamp* records the time the event occurs. *Performer* is the user carrying out the task. *{BO}* represents business objects involved in the event.

**Definition 2** *Execution Instance, Event log*

Let $E$ be the set of events and $E^*$ be the set of all sequences composed of zero or more events from E. Then, $\sigma \in E^*$ denotes the execution instance, while $W \subseteq E^*$ denotes an event log.

**Definition 3** *Business Object (Bo)*

As mentioned above, business objects are the abstractions of virtual and physical entities that are involved in the business processes.

$$Bo = \langle Bo-\_ID, A\rangle;$$
$$A = \{a_1, a_2, \ldots a_m\} \quad (i = 1, 2, \ldots, \mathrm{m});$$

*Bo_ID* is the unique identification of the business object. *A* is the set of attributes of the business object.

**Definition 4** *Business Process (BP)*

A business process is defined as the tuple of the unique identifier of the business process *BP_ID*, a set of scenarios denoted as *SceS* and their relations are denoted as *SceR*.

$$BP = \langle BP\_ID, SceS, SceR\rangle$$

**Definition 5** *Scenario*

The main emphasis of business process re-engineering should be put on the scenarios. A scenario is formed from a role, a set of ordered activities and a set of business objects. And the scenario should stipulate the correspondence between activities and the business objects. Thus, the scenario is defined as the following tuple:

$$Scenario = \langle Sce\_ID, AS, AR, \{BO\}, ABo\rangle$$

*Sce_ID* is the unique identifier of the scenario. *AS* names the activities included in the scenario and can be denoted as the following.

$$AS = \{Act_1, Act_2, Act_3, \ldots\}$$
$$Act_i = \langle ActName\rangle$$

*AR* defines the relations between the activities in *AS* so that the business process topology is defined by *AR*.

$$AR = \{\langle Act_i, Act_j\rangle | Act_i, Act_j \in AS \text{ if } \\ \mathrm{Act_i \text{ is related to } Act_j}\}$$

$\{BO\}$ are the business objects involved in the scenario.

*ABo* is a mapping function that maps an activity $Act_i$ to a set of business objects used by the activity $Act_i$ in this scenario.

$$ABo : AS \rightarrow 2^{|BS|};$$
$$ABo(Act_i) = \{Bo_1, Bo_2, Bo_3, \ldots\};$$

**Definition 6** *Role*

Usually in an organization, employees are grouped into roles according to their assigned jobs, as well as the application systems. Different roles will have different allocated tasks and occupy different business objects.

$$Role = \langle Role\_ID, AS, \{BS\}, ABo\rangle;$$

The *AS* represents all activities carried on by this role in a business process. *{BS}* are business objects used by the role, and *ABo* is the mapping function from one activity $\mathrm{Act_i}$ to a set of business objects used by the activity.

Based on above definitions, the business scenario model and the business elements in the model are shown in Fig. 2.

As shown in Fig. 2, in order to generate the scenarios from the event logs, we use process mining technology from the both organizational structure and the business process perspective. The arrow heads in Fig. 2 represent foreign key relationships. The organizational structure perspective focuses on business objects related to the tasks, such as who perform the tasks and how the task performers are related. The business process perspective focuses on the routines of the business process instances, for example when manufacture orders are fulfilled, what kind of data flows would be involved.

**Fig. 2** Business scenario model for business process mining

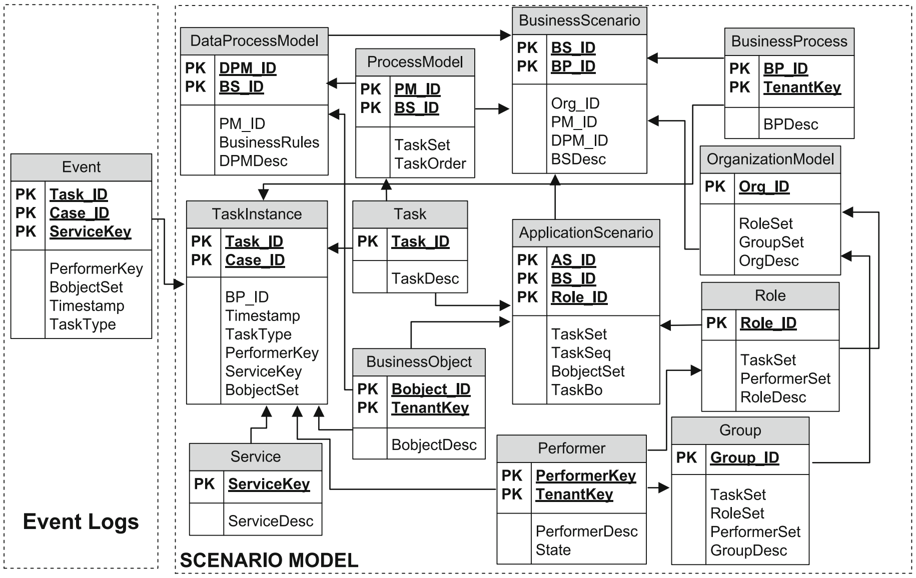


**Table 1** Fragment of event log

| Case 1 | | | |
|---|---|---|---|
| Event Id | Timestamp | Performer | Business objects |
| A | Case1Time1 | P1 | bo1, bo2 |
| B | Case1Time2 | P2 | bo1 |
| C | Case1Time3 | P3 | bo2 |
| D | Case1Time4 | P4 | bo2 |
| E | Case1Time5 | P5 | bo3 |

## 4 Scenario context generation using process mining

### 4.1 Event logs preprocessing

Currently, information systems such as ERP (enterprise resource planning) systems, PDM (product data management) systems, CRM (customer relation management) systems exist widely in various kinds of enterprises [25]. The event logs of these information systems record data about performed activities in the enterprises including time performed, actors and related business objects, as shown in Table 1.

In SOA-based information systems, business process instances may be carried out by different services offered by different service developers. This brings challenge of understanding and analyzing business processes because event data are often scattered over different data sources and they are hard to be collected.

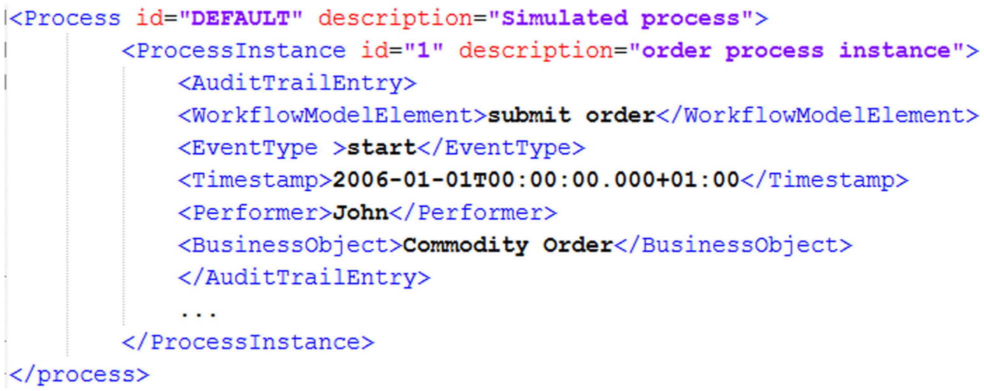
In order to address the above challenge, we use service invocation event logs as business process mining data sources, instead of using scattered running logs of information system. These SOA services invocation logs could be collected centrally on severs sides by recording the message exchanges [26] (e.g., SOAP messages) and read and write actions [27]. Since service-oriented applications are generally deployed on centralized runtime environments, such as Microsoft Azure Services Platform and Google App Engine, the adoption of such centralized runtime environments monitor the running of distributed services-oriented applications and obtain centralized service invocation event logs. These event logs can be transformed to a standard format, such as MXML (Mining eXtensible Markup Language) for further usage.

According to Definition 1, event log files contain the triggering timestamps of the SOA services, names of the SOA services carrying on the activities, the ids of the business process instances including the activities and the performers who initiate or perform the activities. We use MXML to represent these events logs, as shown in Fig. 3.

### 4.2 Mining control flow

We use Petri net to represent the control flow of the SOA services invocations. We mine the sequential structure, parallel structure and selective structure of the SOA services invocations using α algorithm [28].

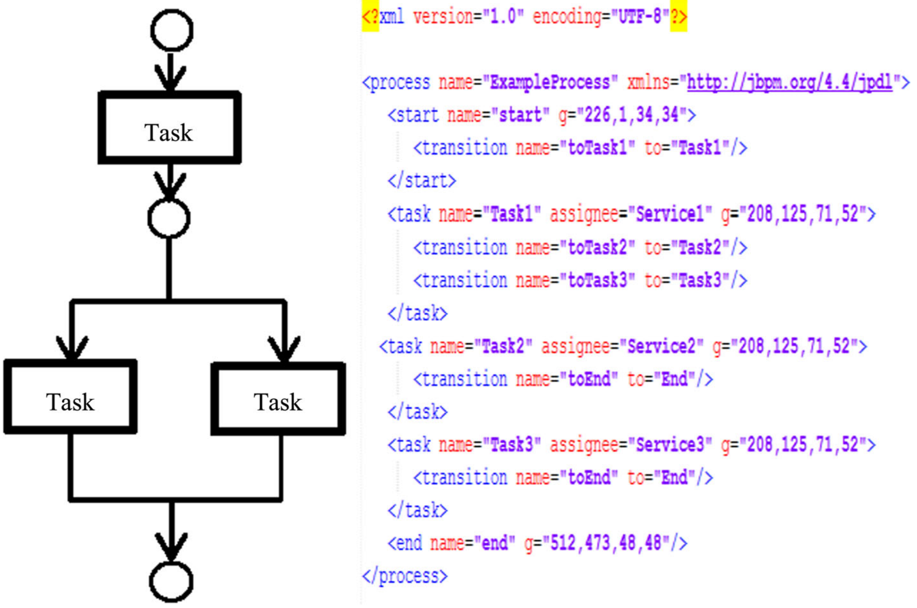
Using Petri nets, a SOA services invocation control flow could be defined as $PM = \langle P, T, S, F\rangle$ where $P$ is the finite set of *places*. The *source place i* and *sink place o* correspond to the start and end points in WF-net (WorkFlow-net). $T$ stands for the finite set of transition in *PM*. Each transition corresponds to a single task in the business process. $S$ is the set of resources of services used to perform the tasks in *PM*. We use $s_i$ to denote the resource used to fulfill the *ith* task. $F$ describes the relations between the *places* and the *transitions* denoted as $F \subseteq (P^*T) \cup (T^*P)$.



```
<Process id="DEFAULT" description="Simulated process">
        <ProcessInstance id="1" description="order process instance">
            <AuditTrailEntry>
            <WorkflowModelElement>submit order</WorkflowModelElement>
            <EventType >start</EventType>
            <Timestamp>2006-01-01T00:00:00.000+01:00</Timestamp>
            <Performer>John</Performer>
            <BusinessObject>Commodity Order</BusinessObject>
            </AuditTrailEntry>
            ...
        </ProcessInstance>
</process>
```

**Fig. 3** MXML format for an event entry in log file

**Fig. 4** Control flow of SOA services invocation using WF-net/JPDL



```
<?xml version="1.0" encoding="UTF-8"?>

<process name="ExampleProcess" xmlns="http://jbpm.org/4.4/jpdl">
   <start name="start" g="226,1,34,34">
      <transition name="toTask1" to="Task1"/>
   </start>
   <task name="Task1" assignee="Service1" g="208,125,71,52">
      <transition name="toTask2" to="Task2"/>
      <transition name="toTask3" to="Task3"/>
   </task>
  <task name="Task2" assignee="Service2" g="208,125,71,52">
      <transition name="toEnd" to="End"/>
   </task>
   <task name="Task3" assignee="Service3" g="208,125,71,52">
      <transition name="toEnd" to="End"/>
   </task>
   <end name="end" g="512,473,48,48"/>
</process>
```

WF-net and JPDL(JBoss jBPM Process Definition Language) are therefore used to display the Petri net result of SOA services invocation control flows mining by α algorithm, which is shown in Fig. 4.

A WF-net and its implementation using JPDL are shown in Fig. 4. In JPDL, label $\langle process\rangle\langle/process\rangle$ stands for an entire process model. Label $\langle task\rangle\langle/task\rangle$ stands for the transition, e.g., the tasks in the business processes. Its attribute *assignee* stands for the service resource related to the task. $\langle start\rangle\langle/start\rangle$ stands for starting a task, while $\langle end\rangle\langle/end\rangle$ stands for end of a task. $\langle transition\rangle\langle/transition\rangle$ is the input/output function of the transition.

### 4.3 Mining the relations between roles

Roles are the actors who perform the tasks in the business processes. Usually, business processes are performed by different roles cooperatively. The relations between the roles often reflect relations between the activities. Therefore, mining the relations between the roles could help to monitor or discover business process changes from the view point of performer relations in addition to the control flows of SOA services invocations. The performers, whether people or software agents, are kinds of enterprise resources; the results of roles mining could help managers to optimize the allocation of human and services resources more efficiently.

In order to establish the role relations of the users, metrics of *performer–activity* are used to discover the relations between the activities and their performers [18]. As users belonging to the same role often conduct similar activities rather than totally different activities, the frequencies of the performers conducting the activities are counted and used to measure the distance between two performers, as shown in Table 2, where $X_{i,j}$ represents the frequency of performer $i$ performing activity $j$.

**Table 2** Performer–activity matrix

| | *A* | *B* | *C* | *D* | *E* |
|---|---|---|---|---|---|
| Performer P1 | $X_{1,1}$ | 0 | 0 | 0 | 0 |
| Performer P2 | 0 | $X_{2,2}$ | $X_{2,3}$ | $X_{2,4}$ | 0 |
| Performer P3 | 0 | 0 | $X_{3,3}$ | $X_{3,4}$ | 0 |
| Performer P4 | 0 | 0 | $X_{4,3}$ | $X_{4,4}$ | 0 |
| Performer P5 | 0 | 0 | 0 | 0 | $X_{5,5}$ |

```
Role Classification Algorithm
Input: P, TV // Input P is the set of performers, TV is the threshold
Output:G = (P; Arc; W ) //Output G is the graph of performers distances,
 for each p in P
     $p_i$ = p;
     for each elseP in P
         $p_j$ = elseP;
         If ($p_i$!=$p_j$) then
             Weight = CalculatePCC($p_i$, $p_j$);
                     //count the correlation coefficient from pi to pj
             If (Weight>TV) then
                 generateArc($p_i$, $p_j$);
             end if;
         end if;
         else continue;
     end for;
 end for;
return G = (P; Arc; W );
```

**Fig. 5** Pseudocode of the roles classification algorithm

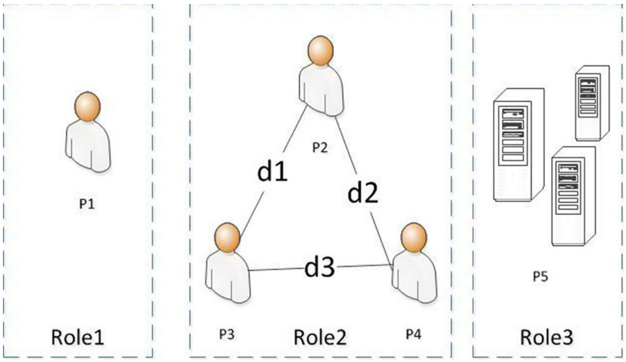


**Fig. 6** Role classifitions derived from event logs

Here, we use Pearson's correlation coefficient to count the distance between two roles. The formula (1) is as follows.

$$\mathrm{PCC}\,(p_1, p_2) = \frac{\sum_{j=1}^{n}\left(\left(X_{1j} - \frac{1}{n}\sum_{j=1}^{n} X_{1j}\right) * \left(X_{2j} - \frac{1}{n}\sum_{j=1}^{n} X_{2j}\right)\right)}{\sqrt{\sum_{j=1}^{n}\left(X_{1j} - \frac{1}{n}\sum_{j=1}^{n} X_{1j}\right)^2 * \sum_{j=1}^{n}\left(X_{2j} - \frac{1}{n}\sum_{j=1}^{n} X_{2j}\right)^2}} \tag{1}$$

Using the above formula, the distances of each two roles is calculated to classify them into different work groups. The pseudocode of the algorithm is shown in Fig. 5.

Using the classification algorithm, the roles shown in Table 2 could be divided into three groups, shown in Fig. 6.

The result shows that *P*1 can be assigned to *Role*1, *P*2, *P*3 and *P*4 can be put into the same group *Role2*, and *P*5 could be assigned to *Role3*.

The relations between the work groups are determined by the control flow structures of the activities. We have three types of control flow structures. They are sequential, parallel and selective. If two work groups perform sequential activities, their relations are defined as *Hierarchical*. If two work groups perform parallel activities, their relations are defined as *Cooperative*. If two work groups perform selective activities, their relations are defined as *Alternative*.

### 4.4 Mining data flow

Data flows are transmission sequences of data between activities in business processes. They reflect the logical relations of the business activities performed by different roles. In many situations, the values of the parameters transmitted in the data flow determine the directions of the business process control and govern the choices of certain cases taking particular business paths [19]. For example, if an order is accepted, the customer should get a notification message sent automatically. These decisions could be made depending on the values assigned to business objects. Therefore, we can analyze the event logs to identify the possible data constraints affecting the structures of business processes, and obtain business rules used for further control of business processes.

We use the Daikon system proposed in [29] to detect the relations between the variables and the assigned values to deduce data constraints affecting the structures of the business processes. These data flows are helpful for the services developers to analyze or re-engineer the business processes.

### 4.5 Scenario generation

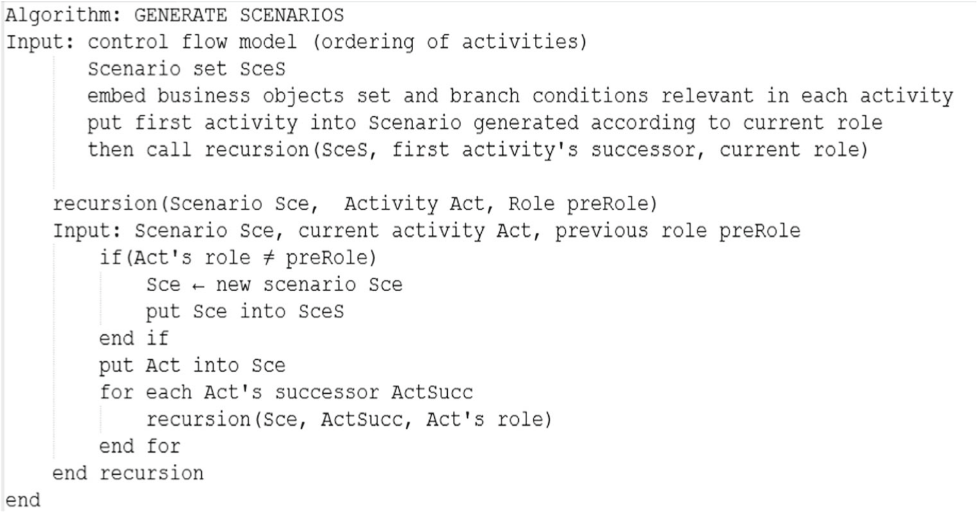
According to control flows, role relations and data flows, scenarios are generated automatically from log data. The scenario could be generated using the algorithm, shown in Fig. 7.

Figure 7 shows that the generation of the scenarios starts by focusing on one of the roles in the scenarios. Then, activities performed by this role are pulled into the same scenarios. After that, related activities according to the control flows are pulled into the scenarios to expand the range of the business processes. The algorithm will recur till all the users are pulled in.

## 5 Case study and discussion

### 5.1 Case study

In this section, we use a prototype to illustrate our approach of scenario generation based on process mining.



```
Algorithm: GENERATE SCENARIOS
Input: control flow model (ordering of activities)
          Scenario set SceS
          embed business objects set and branch conditions relevant in each activity
          put first activity into Scenario generated according to current role
          then call recursion(SceS, first activity's successor, current role)

    recursion(Scenario Sce,  Activity Act, Role preRole)
    Input: Scenario Sce, current activity Act, previous role preRole
        if(Act's role ≠ preRole)
            Sce ← new scenario Sce
            put Sce into SceS
        end if
        put Act into Sce
        for each Act's successor ActSucc
            recursion(Sce, ActSucc, Act's role)
        end for
    end recursion
end
```

**Fig. 7** Scenario generation

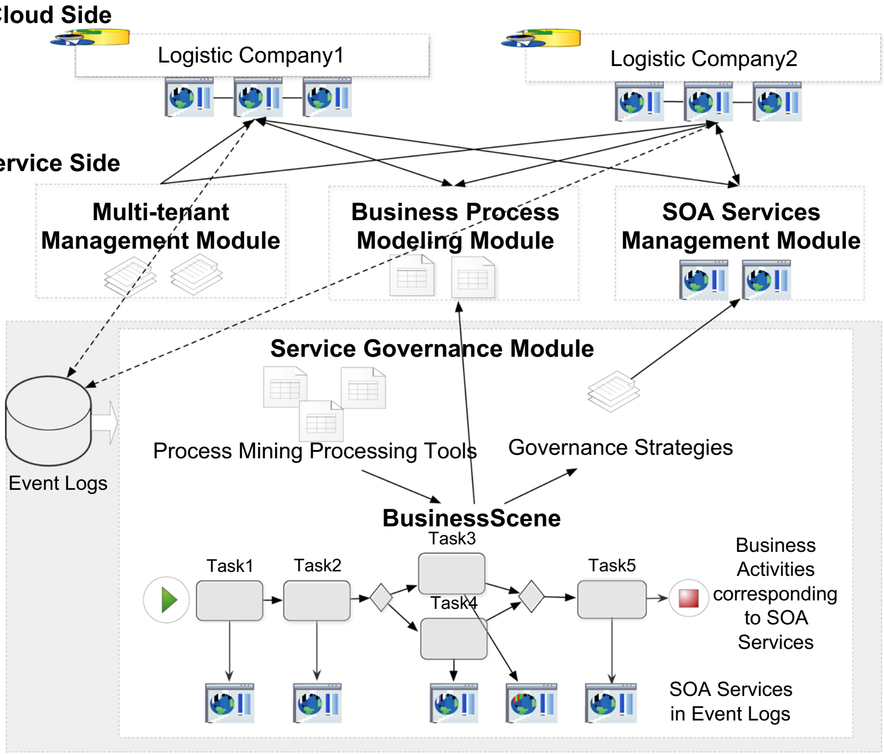

Cloud Side
Logistic Company1
Logistic Company2
Service Side
Multi-tenant Management Module
Business Process Modeling Module
SOA Services Management Module
Service Governance Module
Event Logs
Process Mining Processing Tools
Governance Strategies
BusinessScene
Task1
Task2
Task3
Task4
Task5
Business Activities corresponding to SOA Services
SOA Services in Event Logs

**Fig. 8** Architecture of the cloud logistic management platform

It is believed that cloud computing technologies could improve the efficiency and the performance of resources allocation and many SOA applications are deployed on the cloud computing platforms nowadays [30–32]. The prototype is part of a SOA-based logistics management platform developed for the transportation bureau in one of the provinces in China. In the transportation bureau, logistics companies are registered to convey varieties of logistics transportations. In order to efficiently use the computational resources, the transportation bureau puts forward the project to build a cloud platform so that logistic companies could tenant SOA services from the platform to deploy their logistic management systems.

The architecture of the cloud logistic management platform is shown in Fig. 8.

As shown in Fig. 8, in the cloud side, the logistic companies use the business modeling tools provided by the platform to construct their business process models. On service side, SOA services could be selected to choreograph corresponding logistic information systems for the tenants according

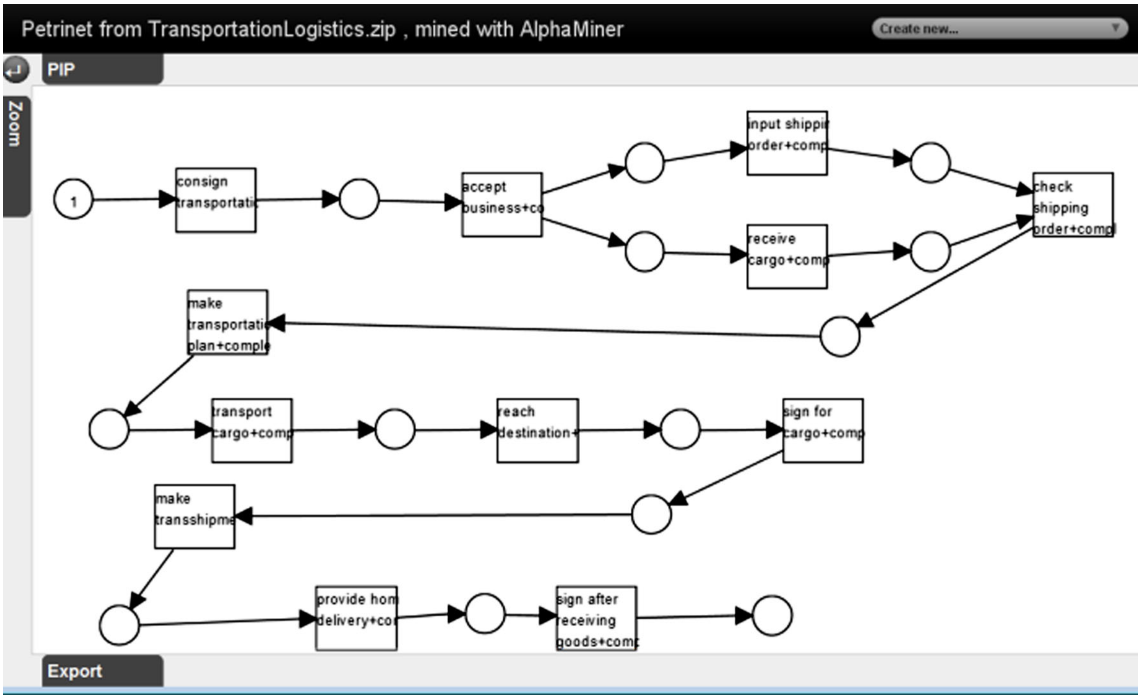


**Fig. 9** Control flow of intra-provincial logistics process

to the business process models. During the execution of the logistics information systems, event logs are collected on the service side. In this prototype system, the process-mining-based scenario generation method is used to analyze the event logs generated by the tenant companies. It could be seen in Fig. 8 that the mined scenarios may help SOA service providers to build SOA governance strategies. Also the scenarios could be references to business process designers.

In the prototype, an open-source tool ProM is used to analyze the event logs [33] as shown in the following steps.

Step 1 Control flow mining.

We choose a plug-in $\alpha$ algorithm of ProM to discover the control flow of the SOA services calling as shown in Fig. 9.

In Fig. 9, the control flow of the business process of intra-provincial logistics is displayed.

Step 2 Role relations mining

Users involved in the business processes are classified and divided into different roles according to the *performer–activity* metrics.

In this step, we use plug-in called social network tool in ProM to calculate the distances between two performers. It could be seen in Fig. 10 that in this case study, the performers could be divided into three types of roles, named drivers, delivers and logistics assistants, belonging to two work groups, named intra-provincial work group and extra-provincial work group.

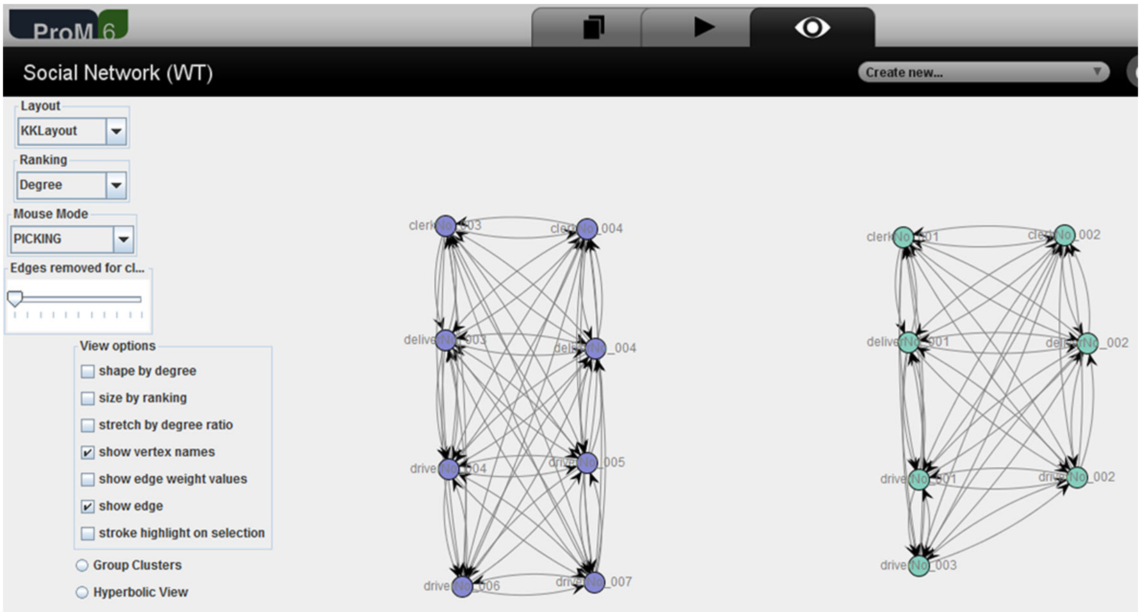
Because the logistics company in this case study has two types of customers running intra-provincial and extra-provincial business separately, the users are divided into two work groups. It could be seen that in this case study, the mining results are consistent with the real situation of the analyzed company. Therefore, the mining results could help the business process designers better understand the business processes of the companies.

Step 3 Data flow mining

In data flows, the constrain conditions which trigger the activities could be obtained by counting the values of the business objects involved in the business processes. In this case study, we use decision-tree algorithm implemented in ProM [19] to calculate the values of the business objects, and the results show that two business objects *TransportPlan* and *ShippingOrder* are critical in this business process, as demonstrated in Fig. 11. They are the attributes related to shipping activities in the logistics process. The result shows that the task "*check shipping order*" will check the shipping destination and current location recorded in *TransportPlan* and *ShippingOrder* , respectively. If *TransportPlan.Destination* != S*hippingOrder.CurrentLocation,* "*transit shipment*" will be carried out, otherwise, "*make transportation plan*" will be carried out.

Step 4 Scenario generation

According to the results obtained in steps 2 and 3, the control flow shown in Fig. 9 could be divided into several



Layout
KKLayout

Ranking
Degree

Mouse Mode
PICKING

Edges removed for cl...

View options
□ shape by degree
□ size by ranking
□ stretch by degree ratio
☑ show vertex names
□ show edge weight values
☑ show edge
□ stroke highlight on selection

○ Group Clusters
○ Hyperbolic View

**Fig. 10** Generated work groups

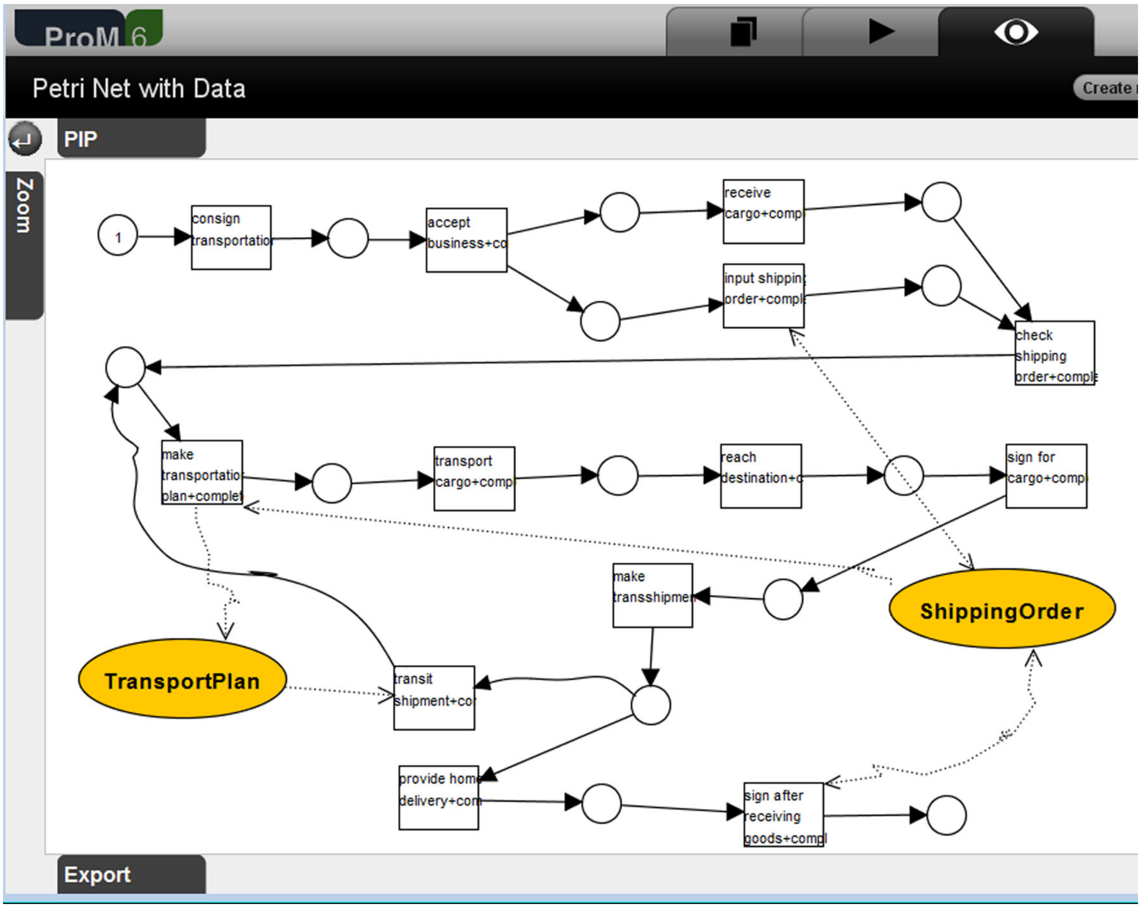


**Fig. 11** Data flow mining

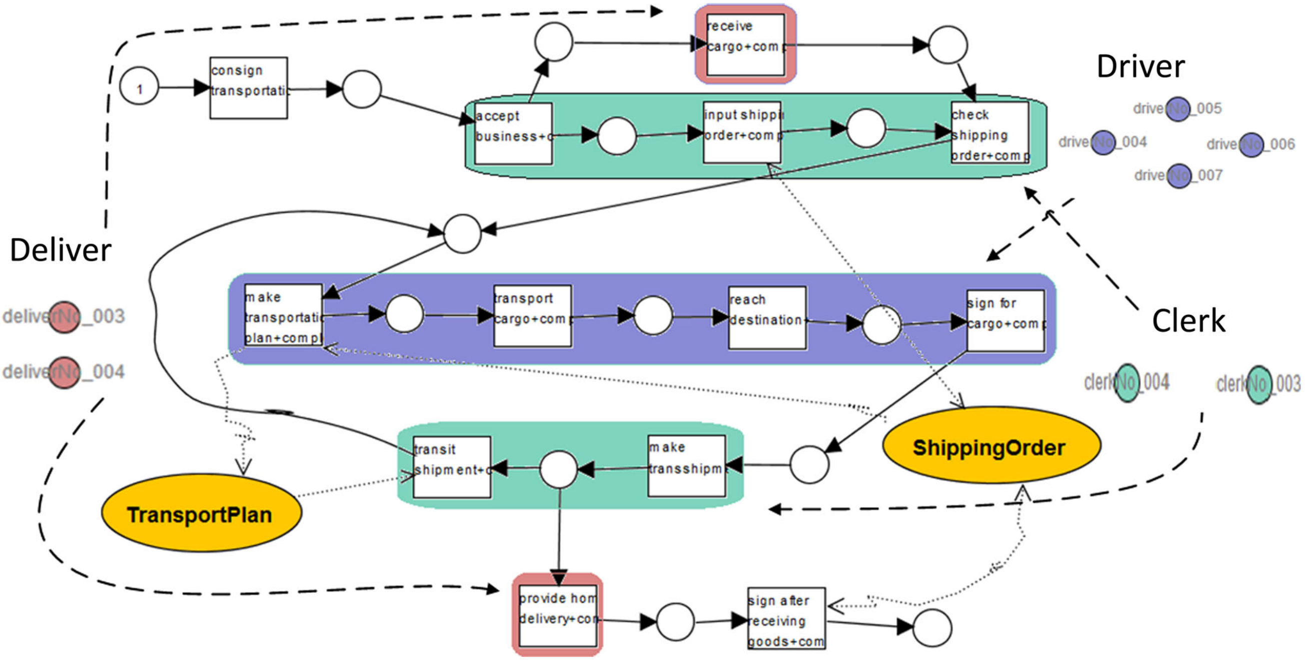


**Fig. 12** Scenario generation

scenarios based on the algorithm shown in Fig. 6. Scenarios for the extra-provincial businesses are synthesized into one comprehensive business processes, as shown in Fig. 12.

In Fig. 12, three scenarios are generated; each scenario corresponds to one type of roles, which are *driver*, *delivery man* and *clerk*, respectively. In these scenarios, business objects, tasks and roles are closely related to show the business processes more clearly to the business processes designers and SOA services developers.

In this case study, it could be seen that business processes are traced in the event logs of SOA service invocations. In our method, we use α algorithm to mine the control flow of the business activities. After that, role relation and data flow mining algorithms are used sequentially at each point of activities along the generated control flow to obtain information about the required human resources and data entities during business process performing. These detailed business contexts may help designers to monitor and optimize the business process more clearly.

### 5.2 Discussion

In the above case study, the prototype of the cloud logistics management platform was developed according to the real requirements of the logistic companies. We have simulated the running of the platform in the laboratory and generated an extra-provincial business scenario using event logs to show the effectiveness and applicability of our proposed method. We compare the scenario-based business process modeling method with multi-view business modeling methods such as ARIS [34] and Artifact [35]. Our comparison is based on provision of various functionalities in the proposed method rather than on comparison of logs collected as a result of running platform.

Table 3 demonstrates that the scenario-based business process model has the following advantages.

1. More flexible for business modeling. Since the scenarios could be regarded as a process-centric artifact, the scenario could be easily encapsulated as business model and transformed for different disposing purposes. Business process management systems may be achieved by the scenarios conveniently and modified through recombining these scenarios or adding new scenarios. For instance, if the business process in the case study should have a new part of asking help from external experts having existed in the scenario repository, it could be directly inserted into the process in accordance with the requirements.
2. A comprehensive view for process optimization or refinement. Since the scenarios contain not only process model but also data objects and roles, we could refine the business processes based on more detailed contexts of run time. This could increase the efficiency of business process modeling.

## 6 Conclusion and future work

In this paper, we emphasize three factors in business process analysis: the roles which invoke the SOA services, the business objects required in the business processes and the control

**Table 3** Comparison between business process modeling methods

| Features | Artifact | ARIS | Our approach |
|---|---|---|---|
| Business process modeling | | | |
| Oriented users | Software engineers | Business process designers | Software engineers and business process designers |
| Process modeling supporting tools | Manually modeling supported by visual tools | Manually modeling by supported by visual tools | Automatically generation by process mining tools |
| Modeling methodology | Abstract expertise of IT engineers | Abstract expertise of domain experts | Analyze event logs |
| Model execution | Support by BPEL or BPMN | None | Support by BPEL after model mapping |
| Model Views | Single view of processes | Multi-views of processes, roles and data | Process-centric multi-views of tasks, data, and roles added in related relations |
| Business process optimization | | | |
| Process optimization support | None | Yes | Yes |
| Optimization method | None | Simulation methods | Process Mining Algorithms |
| Business process monitoring | None | None | Yes |
| Adaption to dynamic business process changes | Low | Low | High |

flows of the activities. We adopt the concept of business scenario to analyze and optimize the business process models based on the data of event logs. Two main contributions are summarized as follows:

1. A method for scenario discovery using three perspectives of process mining is proposed to construct comprehensive business process models. In this scenario generation method, roles are centric. The users are classified into roles and work groups according to their distances calculated on the metrics of *performer–activity*. Activities related to the users are added into the scenarios to expand the covering range of the business process. The control flow structures are used to determine the relations between the roles. The scenarios are enriched by the constrain conditions calculated from the data flows.
2. The scenario model is proposed for analysis and improvement in business process modeling, which is highlighted by its flexible composition and overall comprehensibility for technical or non-technical people. And the usability of scenario-based process improvement is discussed, including the scenario reconstruction and modification.

Although scenario-based technologies used in our research have shown advantages in capturing users' requirements for SOA systems development, the proposed method of scenario generation has limitations. The services invoked in the SOA systems should be well defined and the running traces should be recorded in the log files so that elements such as roles, activities and control flows could be extracted from the log data. In addition, the accuracy of the business process models built on the scenarios in our approach varies depending on the choices of the process mining algorithms.

Our future research will focus on following issues.

1. Algorithms need to be designed for mining more complex relations among the users involved in the business processes. Only simple relations such as clusters and rough hierarchical relations could be discovered in this paper. Algorithms need to be proposed to obtain more accurate sequential and hierarchical relations widely existed in the organization structure from the log files of SOA systems.
2. Methods of merging distributed scenarios need to be studied in future. In cloud computing, services are deployed in distributed environment. In this paper, we use SOA services invoking logs centralized on service sides to analyze the business scenarios. The scattered scenarios are merged according to the control flows mined from the centralized event logs of SOA service invoking. Further research is needed to deal with the situation where event logs are scattered.

**Acknowledgments** This research is supported by the National Natural Science Foundation of China under Nos. 61373030, 71171132.

## References

1. Van Der Aalst WM (2013) Business process management: a comprehensive survey. ISRN Software Engineering: Article ID 507984. doi:10.1155/2013/507984

2. Sheng QZ et al (2014) Web services composition: a decade's overview. Inf Sci 280:218–238
3. van der Aalst WM (2013) Business process management: a comprehensive survey. ISRN Software Engineering, vol 2013, Article ID 507984. doi:10.1155/2013/507984
4. Jimin L, Feng Q, Zhang L (2014) A business process simulation method supporting resource evolution. In: Proceedings of the 2014 international conference on software and system process, pp 169–177
5. Van Der Aalst W, Zhao JL, Wang HJ (2015) Editorial: business process intelligence: connecting data and processes. ACM Trans Manag Inf Syst (TMIS) 5(4):181e:1–181e:5
6. Duan Q et al (2012) A survey on service-oriented network virtualization toward convergence of networking and cloud computing. IEEE Trans Netw Serv Manag 9(4):373–392
7. Yao JT et al (2013) Granular computing: perspectives and challenges. IEEE Trans Cybern 43(6):1977–1989
8. Wang Q, Yung KL, Ip WH (2004) A hierarchical multi-view modeling for Networked Joint Manufacturing System. Comput Ind 53(1):59–73
9. Kwakkel JH, Auping WL, Pruyt E (2013) Dynamic scenario discovery under deep uncertainty: the future of copper. Technol Forecast Soc Chang 80(4):789–800
10. Maghouli P, Hosseini SH, Oloomi Buygi M, Shahidehpour M (2011) A scenario-based multi-objective model for multi-stage transmission expansion planning. IEEE Trans Power Syst 26(1):470–478
11. Ruokonen A, Kokko T, Systa T (2012) Scenario-driven approach for business process development. Int J Bus Process Integr Manag 6(1):77–96
12. De Weerdt J, Schupp A, Vanderloock A, Baesens B (2013) Process Mining for the multi-faceted analysis of business processes—a case study in a financial services organization. Comput Ind 64:57–67
13. Rovani M, Maggi FM, de Leoni M, van der Aalst WM (2015) Declarative process mining in healthcare. Expert Syst Appl 42:9236–9251
14. Folino F, Greco G, Guzzo A, Pontieri L (2011) Mining usage scenarios in business processes: outlier-aware discovery and run-time prediction. Data Knowl Eng 70:1005–1029
15. Li C, Reichert M, Wombacher A (2011) Mining business process variants: challenges, scenarios, algorithms. Data Knowl Eng 70:409–434
16. Baier T, Mendling J, Weske M (2014) Bridging abstraction layers in process mining. Inf Syst 46:123–139
17. Liu Y, Zhang H, Li C, Jiao RJ (2012) Workflow simulation for operational decision support using event graph through process mining. Decis Support Syst 52:685–697
18. Partington A, Wynn M, Suriadi S, Ouyang C, Karnon J (2015) Process mining for clinical processes: a comparative analysis of four Australian Hospitals. ACM Trans Manag Inf Syst 5(4):Article 19
19. De Leoni M, Van Der Aalst WM (2013) Data-aware process mining: discovering decisions in processes using alignments. In: Proceedings of the 28th annual acm symposium on applied computing. pp 1454–1461
20. van der Aalst W (2013) Service mining: using process mining to discover, check, and improve service behavior. IEEE Trans Serv Comput 6(4):525–535
21. Gaaloul W, Baïna K, Godart C (2008) Log-based mining techniques applied to web service composition reengineering. SOCA 2:93–110. doi:10.1007/s11761-008-0023-6
22. Chesbrough H, Spohrer J (2006) A research manifesto for services science. Commun ACM 49(7):35–40
23. Grabowik C, Knosala R (2003) The method of knowledge representation for a CAPP system. J Mater Process Technol 133(1):90–98
24. Yu H, Zhu C, Cai H, Xu B (2009) Role-centric RESTful services description and composition for E-business applications. In: Proceedings of IEEE international conference on e-business engineering. pp 103–110
25. Gunther CW, Rinderle-Ma S, Reichert M, van der Aalst WM, Recker J (2008) Using process mining to learn from process changes in evolutionary systems. Int J Bus Process Integr Manag 3(1):61–78
26. Van Der Aalst WM, Dumas M, Ouyang C, Rozinat A, Verbeek E (2008) Conformance checking of service behavior. ACM Trans Internet Technol 8(3):Article No. 13
27. Diniz PC, Ferreira DR (2008) Automatic extraction of process control flow from I/O operations. In: Proceedings of 6th international conference business process management. pp 342–357
28. Van der Aalst W, Weijters T, Maruster L (2004) Workflow mining: discovering process models from event logs. IEEE Trans Knowl Data Eng 16(9):1128–1142
29. Ernst MD, Cockrell J, Griswold WG, Notkin D (2001) Dynamically discovering likely program invariants to support program evolution. IEEE Trans Softw Eng 27(2):99–123
30. Rahimi MR et al (2014) Mobile cloud computing: a survey, state of art and future directions. MONET 19(2):133–143
31. Rahimi MR et al (2013) MuSIC: mobility-aware optimal service allocation in mobile cloud computing. IEEE CLOUD 2013:75–82
32. Wei G et al (2010) A game-theoretic method of fair resource allocation for cloud computing services. J Supercomput 54(2):252–269
33. Verbeek HMW, Buijs Joos CAM, Van Dongen BF, Van Der Aalst WM (2011) Xes, xesame, and prom 6. Information Systems Evolution. Lecture Notes in Business Information Processing 72:60–75
34. Repa V, Železník O (2014) Methodological limitations of modeling languages BPMN and ARIS. In: The proceedings of 15th international carpathian control conference. pp 507–512
35. Mens T, D'Hondt T (2000) Automating support for software evolution in UML. Autom Softw Eng 7(1):39–59