Computers in Industry 60 (2009) 416–427

Contents lists available at ScienceDirect

Computers in Industry

journal homepage: www.elsevier.com/locate/compind

A Web services and process-view combined approach for process management of collaborative product development

Ping Jiang, Xinyu Shao, Haobo Qiu, Liang Gao *, Peigen Li

The State Key Laboratory of Digital Manufacturing Equipment and Technology, Huazhong University of Science and Technology, 430074 Wuhan, PR China

ARTICLE INFO

Article history:
Available online 21 March 2009

Keywords:
Collaborative product development
Process management
Process-view
Web services

ABSTRACT

Collaborative Product Development (CPD) across enterprises has been recognized as an effective strategy to connect multi-enterprise wide project teams to develop more competitive products. Process management is vital to the successful implementation of CPD in a multi-enterprise environment. In this paper, a novel Web Services (WSs) and process-view combined approach has been developed to manage the dynamic and distributed process of CPD to enhance the workflow interoperability among heterogeneous Workflow Management Systems (WMSs) of enterprises. Meanwhile, a hybrid P2P based WMSs framework has been built to provide an open and scalable architecture to deploy inter-enterprises collaboration. A software prototype system has been implemented using the open JXTA platform, and the collaborative development of a motorcycle has been used as a test case to validate the proposed approach and the prototype system.

1. Introduction

Collaborative Product Development (CPD) has been widely accepted as an advanced paradigm to combine distributed project teams from different enterprises to develop products in an effective means. As one of the important enabling technologies for CPD, Workflow Management Systems (WMSs), which can automate business processes through coordinating and controlling process and information flows between functional departments, have been successfully applied for intra-enterprise management [1–5]. For inter-enterprises collaboration, however, it is still a challenging research issue due to the dynamic and distributed nature of the problem. Web Services (WSs) are a set of new standards to support distributed computing and inter-enterprises collaboration. Some architectures, approaches and systems based on WSs have been introduced to manage inter-enterprises CPD [6–10]. However, the existing works are more suitable for managing relatively simple inter-enterprises processes. More research works based on WSs are expected to support complex collaborative activities such as global mechanical and electronic products development.

In the process management for inter-enterprises collaboration, especially for CPD, some research works have been reported recently. In the Workflow Management Coalition (WfMC) project, specifications were proposed to enable interoperability between heterogeneous workflow systems [11]. In the CrossFlow project, a service contracts-based approach was developed to support cross-organizational workflows interoperability between service providers and consumers, and an architecture for exchanging business processes as business services was designed [12,13]. Li and Qiu reviewed the state-of-the-art of asynchronous co-design process management [14]. Van der Aalst focused on Petri nets-based approaches to model inter-organizational workflows based on combined Petri nets and UML [15,16]. An improved TCPN approach was proposed to model dynamic product development processes, and the process models were constructed by composing some predefined basic TCPN workflow modules [17]. Tan et al. reported a Petri nets-based dynamic workflow model fragmentation approach for the execution of distributed workflows, and focused on the consistency of workflow models [18]. Grefen et al. discussed the dynamic business network process management in instant virtual enterprises to facilitate the composition of existing workflow models [19]. However, the interoperability of heterogeneous workflow models of cooperative enterprises, which is one of the major challenges in the research and applications of inter-enterprises collaboration and CPD, has not been effectively addressed in the above works.

To facilitate cross-organizational workflows interoperability and management, a Petri nets-based process-view approach was developed in our previous projects [20,21]. With the approach, enterprises can reuse their internal workflow models to quickly and flexibly construct the entire cooperative workflow models with partner enterprises according to different market requirements. However, the constraint of the approach is that the workflow modeling methods of the participating enterprises need

* Corresponding author.
E-mail address: gaoliang@mail.hust.edu.cn (L. Gao).

0166-3615/$ – see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.compind.2009.02.008

to be consistent, which is difficult to be satisfied in the real world. As a self-describing, self-contained and platform independent technology, WSs encapsulate implementation details so that they provide a reasonable solution to integrate workflows with different modeling methods. As thus, in this paper, a novel WSs and process-view combined approach has been developed to enable better inter-enterprises collaboration and CPD process management. The approach provides a more universal way to support inter-enterprises collaboration on the workflow level. In this approach, the individual workflow models of participating enterprises are first mapped to process-view workflow models. Then, these process-view workflow models are represented in Web Services Description Language (WSDL). Finally, the entire CPD process is built up through Business Process Execution Language for Web Services (BPEL4WS). Based on the proposed approach, a hybrid P2P based WMSs framework has been developed using the open JXTA platform. The collaborative development of a motorcycle has been used as a test case for systems validation.

The rest of this paper is organized as follows. Section 2 introduces the background of the process management of CPD, combination of WSs and process management, and process-view approach for B2B workflow interoperability. Section 3 presents the process-view based on Timed Coloured Petri Net (TCPN). Section 4 proposes a novel WSs and process-view combined approach for process management of CPD. Section 5 describes a hybrid P2P based WMSs framework. Section 6 presents a software prototype system and a case study of the collaborative development of a motorcycle. Section 7 provides the conclusion and future work.

2. Background

2.1. Requirements for process management of CPD

In order to better adapt to quickly changing market demands, the cooperative relationship among enterprises is moving from tightly coupled one to loosely coupled one. When a new order comes, a temporary enterprise alliance needs to be dynamically established to develop the product collaboratively. Usually, the WMSs of participating enterprises are heterogeneous in terms of workflow modeling method. To ensure the success of CPD across enterprises, it is vital to support the workflow interoperability among those heterogeneous WMSs. Under this scenario, the following new requirements are to be addressed:

- A tradeoff should be made between collaboration and autonomy when designing cooperative processes among participating enterprises, considering each enterprise is an independent economy entity.
- A universal and flexible workflow modeling method is needed to describe the entire process of CPD.
- Compared with traditional centralized WMSs, an advanced WMSs architecture should be designed to efficiently manage the dynamic and distributed CPD processes.
- Participating enterprises can execute their own business processes concurrently with partners.

2.2. Combination of WSs and process management

WSs provide an interoperative means between different software applications to be executed on a variety of platforms and frameworks [7]. WSs are composed of three open XML-based standards, i.e., Simple Object Access Protocol (SOAP), which is a services invocation protocol, WSDL, which describes the operational information about services, and Universal Description, Discovery and Integration (UDDI), which defines a common means to publish available services. WSs streamline business processes through creating an open, distributed system environment [22]. A lot of standards have been proposed to support interoperability of business processes using WSs, such as XPDL (XML Process Definition Language), WSCI (Web Services Choreography Interface), and BPEL4WS and so on. For instance, BPEL4WS is a formal specification of business processes and business interaction protocols to support the modeling of both executable and abstract processes.

Up to now, the existing workflow technology lacks a universal solution for distributed business processes management. The complementary relationship between WSs and process management was identified based on the state-of-the-art analysis of WSs. The combined strength of the two techniques was explored for inter-enterprises collaboration [23].

2.3. Process-view approach for B2B workflow interoperability

Motivated by the view concept of database, process-view has been proposed to facilitate B2B workflow interoperability. According to the different cooperative requirements of business, the private business process of an enterprise can be mapped to process-views to encapsulate sensitive business details and expose necessary business to partner enterprises for better cooperation management. The process-views act as workflow interoperability interfaces to realize inter-enterprises business collaboration while keeping autonomy of participating enterprises. Liu et al. proposed an order-preserving process-view approach to B2B workflow interoperability [24,25]. Chiu et al. adopted a workflow view approach to drive cross-organizational workflows interoperability in the WSs environment [26,27]. Karsten et al. developed a workflow view approach to support cross-organizational workflows, and focused on the communication between the entities of view-based workflow models [28].

To carry out successful business collaboration among cooperative enterprises, the sharing of information is necessary. On the other hand, participating enterprises need to keep sensitive details of private processes to preserve autonomy. A trade-off between collaboration and autonomy needs to be made by process modelers. In this research, a process-view approach has been developed to provide a multi-granularity abstraction mechanism to flexibly conceal information of internal private processes according to different business collaboration requirements. At the same time, the workflow models of enterprises internal business processes are reusable in the process-view approach, which avoids expensive computing and cost on remodeling.

3. Process-view based on TCPN

In this section, a process-view approach based on TCPN is described. The definition of TCPN-based process-view is first introduced. The mapping from TCPN workflow models to process-view workflow models is then presented. Finally, the collaborative execution of CPD workflow instances is implemented.

3.1. The definition of process-view based on TCPN

TCPN owns powerful modeling capability and supports the modeling of business processes.

Definition 1. A process-view based on TCPN can be defined as a bipartite directed graph represented by a nine-tuple $PV = (\Sigma, P', T', A, C, G, E, FT, I)$ satisfying the requirements below:

(1) Σ is a finite set of non-empty types, called colour sets.
(2) $P' = (BP, AP)$ is a finite set of places, BP is a finite set of base places that are the same as those in TCPN, and AP is a finite set of *abstract places*.

(3) $T' = (BT, LT)$ is a finite set of transitions, BT is a finite set of base transitions that are the same as those in TCPN, LT is a finite set of *logic transitions*, and $P' \cap T' = \phi$, $P' \cup T' \neq \phi$.
(4) A is a finite set of arcs.
(5) C is a colour function. It is defined from P' into Σ.
(6) G is a guard function. It is defined from T' into expressions such that:

$$\forall t \in T' : [Type(G(t)) = B \wedge Type(Var(G(t))) \subseteq \sum], B = \{true, \ false\}.$$

(7) E is an arc expression function. It is defined from A into expression such that:

$$\forall a \in A : [Type(E(a)) = C(p(a))_{MS} \wedge Type(Var(E(a))) \subseteq \sum].$$

(8) FT is a finite set of firing time, specifying the deterministic duration of the firing of each transition.
(9) I is the initialization function. It is defined from P' into closed expressions such that:

$$\forall \ p' \in P' : [Type(I(p')) = C(p')_{MS}].$$

Some syntax of this definition can be referred to [29]. In PV workflow models, each base transition and *abstract place* has a timestamp attached to it. The timestamp attached to a base transition specifies the deterministic firing duration of the base transition, and the timestamp attached to an *abstract place* specifies the duration of firing all base transitions that are contained by the *abstract place*.

Definition 2. An *abstract place* represents a sub-model of the TCPN workflow model, and is used to replace the sub-model to conceal some sensitive business details.

Since the *abstract place* may damage the integrity of a PV workflow model as a TCPN workflow model, the *logic transition* has been proposed to ensure that the PV workflow model is an integrity TCPN workflow model.

Definition 3. A *logic transition* does not represent a real activity, and is just used as logic control. The firing of a *logic transition* does not consume and produce tokens.

Fig. 1 describes the TCPN-based process-view approach for CPD workflows design. There are two temporary Enterprise Alliances (EAs) for CPD: EA1 and EA2. EA1 consists of enterprises A, B and C, and EA2 contains enterprises A, D and E. Each participating enterprise in EA maps its internal TCPN workflow model to a PV workflow model according to business collaboration requirements respectively, and then the integrated PV workflow model of EA can be obtained by integrating these PV workflow models. For the enterprise A, which is a member of EA1 as well as a member of EA2, its internal TCPN workflow model can be mapped to a corresponding PV workflow model (such as PV workflow model to EA1 or PV workflow model to EA2) when designing different CPD workflows.

Fig. 2 describes TCPN workflow models, PV workflow models and the integrated PV workflow model. Fig. 2(a) represents a TCPN workflow model that describes the design-manufacturing process of a product in an enterprise A. This product is made up of three parts. Part1 and Part2 are manufactured in the enterprise A, and Part3 is customized from its partner enterprise B. Fig. 2(e) represents a TCPN workflow model for the design-manufacturing process of Part3 in the enterprise B. To implement collaboration while keeping privacy, the two TCPN workflow models are mapped to PV workflow models according to business collaboration requirements respectively. Fig. 2(b) shows the PV workflow model mapped from the TCPN workflow model of the enterprise A, and

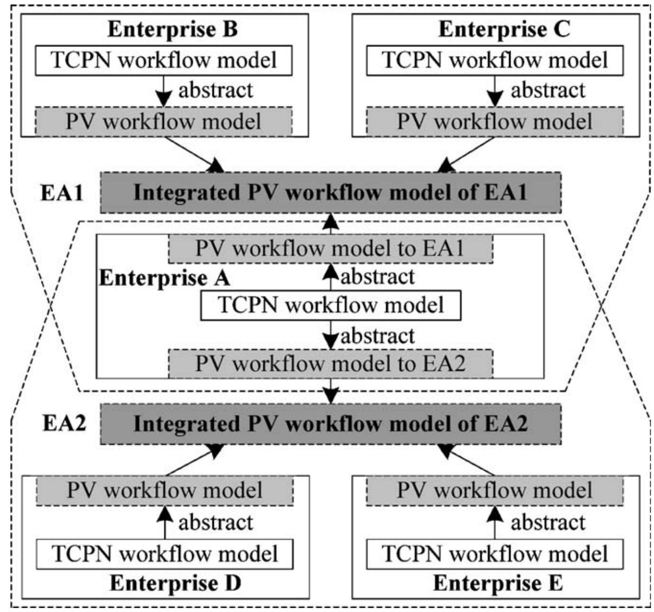


Fig. 1. The TCPN-based process-view approach for CPD workflows design.

the *abstract place* Part 1&2 that is described in the dashed circle hides the details of manufacturing processes of Parts 1 and 2. Fig. 2(d) describes the PV workflow model mapped from the TCPN workflow model of the enterprise B. The integrated PV workflow model represents the entire business collaborative process of a temporary EA, and it can be obtained by integrating the corresponding PV workflow models of enterprises A and B. Fig. 2(c) describes the integrated PV workflow model of enterprises A&B, which is obtained by integrating the two PV workflow models represented in Fig. 2(b) and (d).

PV workflow models and the integrated PV workflow model are the workflow interoperability interfaces among participating enterprises. The collaborative execution of the TCPN workflow instances, PV workflow instances and the integrated PV workflow instance can realize the business collaboration of a temporary enterprise alliance, which will be discussed in Section 3.3.

3.2. Mapping TCPN workflow models to PV workflow models

In order to automate the mapping from TCPN workflow models to PV workflow models, a TCPN algebra model and a character string mapping algorithm have been proposed.

Definition 4. A TCPN algebra model is a character string L consisting of P*i*, A*j*, T*k*, L*m*,+, –, *, /, [], () and N, where:

(1) P*i*, A*j*, T*k* and L*m* are the starting and ending characters representing base place, *abstract place*, base transition and *logic transition* respectively, where *i*, *j*, *k* and *m* are the sequence number.
(2) +, –, *, / represent the order structures of Or-Join, Or-Split, And-Join and And-Split respectively.
(3) [] represents the Loop structure, the places and transitions that are contained in it are executed repeatedly.
(4) () represents the parallel branch.
(5) N represents nothing.

The TCPN algebra model provides a cognizable way for computers to process a graphical TCPN workflow model. The automation of the mapping from a TCPN workflow model to a PV

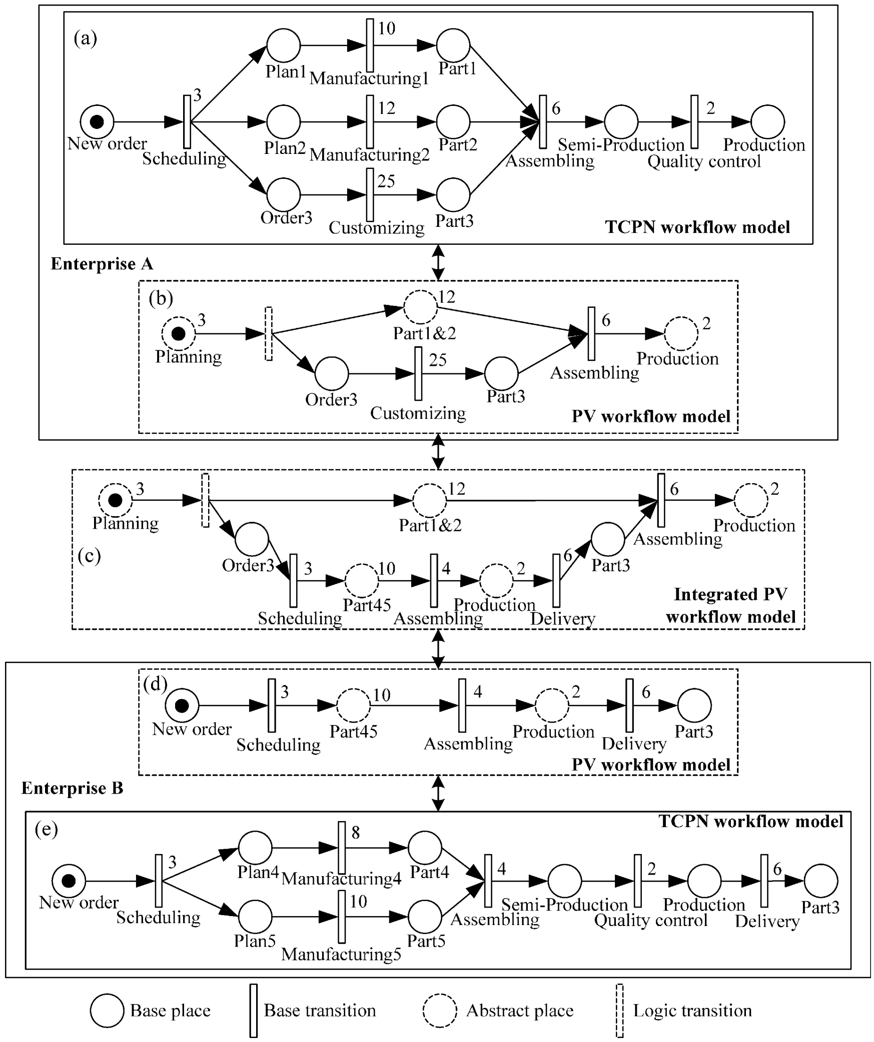


Fig. 2. TCPN workflow models, PV workflow models and the integrated PV workflow model.

workflow model can be implemented by intelligently processing character strings.

Fig. 3 describes a mapping case, and the mapping steps are as follows:

(1) Model the TCPN workflow model described in Fig. 3(a).
(2) Get the corresponding TCPN algebra model L described in Fig. 3(b): P01T01P02-(T02P03)(T03P04)*T04P05[T05P06T06P07]P08.
(3) Select the sub-models that are needed to be hidden in the TCPN workflow model. In Fig. 3(c), the sub-models {P02, T02, P03} and {T05, P06, T06, P07} surrounded by curving dash lines are the selected sub-models.
(4) Add signs to the TCPN algebra model L to denote the selected sub-models described in Fig. 3(d), where "{}" is chosen as the sign, and get the middle character string L′: P01T01{P02-(T02P03)}(T03P04)*T04P05{[T05P06T06P07]}P08.
(5) Process L′ by character string mapping algorithm, and the TCPN algebra model L″ of the PV workflow model can be obtained described in Fig. 3(e): P01T01A01-(N)(T03P04)*T04P05L01A02L02P08, where the sub-models {P02-(T02P03)} and {[T05P06T06P07]} in L′ are replaced by *abstract places* A01 and A02 in L″ respectively, and *logic transitions* L01 and L02 are added to L″.
(6) Get the PV workflow model from the TCPN algebra model L″ described in Fig. 3(f).

The key idea of the character string mapping algorithm is the reduction of graph complexity, and the matrix is adopted as the middle processing object in the mapping to simplify the algorithm. The main procedures of the character string mapping algorithm are as follows:

```
input ≫ TCPN1;                 // Input initial character string
TCPN1 = find_nodes(TCPN1);     // Find out all nodes
out_in_way(TCPN2);             // Search the split and join nodes
creat_graph(TCPN2);            // Transform the character string to a matrix
combine_graph(TCPN1);          // Replace the nodes that are needed to be
                                  hidden with necessary abstract place nodes
insert_L();                    // Add some necessary logic transition nodes
ans = output_graph();          // Transform the matrix into a character string
output(ans).                   // Output the last character string
```

3.3. Collaborative execution of CPD workflow instances

The TCPN-based process-view approach provides a way to design the integrated process model for enterprise collaboration. In order to realize the collaborative execution of CPD workflow instances (including the TCPN workflow instances, PV workflow

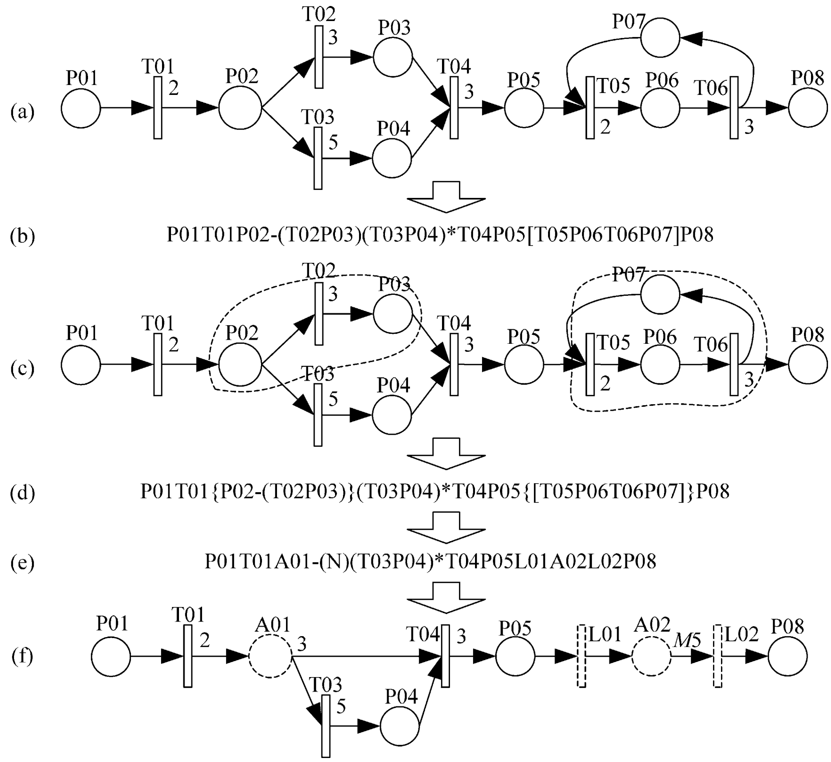


Fig. 3. The mapping from a TCPN workflow model to a PV workflow model: (a) The TCPN workflow model; (b) The TCPN algebra model L; (c) The TCPN workflow model with sub-models needed to be hidden; (d) The middle character string L′; (e) The TCPN algebra model L″; (f) The PV workflow model.

instances and integrated PV workflow instances), the *Synchronization Points* (SPs) are designated in the integrated PV workflow models. SPs are some base transitions in the integrated PV workflow models, and they can facilitate the exchange of workflows status information and the coordination of the workflows progress among the corresponding three kinds of workflow instances. A Multi-Agent System (MAS) was proposed in our previous projects [21] to manage these CPD workflows.

4. WSs and process-view combined approach for process management of CPD

The TCPN-based process-view approach merges the powerful modeling and simulation abilities of TCPN and the merits of process-view discussed in Section 2.3, which provides an effective way to facilitate cross-organizational workflows interoperability for CPD. However, the intrinsical constraint still exists. The TCPN workflow modeling method should be adopted by each participating enterprise. In order to solve this problem, a WSs and process-view combined approach has been developed to provide a flexible and universal way to manage the process of CPD.

4.1. Combination of WSs and process-view

There exist various workflow modeling methods in enterprises, such as Petri nets, UML, IDEF and so on. Some private workflow modeling methods are also proposed and applied in some enterprises. The inconsistency of workflow modeling methods adopted by cooperative enterprises embarrasses the workflows interoperability. It is infeasible to rebuild the WMSs of cooperative enterprises to provide a consistent platform for inter-enterprises collaboration. Facilitating the interoperability among existing WMSs of cooperative enterprises provides a promising way to realize inter-enterprises collaboration, which avoids much time and cost spent on developing new WMSs.

WSs provide an advanced interoperable platform for inter-enterprises collaboration, and the just-in-time composition of business processes supported by WSs architecture is suitable for dynamic process management of CPD. The novel WSs and process-view combined approach for CPD process management is described Fig. 4. There are four participating enterprises, and the workflow modeling methods adopted by all these enterprises are different. In order to realize business collaboration, the private workflow models of all participating enterprises are first mapped to corresponding process-view workflow models. These process-view workflow models are then encapsulated into WSs. Finally, the entire process of CPD can be obtained by composing these WSs. The WSs and process-view combined approach provides a universal and flexible way for cross-organizational workflows interoperability among heterogeneous WMSs while keeping the privacy and autonomy of participating enterprises.

We will introduce the mapping from PV workflow models to WSs in Section 4.2, and the composition of WSs to obtain the entire CPD process will be stated in Section 4.3.

4.2. Mapping PV workflow models to WSs

A WSDL document defines services as collections of network endpoints or ports, which is an XML grammar for specifying properties of a WS such as what it does, where it is located and how it is invoked. In WSDL, the abstract definition of endpoints and messages is separated from their concrete network deployment or data format bindings, which allows the reuse of abstract definitions [30]. A WSDL document uses the following elements in the definition of network services:

- Types. A container for data type definitions using some type system (such as XSD).
- Message. An abstract, typed definition of the data being communicated.

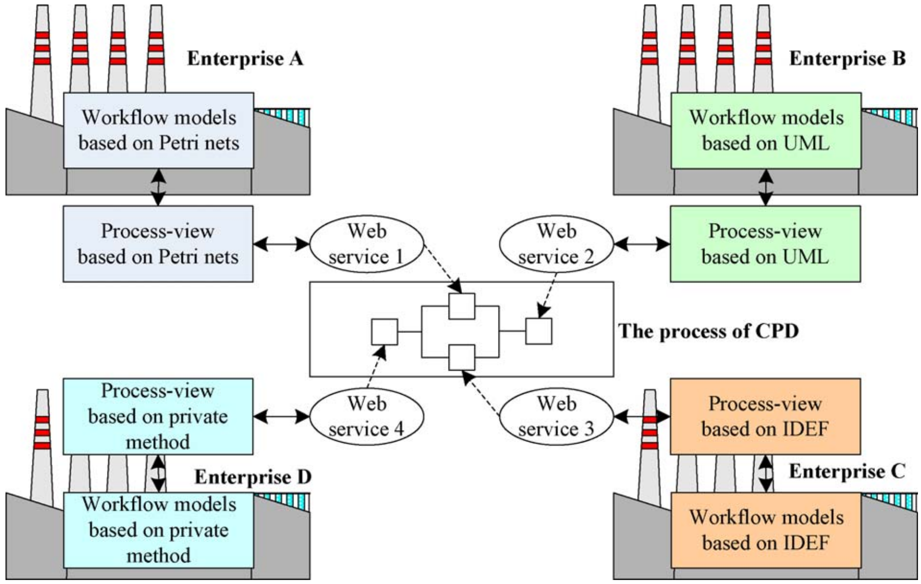


Fig. 4. The WSs and process-view combined approach for process management of CPD.

- Operation. An abstract description of an action supported by the service.
- Port Type. An abstract set of operations supported by one or more endpoints.
- Binding. A concrete protocol and data format specification for a particular port type.
- Port. A single endpoint defined as a combination of a binding and a network address.
- Service. A collection of related endpoints.

Petri Net Markup Language (PNML) is an XML-based interchange format for Petri nets. The places, transitions and arcs of Petri nets are all objects in PNML, and each object has a unique identifier which can be used to refer to this object. Some labels are attached to each object for assigning further meaning to the object, such as the initial marking of a place, the guard of a transition, and so on. More details about PNML can be referred to [31,32].

Fig. 5 shows the steps of the mapping from PV workflow models to WSs in WSDL, where the PNML file is adopted as the middle object.

Besides base places and base transitions, there exist reference places and reference transitions in PNML files, and they are used to describe the large size Petri nets system. To the PNML files that are mapped from the PV workflow models, the reference places and reference transitions are eliminated due to the PV workflow models are abstract models with small size, and *abstract places* and *logic transitions* are added to the PNML files.

After obtaining PNML files from the PV workflow models, the corresponding WSDL files can be mapped from these PNML files. The steps of the mapping from a PNML file to a WSDL file are as follows:

(1) Each base place object and *abstract place* object is mapped to a WSDL "message".
(2) Each base transition object and *logic transition* object is mapped to a WSDL "operation".

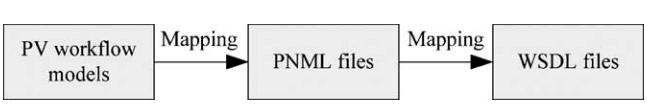


Fig. 5. The steps of mapping PV workflow models to WSDL files.

(3) Other information of the PNML file is mapped to corresponding elements of the WSDL file to realize the consistency.

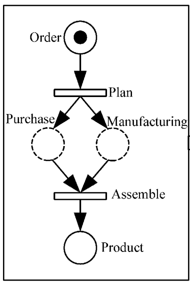
Fig. 6 describes the mapping from a PV workflow model to its corresponding WSDL file. Fig. 6(a) describes the PV workflow model of order processing, where *abstract places* Purchase and Manufacturing are used to conceal business details. The PNML file mapped from the PV workflow model is described in Fig. 6(b). Fig. 6(c) describes the WSDL file mapped from the PNML file, where the objects of base place Order, *abstract place* Purchase, *abstract place* Manufacturing and base place Product are mapped to WSDL "messages", and the objects of base transitions Plan and Assemble are mapped to WSDL "operations".

PNML has been proposed to facilitate the interoperability of different kinds of Petri nets, so the mapping methods presented above are suitable not only to the mapping from PV workflow models to WSDL files, but also to the mapping from other kinds of Petri nets-based process-view workflow models to WSDL files. The principles of the mapping from other workflow modeling methods (such as UML and IDEF) based process-view workflow models to WSDL files are similar to those of the mapping from PV workflow models to WSDL files, which are not discussed here due to the length limitation of this paper.

4.3. Composition of WSs

Composition of WSs enables collaboration among autonomous business organizations such that they can integrate their services to perform collaborative business activities [33]. After the process-view workflow models of enterprises have been encapsulated into corresponding WSs, then these WSs will be published on UDDI. The entire process of CPD will be obtained by dynamically composing some suitable WSs that are discovered from UDDI.

The process described in BPEL4WS is a flow-chart, in which each element is either a primitive or structured activity. A BPEL4WS process definition uses one or more WSDL services, and provides the description of the behavior and interaction of a process instance relative to its partners and resources through WSs interface [34]. BPEL4WS is utilized to describe the entire CPD process by composing suitable WSs in this paper. The synchronous and asynchronous collaboration among participating enterprises in CPD can be efficiently described by BPEL4WS process.



(a) PV workflow model

```
<pnml>
 <place id= "p1">
   <name>
     <value>order</value>
   </name>
   <initialMarking>
     <value>1</value>
   </initialMarking>
 </place>

 <arc id= "arc1" source= "p1" target= "t1">
   <inscription>
     <value>0</value>
   </inscription>
 </arc>

 <transition id= "t1">
   <name>
     <value>plan</value>
   </name>
 </transition>

 <arc id= "arc2" source= "t1" target= "a1">
   <inscription>
     <value>0</value>
   </inscription>
 </arc>

 <abstract place id= "a1">
   <name>
     <value>purchase</value>
   </name>
   <initialMarking>
     <value>0</value>
   </initialMarking>
 </abstract place>

 <arc id= "arc3" source= "t1" target= "a2">
   <inscription>
     <value>0</value>
   </inscription>
 </arc>

 <abstract place id= "a2">
   <name>
     <value>manufacturing</value>
   </name>
   <initialMarking>
     <value>0</value>
   </initialMarking>
 </abstract place>

 <arc id= "arc4" source= "a1" target= "t2">
   <inscription>
     <value>0</value>
   </inscription>
 </arc>

 <arc id= "arc5" source= "a2" target= "t2">
   <inscription>
     <value>0</value>
   </inscription>
 </arc>

 <transition id= "t2">
   <name>
     <value>assemble</value>
   </name>
 </transition>

 <arc id= "arc6" source= "t2" target= "p2">
   <inscription>
     <value>0</value>
   </inscription>
 </arc>

 <place id= "p2">
   <name>
     <value>product</value>
   </name>
   <initialMarking>
     <value>0</value>
   </initialMarking>
 </place>
</pnml>
```

(b) PNML file

```
<types>
    <XML Schema>
</types>
<message name= "order"/>
<message name= "purchase"/>
<message name= "manufacturing"/>
<message name= "product"/>
<portType name= "orderprocessingPortType">
    <operation name= "plan">
      <input message= "order"/>
      <output message= "purchase"/>
      <output message= "manufacturing"/>
    </operation>
    <operation name= "assemble">
      <input message= "purchase"/>
      <input message= "manufacturing"/>
      <output message= "product"/>
    </operation>
</portType>
<binding name= "orderprocessingBinding" type= "orderprocessingPortType">
    <soap:binding transport= http://schemas.xmlaosp.org/soap/http/>
    <operation name= "plan">
      <soap:operation soapAction= "http://hust.edu.cn/planorderprocessing.xsd">
    …
    </operation>
    <operation name= "assemble">
    …
    </operation>
</binding>
<service name= "orderprocessing">
    <port name= "orderprocessingPort" binding= "orderprocessingBinding">
      <soap:address location= http://www.pvtowsdl.hust.edu.cn/services/orderprocessing/>
    </port>
</service>
```

(c) WSDL file

Fig. 6. The mapping from a PV workflow model to a WSDL file.

In order to realize multi-level business collaboration among cooperative enterprises, different activities in BPEL4WS process can be bound to the same WSs and these activities use different segments of the service to implement different functions, which are called the WSs rebinding mechanism. In Fig. 7, activities 3, 4 and 6 are all bound to the same WSs, where activity 3 uses the segment 1 of the WSs, activity 4 uses the segment 2 of the WSs, and activity 6 uses the segment 3 of the WSs.

The collaborative development of a motorcycle has been studied to introduce the composition of WSs. The motorcycle is made up of five main systems, namely control system, fuel system, driving system, engine and power train. The control and fuel systems can be developed independently, but the driving system, engine and power train need to be developed collaboratively. Through the services discovery, five design-manufacturing WSs of the five systems are selected, which belong to three enterprises, namely the motorcycle enterprise, the engine enterprise and the power train enterprise. The BPEL4WS process described in Fig. 8 shows the entire process of the motorcycle collaborative development, where:

(1) Activity A represents the receiving of the order.
(2) Activity B represents the design-manufacturing of control system, which is bound to the WSs of control system.
(3) Activity C represents the design-manufacturing of fuel system, which is bound to the WSs of fuel system.
(4) Activities D, E and F represent the primitive design of the driving system, engine and power train respectively, which are bound to the WSs of driving system, engine and power train respectively.
(5) Activity G represents the discussion of the primitive design schemes of the three systems among the three enterprises, which is bound to all the three WSs, namely the WSs of driving system, engine and power train.
(6) Activities D′, E′ and F′ represent the manufacturing of the driving system, engine and power train respectively, which are

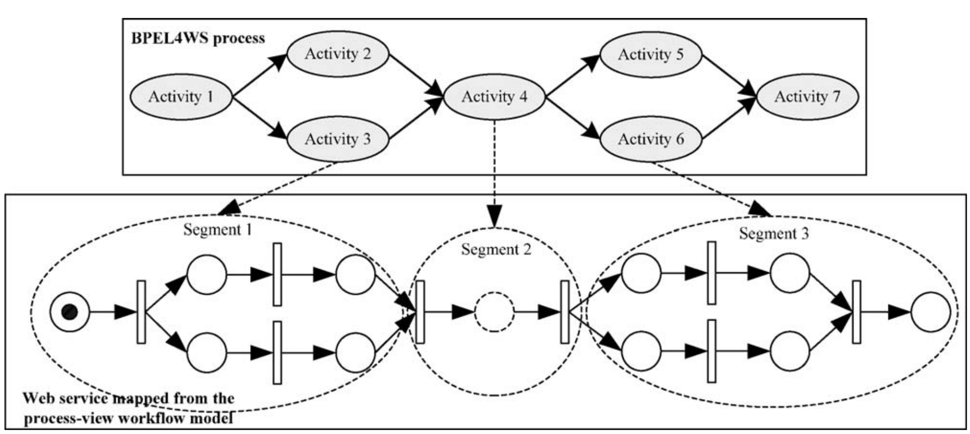


Fig. 7. WSs rebinding.

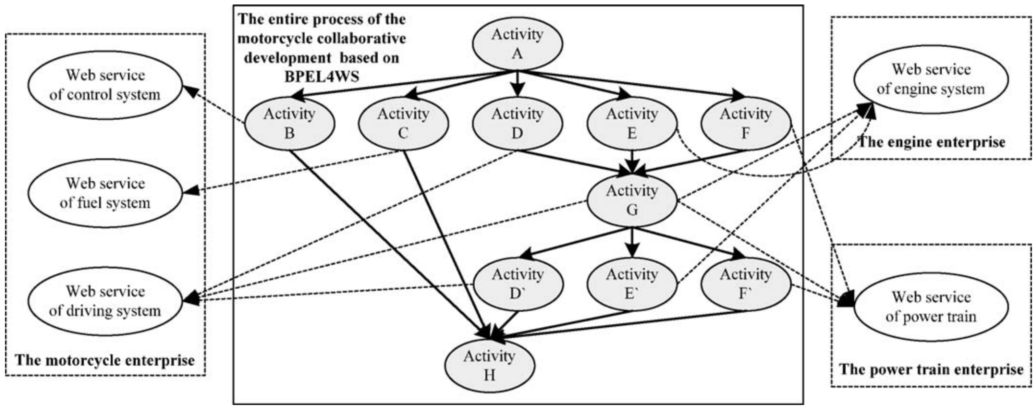


Fig. 8. The WSs composition of the motorcycle collaborative development.

bound to the WSs of driving system, engine and power train respectively.

(7) Activity H represents the assembling of the whole motorcycle.

In Fig. 8, the WSs of driving system is bound to the activities D, G and D′, the WSs of engine system is bound to the activities E, G and E′, and the WSs of power train is bound to the activities D, G and D′.

The process modelers of CPD can utilize the WSs rebinding mechanism to construct optimal business collaboration granularity among cooperative enterprises to realize maximal output.

5. Hybrid P2P based WMSs framework

In Section 4, a novel WSs and process-view combined approach is discussed. The approach provides a universal way to manage the dynamic and distributed process of CPD. In order to further facilitate the deployment and composition of process-view based WSs, a hybrid P2P based WMSs framework has been developed and is presented in this section.

Almost all the current WMSs adopt the dominating Client/Server (C/S) architecture, which provides centralized coordination and control [35]. These WMSs exert great effects in better process control and improvement in enterprises. However, the centralized WMSs have many limitations in managing dynamic and distributed inter-enterprises processes, such as single point of failure, limited scalability, poor performance and unsatisfactory system openness. P2P computing facilitates the sharing of computer resources and services by directly exchange between systems [36], which is a new and promising paradigm to manage the distributed workflow.

In order to leverage the advantages of C/S systems and P2P systems, a hybrid P2P WMSs framework based on the WSs and process-view combined approach has been developed to manage the process of CPD, which is described in Fig. 9. A super peer acts as the server and the extended WMSs of enterprises act as participant peers in the proposed framework. The connectivity of all peers is achieved through the Sun Microsystems's JXTA network interfaces. To the participant peer, an interface is added over the existing WMS, which consists of a Process-view Definition module, a Process-view Engine, a Process-view Instances Repository and a WSs Interface. The super peer consists of a UDDI, a BPEL4WS Process Engine, a BPEL4WS Process Definition module, a Process Instances Repository and an Exception Manager.

In order to realize business collaboration among enterprises in the hybrid P2P WMSs framework, process-view workflow models of enterprises are encapsulated into WSs and published on the UDDI through the WSs Interface. When a new order comes, the super peer designs the entire workflow model of CPD by composing suitable WSs that are discovered from the UDDI, and

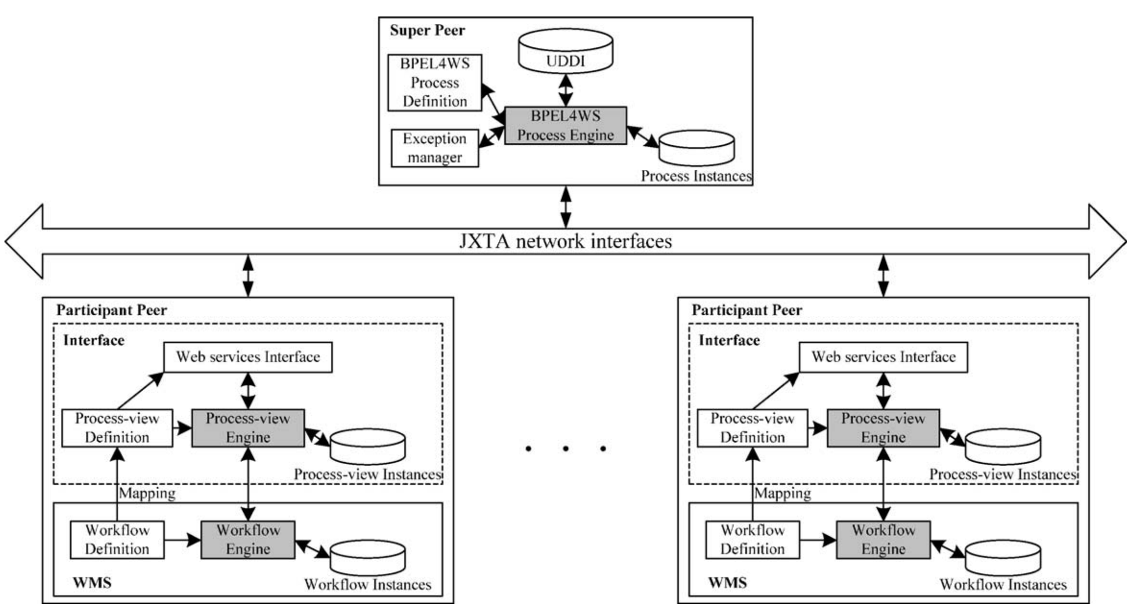


Fig. 9. Hybrid P2P based WMSs framework.

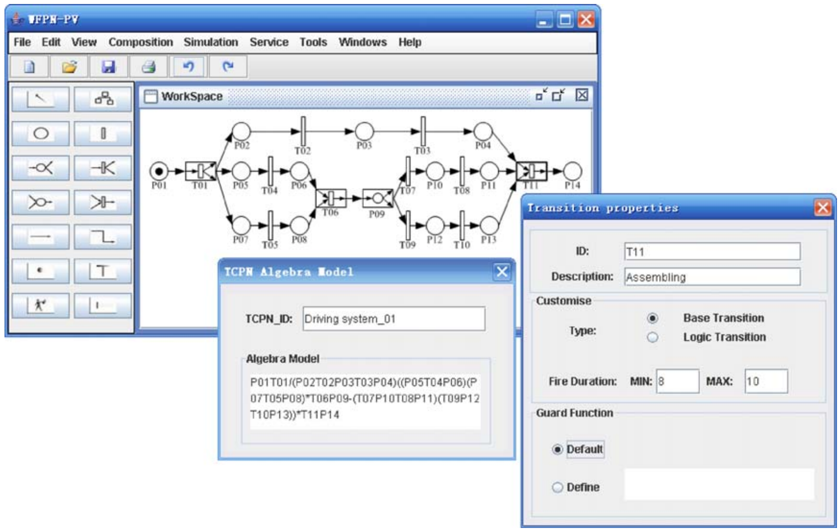


Fig. 10. TCPN workflow model of the driving system.

the providers of these WSs form a temporary EA for this CPD project. The execution of the CPD workflow instances (including the BPEL4WS workflow instance of the entire EA, the corresponding process-view workflow instances and private workflow instances of participating enterprises) have been implemented by the collaborative work among BPEL4WS Process Engine, the Process-view Engines and Workflow Engines. At the same time, the Exception Manager of the super peer takes charge of monitoring the execution of the BPEL4WS workflow instance and dealing with the failures and exceptions.

The hybrid P2P based WMSs framework provides an open and scalable architecture for process management of CPD, where temporary EA can be dynamically constructed and dissolved to adapt to quickly changed market demands.

6. Prototype system implementation and case study

The software prototype environment of this hybrid P2P based WMS has been implemented. The environment consists of two sub-systems: one is a TCPN and PV based WMS called WFPN-PV, the other is a WSs platform for CPD called CPD-WS.

WFPN-PV acts as the participant peer in the hybrid P2P based WMSs prototype system, where Java is the primary programming language for system implementation. The main functions of WFPN-PV are as follows:

(1) Model the processes of product development in TCPN workflow models.
(2) Encapsulate TCPN workflow models into WSs.
(3) Execute CPD workflow instances with super peer and other partner participant peers cooperatively.

CPD-WS acts as the super peer in the hybrid P2P based WMSs prototype system, where JWSDP (Java Web Services Development Pack) has been adopted to develop WSs. The main functions of CPD-WS are as follows:

(1) Act as UDDI server.
(2) Manage orders.
(3) Discover suitable WSs and design the entire CPD workflow model by composing these WSs.
(4) Execute CPD workflow instances with participant peers cooperatively.

The current hybrid P2P based WMSs prototype system is placed on the top of the open JXTA platform and the connectivity of all peers is achieved through the JXTA network interfaces. The providers of WSs for a tangible CPD project and the super peer form a peer group in JXTA virtual network. The communication between peers uses XML messages.

In the following subsections, we will demonstrate the prototype system using the case study of a motorcycle collaborative development.

6.1. Workflow modeling in WFPN-PV

As described in Section 4.3, there are five main systems in a motorcycle. The driving system is chosen to demonstrate the workflow modeling and WSs encapsulation using WFPN-PV. The driving system consists of a shock absorber, a wheel axle and accessories. The shock absorber and wheel axle need to be designed collaboratively in the motorcycle enterprise. Fig. 10 describes the design-manufacturing process of the driving system that is modeled in TCPN workflow model, and the id is Driving system_01. The corresponding TCPN algebra model is: P01T01/(P02T02P03T03P04)((P05T04P06)(P07T05P08)*T06P09-(T07P10T08P11)(T09P12T10P13))*T11P14. Fig. 11 describes the PV workflow model, the corresponding PNML file and WSDL file. The PV workflow model is first mapped from the TCPN workflow model of Driving system_01, and then the PNML file is obtained from the PV workflow model, finally, the WSDL file is mapped from the PNML file. Table 1 describes all the place and transition objects in Figs. 10 and 11.

After PV workflow models have been encapsulated into WSs in WFPN-PV, these WSs will be published on the CPD-WS platform with some added profile information.

6.2. WSs composition in CPD-WS platform

Fig. 12 describes the service profile of the driving system on the CPD-WS platform, where some basic descriptions and binding information are given.

When the order of the motorcycle comes, the manager of the CPD project will select suitable WSs from the CPD-WS platform. Fig. 13 shows the information about the CPD project and the five selected WSs. The providers of the five WSs form a temporary EA, and a collaboration contract will be signed. The entire process model of the

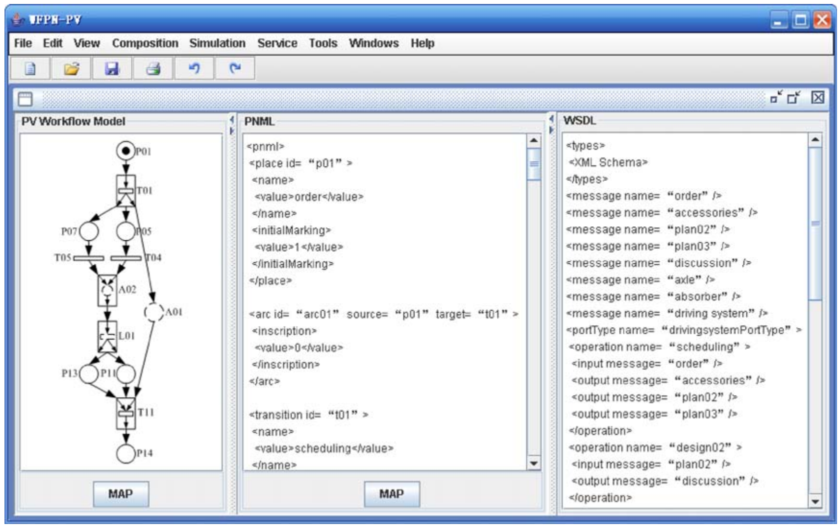


Fig. 11. The WSs encapsulation of PV workflow model.

Table 1
Object descriptions for Figs. 10 and 11.

Objects	Object descriptions	Objects	Object descriptions
P01	Order	A01	Design-manufacture of Accessories
P02	Accessories plan	A02	Design-manufacture of shock absorber and wheel axle
P03	Accessories design result	T01	Scheduling
P04	Accessories	T02	Design accessories
P05	Shock absorber plan	T03	Manufacture accessories
P06	Preliminary design result of shock absorber	T04	Design shock absorber
P07	Wheel axle plan	T05	Design wheel axle
P08	Preliminary design result of wheel axle	T06	Discuss the preliminary design results of shock absorber and wheel axle
P09	Improving schemas for shock absorber and wheel axle	T07	Improve preliminary result of shock absorber
P10	Final design result of shock absorber	T08	Manufacture shock absorber
P11	Shock absorber	T09	Improve preliminary result of wheel axle
P12	Final design result of wheel axle	T10	Manufacture wheel axle
P13	Wheel axle	T11	Assembling
P14	Driving system	L01	None

motorcycle collaborative development is designed by composing the five WSs in BPEL4WS, which have been described in Fig. 8.

6.3. Execution of CPD workflow instances

After all the workflow models of motorcycle collaborative development have been designed, the collaborative execution of corresponding workflow instances will be implemented through the communication between peers in the peer group of JXTA virtual network. The BPEL4WS Process Engine on the CPD-WS platform, the Process-view Engines and Workflow Engines in the WFPN-PVs work to implement the collaborative execution of all the CPD workflow instances.

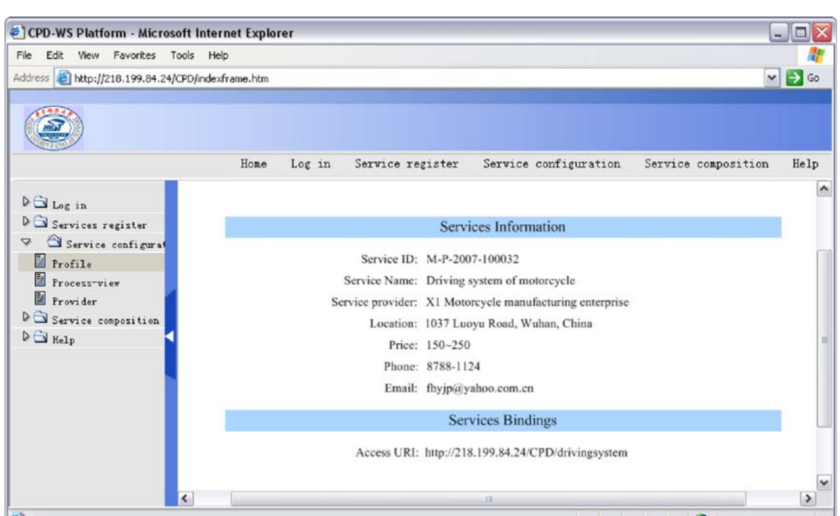


Fig. 12. The service profile of the driving system.

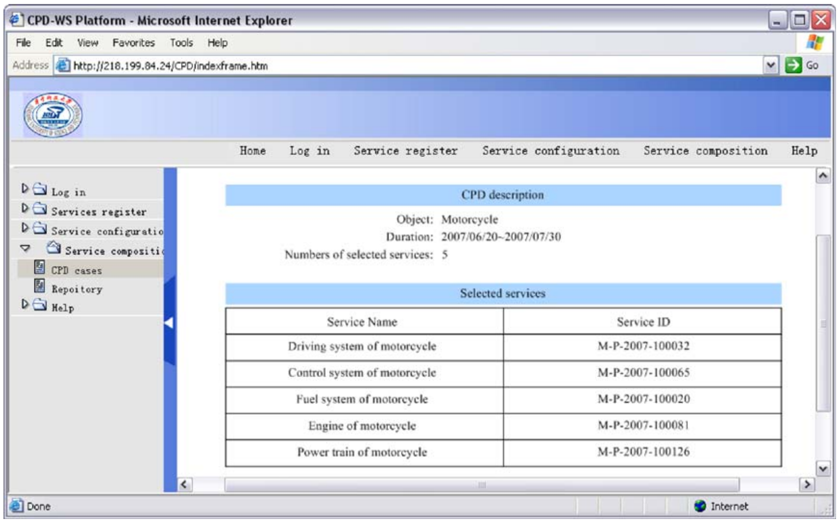


Fig. 13. The selected services for the collaborative development of a motorcycle.

7. Conclusion

In this paper, a novel and universal WSs and process-view combined approach has been developed to manage the dynamic and distributed process of CPD. The approach can facilitate the workflow interoperability between heterogeneous WMSs. In the research, the mapping from TCPN-based process-views to WSs in WSDL has been developed. BPEL4WS has been utilized to compose the suitable WSs to construct an entire CPD process. The WSs rebinding mechanism provides an efficient way to realize multi-level business collaboration between cooperative enterprises. The developed hybrid P2P based WMSs prototype system establishes an open and scalable architecture to support inter-enterprises collaboration.

The advantages and contributions of this research are summarized below:

(1) With the approach and the system, cooperative enterprises can realize workflow interoperability without changing their internal private workflow models. This characteristic enables enterprises to focus on their core business processes without investing further to remodel the processes.
(2) The WSs mechanism provides a neutral way to represent information between heterogeneous systems. The heterogeneous workflow models from enterprises are encapsulated using WSs, and therefore the interoperability of enterprise information is enhanced significantly.
(3) Cooperative enterprises can realize the multi-granularity workflow collaboration while keeping autonomy through the flexible information concealing mechanism of process-view.
(4) The rebinding mechanism of WSs can support the multi-level business collaboration between cooperative enterprises effectively. The mechanism provides an efficient way for designing the entire process of CPD to achieve the optimal output of collaboration.
(5) The hybrid P2P based WMSs prototype system provides an open and scalable architecture allowing enterprises to dynamically construct the temporary enterprise alliance according to the different market demands.

To apply the approach to more complex industrial situations, the multi-step mappings of workflow models are required when encapsulating private workflow models into WSs, and mapping algorithms and assistant systems need to be developed as well. Meanwhile, the collaborative execution mechanisms of CPD workflow instances are to be enhanced. Further research is being carried out towards the aspects.

Acknowledgments

This research has been supported by the National Basic Research Program (973 Program) of China under grant No. 2004CB719405, the Hi-Tech Research and Development Program of China under grant No. 2007AA04Z120, the Natural Science Foundation of China under grant No. 50705032 and No. 50675082, and the Ph.D. Programs Foundation of Ministry of Education of China No. 20070487081. The authors would like to thank the anonymous referees and the editors of this special issue for their valuable comments.

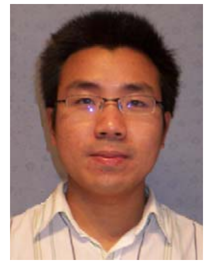
Appendix A. List for acronyms in this paper

AP	Abstract Place
EAs	Enterprise Alliances
BP	Base Place
BPEL4WS	Business Process Execution Language for Web Services
BT	Base Transition
CPD	Collaborative Product Development
C/S	Client-Server
JWSDP	Java Web Services Development Pack
LT	Logic Transition
MAS	Multi-Agent System
PNML	Petri Net Markup Language
PV	Process-View based on TCPN
SOAP	Simple Object Access Protocol
SPs	Synchronization Points
TCPN	Timed Coloured Petri Net
UDDI	Universal Description, Discovery and Integration
WMSs	Workflow Management Systems
WSCI	Web Services Choreography Interface
WSDL	Web Services Description Language
WSs	Web Services
XPDL	XML Process Definition Language

References

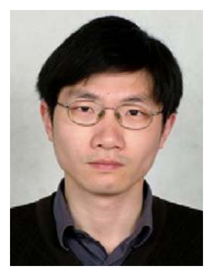
[1] E. Stohr, J. Zhao, Workflow automation: overview and research issues, Information Systems Frontiers 3 (3) (2001) 281–296.
[2] R. Fung, A. Au, A. Ip, Petri net-based workflow management systems for in-process control in a plastic processing plant, Journal of Materials Processing Technology 139 (2003) 302–309.

[3] W. Van der Aalst, On the automatic generation of workflow processes based on product structures, Computers in Industry 39 (1999) 97–111.
[4] J. Bae, S. Jeong, Y. Seo, Y. Kim, S. Kang, Integration of workflow management and simulation, Computers and Industrial Engineering 37 (1999) 203–206.
[5] A. Ortiz, F. Lario, L. Ros, M. Hawa, Building a production planning process with an approach based on CIMOSA and workflow management systems, Computers in Industry 40 (1999) 207–219.
[6] S. Wang, W. Shen, Q. Hao, An agent-based Web service workflow model for inter-enterprises collaboration, Expert Systems with Applications 31 (2006) 787–799.
[7] W. Kim, M. Chung, K. Qureshi, Y.K. Choi, WSCPC: an architecture using semantic Web services for collaborative product commerce, Computers in Industry 57 (2006) 787–796.
[8] J. Jung, H. Kim, S. Kang, Standards-based approaches to B2B workflow integration, Computers and Industrial Engineering 51 (2006) 321–334.
[9] K. Baina, K. Benali, C. Godart, DISCOBOLE: a service architecture for interconnecting workflow processes, Computers in Industry 57 (2006) 768–777.
[10] J. Lee, S. Lee, K. Kim, H. Kim, C. Kim, Process-centric engineering web services in a distributed and collaborative environment, Computers and Industrial Engineering 51 (2006) 297–308.
[11] Workflow Management Coalition, Interface 1: process definition interchange process model, Technical report, WfMC TC-1016-P, (1998).
[12] P. Grefen, K. Aberer, Y. Hoffner, H. Ludwig, CrossFlow: cross-organizational workflow management in dynamic virtual enterprises, Computer Systems Science and Engineering 15 (5) (2000) 277–290.
[13] IBM Research, CrossFlow architecture description, Technical report, ESPRIT CrossFlow EP 28653, (1999).
[14] W. Li, Z. Qiu, State-of-the-art technologies and methodologies for collaborative product development systems, International Journal of Production Research 44 (31) (2006) 2525–2559.
[15] W. Van der Aalst, Loosely coupled interorganizational workflows: modeling and analyzing workflows crossing organizational boundaries, Information and Management 37 (2000) 67–75.
[16] W. Van der Aalst, Interorganizational workflows: an approach based on message sequence charts and Petri nets, Systems Analysis–Modelling–Simulation 34 (3) (1999) 335–367.
[17] S. Ha, H. Suh, A timed colored Petri nets modeling for dynamic workflow in product development process, Computers in Industry 59 (2008) 193–209.
[18] W. Tan, Y. Fan, Dynamic workflow model fragmentation for distributed execution, Computers in Industry 58 (2007) 381–391.
[19] P. Grefen, N. Mehandjiev, et al., Dynamic business network process management in instant virtual enterprise, Computers in Industry 60 (2009) 86–103.
[20] X. Shao, P. Jiang, H. Qiu, L. Gao, Workflow modeling for virtual enterprise: a Petri net based process-view approach, in: Proceedings of the 10th International Conference on Computer Supported Cooperative Work in Design (CSCWD), 2006, pp. 1213–1218.
[21] P. Jiang, X. Shao, H. Qiu, P. Li, Interoperability of cross-organizational workflows based on process-view for collaborative product development, Concurrent Engineering: Research and Applications 16 (1) (2008) 73–87.
[22] K. Gottschalk, S. Graham, H. Kreger, J. Snell, Introduction to Web services architecture, IBM System Journal 41 (2) (2002) 170–177.
[23] J. Zhao, H. Cheng, Web services and process management: a union of convenience or a new area of research? Decision Support Systems 40 (2005) 1–8.
[24] D. Liu, M. Shen, Workflow modeling for virtual processes: an order-preserving process-view approach, Information Systems 28 (2003) 505–532.
[25] D. Liu, M. Shen, Business-to-business workflow interoperation based on process-views, Decision Support Systems 38 (2004) 399–419.
[26] D. Chiu, S. Cheung, S. Till, K. Karlapalem, Q. Li, E. Kafeza, Workflow view driven cross-organizational interoperability in a Web service environment, Information Technology and Management 5 (2004) 221–250.
[27] Z. Shan, D. Chiu, Q. Li, Systematic interaction management in a workflow view based B2B process engine, in: Proceedings of the 38th Hawaii International Conference on System Sciences, 2005.
[28] A. Karsten, E. Maria, Facilitating cross-organizational workflows with a workflow view approach, Data and Knowledge Engineering 51 (2004) 109–147.
[29] K. Jensen, Coloured Petri nets-basic concepts, analysis methods and practical use, second ed., Springer, New York, 1997, 69–73.
[30] WSDL, http://www.w3.org/TR/2001/NOTE-wsdl-20010315.
[31] M. Weber, E. Kindler, The Petri net markup language, LNCS 2472 (2003) 124–144.
[32] J. Billington, et al., The Petri net markup language: concepts, technology, and tools, LNCS 2679 (2003) 483–505.
[33] M. Younas, I. Awan, D. Duce, An efficient composition of Web services with active network support, Expert Systems with Applications 31 (2006) 859–869.
[34] J. Shen, G. Grossmann, Y. Yang, M. Stumptner, M. Schrefl, T. Reiter, Analysis of business process integration in Web services context, Future Generation Computer Systems 23 (2007) 283–294.
[35] J. Yan, Y. Yang, G. Raikundalia, SwinDew-a p2p-based decentralized workflow management system, IEEE Transaction on Systems, Man, and Cybernetics-Part A, Systems and Humans 36 (5) (2006) 922–935.
[36] G. Fakas, B. Karakostas, A peer to peer (P2P) architecture for dynamic workflow management, Information and Software Technology 46 (2004) 423–431.



Ping Jiang received his BS and PhD degrees from the School of Mechanical Science and Engineering, Huazhong University of Science and Technology (HUST) in 2004 and 2008. He is now working on postdoctor study in the state key laboratory of digital manufacturing equipment and technology, HUST. His research interests include collaborative design, process management of product development, cross-organizational workflow, Petri nets, etc.

Xinyu Shao became a PhD candidate of the School of Mechanical Science and Engineering, Huazhong University of Science and Technology (HUST), China, in 1992. From 1995 to 1998, he had been studying and working in the University of Michigan and Ford Motor Company as a research associate to continue his PhD program and research work. He got his PhD degree and became an associate professor of HUST in 1998. He is now a full professor and the dean of the School of Mechanical Science and Engineering of HUST. His research interests include CAD/CAPP/PDM/ERP, CIMS, Concurrent Engineering, Agile Manufacturing, electric vehicles, etc. He has published more than 50 research papers.



Haobo Qiu received his BS and MS degrees from Nanjing University of Aeronautics and Astronautics in 1997 and 2000. In 2004, He received his PhD degree in Shanghai Jiaotong University. He is now a member of National key laboratory of Digital Manufacturing Equipment and Technology, Huazhong University of Science and Technology (HUST). His research interests include collaborative design methods and system, process management in design, design and manufacturing integration etc. He has published over 30 technical and conference papers.

Liang Gao received his BS degree in mechatronic engineering from Xi'dian University, China, in 1996, and the PhD degree in mechatronic engineering from Huazhong University of Science and Technology (HUST), China, 2002. He is now a full professor in the Department of Industrial & Manufacturing Systems Engineering, School of Mechanical Science and Engineering, HUST. His research interests include collaborative design, shop scheduling and swarm intelligence, etc. He has published more than 50 research papers.

Peigen Li received his MS degree from Huazhong University of Science and Technology (HUST) in 1981, and PhD degree at University of Wisconsin-Madison in 1987. Since 1987, he has been working at HUST as a faculty member. Dr. Li became a professor in 1991 and the dean of the School of Mechanical Science and Engineering of HUST in 1995. He is now the president of HUST. His research interests include CAD, CAPP, CIMS, etc. In 1999, HUST got University Lead Award given by SME/CASA due to a large CIMS program of which Dr. Li was one of the key members.