

# An Extended Petri-Net Based Approach for Supply Chain Process Enactment in Resource-Centric Web Service Environment

Xiaodong Wang<sup>2</sup>, Xiaoyu Zhang<sup>2</sup>, Hongming Cai<sup>1</sup>, and Boyi Xu<sup>3</sup>

<sup>1</sup> School of Software, Shanghai JiaoTong University, Shanghai, China

<sup>2</sup> BIT Institute, University of Mannheim, Mannheim, Germany

<sup>3</sup> Antai College of Economic&Management, Shanghai JiaoTong University, Shanghai, China

{wangxd.sjtu,xiaoyu.zhang.cn}@gmail.com,  
{hmcai,byxu}@sjtu.edu.cn

**Abstract.** Enacting a supply-chain process involves variant partners and different IT systems. REST receives increasing attention for distributed systems with loosely coupled resources. Nevertheless, resource model incompatibilities and conflicts prevent effective process modeling and deployment in resource-centric Web service environment. In this paper, a Petri-net based framework for supply-chain process integration is proposed. A resource meta-model is constructed to represent the basic information of resources. Then based on resource meta-model, XML schemas and documents are derived, which represent resources and their states in Petri-net. Thereafter, XML-net, a high level Petri-net, is employed for modeling control and data flow of process. From process model in XML-net, RESTful services and choreography descriptions are deduced. Therefore, unified resource representation and RESTful services description are proposed for cross-system integration in a more effective way. A case study is given to illustrate the approach and the desirable features of the approach are discussed.

**Keywords:** supply chain, RESTful web service, Petri-Net, XML-Net, WADL.

## 1 Introduction

Supply Chain Process is a special business process which interrelates production, logistics and information. Distinguishing from other business processes, supply chain processes span usually departments and organizations. They are generally performed by different independent firms in order to produce and deliver a specified range of goods and services. A supply chain process is normally very complex, while it often involves lots of people, data, materials, products, rules and the relationships among them. The supporting IT systems are accordingly always distributed and heterogeneous. Furthermore, unstructured applications and data also make it difficult to integrate supply chain processes with business entities and services provided by various sources.

Service Oriented Architecture (SOA) [1] is now accepted as a popular construction approach for inter-organizational business process integration and composition of

applications across systems. In the paradigm of SOA, business services are mapped to IT services, which are mainly implemented based on SOAP (Simple Object Access Protocol) protocol. However, SOAP services are generally complicated. Hence discovering and matching services for integration of business processes are arduous work. Moreover, the representation for the semantics of business entities in SOA framework is still missing.

Currently, REST (Representational State Transfer) [2] receives increasing attention. Compared with SOA, RESTful architecture requires much less work for development. The Resource-Oriented Architecture (ROA) is based on the concept of the resource, which is an abstraction of information [3] and is referenced with a global identifier (URI) containing the name and the address of the resource. From the resource-oriented viewpoint, the entities engaged in business process could be abstracted as resources. Based on this standpoint, business process will be regarded as interactions among resources and resources' states transformation. Accordingly, in a resource-oriented environment, the access to resources is achieved through RESTful Web Services using the Hypertext Transfer Protocol (HTTP).

Integrating supply chain process across organizational and heterogenous systems requires the unified resource modeling and representation. Correlatively, widely accepted modeling tools for describing interaction processes of resources are indispensable. Finally, universal description for RESTful services and their compositions should be generated according to resource models, so as to encapsulate the various implementations from different systems.

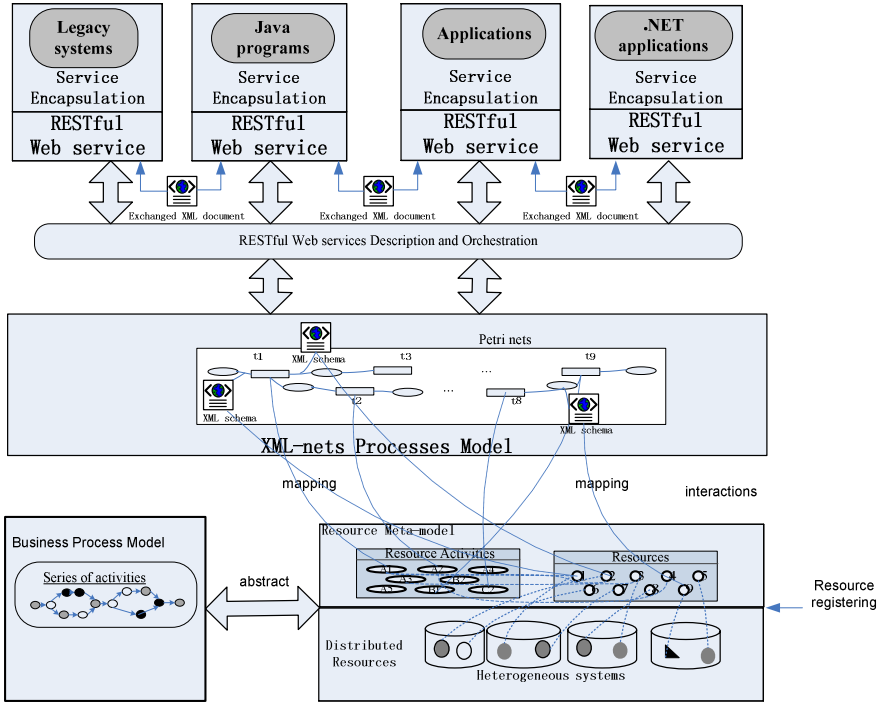
In this paper, an extended Petri-net based approach is proposed for modeling and integrating supply chain process in RESTful service environment. To obtain unified definitions of resources, a resource meta-model is built to represent the basic information elements of resources, so as to conceal the incompatibilities among resources from heterogeneous applications. According to this resource meta-model, XML schemas and documents are derived, which describe resources and their states in Petri nets. As self-described messages, the information encapsulated in XML could be simply exchanged through HTTP, which makes inter-operability across systems possible. A high level Petri-net, so-called XML-net is employed for modeling processes of interactions among resources. The operations on resources could be then translated into manipulations of XML documents. Finally, the unified RESTful services description in WADL (Web Application Description Language) and orchestration in extended BPEL are deduced from process models in XML-net. Thereby diversiform applications could be encapsulated in RESTful services and integrated into complete supply chain process. Finally, the software architecture for implementation and a case study are given to verify and illustrate the approach.

## **2 Overview of the Process Integration Approach Based on XML-Nets**

From the resource-oriented viewpoint, business processes could be regarded as a set of activities, which change resource properties and transform their states. Therefore, resource, the interaction of resources with process, and IT implementation of process models are three crucial factors to realize process integration in a RESTful service environment. Hereby unified definition and representation of resource model are

required firstly. In our approach, resource meta-model describes the structures of resources. Thereafter, a high-level Petri-net is employed to model and represent interactions among resources and their states transformation. Finally, for execution and deployment, RESTful web services description and orchestration are derived from process model in Petri-net, in this way the diversiform applications will be encapsulated in RESTful services and organized to complete execution of business processes.

Fig. 1 illustrates the main idea of our approach.



**Fig. 1.** The main idea of integrated framework based on XML-nets processes model

Firstly, the information of business entities is mapped as resources. To achieve supply chain process integration, all resources coming from existing distributed and heterogeneous systems have to be identified, analyzed and encapsulated, so as to hide the data differences among different systems. Therefore, it is very necessary to define a unified structure for resource modeling. Based on workflow analysis, a resource meta-model is built, which defines not only the internal attributes and properties of a specific resource, but also its external interfaces and behaviors. The resource meta-model contains four main components: Resource Properties, Operation Set, Activity State and Rule Model. The Resource Properties and Operation Set represent separately the internal attributes and operations of a resource, while Activity State and Rule Set define the external behaviors of a resource. A business process contains several activities, in which the correlative resources can be invoked by RESTful service. In section 3, the formal expression of resource meta-model will be presented.

After modeling resources with resource meta-model, XML schemas for corresponsive resources are derived. These XML schemas act as resource views in Petri nets during the process modeling. Likewise, concrete resource objects and their states are described in XML documents.

In a REST framework, the control flow of process is essentially regarded as resources states and their transformation. To describe the states transformation of resources and enact supply chain processes, a specific class of high-level Petri-net, so-called XML-net, is employed. XML-net is a formal, graphical modeling language which was developed by Oberweis and Lenz [4] [5]. It integrates the concept of complex data objects into Petri-net formalism [6]. In XML nets, the static components are typed by XML schema diagrams, each of them representing a XML schema. Places can thus be regarded as containers for XML documents that are valid for the corresponding XML schema. These XML documents conserve the resources states in runtime. The flow of XML documents is defined by occurrences of transitions, i.e., the activities taking place. So-called filter schemas label the adjacent arcs and thereby describe the documents that are related to the activities and the way these documents are manipulated. In our approach, places in XML-net could be regarded as containers for XML-documents and transitions in XML-net represent the operations on resources. With XML-net, control flow of supply chain processes and the states transferring of resources can be integrated well into process models.

However, a supply chain process model in XML-net could not be directly executed in a RESTful services system. Therefore it is necessary to interpret such processes with machine-processable specifications for HTTP based web applications [7]. In this paper, RESTful Web services are described with WADL, which is a resource-centric description language and can hereby combine resources representation with Web service description more appropriately. Briefly speaking, the resource definitions in WADL are derived from XML schemas. The operation definitions could be deduced from transitions, and operations set of resource builds up the port of correlative RESTful web service.

The Business Process Execution Language (BPEL) [8] is used as a standard service composition for modeling and executing business process. However it cannot be applied directly in RESTful service environment. To composite RESTful Web services into processes, a BPEL extension for RESTful is proposed in [9]. Based on process models in XML nets, the RESTful services description in WADL and services composition in extended BPEL could be automatically generated. In this way, RESTful web services provide a resource-oriented platform for interoperability among legacy systems, java programs, application interfaces, and .net applications.

The service descriptions in WADL provide general interfaces for accessing to resources, through which diversiform applications in heterogeneous systems are encapsulated into RESTful services. And exchanged data are packed in self-described XML documents, which facilitate the communication and cooperation among participants in supply chain process.

### 3 Resources and Process Modeling

In order to realize process integration for supply chain application in a resource-centric environment, resource model, interactive mechanism between resource and

activities, and services description generation are main three theses of our approach. Therefore, resource model and its formal expression are given in section 3.1. Based on the XML-net, interactive mechanism is given in section 3.2. The last section is used to build WADL and BEPL expression for execution in the application.

### 3.1 Resource Modeling and Formal Expression

Based on information rather than process, the nature of business process is the executing and transforming process of resources. Generally speaking, a business process step includes some procedures, in which business information of resource models is carried out and properties of resource models are changed according to each certain operation, and the corresponding parameters along with the messages are transferred to the next business process step. To achieve the goal of business process integration, all resources in existing distributed and heterogeneous systems have to be firstly identified, analyzed and encapsulated. Then all encapsulated resources will be registered into the resource integration platform under the unified management, so as to hide the data differences among different systems. Finally the resources are able to achieve interoperability in the way of RESTful Web Services. Therefore, it is very necessary to define a unified structure for resource modeling.

Based on workflow analysis, a resource meta-model is proposed. With the aid of this method of resource modeling, the integrated platform can organize and manage the static and dynamic messages in the enterprise system, as well as establish an open and extensible framework for business application system so as to satisfy the integration requirements of distributed, dynamic, flexible, and typically heterogeneous systems.

Fig. 2 illustrates the resource meta-model. The resource meta-model contains four main components: Resource Properties, Operation Set, Activity State and Rule Model. The Resource Properties and Operation Set represent separately the internal attributes and operations of a resource, while Activity State and Rule Set define the external behaviors of a resource. A business process contains several activities, in which the correlative states transformation of resources can be implemented through RESTful service. All these components are defined accurately as following.

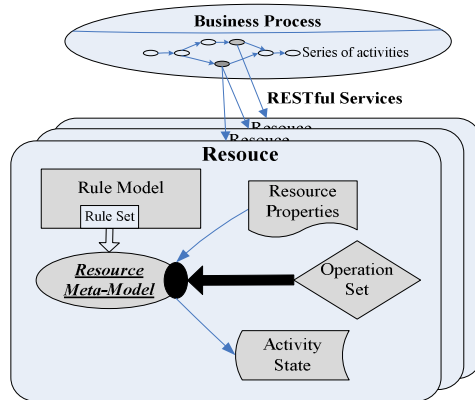


Fig. 2. Resource Meta-Model

**Definition 1.** Resource Meta-Model (RMM)

The Resource Meta-Model is defined as a set of Resource Property (RP), Operation Table (OT), Rule Set (RS) and Activity State (AS). Thus the Resource Meta-Model (RMM) is defined as the following tuple:

$$\text{RMM} = \langle \text{RP}, \text{OT}, \text{RS}, \text{AS} \rangle$$

**Definition 2.** Property Set (PS)

The Property Set (PS) is defined as a following set of properties.

$$\text{PS} = \{p_1, p_2, \dots\}, |\text{PS}| > 0.$$

For each property  $p_i \in \text{PS}$ ,  $p_i = \langle \text{PropertyName}, \text{InitialValue} \rangle$

Hereby  $p_i$  indicates the content of a single property, and the PS is a set of several properties. Every property  $p_i$  uses two main fields to keep its name and its initial value separately.

**Definition 3.** Resource Property (RP)

The Resource Property is defined by the following tuple:

$$\text{RP} = \langle \text{ResID}, \text{ResType}, \text{SProps}, \text{DProps} \rangle$$

Where  $\text{SProps} = \{rp_i \mid rp_i \in \text{PS} \wedge \text{Static}(rp_i) = \text{TRUE}\}$ ,

$\text{DProps} = \{rp_i \mid rp_i \in \text{PS} \wedge \text{Static}(rp_i) \neq \text{TRUE}\}$ .

Hereby the ResID is the unique identifier of a resource entity. ResType is the underlying category, which describes the type of a resource with a specific static property and can be organized by using a hierarchical tree method. SProps includes the static properties of a resource, and here the “static” means that the values of these properties will never be changed once created, e.g. the manufacturing data of a product. Correspondingly DProps includes the dynamic properties, which are used to describe the real-time status, usage calendar and lifecycle status of resources and so on, e.g. the status (created, locked, outgoing, and archived) of orders in a sales process. Dynamic properties can also be used to describe the status of resources in business process steps, because that they are suitable for workflow modeling.

**Definition 4.** Operation Table (OT)

The Operation Table is defined as the following set of operations.

$$\text{OT} = \{o_1, o_2, \dots\}, |\text{OT}| > 0$$

where  $o_i = \langle \text{OpName}, \text{OpBody}, \text{VariableTable} \rangle$

Each operation  $o_i$  includes three fields: OpName indicates the name of an operation, OpBody defines the body of the operation and all the variables used by the operation are defined in the field VariableTable.

**Definition 5.** Activity State (AS)

The Activity State is defined by the following tuple:

$$\text{AS} = \langle s_{\text{start}}, s_{\text{end}}, S_n \rangle$$

where  $S_n = \{s_1, s_2, s_3, \dots\}$ ,  $|S_n| > 0$ ,  $s_{\text{start}} \in S_n$ ,  $s_{\text{end}} \in S_n$

$s_i = \langle \text{StateName} \rangle$

$s_{\text{start}}$  indicates the initial state of a resource in the state diagram, and  $s_{\text{end}}$  means the end state. Each  $s_i$  in the set  $S_n$  defines an intermediate state of the resource.

**Definition 6. Rule Set (RS)**

Rules Set is used to reflect the rules and restricts of a resource status transition in the process.

$$RS = \{r \mid r = \langle \langle s_i, RP, \rangle, s_j \rangle, (s_i \in \{\emptyset, s_{start}\} \cup S_n) \wedge (s_j \in \{s_{end}\} \cup S_n), c_k \in 2^{CS} \}$$

Here  $s_i$  indicates the current workflow state, which is generally defined in AS. If the  $s_i$  in the corresponding rule is null, the rule only needs to check the current resource property and the constraints defined by  $c_k$ .  $c_k$  defines a constraint combination. The status will be set to  $s_j$  if the constraint is satisfied with the current resource property. E.g., when the order's goods are available in the inventory, the status can be converted as activated. Business rules are organized based on resources, and the choreography of resource services can be more convenient through compounding with resource rules in the integration of processes.

**3.2 XML-Net Component Derivation for Resource Process Interaction**

In order to analyze the interactive mechanism of resources and resource-based activities, XML-net is introduced to act as a media. In a resource-based environment, all messages exchange between partners in supply chain via HTTP request/response interactions. Such interaction processes are modeled in XML-net. The formal definition of XML-net is presented as follows:

**Definition 7. XML-Net (XN)**

A XML net is a tuple =  $\langle P, T, F, \text{SchemaXN}, \text{FuncP}, \text{FuncT}, \text{FuncURI} \rangle$  where

- $P$  and  $T$  are disjoint sets of places and transitions
- $F \subseteq (P \times T) \cup (T \times P)$  is the flow relation
- $\text{SchemaXN}$  are the XML schema related to places.
- A function  $\text{FuncP}$  assigns to each place an XML schema  $\text{SchemaXN}$
- $\text{FuncT}$  is a function assigning guard conditions to transitions, where a condition  $\text{FuncT}(t): (\bullet t \rightarrow \text{XML})$  specifies combination of input XML documents
- $\text{FuncURI}$  is a function assigning URIs to tuples of communication transitions and combinations of input of input documents, i.e.  $\text{FuncURI}(t) : (\bullet t \rightarrow \text{XML}) \rightarrow \text{URI}$
- A set of valid XML documents are assigned to each place of XML-net as initial marking.

Based on resource meta-model, process models in XML-net could be deduced. In a XML-net process model, XML schemas with unified definition are deduced from resources model. They prescribe the properties of a resource and possible value ranges. Thereafter these XML schemas are utilized for verifying the exchanged XML documents between transitions in XML-net process models. During runtime, XML documents are generated and contain the states of resources. And Places in XML-net could be regarded as containers for XML documents. Similarly, transitions in XML-net are derived from the operations set on resources. In Resource Meta-Model, each resource has its own operations set, in which operations are defined. These operations read or modify the properties values of the resource object, so that the state of

resource as well as the state of process will also be changed. In XML-net, the operations which read or change states of resource will be interpreted as transitions. The rules in resource meta-model prescribe the states transferring. Combining with operations and rules set, the flow relation F in XML could be also deduced.

In process models, XML documents describing resource states are generated according to the XML schemas and the data fields will be evaluated according to resource model.

The mapping functions could be defined as follows:

**Definition 8. XML schema Mapping**  $\phi_R(\mathbf{R}) = \{(\mathbf{R}_i, \text{SchemaXN}_i)\}$

The mapping function  $\phi_R(\mathbf{R})$  defines the mapping relation between resource models and XML schemas in XML-net. For each resource, there is a corresponsive XML schema which describes it.

**Definition 9. Transition Mapping**  $\phi_T(\mathbf{R}, \mathbf{OT}) = \{((\mathbf{R}_i, \mathbf{o}_k), \mathbf{t}_j) \mid \mathbf{o}_k \in \mathbf{OT}_i, \mathbf{t}_j \in \mathbf{T}\}$

For operation  $\mathbf{o}_k$  of resource model  $\mathbf{R}_i$ , there is a corresponding transition  $\mathbf{t}_j$  in XML-net, which describes the transformation of the correlative resource states.

**Definition 10. XML document Mapping**  $\phi_P(\mathbf{R}, \mathbf{AS}) = \{((\mathbf{R}_i, \mathbf{r}_{p_i}, \mathbf{s}_j), \mathbf{p}_k) \mid \mathbf{r}_{p_i} \in \mathbf{PS}_i, \mathbf{s}_j \in \mathbf{AS}_i, \mathbf{p}_k \in \mathbf{P}\}$

For the properties and states of resource, a XML document in XML net is used to describe resource runtime state. Places in XML net act as container for XML document. Therefore we could build up the function  $\phi_P$  which maps from resource activity states with corresponding properties to places in XML net.

**Definition 11. F relation Mapping**  $\phi_F(\mathbf{R}, \mathbf{AS}, \mathbf{RS}, \mathbf{OT}) = \{((\mathbf{R}_i, \mathbf{s}_j, \mathbf{r}_k, \mathbf{o}_l), \mathbf{f}_m) \mid \mathbf{s}_j \in \mathbf{AS}_i, \mathbf{r}_k \in \mathbf{RS}_i, \mathbf{o}_l \in \mathbf{OT}_i\}$

The relation F between transitions and places could be also mapped from the states, operations and rules of resources.

Fig.3 illustrates the course of generation of XML schemas and XML documents from resource meta-model.

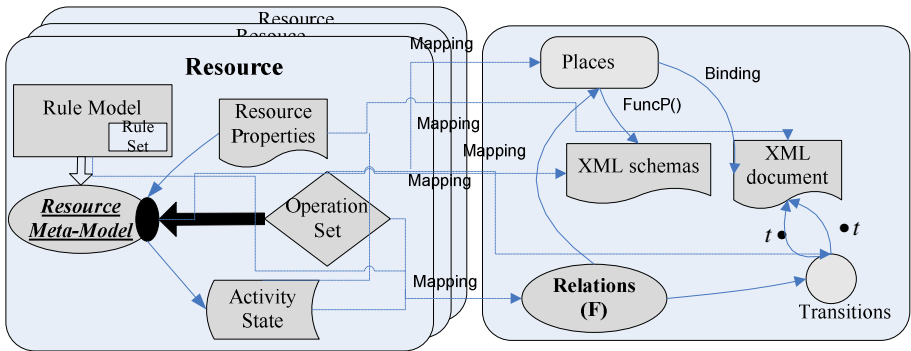


Fig. 3. Generation of XML-net Components for Process Model

### 3.3 RESTful Services Description and WADL Generation

As explained above, supply chain processes are modeled with XML-nets. The correlated resources and their states are described with XML documents. It implies that both data objects and control flow of processes are modeled in XML-nets completely. From such model, we could deduce RESTful services descriptions.

Currently there are two specifications used to describe REST services: WADL (Web Application Description Language) and WSDL 2.0 (Web Services Description Language) HTTP binding extension. WADL [10], proposed by Marc J. Hadley at Sun Microsystems, is a description language especially for HTTP-based web applications that follow REST architectural style.

The root of WADL is the application element, which consists of grammars and resources elements. The grammars element contains definition of the exchanged data format, e.g. in the language of XML schema or Relax NG. The resources element is a container for all resources and defines the base URI for all resources. Each resource is then represented as the resource element, which is identified with a path to the base URI. A resource element contains a list of HTTP method elements that are allowed to operate on this resource. Each method element contains a request element and a response element. The request element is composed by a few parameters. The response element contains the representation of the resource and a fault element if something goes wrong during the execution.

Fig.4. shows the mapping process from transition to method description in WADL. For simplicity we only focus on the main elements and ignore other less important elements in WADL such as doc, resource-type etc.

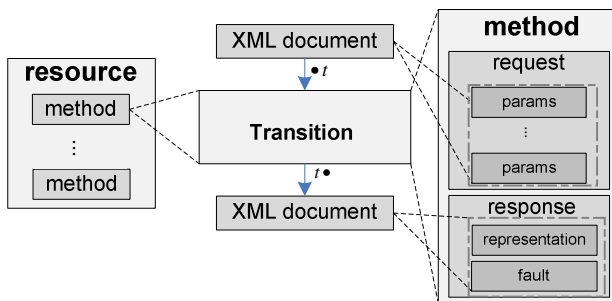


Fig. 4. Mapping WADL description

The definition of resource could be derived from XML schema. The methods are deduced from description of transitions. Correspondingly, the XML documents related to transition contain the parameters and return value of correlative deduced method.

With WADL, the stub and skeleton code for manipulation of resource can be easily generated to simplify the development of RESTful services.

BPEL is the standard composition language in the context of SOA. However it cannot be applied directly to REST. To composite RESTful Web services into processes, BPEL is extended for RESTful in [9]. And the new extension called BPEL for REST extends BPEL in the following two ways:

- Four activities <get>, <post>, <put> and <delete> are introduced to invoke RESTful web services, which corresponds the HTTP methods directly so that the client can use the familiar way to manipulate appropriate resources.
- The <resource> container element is introduced to declare resources involved in the process dynamically. The client can then access the published resources if they are available, that is, the process is being executed in their scope. Global resource is always available. Within the <resource> element, four elements, i.e. <onGet>, <onPost>, <onPut>, <onDelete>, are defined to declare the handlers that are triggered when the process receives the corresponding HTTP request.

With these two ways of extensions, BPEL is also appropriate to describe a business process implemented with RESTful web services. A. Koschmider proposed an approach to deduce BPEL components from XML nets [11]. Similarly, BPEL extension code could be also deduced from process model in XML nets. Especially, the <resources> container elements in BPEL extension are derived from XML schema. And the activities in BPEL extension are derived from transitions definitions in XML net.

## 4 Software Architecture and Case Study

### 4.1 Software Architecture

Based on resource model and process model in XML-net, Fig.5 illustrates the main framework supporting our idea, which involves five main modules: Heterogeneous Resources Adapter, Resource Model Management, Resource Object Management, XML-nets Processes Model Engine and RESTful Services Management.

The module Heterogeneous Resources Adapter supports heterogeneous systems to encapsulate operational resources, and then registers them to the Resources Registration Center on the integrated platform in order to make sure that the integrated platform can use the unified resource identifier to carry out operations on the heterogeneous resources.

The module Resource Model Management is responsible for the management of the resource model information from various heterogeneous information systems. The meta-information of the resource model and resource mapping strategies are stored in XML files and can be used by the resource object model. The module Resource Object Management manages the resource object information, which is obtained from different heterogeneous systems. Resources representation in XML documents, which is the result of the resources requests, is generated and sent to the XML-net model engine.

Based on resources model, the module XML-nets Process Model Engine is responsible for modeling processes. Thereafter, the RESTful services descriptions and orchestration descriptions are automatically generated according to process models.

The authority, transaction, discovery and matching of RESTful services are managed within this module.

### 4.2 A Case Study

In this section, a supply chain process with type “Make-to-Order” is proposed. This process could be seen as the states transformation of correlative resources. For simplicity, the payment processes and correlative resources are not considered in this

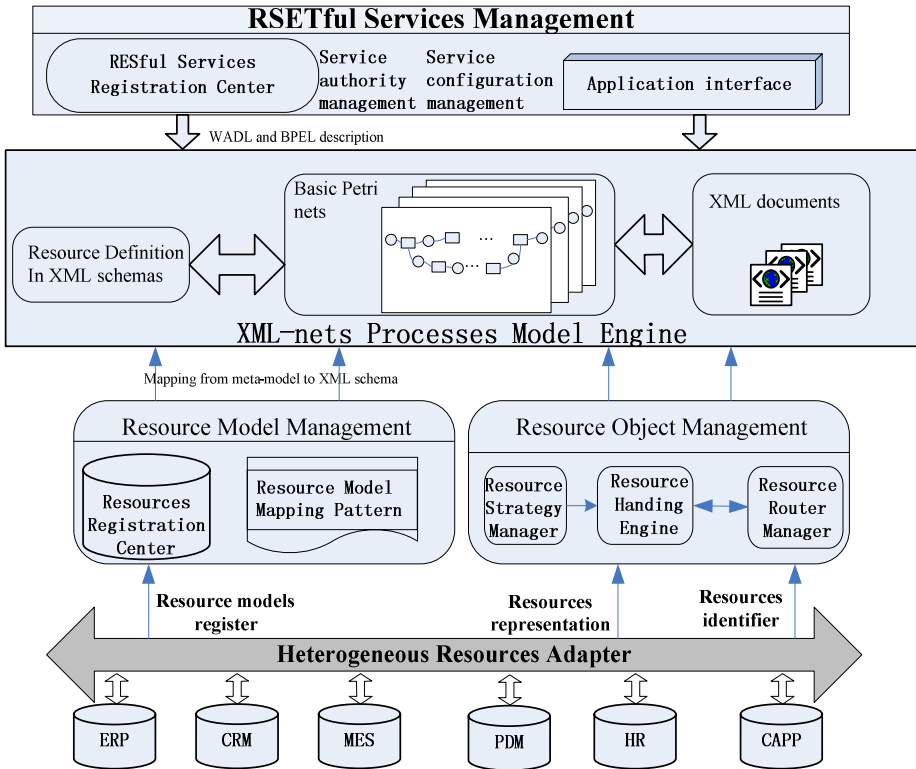


Fig. 5. The software architecture of the resource-based integration platform

paper. Several participants are involved in this scenario: at the beginning, a customer sends a purchase order request to manufacturer. The manufacturer receives the purchase order, checks inventory and organizes productions, including creating MRP (Material Request Planning). To prepare for production, the manufacturer authorizes a raw material supplier and sends raw materials order to the supplier. The material supplier deliveries raw materials to manufacturer and updates the state of his stock. The manufacturer checks the state of raw materials, when its state becomes to be “arrive”, the product will be manufactured and then be delivered to the customer by the deliver. The corresponsive invoice is also created and sent to the customer.

Based on the process description above, there are 7 resources involved in the process. The Resource set is:

$$ResS_t = \{Purchase\ Order, Inventory, Material\ Order, Stock, Raw\ Material, Product, Invoice\}$$

Each resource should be further defined based on resource meta-model using Definition 1, 2 and 3. For example, the meta-model of resource Purchase Order can be depicted as:

$R_{PURCHASEORDER} = \langle R_{URI}, OS_{PURCHASEORDER}, PS_{PURCHASEORDER}, AS_{PURCHASEORDER}, RS_{PURCHASEORDER} \rangle$

Which  $R_{URI}$  is the URI of resource Purchase Order,  $OS_{PURCHASEORDER}$  is the operations set, here  $OS_{PURCHASEORDER} = \langle OT_{CREATE}, OT_{UPDATE}, OT_{GET}, OT_{POST}, OT_{DELETE} \rangle$ . To adapt to RESTful services environment, the operations of a resource degenerate to simplified HTTP verbs.

$PS_{PURCHASEORDER}$  describe the properties set of Purchase Order. In this example, it includes basic information of customer, product and vender.

$AS_{PURCHASEORDER}$  defines activity states of Purchase Order, in this example,  $AS_{PURCHASEORDER} = \langle S_{START}, S_{CREATE}, S_{VERIFIED}, S_{FULFILED}, S_{END} \rangle$

$RS_{PURCHASEORDER}$  defines the rules set of resource states transferring. In this example  $RS_{PURCHASEORDER} = \langle \langle S_{START}, S_{CREATE} \rangle, \langle S_{CREATE}, S_{VERIFIED} \rangle, \langle S_{FULFILED}, S_{END} \rangle \rangle$

From meta-model, the corresponsive XML schema is derived. In XML-net, XML schemas are represented in a graphical XML schema definition language (GXSL) [5]. Details about GXSL and its utility in XML-net could be found in [5] [6]. For example, the schema of resource Purchase Order in GXSL could be depicted as Fig.6:

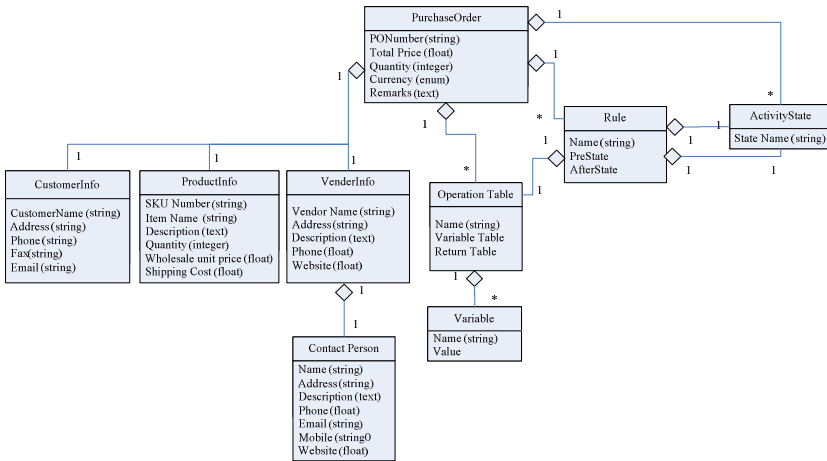


Fig. 6. The schema of resource Purchase Order in GXSL

XML documents in XML-net contain states of resources and are generated according to their corresponding XML schemas. Illustratively, a XML document describing verified Purchase Order is shown as Fig.7:

```

<?xml version="1.0" encoding="UTF-8" ?>
- <PurchaseOrder xmlns:xs="http://www.w3.org/2001/XMLSchema" xml:
  <PONumber>2009047254744711</PONumber>
  <TotalPrice>302,452</TotalPrice>
  <ActivityState>Verified</ActivityState>
  <Quantity>240</Quantity>
  <Currency>Euro</Currency>
  <Remarks />
- <CustomerInfo>
  <Customername>Autoankauf.Co</Customername>
  <Address>Feldberg str.22a Mannheim</Address>
  <Phone>+49 621 3547134</Phone>
  <Fax>+49 621 3547135</Fax>
  <EMail />
</CustomerInfo>
- <ProductInfo>
  <SKUNumber>223443562347634</SKUNumber>
  <Itemname>Zahnriehmen</Itemname>
  <Description>VW Zahnriehmen</Description>
  <Quantity>240</Quantity>
  <Wholesaleunitprice>1000</Wholesaleunitprice>
  <Shippingcost>62,452</Shippingcost>
</ProductInfo>
+ <VenderInfo>
- <OperationTable name="create">
  <Variable name="input" />
  <Variable name="input2" />
  <ReturnValue name="output" />
</OperationTable>
- <OperationTable name="get">
  <Variable name="input" />
  <Variable name="input2" />
  <ReturnValue name="output" />
</OperationTable>
+ <OperationTable name="post">
+ <OperationTable name="update">
+ <OperationTable name="delete">
+ <Rule>
</PurchaseOrder>

```

Fig. 7. XML documents describing verified Purchase Order

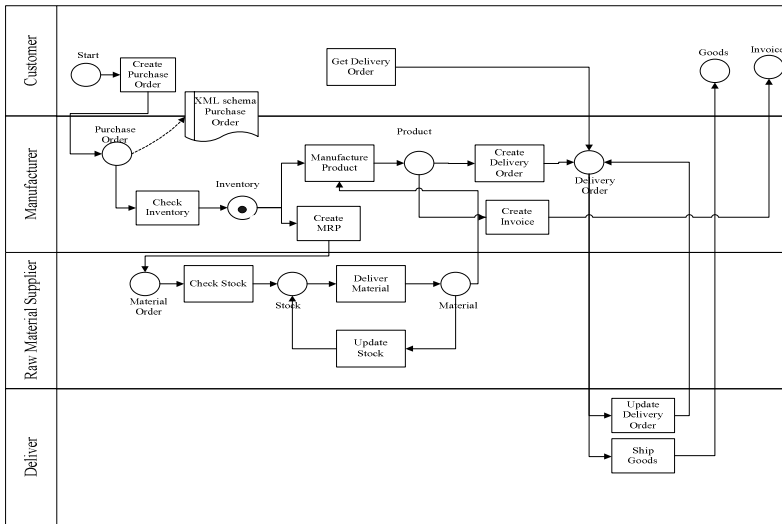


Fig. 8. Process model in XML-nets

The correlated places in XML-net is the set  $P_{\text{PURCHASEORDER}} = \langle P_{\text{PURCHASEORDER}} \rangle$ , and the correlated transitions in XML-net is the set  $T_{\text{PURCHASEORDER}} = \langle t_{\text{CREATEPURCHASEORDER}}, t_{\text{UPDATEPURCHASEORDER}} \rangle$

Analogically, the other places, transitions and correlative XML schemas for resources could be derived. Hence, the complete XML-net which describes this supply chain process is shown as Fig.8.

From process model in XML-net, RESTful service descriptions in WADL and choreograph description in BPEL extension are deduced. Fig.9 shows the fragment of service description related to resource Purchase Order.

```

- <application xmlns:xsi="http://wss.w3.org/2001/XMLSchema-instance" xsi:schemaLocati
- <resources base="http://example.org/customer">
  - <resource path="PurchaseOrder">
    - <methode name="CREATE">
      - <request>
        <param name="customerid" type="xsd:string" required="true" />
        <param mediaType="application/xml" element="order" />
      </request>
      - <response>
        <representation mediaType="application/xml" element="order" />
      </response>
    </methode>
    - <methode name="GET">
      - <request>
        <param name="porderid" type="xsd:string" required="true" />
      </request>
      + <response>
      </methode>
    - <methode name="UPDATE">
      + <request>
      + <response>
      </methode>
    + <methode name="DELETE">
    </resource>
  </resources>
</application>

```

Fig. 9. Service description in WADL

```

<?xml version="1.0" encoding="UTF-8" ?>
- <process name="replenishProccess">
  - <resource uri="PurchaseOrder">
    <variable name="inventory" />
    <variable name="invoice" />
    <variable name="products" />
    - <onPut>
      <post uri="inventory/materialInventory" response="inventory" />
      - <if>
        <condition>$inventory.amount</condition>
        - <scope name="requestRawmaterial">
          - <resource name="MaterialOrder">
            <POST uri="http://www.supplier.com/MaterialOrder" />
            <resource uri="Stock" />
            - <flow>
              <PUT uri="http://www.supplier/stock" response="stock" />
              <PUT uri="http://www.manufacture.com/rawmaterial" />
            </flow>
          </resource>
        </scope>
      </if>
      + <scope name="production">
      + <flow name="sendGoods">
      </onPut>
    </resource>
  </process>

```

Fig. 10. Fragment of choreograph description in BPEL

With the extension of BPEL, process implemented with RESTful web services could be also described appropriately. Fig.10 shows the fragment of extended BPEL code that composites RESTful services into process.

## 5 Discussion

Compared with other approaches, which integrate business processes in resource-oriented architectures, our approach has the following features.

Firstly, resource meta-model provides a unified information framework to describe resource. By introducing resource adapter, resources from heterogeneous environments can be represented and managed in a uniform formats, which is the basis for resources manipulation across different systems.

Secondly, XML-net is employed both for representing data flow and modeling control flow. XML schemas derived from resource meta-model provide a unified view for resources with diversiform sources. XML documents encapsulate resources states, thereby the messages are self-descriptive and using HTTP protocol could be a standard protocol for data exchanging.

Last but not least, RESTful service description in WADL and services orchestration description can be automatically deduced from process model in XML-net. Hence, the heterogeneous systems are encapsulated in RESTful service, which makes it possible to integrate process cross organizations and systems.

## 6 Related Work

A high-level Petri-net, so-called XML-net, is used for integration and deployment supply chain process in a RESTful service environment. The development of XML nets has been done in [5]. Here, XML documents carry information and are transferred between transitions. The work of employing XML nets for cross organizational workflow modeling could be found in [6]. Similarly, in [12], XML net is used for process management in supply chains, but it lacks of consideration on IT implementations.

XML nets could be also used for Web services description and composition. In [11], Koschmider developed a XML nets based approach for deduction of Web service description in WSDL and orchestration description in BPEL. Nevertheless, these works are based on traditional WfMS (Workflow Management System) and towards SOA architecture.

Recently, resource-oriented architecture and RESTful Web service have received increasing attention for business process modeling and integration. A few papers that use REST principles to support the business process modeling and execution have been published [14] [15] [16]. Business process is then viewed as the evolution of the states of a collection of business entities. However, the unified resource representation is still missing.

To adapt to RESTful services environment, a few specifications for services description and composition appear, e.g. WADL [10] and WSDL 2.0 can be used to describe web application and also to describe RESTful services. Details about the comparison between WADL and WSDL 2.0 HTTP binding extension can be found in [7].

In [9] BPEL is extended to support the composition of RESTful services natively. The extension is achieved in the following two aspects. Firstly, a process can be published as resources dynamically by adding a new <resource> element. Secondly, the RESTful services with BPEL can be invoked by adding four activities according to the standard HTTP methods.

However, in these works, there is still lack of unified representation and management of resource from various service providers. Service description and orchestration that could be generally accepted by supply chain participants are also in absence. Such issues prevent the communication and coordination among supply chain participants. Hence further across-organizational integration of supply chain process could not be effectively achieved. It is therefore necessary to describe resources in heterogeneous systems formally and encapsulate of services so that the across-organizational integration of supply chain process becomes possible.

## 7 Conclusion

In this paper we propose a XML-net based approach for enactment of supply chain process in RESTful service environment. A resource meta-model is constructed to describe resources from various sources. XML-net is employed for resource representation, process modeling and generation of RESTful services description. Our approach is highlighted by its integration of unified representation for resources from heterogeneous systems. And by employing XML nets, both process modeling and the generation of RESTful services description can be enhanced, which facilitates the enactment of cross-organizational collaborative business process.

## Acknowledgment

This paper is supported by the National High Technology Research and Development Program of China ("863" Program) under No.2008AA04Z126, the National Natural Science Foundation of China under Grant No.60603080, No.70871078, and the Aviation Science Fund of China under Grant No.2007ZG57012.

## References

1. Cherbakov, L., Galambos, G., Harishankar, R., Kalyana, S., Rackham, G.: Impact of service orientation at the business level. *IBM Systems Journal* 44(4), 653–668 (2005)
2. Fielding, R.T.: *Architecture Styles and the Design of Network-based Software Architectures*. Doctoral dissertation, University of California, Irvine (2000)
3. Kumaran, S., Liu, R., Dhoolia, P., Heath, T., Nandi, P., Pinel, F.: A RESTful Architecture for Service-Oriented Business Process Execution. In: *Proceedings of IEEE International Conference on e-Business Engineering 2008*, pp. 197–204 (2008)
4. Lenz, K., Mevius, M., Oberweis, A.: Process-oriented Business Performance Management with Petri Nets. In: *Proceedings of the 2nd IEEE Conference on e-Technology, e-Commerce and e-Services, Hong Kong*, pp. 89–92 (2005)

5. Lenz, K., Oberweis, A.: Modeling Interorganizational Workflows with XML Nets. In: Proceedings of the 34th Hawaii International Conference on System Sciences (2001)
6. Lenz, K., Oberweis, A.: Inter-organizational Business Process Management with XML Nets. In: Ehrig, H., Reisig, W., Rozenberg, G., Weber, H. (eds.) *Advances in Petri Nets*. LNCS, vol. 2472, pp. 243–263. Springer, Heidelberg (2003)
7. Takase, T., Makino, S., Kawanaka, S., Ueno, K., Ferris, C., Ryman, A.: *Definition Languages for RESTful Web Services: WADL vs. WSDL 2.0*. IBM Research (2008)
8. *Web Service Business Process Execution Language 2.0*, OASIS, <http://docs.oasis-open.org/wsbpel/2.0/OS/wsbpel-v2.0-OS.pdf>
9. Pautasso, C.: BPEL for REST. In: Dumas, M., Reichert, M., Shan, M.-C. (eds.) *BPM 2008*. LNCS, vol. 5240, pp. 278–293. Springer, Heidelberg (2008)
10. Hadley, M.J.: *Web Application Description Language (WADL)*. Sun Microsystems Inc. (2006), <https://wadl.dev.java.net/wadl20061109.pdf>
11. Koschmider, A., Mevius, M.: A Petri Net Based Approach for Process Model Driven Deduction of BPEL Code. In: Meersman, R., Tari, Z., Herrero, P. (eds.) *OTM-WS 2005*. LNCS, vol. 3762, pp. 495–505. Springer, Heidelberg (2005)
12. Mevius, M., Pibernik, P.: Process Management in Supply Chains – A New Petri-Net Based Approach. In: Proceedings of the 37th Hawaii International Conference on System Sciences (2004)
13. Richardson, L., Ruby, S.: *RESTful Web Services*. O’Reilly, Cambridge (2007)
14. Xu, X., Zhu, L., Liu, Y., Staples, M.: Resource-Oriented Architecture for Business Processes. In: *15th Asia-Pacific Software Engineering Conference 2008*, pp. 395–402 (2008)
15. Kumaran, S., Liu, R., Dhoolia, P., Heath, T., Nandi, P., Pinel, F.: A RESTful Architecture for Service-Oriented Business Process Execution. In: *Proceedings of IEEE International Conference on e-Business Engineering 2008*, pp. 197–204 (2008)
16. Muehlen, M., Nickerson, J.V., Swenson, K.D.: Developing web services choreography standards: the case of REST vs. SOAP. *Decision Support Systems* 40(1), 9–29 (2005)