Research on Semantic Integrity and Consistency of Process Relocation

Yuan Gang, Sun Ruizhi*, Wang Wendi, Dai Yizhou

College of Information and Electrical Engineering, China Agricultural University, Beijing, 100083, China

***Abstract* — In the agricultural products trading platform, a large number of transaction processes have similar structures and functions. With the development of information technology and the increasing demand of the market, the basic infrastructure and information system of agricultural product market will be continuously improved, upgraded and merged, and the existing business systems need to be migrated to a new platform. For the workflow systems' upgrading, as well as simulating and testing the existing processes in different environments, it needs to re-model a large number of workflow process definitions. To address the problem, this paper propose a method to export and import business processes based on process templates. In order to ensure the integrity of the exported process' structure and semantic, and the imported process' consistency with the current system's resource, a process template was proposed to describe the business process. By the comparison of systems' resources and executive roles which based on the constraint relationship with business process and resource conditions, this paper discuss several conversion rules to convert the processes and make it conform to the semantic of local template. This paper also propose a relational schema model for the new process files' storage, while identifying the particular entity of introduced processes based on their dependences and constraints, and checking the conflicts of the data between local and introduced processes. Ultimately it achieve the purpose of process definitions' relocation between isomorphic workflow platforms through a case study of agricultural product trade flow.**

Keywords — Workflow; Relocation; Process template; Transformation rule; Export and import

I. INTRODUCTION

At present, as the middleware of enterprise business management, workflow management system has been widely used in enterprise information systems. With the development of technology, enterprise' information system and infrastructure will be continuously improved and upgraded, so the existing business systems need to be migrated to a new platform. Even with the development of cloud computing technology, enterprise business systems will be migrated to the cloud platform. After years of process executions' accumulation, there will be a large number of operated processes in the original system whose process definitions and related supporting resources are perfect. To avoid redefining these business processes in new process management system, there needs an export and import tool that could efficiently achieve process definition's relocation.

Most workflow management systems can provide the tool for processes' export and import, but their functions are mainly for process' saving and backup, and that is the processes could be exported and imported in the same workflow management system. Considering the process definition's migration across different process management platforms, this article studied on the situation that a process definition was exported from the original system and then introduced into a new environment. Because of the difference of different workflow managements and the heterogeneity of process models, the original process could not be directly used after importing into a new system, it should to be mapped, converted and semantic checked according to the new platform's process template before put into use.

II. RELATED WORK

There are two questions should be solved when a process definition is imported into a new workflow platform: 1) the introducing processes should match the restriction of new process template's structure; 2) the semantic of original process definitions' resources and data should match the constraints of new platform's business rules and resource conditions.

Currently, the main method of different process models' conversion is about mapping a process' key elements, and then it will be deployed on the new workflow platform before the business process structures and semantics' matching, qualifying and transforming [1,2,3]. Zha [4] checked the processes' control logic under the condition of converting itself to Petri net, but it still should consider that whether the relations of original process' control logic, resources and data dependencies would meet the constraints of new business' rules and resource conditions, and whether the loss of semantic components during the conversion would affect the processes' normal operations. So when a process would be transformed and introduced into a new system, we should mainly think about the logical relationship between activities and resource constraints of activities' invoking and implementation.

Workflow semantic includes control flow, data flow, resources, and the semantic of their combinations[5].

DOI 10.5013/IJSSST.a.17.27.05 ISSN: 1473-804x online, 1473-8031 print

Workflow model is actually a description of a series of business rules which represent process logic, it describes all the possible circumstances of business' scenarios by formalizing business rules' semantic into constraint sets which are used to verify the process' semantic[5,6]. In addition to the events and activities' response rules, properties of original process activities may also be inconsistent with new system's business rules and resource constraints, such as whether the forms or Web services that invoked by activities are still exist, restricted to use and so on.

The logical relationship between activities and process statuses' change are almost reflected by data's transmission. Harry[7] proposed some constraint rules and focused on the process exceptions which may be brought by three kinds of operations, such as addition, modification and deleting. But he only considered the data which passed through process nodes, the random data generated during process modeling and the change of process nodes' intrinsic attributes that may bring exceptions were not mentioned. Reference[8,9,10] described the processes' information through process template and data model's schema, they defined and realized the mapping, matching and data conversion between the models, but there were lack of data semantic validation after conversion. Liu[11] studied on the data dependency, definition conflicts and constraint conflicts in business process, and proposed a method of data model exceptions' detection, but the impact of business resources and system environment's changes on the data model was not considered.

Based on the above-mentioned research of process' conversion, semantic checking and process data management, by analyzing the processes' relocation between different workflow platforms which locate in the same business domain and from the perspective of process files' export and import, this paper put forward process partial semantics' checking and restriction which based on process template, identification and modification of particular entities in process files according to the relevance between business processes and resources, and the conflict verification between local and introduced processes, and finally it realized the effective relocation of process definitions.

III. RESEARCH ON PROCESS DEFINITION'S RELOCATION

The workflow systems may have various workflow languages, resources, designs of database tables and so on, which may lead to overlapping dislocation of the flow chart after importing a process, and bring some errors of processes' saving and operation when the business semantics do not meet current workflow system's resource constraints. For example, the forms' link which bounded to business activities have changed, the system environment can't provide the equipments and services that ever required for the original process definitions, and the business organization has changed and so on. Therefore, when we want to relocate some process definitions which could achieve a certain business goal, the processes' original structure and semantic information should be preserved as much as possible when we export them from source system, as well as it should be verified whether the processes' semantic fit the new workflow platform's resource constraints when we import them into a new workflow system. In the following sections, parts of the process semantics' conversion and checking, and process' storage will be discussed.

A. Process Description Based on Process Template

Most workflow systems use XML file as a carrier of information exchange across platforms. As XML Schema can define XML document's structure, and it also can support multiple data types, describe data's complex relationships, expand function and so on. So this paper took XPDL, BPMN and BPEL these three common process modeling languages which are in accordance with the specification of XML Schema.

Business process is the combination of a series of business activities which would achieve a certain goal. In general, the templates which based on the type of process flow chart style (such as XPDL and BPMN) are mainly composed of root node, active node, transfer line and other process elements. The BPEL process template which based on block structure is mainly composed of root node, active node, and other process elements. Each template only has one root node, which represents a process' namespace. Process' namespace is the standard specification that the workflow process should follow by. Other process elements express some specific elements of some workflow, such as Pool in BPMN.

Definition 1(Process Mapping): $P \xrightarrow{TRC} Q$, represents the conversion from process P to process Q through TRC mapping rules, Where TRC = {TR1, ..., TRn, CR1, ..., CRn} is a set of conversion and restrictive rules. It is the association mapping of each element in the process which before and after the introduction, as well as a description of the process' primary and foreign key constraints.

There exist three relationships between the old and new workflow platform's process template, we assumed that SchemQ and SchemP represent the destination and source workflow platform's process template:

(1)Equal：SchemQ=SchemP。

(2)Compatible：SchemaP is compatible with SchemQ, and SchemaP is contained in SchemQ, that is they have the same nodes, but the contents and items of these nodes' attributes have been expanded or added in SchemQ.

(3)Different：SchemQ and SchemP have different structure. This situation is complicate, and it needs to be mapped and converted between different workflow languages and models.

Among the above, (1) and (2) belong to the same kind of workflow process template, that is SchemQ and SchemP's root node are identical. And (3) belongs to different kind of workflow process template, that is SchemQ and SchemP have different root node. Due to the different languages of different workflow process modeling tools which support different activities' icons and meanings, not all elements of the process and process models could be mapped and converted entirely, there must bring loss of information[13]. Therefore, in order to keep the processes' original structure and semantic as much as possible, it should to map and convert the process' business activity and control flow, including activity node, routing node and the transfer line which used to connect the nodes.

Definition 2(Process Template Tree):For a process, its template which based on XML Schema is defined as a tree called PST = <Root, PN, AN, RN>. Among them, Root is the unique one root of PST, and Root = <Name, NSpace> represents the process name, logo and name space; PN is the limit to all of the process nodes' types and levels, such as the process activities and transfer lines; AN is the collection of each PN's element attributes, including the name, type, icon coordinates, value range, number of occurrences, the relationship between front and rear logical conditions' restriction and so on; RN is the set of condition rules, that means it is the enumeration and restriction of a system's business resource series, logical conditional expressions and the actors.

Definition 3(Business Resource):Business resource RS=<SL, SP>, refers to the information resources and organizations which need to be relied or used in the processes' execution, including person, procedures, equipments, services and so on. SL is the resource items for workflow system; SP represents the attributes and status of resources, including the static properties (such as products' production date) and the dynamic properties (such as the number of products) which would change after the business activities' executions.

Definition 4(Control Flow)**:**Control flow CF=<NNode, NType, PreNode, PostNode, RS, LCE, Args>, which determines the process activities' execution sequence and direction. Among them, NNode=<BA, route, TransLine>, if it is the transfer line(TransLine), both PreNode and PostNode represent the pre and post process node, and NType is the transfer line's type, LCE represents transfer conditions. If NNode is a business activity(BA) or routing activity(route), NType indicates business activity's type and branch type of structure activity respectively, both PreNode and PostNode are its pre and post process node, and LCE represents global variables, environmental variables and expressions. Args is the data which are bound in activities; RS is resource set of business activities, including equipments, services, forms and so on.

Definition 5(Constraint Condition)**:**RN=<RS and LCEs, OPS, type, PreNode, PostNode, ORS> is constraint conditions, it defines the constraint conditions to the variable factors which would influence process' execution according to different business needs and the fields' characteristics; LCES is the logical conditional expression sequence set; OPS is the role set of activities; Type represents the activities' dependency, exclusiveness, nonexistence and so on; PreNode represents the last activity of current activity; PostNode represents the next activity of current activity; ORS represents the collection of resources' operations.

B. Process Quotation and Conversion for Semantic Consistency

To import a process P, firstly, when its namespace is consistent with the namespace of current workflow platform's process template, we can know that the current process platform could support process P, and it belongs to the situation that the old and new template are equal or compatible. So it only need to traverse the process P's each node, and then initialize and modify them according to SchemQ; If they are not match, it represents that process P and the platform are heterogeneous, so process P needs to be mapped and transformed. Secondly, we obtain the information of node elements which represent activities, tasks and events. And then we judge that the current activity is start, end, business or routing activity according to the number of nodes which locate before or after the current activity node. At last, it will be judged manually whether other process elements can be mapped, if not, it will be discarded. The conversion algorithm is shown in Algorithm 1 as follow:

Algorithm 1：Process conversion based on process template — ProcessSwitch(Xml_file, schema_file)

```
ProcessSwitch(Xml_file, schema_file)

Input：Xml_file represents the process' XML file for
       conversion; schema_file is the workflow
       system's template file
Output：Process' XML file
Function：For converting the process file according to
          the destination workflow system's process
          template

Step1：Parse the process P's XML file, obtain its root
       node's information, and compare it with
       SchemaQ's root node, if they are different,
       then jump to Step3.
Step2：Traverse other unhandled nodes of process P,
       obtain their attributes and then construct them
       to the object which according to SchemaQ.
       Add the corresponding node attributes' field in
       SchemaQ, and assign them to the default or
       null value. If there do not have any unhandled
       nodes, and then jump to Step10.
Step3：P.Root←SchemaQ.Root. Parse node elements'
       information of the activities, tasks and events
       which are represented in process P, and obtain
```

the front and rear active nodes of current active node. According to the active node without front or rear active node, we can obtain the "start activity" and "end activity", and then initialize its coordinates. Next judge SchemaQ.Root, if it belongs to XPDL or BPMN, then jump to Step7.

Step4: Parse the next activity node of "start activity" and obtain the number of its pre and post activity node. If the number of the front or rear node is greater than 1, the active node is a routing activity; And if the number is equal to 1, the active node is a general business activity.

Step5: Obtain other unhandled activity nodes and parse each one. If it is a routing activity, it will correspond to BPEL's structural activity, and the routing activity's post activities are nested in structure activity block. If it is a sub-process activity, then jump to Step5.

Step6: Obtain other unhandled activity nodes and judge whether it could be mapped to BPEL. If can then obtain its attribute values and build into an object which according to SchmeaQ. If can't then jump to Step10.

Step7: Obtain the information of the process' common activities and structure routing activities.

Step8: Convert the routing activities to the gateways or route activities which described in XPDL or BPEL, and add the transfer lines between the activities. If it is a sub-process activity, then jump to Step7.

Step9: Obtain other unhandled activity nodes, judge whether it can be mapped to XPDL or BPMN. If can then obtain its attribute values and build the object that is described in SchemaQ.

Step10: End.

After the conversion, the process could meet the requirement of new platform in structure. To ensure the imported process' normal operation, it is important to consider the original process' related semantics, such as control, data, and resources which should still be met with the new system's environment.

(1) control flow semantics' checking: 1) maintenance of the original business activities' behavior and attributes after transform; 2) checking the extended elements' behavioral semantics; 3) conditions and rules before and after the activities' operations; 4) the change of branch types and control variables which caused by the modification of business' rules is not under consideration, we only consider the branch control conditions' alteration.

(2) activity attribute's checking: 1) modification of process definition activities' attributes, such as the change of participant property; 2) the resource environmental semantics' checking.

(3) process data's checking: In fact, during the process' importing and exporting, it always works on the data which has been defined in Schema and related to the process(data attributes and some random data, such as ID, etc.). It needs to check and modify the workflow control flow and related data (such as the semantic of a branch's control flow, etc.), process activities' attributes (type of activities, participants, and access to resources, etc.) and real-time generated data.

In order to achieve the dynamic assignment and authorization of the workflow, and reduce the occurrence of conflicts, Ma[12] did some full analysis on constraint relationships of user, role and authority, but the method was complicated. In order to facilitate the understanding and implementation, this paper combined roles and permissions, and made the executable permissions attach to the role. The event producers are identified by the role which own some executable permissions while modeling. In this section, the role is seen as an abstract unit that according to the participants and power groups' abstraction, which have some certain tasks, authorities and parameters, and also can take the initiative to complete a series of related operations. According to the Definition 1 to 5, the following conversion rules are defined.

Rule TR1: SW and DW have exactly the same resource and activity executive authority, that is $\forall$SW(RS,OPS)= DW(RS,OPS) $\Rightarrow$ $SMP \xrightarrow{PST(Root,PN,AN,RN)} DMP$ 。

Where:

1) DW and SW represent the source and destination workflow system;

2) SMP is exported from SW and needs to be imported into DW. DMP is the process that imported into DW after its conversion based on the process template.

3） PST(Root,PN,AN,RN) and"→"indicate that the corresponding elements of the SMP file would be modified according to the process template PST of DW. The definition of Root, PN, AN and RN are defined in Definition2 in this paper, which mean the root node, process node, node attribute and conditional rule set of process template.

Rule TR2: SW and DW have the same resources, that is $\forall$SW(RS)=DW(RS), and their execution permissions are not identical.

Rule TR2-1: SW and DW's executive authority repel each other, that is SW(OPS)$\wedge$DW(OPS)=$\varnothing$ $\Rightarrow$

$$SMP \xrightarrow{PST(Root,PN,RN)\wedge(FlowNode.OPS=Null|AN)} DMP$$

Rule TR2-2: SW and DW's executive authority are inclusive to each other, that is $\exists$SW(OPS)$\subseteq$DW(OPS) $\Rightarrow$

$$SMP \xrightarrow{PST(Root,PN,AN,RN)} DMP$$

Rule TR2-3: SW and DW's executive authority are inclusive to each other, that is $\exists$SW(OPS)$\supseteq$DW(OPS) $\Rightarrow$

$$SMP \xrightarrow{PST(Root,PN,RN)\wedge(FlowNode.OPS\in DW.OPS|AN)} DMP$$

Rule TR2-4: SW and DW's executive authority are overlapped, that is $\exists$SW(OPS)$\wedge$DW(OPS)≠□ AND

SW(OPS)□DW(OPS) $\Rightarrow SMP \xrightarrow{PST(Root,PN,RN)\wedge(FlowNode.OPS\in SW.OPS\wedge DW.OPS|AN)} DMP$

Rule TR3：SW (DW)'s resources are contained in DW (SW), that is ∀SW(RS) ⊆ DW(RS)（or SW(RS) □ DW(RS)）, and their executive permissions are not identical.

Rule TR3-1：SW and DW's executive authority are equal, that is □SW(OPS)=DW(OPS) $\Rightarrow SMP \xrightarrow{PST(Root,PN,AN,RN)} DMP$ (or $SMP \xrightarrow{PST(Root,PN,RN)\wedge(FlowNode.RS\in DW.RS|AN)} DMP$)

Rule TR3-2：SW and DW's executive authority repel each other, that is ∃ SW(OPS)∧DW(OPS)=∅ ⇒ $SMP \xrightarrow{PST(Root,PN,RN)\wedge(FlowNode.OPS=Null|AN)} DMP$ (or $SMP \xrightarrow{PST(Root,PN,RN)\wedge(FlowNode.OPS=Null|AN)\wedge(FlowNode.RS\in DW.RS|AN)} DMP$)

Rule TR3-3：SW and DW's executive authority are inclusive to each other, that is ∃ SW(OPS) ⊆ DW(OPS) ⇒ $SMP \xrightarrow{PST(Root,PN,AN,RN)} DMP$ (or $SMP \xrightarrow{PST(Root,PN,RN)\wedge(FlowNode.RS\in DW.RS|AN)} DMP$)

Rule TR3-4：SW and DW's executive authority are inclusive to each other, that is ∃ SW(OPS) ⊇ DW(OPS) ⇒ $SMP \xrightarrow{PST(Root,PN,RN)\wedge(FlowNode.OPS\in DW.OPS|AN)} DMP$ (or $SMP \xrightarrow{PST(Root,PN,RN)\wedge(FlowNode.RS\in DW.RS|AN)\wedge(FlowNode.OPS\in DW.OPS|AN)} DMP$)

Rule TR3-5：SW and DW's executive authority are overlapped, that is □ SW(OPS)∧DW(OPS)≠□ AND SW(OPS)□ DW(OPS) $\Rightarrow SMP \xrightarrow{PST(Root,PN,RN)\wedge(FlowNode.OPS\in SW.OPS\wedge DW.OPS|AN)} DMP$ (or $SMP \xrightarrow{PST(Root,PN,RN)\wedge(FlowNode.RS\in DW.RS|AN)\wedge(FlowNode.OPS\in SW.OPS\wedge DW.OPS|AN)} DMP$)

Rule TR4：The resources of SW and DW are not identical, that is ∀ SW(RS)∧DW(RS)≠□ AND SW(RS)□DW(RS), and their executive permissions are not identical

Rule TR4-1：SW and DW's executive authority are equal, that is □SW(OPS)=DW(OPS) $\Rightarrow SMP \xrightarrow{PST(Root,PN,RN)\wedge(FlowNode.RS\in SW.RS\wedge DW.SI|AN)} DMP$

Rule TR4-2：SW and DW's executive authority repel each other, that is ∃ SW(OPS)∧DW(OPS)=∅ ⇒ $SMP \xrightarrow{PST(Root,PN,RN)\wedge(FlowNode.RS\in SW.SI\wedge DW.RS|AN)\wedge(FlowNode.OPS=Null|AN)} DMP$

Rule TR4-3：SW and DW's executive authority are inclusive to each other, that is ∃ SW(OPS) ⊆ DW(OPS) ⇒ $SMP \xrightarrow{PST(Root,PN,RN)\wedge(FlowNode.RS\in SW.RS\wedge DW.RS|AN)} DMP$

Rule TR4-4：SW and DW's executive authority are inclusive to each other, that is ∃ SW(OPS) ⊇ DW(OPS) ⇒ $SMP \xrightarrow{PST(Root,PN,RN)\wedge(FlowNode.RS\in SW.RS\wedge DW.RS|AN)\wedge(FlowNode.OPS\in DW.OPS|AN)} DMP$

Rule TR4-5：SW and DW's executive authority are overlapped, that is ∃ SW(OPS)∧DW(OPS)≠□ AND SW(OPS)□ DW(OPS) ⇒ $SMP \xrightarrow{PST(Root,PN,RN)\wedge(FlowNode.RS\in SW.RS\wedge DW.RS|AN)\wedge(FlowNode.OPS\in SW.OPS\wedge DW.OPS|AN)} DMP$

C. Process Storage Management Based on Relational Schema

The export of local processes and the import of external processes should be mapped with the local relational database based the process template, and then the database access is realized. In general, the business resources will affect the process executions' variable factors, and the method of using relational database to store enterprise resource model could control the resource models' association by the database tables' foreign key. Assume that the process template has already been mapped the local database, the relation pattern is defined as follows in order to ensure the integrity of process data's access.

Definition 6(Relation Pattern):RP represents the relation patterns, and it describes the relationship between database and process' elements. RP=<TN, CN, PK, FK>, among them, TN is the set of existing data tables' names. CN is the set of columns name whose table's name belong to TN; TD is the type definitions of CN, such as data types, is unique, is null, restricted ranges and default values and so on. PK is the columns name which correspond to primary key; FK is the columns name which correspond to foreign key.

The XML data file does not have primary key, it needs to be added unique identification to denote some entity object's uniqueness. So it will bring some temporary, independent and random data when the process is relocated. When we would save the process to current workflow system after the semantic conversion, the identification data for these conflicts should be checked, and we could use the association rules to manage the imported processes' storage.

(1) The process is considered as an independent process definition which would be saved to local database, and then a record is added to the original process. At first, we will extract process files' information which is related to the database tables. Secondly we will check the original primary keys and foreign keys' conflicts, and then modify the related information when the records are added.

Rule CR1：

$$DMP \xrightarrow[if(T(i)(PK_m\vee FK_n)=SMP.AN_j)thenMODIFY(SMP.AN_j)]{INSERT} t(i)$$

(2) The imported flow is used to replace the existing process definition. The original process primary and foreign keys' values will be reserved, and the rest part of it will be replaced by the introduced process information.

Rule CR2：

$$DMP \xrightarrow[UPDARTEofT(i)withSMP.AN_jExcept(PK_m,FK_n)]{UPDATE} t(i)$$

When a new process definition is introduced, it involves process data's acquisition, merger, format conversion and so on, and we also need to consider whether the introduced process has conflict or correlation with original processes. The method of semantic checking and the introduced process' import and storage is shown in Algorithm 2 as follow:

Algorithm 2：Semantic checking and importing based on constraint rules — Check4Store（Xml_file, rule_file）

***Check4Store*（*Xml_file, rule_file*）**

Input：Xml_file represents the process' XML file for conversion; rule_file is the defined constraint rule file

Output：Process' XML file

Function：Check the process file based on the defined conversion rules, the eligible process would be stored in database

Step1：Change the process' namespace according to the template.

Step2：Parse process P's XML file to get the "start activity", and then initialize its coordinates.

Step3：Traverse other unhandled nodes(NNode) of process P, and construct them to the object according to SchemaQ. If there has no unhandled node, then jump to Step10.

Step4：Parse the nodes' types and obtain their attribute values. Add the corresponding nodes attributes' field in SchemaQ, and then assign them to the default or null value.

Step5：View the process P's resource items(RS) and business activities executive roles(OPS), and then compare the resource rules sets(RN) of source workflow process template tree(SW.PST) with destination workflow process template tree(DW.PST).

Step6：According to the relationship between OPS and RS in SW.PST and DW.PST, the conversion of rule TR1~TR4 is carried out.

Step7：Return the next node, if it is unhandled then jump to Step4.

Step8：Extract the information of process Q which is related to database table TN(i).

Step9：To meet the INSERT or UPDATE need of process Q in database, it will do some operation according to CR1 and CR2 rule, and check the RP.TD's consistency when records are added. If there has a demand, it can extract the relevant fields from database tables, and then re-assemble the generated process XML file based on PST.

Step10：End.

In order to realize the unified management of the imported processes in local workflow system, firstly, the process files are parsed and traversed. In order to ensure the accordance with the local process template's semantic restrictions, the processes would be converted according to the template resource rule sets. Secondly, we should identify the specific entities, extract the effective attribute data and check the conflicts when we store the processes in database. We also should do some pretreatment and correction for process data and database tables' primary and foreign keys' values. Finally, the process files which located in database could be assembled and generated based on process template. The process' importing and transformation block diagram is shown in Fig.1.

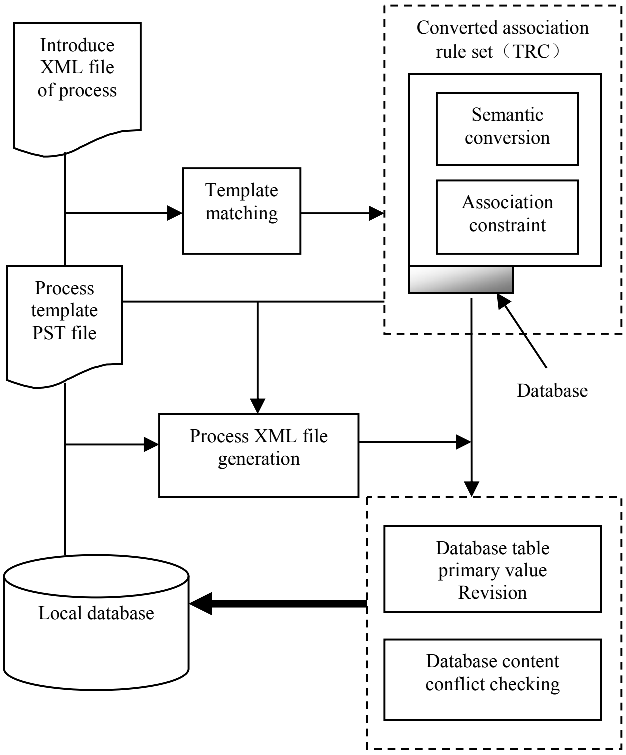
Introduce XML file of process
Template matching
Converted association rule set（TRC）
Semantic conversion
Association constraint
Process template PST file
Database
Process XML file generation
Local database
Database table primary value Revision
Database content conflict checking

Fig.1 Process Import and Conversion Block Diagram

IV. CASE STUDY

Assuming in the local system, there has a "booking flow" process which is described by workflow language XPDL as shown in Fig.2, it has an activity which called "payment" is a sub process. Fig.3 is an "order payment flow" process which locates in other departmental workflow system. And now, there has another process which locates in local system and needs to use the "order payment flow", so we can move Fig.3's process to local system. In order to maintain the integrity and consistency of the original process' structure and semantic, and ensure the introduced processes' proper operation, the process semantic should be converted based on process template and constraint rules, and at last it will replace the Fig.4's sub-process.

Because these two workflow systems' process templates have the same structure, it only needs to traverse every node of the process, and then the corresponding nodes' attributes would be initialized or modified. After the method of process' import and export as Fig.5 shows, we can export all the details of process activities (including definitions of process and relevant details) to an XML file, and save the XML file's contents into database without manual modeling when a process is imported.

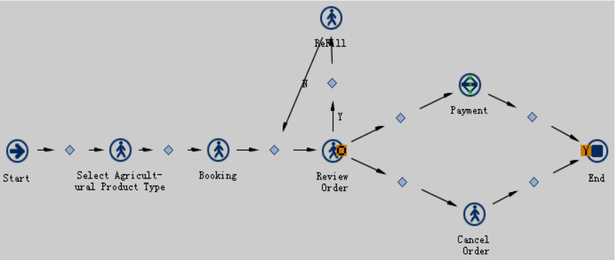

Fig.2 Booking Flow Chart

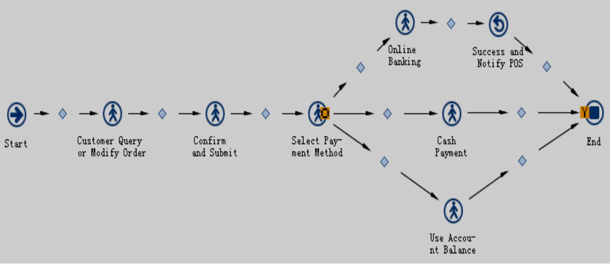

Fig.3 Order Payment Flow Chart

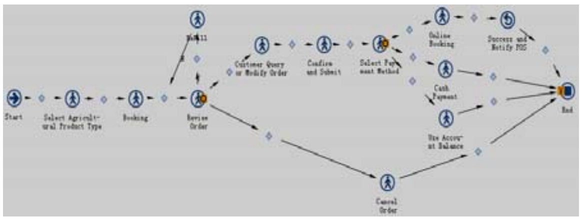

Fig.4 After Replacing the Booking Process

As shown in Fig.3, after its conversion into the local system, the "order payment flow" process' structure remains unchanged as shown in modeling tool. In order to ensure that the imported process could satisfy local system's environment and execute properly, we also need to convert its semantics according to the process template and transformation rules. Assuming that the source and destination workflow system have the same business resources, and there exist four kinds of role, which are "manager", "vvipUser", "vipUser" and "normalUser". In source workflow system, the executive participant roles are defined as "vvipUser", "vipUser" and "normalUser". But in the destination workflow system, the executive participant roles are defined as "vvipUser" and "vipUser". That is, SW and DW's executive authority are inclusive to each other. So, according to the transformational rule TR2-3, after the introduction of the "order payment" process, the activities of the executive role are changed to "vvipUser" and "vipUser" as shown in Fig.6.

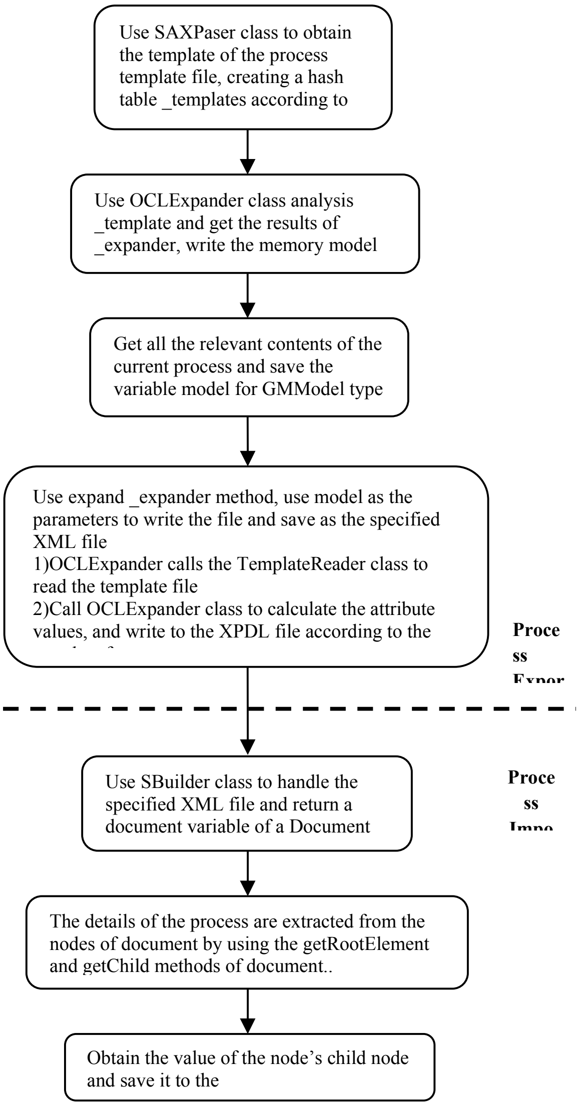
Use SAXPaser class to obtain the template of the process template file, creating a hash table _templates according to
Use OCLExpander class analysis _template and get the results of _expander, write the memory model
Get all the relevant contents of the current process and save the variable model for GMModel type
Use expand _expander method, use model as the parameters to write the file and save as the specified XML file
1)OCLExpander calls the TemplateReader class to read the template file
2)Call OCLExpander class to calculate the attribute values, and write to the XPDL file according to the
Proce ss Expor
Use SBuilder class to handle the specified XML file and return a document variable of a Document
Proce ss Impo
The details of the process are extracted from the nodes of document by using the getRootElement and getChild methods of document..
Obtain the value of the node's child node and save it to the

Fig.5 Method of External Processes' Import

```
<xs:element name="OperationSets" type=" ParticipantType "/>
<xs:simpleType name=" ParticipantType ">
  <xs:restriction base="xs:string">
            <xs:enumeration value="vvipUser"/>
    <xs:enumeration value="vipUser"/>
    <xs:enumeration value="normalUser"/>
  </xs:restriction>
</xs:simpleType>
```

```
<xs:element name="OperationSets" type=" ParticipantType "/>
<xs:simpleType name=" ParticipantType ">
  <xs:restriction base="xs:string">
       <xs:enumeration value="vvipUser"/>
    <xs:enumeration value="vipUser"/>
  </xs:restriction>
</xs:simpleType>
```

In order to ensure the processes' integrity, consistency and validity when relocate them between homogeneous workflow platforms, this paper researched on the different resources which owned by different workflow systems and put forward a method of imported process' semantic conversion based on process template. We also identified the specific entity and extracted the defective data from imported process by relational model, and then checked the conflicts with local system's process data. Finally it realized the multi processes' organic integration and unified management in local workflow system.

CONFLICT OF INTEREST

The authors confirm that this article content has no conflicts of interest.

ACKNOWLEDGMENT

This work is supported by the National Science and Technology Support Program (2012BAK17B09, 2015BAK04B01).

REFERENCES

[1] L. Wu and J. Ma, "Developing research of workflow management system based on model-driven architecture," Computer Technology and Development, vol. 21, pp. 89-94, June 2011.

[2] X. Guo, Z. Lin and W. Yao, "Workflow technology based on rule base and message driven mechanism," Journal of Computer Applications, vol. 34(S1), pp.207-272,276, June 2014.

[3] H.Y. Chen, Z. Liu and J.X. Liu, "A xml-based workflow model and its mapping algorithm for exchange," Computer Engineering&Science, vol. 32, pp. 147-150, July 2010.

[4] H.P. Zha, W.M.P. Aalst, J.M. Wang, L. Wen and J. Sun, "Verifying workflow processes: a transformation-based approach," Software and Systems Modeling, vol. 10, pp. 253-264, 2011.

[5] H.B. Li, D.C. Zhan and X.F. Xu, "Integration verification of workflow business rule semantic," Journal of Computer Research and Development, vol. 46, pp. 1143-1151, July 2009.

[6] W. Roover and J. Vanthienen, "A transformation from SBVR business rules into event coordinated rules by means of SBVR patterns," Lecture Notes in Computer Science, vol. 6569, pp. 172-179, 2011.

[7] H.J. Wang and J.L. Zhao, "Constraint-centric workflow change analytics," Decision Support Systems, vol. 51, pp. 562-575, March 2011.

[8] H.W. He, Y.Z. Zheng and Y.H. Yang, "Research and design of heterogeneous data exchange system in E-Government based on XML," Advances in Intelligent and Soft Computing, vol. 169, pp. 31-36, 2012.

[9] Y.S. Chae, S.S. Hyun and R.Y.C. Kim, "Model transformation rule for generating automatic database schema of business process framework," International Journal of Software Engineering and Its Applications, vol. 8, pp. 47-54, March 2014.

[10] H.S. Xu and R.L. Zhang, "Research on data integration of the semantic Web based on ontology learning technology," Journal of Electrical Engineering, vol. 12, pp. 167-178, January 2014.

[11] Z.Q. Liu，H.Y. Li，L. Wang and Q. Qu, "A data model anomalies detection method for business process model," Chinese Journal of Computers, vol. 33, pp. 1349-1358, August 2010.

[12] C.H. Ma, J. Wang, J. Qiu and G.D. Lu, "Flexible context-constraint-based access control model for workflow," Journal of Zhejiang University(Engineering Science), vol. 44, pp. 2297-2308, December 2010.

[13] X.N. Feng, Q.X. Li and Z. Wang, "A mapping method from BPMN models to BPEL," Journal of Computer Research and Development, vol. 50(Suppl.), pp. 44-52, 2013.