# A Mapping Algorithm of Scientific Workflow from BPMN to BPEL

X. Tian, H.S. Li, Y.J. Huang, Q. Cai
School of Computer and Information Engineering
Beijing Technology and Business University
Beijing, China

***Abstract*—To get a general mapping algorithm for any BPD(Business Process Diagram),depth-first traversal algorithm based component is designed to deal with any BPMN(Business Process Modelling Notation) model which is composed of the tasks, events, parallel join gateway, parallel fork gateway, data-based XOR decision gateway and XOR merge gateway mapping BPEL. The generated BPEL code has good maintainability and extensibility. The algorithm is implemented in the SWF system developed by Python and its validity and usability has also been proved in practical application.**

***Keywords-SWF; mapping algorithm; BPMN; BPEL***

## I. INTRODUCTION

BPEL(Business Process Execution Language For Web Services) has become a standard language of the business process in the field of web service [1]. The Business Process Modeling Notation (BPMN) [2] is a graph-oriented language so that it is easy to understand the process. Mapping algorithm from BPMN to BPEL has become an important research topic of workflow modeling .

BPEL can support 20 kinds of control models of business workflow. In the field of scientific workflow [3], workflow language is mostly designed for a specific computing environment, all implementations of which lay emphasis up on mass calculation and the performance of the intensive use of parallel architecture. The paper,[4] summarized the data models and control models of SWF(Scientific Workflow) that BPEL supports to illustrates that how BPEL to adapt to the special geographic information process requirements of BPEL. Asif Akram [5] proved the possibility of BPEL becoming scientific workflow definition language by setting forth six demands.

The research,[6,7,8,9,10],introduce the four business workflow mapping algorithms.

The four kinds mapping algorithms above divide BPD into components and map BPD into BPEL through different mapping method. However, due to the randomness of these mapping methods, different readers can create different BPEL for the same BPD. So a mapping algorithm is proposed in this paper: Depth-first traversal algorithm based component. The mapping process is the only one for different readers for the same BPD.

By analyzing the feasibility of BPEL in SWF and the mapping algorithms of BWF, the paper presented:

Transform the traversal problem of BPMN model into a kind of important none-linear data structure: the search problem of graph structure. And apply the depth-first traversal algorithm to the algorithm that BPEL generated.

Depth-first traversal algorithm based on component makes the two main business standards that are BPMN and BPEL applied in SWF mapping algorithm. The algorithm has been implemented in SWF developed by Python and its validity and feasibility has been verified.

## II. RELATED WORK

The two main current process specification standards are BPMN [11] that is a graph-oriented language and BPEL [12] that is a block-structured language. The BPMN is easy to understand for process designer, but there is no system that can execute BPMN model directly. BPEL has become a standard language of the business process in the field of web service, and its advantage is suitable for parsing and executing the process engine. But considering the grammatical complexity, BPEL is not easy to understand and use for process designer. Because different modeling tools have different implementations, and there is no unified definition for graphical representation of each element of BPEL [13].

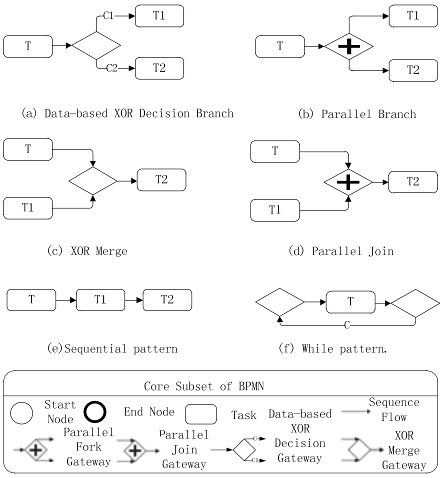
SWF originated in BWF, but SWF can't completely use technology and standard of BWF. The main reason is that BWF is control flow oriented and it focuses on the change and its dependency of program module in the transaction process[14], while the SWF is data-oriented, its focus is the change of each node in the process of scientific data and its influence to the scientific research [15]. The BPEL process is similar to a flow chart that expresses algorithm, and each step of the process is called an activity. The main activities of BPEL are introduced in [16].

BPMN uses BPD (Business Process Diagram) to describe business processes. A BPD is made up of elements of BPMN. The main process structures of BPD are respectively sequential pattern, exclusive pattern, simple parallel branch, synchronization pattern, merger pattern, circulation pattern which is shown in Figure 1.

## III. MAPPING ALGORITHM

### *A. Concept of Component*

Component is control mode. It is the component unit of BPMN process model, and it can be completed by different elements of BPMN. In the six kinds of SWF patterns shown in Figure 1, (a) and (d) constitute the switch-component,(b) and (c) constitute flow-component,(e) constitutes sequence-component,(f) constitutes while-component. Table 1 shows the four components and respectively corresponding BPEL code.



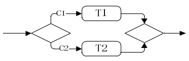
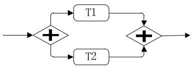
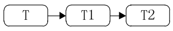
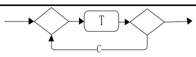
(a) Data-based XOR Decision Branch (b) Parallel Branch

(c) XOR Merge (d) Parallel Join

(e)Sequential pattern (f) While pattern.

FIGURE I. SIX PATTERNS OF BPD

TABLE I. FOUR STRUCTURED ELEMENT AND ITS BPEL CODE

| Type of Component | BPD of Component | BPEL Code |
|---|---|---|
| **Switch-component** |  | <switch><br><case condition=C1><br><invoke name=T1><br><case condition=C2><br><invoke name=T2><br></switch> |
| **Flow-component** |  | <flow><br><sequence><br><invoke name=T1><br></sequence><br><sequence><br><invoke name=T2><br></sequence><br><flow> |
| **Sequence-component** |  | <sequence><br><invoke name=T><br><invoke name=T1><br><invoke name=T2><br></sequence> |
| **While-component** |  | <while condition=C><br><invoke name=T><br></while> |

## *B. Depth-First Traversal Algorithm*

The main graph traversal algorithms are depth-first traversal [17] and breadth-first traversal [18].We used the depth-first traversal based on the structured activity to traverse the entire BPMN process. As shown in Figure 2, firstly, make v1 the start node to traverse. Secondly, using depth-first traversal traverse the nodes that are adjacent to start node but have not been accessed. The traversal order of nodes in Figure 2 are: v1、v2、v4、v8、v5、v3、v6、v7.

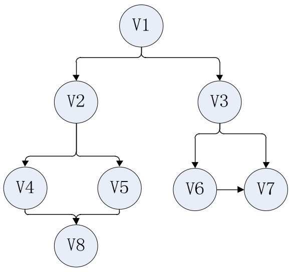


FIGURE II. DIRECTED GRAPH

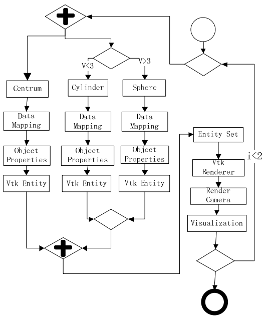


FIGURE III. PROCESS INSTANCE OF SWF

## *C. Principle of Mapping Algorithm*

This paper combined the principle of component and the depth-first traversal algorithm together as the transformation algorithm from BPMN to BPEL. This mapping algorithm is a kind of strategy to hierarchical decomposition of the BPMN process. The main idea of the algorithm is regarding the entire BPMN process as a sequential structure, whose process is {start event, the sub process, end event}.The first step of algorithm is traversing the process tree of BPMN using depth-first traversal algorithm. We will add the task node to the sequence-component if the subsequent node of parent node is a task node; and we will add the gateway node to the corresponding component if the subsequent node of parent node is a gateway node, and merge the corresponding gateway node as the corresponding component. Then traverse the internal nodes of component using depth-first traversal algorithm. When the merge node is traversed, the mapping process of the component is completed. Before recursion is completed, continue to traverse the subsequent nodes. The result of the mapping process is the BPEL code.

## *D. Description of Mapping Algorithm*

(1) Make sure that the process model of BPMN is a well-formed BPD. In other words, there are only one start node and end node in BPD, which can be composed by the four components as shown in table 1.

(2) Create main process of sequential structure and add the start node and end node up to the main process. Scan the start

node and create the corresponding BPEL tags that is <sequence> and its child node tag <start-event>.Then scan the successor nodes of start node.

(3) When the next node of process is empty, the next node is end node. Create the corresponding BPEL tag that is <end-event>. The end node means the end of the process. Add the tag of </sequence> and go to the step (5).

(4) If the next node is not empty, this algorithm can be divided into following several ways for different subsequent node type:

(4-a) if the subsequent node is parallel fork gateway, according to the parallel fork structure in upper segment, we can regard it and its corresponding parallel join gateway as component, and record the numbers of branch and node types. Create the tag of BPEL <flow>, we should add the tag <sequence> before traverse the branch if the number of branches is more than 1.Then we should scan the subsequent nodes and go to the step (3).

(4-b) if the subsequent node is data-based XOR decision gateway, according to the exclusive structure in upper segment, we can regard it and its corresponding XOR merge gateway as component, and record the number of branch and node types. Create the tag of BPEL <switch>. We should add the tag <sequence> if the number of branches is more than 1.Then we should scan the subsequent nodes and go to the step (3).

(4-c) if the subsequent node is parallel join gateway, create the tag of BPEL </flow>.Then we should scan the subsequent nodes and go to the step (3).

(4-d) if the subsequent node is XOR merge gateway, create the tag of BPEL </switch>.Then we should scan the subsequent nodes and go to the step (3).

(4-e) if the subsequent node is loop structure, we can regard it as a while-component. Create the tag of BPEL <while>, then we should scan the subsequent nodes and go to the step (3).

(4-f) if the subsequent node is task node, create the tag of BPEL <invoke>.Then we should scan the subsequent nodes and go to the step (3).

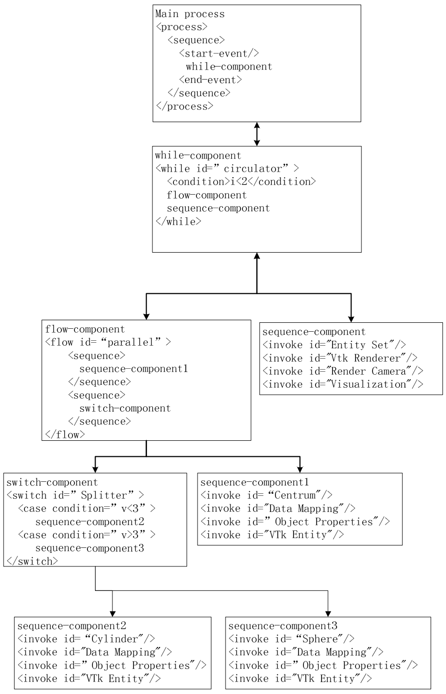
(5) When the traversal is completed, the result of the mapping algorithm is the code of BPEL.

## IV. Experiments

This mapping algorithm is illustrated by taking the example of the visualization of the SWF process about three-dimensional volume rendering. The process of SWF is shown in Figure 3.

As the mapping algorithm has been described, Figure 4 is the process of mapping algorithm of the SWF process shown in Figure 3.

What shown in Figure 5 is the display about three dimensional stereogram with the help of VTK(visualization toolkit).



```
Main process
<process>
  <sequence>
    <start-event/>
     while-component
    <end-event>
  </sequence>
</process>
```

```
while-component
<while id=” circulator” >
  <condition>i<2</condition>
  flow-component
  sequence-component
</while>
```

```
flow-component
<flow id= “parallel” >
    <sequence>
      sequence-component1
    </sequence>
    <sequence>
      switch-component
    </sequence>
</flow>
```

```
sequence-component
<invoke id=″Entity Set″/>
<invoke id=″Vtk Renderer″/>
<invoke id=″Render Camera″/>
<invoke id=″Visualization″/>
```

```
switch-component
<switch id=” Splitter” >
  <case condition=” v<3” >
     sequence-component2
  <case condition=” v>3” >
     sequence-component3
</switch>
```

```
sequence-component1
<invoke id= “Centrum″/>
<invoke id=″Data Mapping″/>
<invoke id=” Object Properties″/>
<invoke id=″VTk Entity″/>
```

```
sequence-component2
<invoke id= “Cylinder″/>
<invoke id=″Data Mapping″/>
<invoke id=” Object Properties″/>
<invoke id=″VTk Entity″/>
```

```
sequence-component3
<invoke id= “Sphere″/>
<invoke id=″Data Mapping″/>
<invoke id=” Object Properties″/>
<invoke id=″VTk Entity″/>
```

FIGURE IV. PROCESS OF MAPPING ALGORITHM

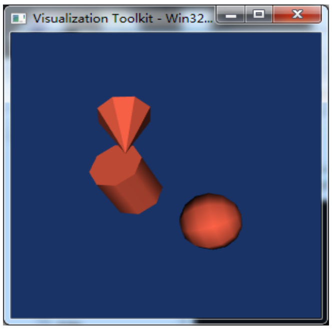


FIGURE V. VISUALIZATION OF SWF PROCESS

## V. Summary

Comparing with the four mapping algorithms mentioned in the introduction of this paper, the depth-first traversal algorithm based on component is much simpler to use when mapping the logical structure. Meanwhile, we avoid the complicated locating and judging in loop structure because of using the branch structure and the sub-process of loop structure. The algorithm can deal with tasks, events, parallel join gateway, parallel fork gateway, data-based XOR decision gateway and XOR merge gateway mapping BPEL. Because of the hierarchical decomposition of components to generate the BPEL code, the generated code has good maintainability and expansibility.

## References

[1] Harris P A, Taylor R, Thielke R, et al. Research electronic data capture (REDCap)—a metadata-driven methodology and workflow process for providing translational research informatics support[J]. Journal of biomedical informatics, 2009, 42(2): 377-381.

[2] Zhang H, Fan X, Zhang R, et al. Extending BPEL2. 0 for Grid-Based Scientific Workflow Systems[C]//Asia-Pacific Services Computing Conference, 2008. APSCC'08. IEEE. IEEE, 2008: 757-762.

[3] Sonntag M, Karastoyanova D. Model-as-you-go: An Approach for an Advanced Infrastructure for Scientific Workflows[J]. Journal of grid computing, 2013, 11(3): 553-583.

[4] Yu G E, Zhao P, Di L, et al. BPELPower—A BPEL execution engine for geospatial web services[J]. Computers & Geosciences, 2012, 47: 87-101.

[5] Akram, A., Meredith, D., Allan, R., 2006. Evaluation of BPEL to Scientific Workflows,In: Turner, S.J., Lee, B.S., Cai, W. (Eds.), Sixth IEEE International Symposium on Cluster Computing and the Grid, 2006 (CCGRID 06). IEEE Computer Society, Singapore, pp. 269–274

[6] White S. Using BPMN to model a BPEL process[J]. BPTrends, 2005, 3(3): 1-18.

[7] Dumas M. Case study: BPMN to BPEL model transformation[C]//5th International Workshop on Graph-Based Tools–GraBaTs. 2009.

[8] Recker J C, Mendling J. On the translation between BPMN and BPEL: Conceptual mismatch between process modeling languages[C]//The 18th International Conference on Advanced Information Systems Engineering. Proceedings of Workshops and Doctoral Consortium. Namur University Press, 2006: 521-532.

[9] White S A. Introduction to BPMN. 2004[J]. IBM Corporation, 2004.

[10] Ouvans C, Dumas M, Ter Hofstede A H M, et al. From BPMN process models to BPEL web services[C]//Web Services, 2006. ICWS'06 International Conference on. IEEE, 2006: 285-292.

[11] Chinosi M, Trombetta A. BPMN: An introduction to the standard[J]. Computer Standards & Interfaces, 2012, 34(1): 124-134.

[12] Grigori D, Corrales J C, Bouzeghoub M, et al. Ranking bpel processes for service discovery[J]. Services Computing, IEEE Transactions on, 2010, 3(3): 178-192.

[13] WEI Ming, XIA Yong Lin, WEI Jun. Model transformation from BPMN to BPEL 2.0[J]. Application Research of Computers, 2008, 25(11): 3363-3366.

[14] Barseghian D, Altintas I, Jones M B, et al. Workflows and extensions to the Kepler scientific workflow system to support environmental sensor data access and analysis[J]. Ecological Informatics, 2010, 5(1): 42-50.

[15] WANG Chun-jie, CAO Jian.Scientific Workflow-oriented Application Integration Framework[J]. Computer Engineering, 2009, 35(20): 258-260.

[16] Voss E, Jin Q, Overend M. A BPMN-based process map for the design and construction of façades[J]. Journal of Facade Design and Engineering, 2013, 1(1): 17-29.

[17] Ni Taoyong, Jin Naiyong. Detection Algorithm for Combinational Logic Loop Based on Digraph Depth First Traversal[J]. Computer Applications and Software, 2008, 25(6): 76-77.

[18] Downey R G, Fellows M R. Depth-First Search and the Plehn–Voigt Theorem[M]//Fundamentals of Parameterized Complexity. Springer London, Springer London, 2013: 291-300.