Journal of Theoretical and Applied Information Technology
28th February 2015. Vol.72 No.3

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

DEVELOPING A WORKFLOW MANAGEMENT SYSTEM FOR ENTERPRISE RESOURCE PLANNING

[1]RIYANARTO SARNO, [2]CHAIRAJA ALMAS DJENI, [3]IMAM MUKHLASH, [4]DWI SUNARYONO

[1,2,4]Department of Informatics, Sepuluh Nopember Institute of Technology, Surabaya Indonesia

[3]Department of Mathematics, Sepuluh Nopember Institute of Technology, Surabaya Indonesia

E-mail: [1]riyanarto@if.its.ac.id, [3]imamm@matematika.its.ac.id, [4]dwi@its-sby.edu

ABSTRACT

The company generally requires an application that can be easily configured. This can happen if the application is supported by the Workflow Management System (WFMS). WFMS which built serves to store a variety of common workflows and varied. In addition to working as a storage place, WFMS also has the ability to generate an event log that can be analyzed with the application process workflow mining and semantic search. Semantically workflows searching are e by annotated workflow metadata using the Ontology Web Language for Services (OWL-S). The capabilities of workflows searching are using SPARQL Query WordNet database. SPARQL query is used to extract data from OWL-S and database WordNet is used in calculating the value of keywords and data semantics of OWL-S. Results of the testing showed that the WFMS can do a search workflow with high accuracy from a variety of business processes in Enterprise Resource Planning (ERP). The results of the testing also show the event log generated can be analyzed using the technique of mining process.

Keywords: *Workflow Management System, OWL-S, process mining, ERP*

1. INTRODUCTION

The needs companies to business processes from ERP systems tend to fluctuations. ERP required to be quickly configured to changes in business processes. Changes in organizational structure and operating standards are among the reasons that changing business processes. In other words, applications are required to be configured flexibly.

WFMS [1] is a media that can support flexible configuration of an application. This is because the WFMS storing various types of workflows and can be compounded further. However, the use of WFMS as the application base is not without obstacles. There are two problems that exist in the use of WFMS, the search process workflow for further composition and activity analysis workflows.

Problems workflow searching process is the first issue of the background for this study. Existing WFMS as SHIWA[2] and APROMORE [3] has applied the retrieval methods in their architecture. However, the architecture in both the WFMS are closed and cannot be accessed by other systems to do a search on their system. This is due to the closing of the search system workflow storage in the form of a relational database and workflow attributes or metadata structures are not standardized. Therefore, the searching method on WFMS architecture in this research uses a standardized approache of ontology, OWL-S [4] to annotate workflows. In addition, this search method using SPARQL query language and ontology WordNet database so that it can perform semantic searches. SPARQL is used to retrieve data from an existing workflow annotation in OWL-S, while the database is used to calculate the proximity WordNet semantic similarity between words.

The second problem addressed in this research is the analysis of workflow activity. One of the techniques of analysis of the emerging workflow activity is process mining. This technique analyzes the activity of workflows from event logs generated WFMS. The advantage of the process mining is the ability to produce analytical analysis such as workload analysis, bottleneck, and fraud or cheating the system [5]. However, the resulting difficulty of converting logs to the log-standard WFMS mining process complicate the use of process mining

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

techniques [6]. To that end, this WFMS architecture using event log generator that can produce the standard event log of process mining. The event log generator methods using existing event listener on Apache ODE. The method captures its activities and establishes standards to mining process in csv format.

In other words, WFMS architecture which built have two purposes, namely to provide semantic search features workflow and form a standard workflow activity data mining process. The search feature will increase the reusability workflow [7] of workflows because workflows more easily found. The standardized of data activity of process mining can also facilitate further analysis of workflow activity.

2. BASIC CONCEPT AND RELATED RESEARCHES

This section describes the various studies that support this research. The various researches include about WFMS, WSDL matching and OWL-S Web service mining.

2.1 Business process execution language (BPEL)

Business Process Execution Language (BPEL) is an orchestration language proposed by OASIS [8]. OASIS is an organization formed by some of the leading vendors in the field of application of web service orchestration, such as Microsoft, IBM, SAP, and Siebel. This orchestration language is a language that is formed to create a web service orchestration.

BPEL is a language that uses standard XML, so that, for various attributes in the BPEL will be represented in the xml tags. This language was appears as a new requirement of the Service Oriented Architecture is the ability of the various web services in orchestration. During this time, the web service is highly dependent on the vendor poster, thus reducing the interoperability factor on SOA. Before BPEL, there are already several existing standards.

WSDL definition part of BPEL will explain all parts of the WSDL [9] used. This section will explain the various issues related to data types, declaration of partner links, input and output of the various operations. There are some important parts in the WSDL definition including Message, Port type and partner link type. Message part explains the various inputs or outputs that will be used in BPEL. This section is useful to know what parameters are to be sent and received by the client or server side. In a BPEL, sometimes there is more than one operation or web service that is used. To be able to distinguish between the operations and other operations, the use of the port type required. A port type will coordinate with one operation. In the port type used will be declared the operation and message are used. In WSDL, the various operations were declared. However, not all operations are used in BPEL come from one place is equal. Sometimes, for one event, two web services from different places used. Both this web service will be incorporated into a partner link type.

Process definition part describes the flow of activities undertaken in BPEL. This part also describes the various variables used in BPEL. This part consists of several main parts: the Partner Links, Variables, Sequence. Partner Links part explain the partner link used by the BPEL process flow. A partner link contains various partner link type and a role. When an activity partner call this link, the whole operation was in his partner link type will be called automatically. Variable part will contain variable declarations used in the BPEL flow of activity. Various types of these variables must be derived from these types that have been identified in the WSDL definition. Sequences part is a core module of the BPEL. This module describes the flow of a series of activities or workflows in BPEL.

2.2 Web Ontology Language for Services (OWL-S)

Web Ontology Language for Services (OWL-S) is a standard for describing a web service that promotes the ability of a semantic description of a web service [4]. Before OWL-S is formed, there is a standard description for a web service that is Web services Description Language (WSDL) [9]. WSDL describes a web service in terms of technical information such as operations, inputs, outputs, data types, and ports. The shortcoming of WSDL is the absence of information that can be extracted in order to create the semantic web service capabilities. Therefore, OWL-S appears to provide semantic capabilities to the web service.

OWL-S is divided in three major parts, namely the service profile, service grounding, and service models. Service profile is the part that contains the description of the web service such as a name, a brief description, the developer of a web service. Service grounding is the part that contains a variety of information about how to relate to the web service. Service models describe the model of the operation of a web service.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

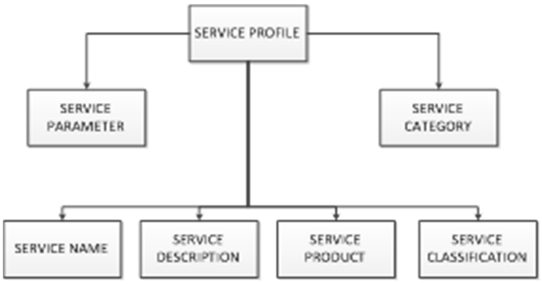

Figure 1: Service profiles on OWL-S

Service profiles are part of OWL-S are queryable. Service profile has several sections as shown in Figure 1. In Figure 1, there are some important parts that must be explained. First, a brief description of the parts such as service classification, service product, service name, text description is a part that is filled manually by the web service developers when creating OWL-S files. The part that is on the left, shows the various properties of the process, such as parameters, condition, result, input, and output. Various parts are needed to determine the model of the process in a nutshell, while the further explanation of the model will be given by the model parts service.

2.3 Workflow management system (WFMS)

Workflow management system has a very broad and diverse definition. Based on Van der Aalst and Van Hee, WFMS is ideas, methods, techniques, and software used to support structured business processes [10]. While based on the Workflow Management Coalition (WFMC) [1], WFMS is a system that is capable to define, create, and manage the execution of a workflow that comes from the use of a software or source code and run both on the same workflow engine or different. Various definitions conclude that a WFMS should run a few main functions, namely running a business process or workflow, manage various business processes or workflows that exist, and create and define business processes.

There are several WFMS that are well known in the development of workflows, such as an Advanced Process Model Repository (APROMORE) [3] and SHIWA [2]. Existing architecture on APROMORE and SHIWA have some similarities. WFMS architecture in both the repository and workflows had a data repository. Workflow repository at SHIWA contains a variety of workflows.

2.4 Matching WSDL and OWL-S Web services

In previous studies, there is a method to connect between the web service and OWL-S in the field of service advertisement [11]. This study offers a method for using OWL-S as an advertisement and query performed on OWL-S and WSDL. This can be done because there is a similarity of WSDL and OWL-S. This study mentions there are some similarities, the operation name, input messages and output messages.

Based on these similarities, the method calculates similarity between WSDL and OWL-S uses two formulas, namely the operation similarity calculation and input-output similarity calculation. Operations similarity calculation formula calculates the value of the similarity between the operation name in the OWL-S and the operation name in the WSDL using semantics. Input similarity calculation is a formula that calculates the input and the input message in the WSDL concept in OWL-S, while the output similarity calculation is a formula for calculating the output concept of OWL-S and the output message of WSDL. The entire value of the formula had to be multiplied by their respective weights. Weighting is determined by the user, the higher the weight, the more important the component values for users.

There is also a system architecture which offers the OWL-S to apply the repository [12]. The concept offered named Ontology Search Engine and Standardization (OSSE). The concept of ontology is seeking files that match keywords and sorted by concept matching of ontologies.

In addition, there is also a method that is offered in the form of broker-based Semantic Search Agent (SSA) [13]. This method focuses on the architecture of a search based on the similarity of input output and UDDI, another standard that represents a web service.

2.5 Web Service Mining

There are several studies that have conducted research on the possibility of doing the analysis or mining of the activity or activities of the web service. The difficulties encountered in the concept of mining the web service as described in the paper service mining [5]. One of the difficulties is in the collection of activity data or log web service. The difficulty in this section range from concept instance in the event log standard mining process, certainty and correctness of the logs generated, and the relationship between the model and the log.

Journal of Theoretical and Applied Information Technology
28th February 2015. Vol.72 No.3

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Several studies have tried to examine the difficulties in this log problem. There are studies that attempt to produce an event log of the web service. For example, the method has been tried in [14]. Various methods were tried as trivial level, which takes data from the header to the web service request. This study also provides a method of logging a variety of services available today.

Process mining is a technique of extracting information using the log as one of the main ingredients in the method [6]. Process mining has three main parts: discovery, conformance, and enhancement. Log is the main ingredient of the process of discovery. In the discovery process, the logs will be processed into a model. Furthermore, important information, such as performance, bottlenecks, and analyze the load of the software can be known from the analysis of the model [5]. For this reason, this paper has a feature to generate a log that can be used for process mining method.

Event logs are used in the mining process is very much different from the standard logs that are known so far. In the process of mining concept, the event log has three mandatory attributes, namely caseID, activity, and timestamp [6]. CaseID is an attribute that indicates the ID of a process. CaseID needed to distinguish the various processes that use the same activity. Activity is an attribute to indicate the name of the activities undertaken. Timestamp is an attribute that indicates the time of an activity. Timestamp can be divided into two, the starting time an activity, and the end of an activity. Attribute event log other than those mentioned above, are optional. An example of a true event log can be seen in Figure 2.

The biggest problem in the event log is the lack of standard log format in information systems or workflow application. It makes difficulties to automatically process analysis of workflow systems. Thus, one of the problems solved in this paper would like is to create a system that can generate an event log that use this standard.

Case id	Event id	Properties	
		Timestamp	Activity
1	35654423	30-12-2010:11.02	Register request
	35654424	31-12-2010:10.06	Examine thoroughly
	35654425	05-01-2011:15.12	Check ticket
	35654426	06-01-2011:11.18	Decide
	35654427	07-01-2011:14.24	Reject request

Figure 2: Event log example of standardized process mining

3. PROPOSED METHOD

3.1 System Architecture

In this study, WFMS architecture used is shown in Figure 3. The application is built in the form of a web application that uses the Java language and the framework used is SPRING. There are several important points of the architecture proposed in this study, namely WFMS architecture has two repositories that BPEL and OWL-S repository. In this system architecture, workflow engine used is APACHE Orchestration Director Engine (ODE). Storage of event logs generated using a relational database.

There is a difference in the architecture proposed in this research with the existing architecture. The differences are the different approaches of the workflow metadata attributes. WFMS such as SHIWA and APROMORE store attributes of the workflow in the form of a relational database, and does not have a standard attribute of the attribute used. While this study use a standard approach to ontology with OWL-S. There are several advantages of using OWL-S standard in the standard attributes of a workflow. The first is the standard OWL-S is general, so that users can understand the existing attributes. The second is, because it is common, OWL-S can be used by different queries, and can improve the independent nature of OWL-S from the WFMS. In other words, the search feature on the WFMS can take OWL-S from the various OWL-S repository- WFMS itself or from other WFMS.

Based on the purpose of this study is to provide architecture concepts that can store a variety of common and varied workflow and assist in the composition of the new workflow with the search process. To store a variety of common and variable workflow, used to BPEL repository storage feature. This architecture will make the service description standard OWL-S from the various existing workflow in BPEL repository. To provide confidentiality in workflow, BPEL workflows based on user repository differentiate and type. Workflow can be public or private. Workflow which is private, will not be queried when the searching feature is activated.

Both repositories contained on this architecture connected by an OWL-S Converter. In other words, the description of standardized OWL-S workflow contained in OWL-S Repository is a description from the BPEL repository that has been generated using OWL-S Converter. OWL-S Converter is a

Journal of Theoretical and Applied Information Technology
28th February 2015. Vol.72 No.3

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

method that takes a description of the repository and workflow of the user and convert it into a standard workflow description OWL-S. This method is one of the important points in this architecture, as it helps the search process workflows to produce a variety of workflow description.

Both storage and repository functions to accommodate workflow uploaded by users. This architecture separates the repository BPEL and OWL-S repository because it is different. BPEL repository is closed and can only be accessed by the web application. This is because the closed nature contains BPEL orchestration and implementation of a sensitive nature. Some tenants may want to keep the implementation and workflow variations used in the company. However, OWL-S serves as an advertisement that contains the profile of the service that not sensitive. Unlike other WFMS, this architecture allows OWL-S repository accessible to the public. BPEL repository is loaded when the user uploading workflow into BPEL.

There are several stages of the groove provided by this architecture. The first stage is the upload BPEL by users. At this stage, users can upload files that have been compressed with the zip's standardized BPEL. The second stage is the implementation of storage and BPEL. BPEL files that have been uploaded are stored in the repository and into the web service is implemented by the workflow engine. The third stage is the conversion of BPEL into OWL-S. At this stage, web services that have been implemented will be changed OWL-S and OWL-S are stored in the repository. The fourth stage is the search workflow. Users who want to search for the required workflow composition may be looking to give something from the service name and description. The final stage is taking the event log. Any activity performed by the workflow will be recorded by the system. This is possible because the architecture is equipped with an event log generator method.

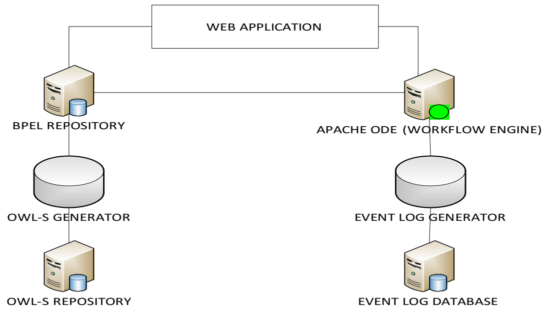

Figure 3: Proposed WFMS architectures

3.2 OWL-S Generator

OWL-S Generator is the method used to generate the OWL-S standard's workflow description. This method uses a library of OWL-S API. This conversion method consists of multiple stages of data collection from users, taking part WSD, and the formation of an OWL-S description.

Flow conversion from BPEL to OWL-S can be seen in Figure 4. The data acquisition phase is useful to retrieve the data needed to create a profile section of OWL-S. At this stage, the user provides the service name and service description. Service name in WFMS consists of three words, not separated by a space, and the first letter of the first word and the second had to wear a capital letter. It is useful to facilitate the solution of the string in the search feature workflow. Service description in WFMS forms a set of words that describe the service in English. After that, the stage of implementation of the decision will be started WSDL. At this stage, the implementation in the form of a WSDL of a BPEL generated from Apache ODE, will be taken. The final stage of this process is the incorporation and formation of OWL-S from the data collected. To complete this phase, OWL-S API will use WSDL translator class to form part of the process and grounding. After that, WSDLtranslator will generate a service description in OWL-S standard. The data provided will be added to the user profile section and are incorporated within the process and grounding. After the OWL-S finished formed, OWL-S files to be incorporated into OWL-S repository.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

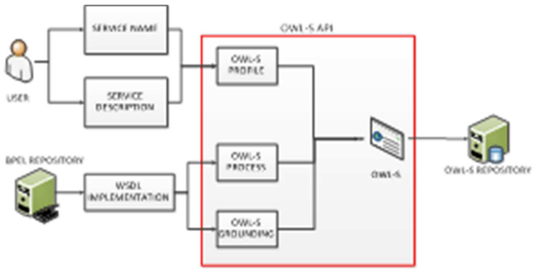

Figure 4: Conversion flow of BPEL to OWL-S

3.3 Workflows Searching using Semantic OWL-S Query

The workflows searching feature are proposed in this study consists of two major phases, namely query phase for OWL-S descriptions using SPARQL and semantic similarity calculation stage using the WordNet database of Princeton that can be seen in Figure 5.

The first stage of the searching process is taking all the information from the file description OWL-S and keywords from the user. Keywords of the user can be obtained from the form shown from the web application. While the description of OWL-S can be obtained using a SPARQL query like the one in Figure 6. Various descriptions of OWL-S can be obtained through two stages. The first stage is to enter the entire OWL-S files in the repository to the OWL-API, so the entire file OWL-S can be queried. The second phase is the query for the entire OWL-S files that have been taken from the repository. At this stage, will produce a variety of SPARQL query service name and service description of each file OWL-S.

The second stage of the searching process is the calculation of semantic similarity. This stage looks for a value of similarity between the keywords and the description of OWL-S using the formula in (1).

$$score = \frac{sem(name\boldsymbol{OWL}, name\boldsymbol{Key}) + sem(desc\boldsymbol{OWL}, desc\boldsymbol{Key})}{2} \quad (1)$$

The formula in (1) assess the similarity between the keyword and description of OWL-S from the value of the similarity between the keyword and the service name in the file and the OWL-S service description of the keywords and OWL-S files. The calculation of the similarity service name and service description are calculated using (1) or which is denoted by the function sem. For the calculation using (1), we need a database that can determine the value of proximity between one word with another. Therefore, the similarity calculation of the WordNet database from the Princeton between the words used in the formula (1) performed. The value of the similarity service name and service description then divided by two to generate a score value that represents the value of the similarity between OWL-S description and keywords.

After the second phase is completed, each file will have the OWL-S score, namely the value of proximity or similarity between OWL-S description and keywords. If a higher score obtained, the higher affinity between description and keywords are given. Obtained score value will determine whether the service described by OWL-S is close enough to be shown to the user. For that, set a threshold to determine the minimum value score from OWL-S descriptions that can be taken. Determination of the threshold value can be seen in the test.

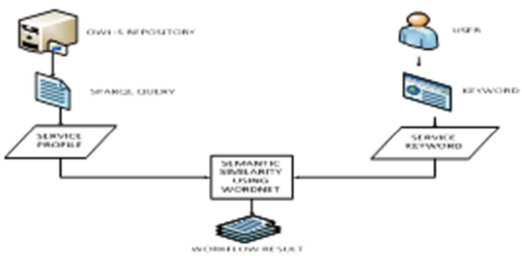

Figure 5: Stages of workflow searching features

```
PREFIX service:<http://www.daml.org/services/owl-s/1.2/Service.owl#>
PREFIX profile:<http://www.daml.org/services/owl-s/1.2/Profile.owl#>

SELECT ?subject ?name ?desc
WHERE { ?subject service:presents ?object.
?object profile:serviceName ?name.
?object profile:textDescription ?desc.
}
```

Figure 6: Query SPARQL

3.4 Event log generator

The second problem in this research is a method for generating event log or event log generator. There are various methods that have been developed and discourse in order to generate the event log of the web service composition. One of the methods is the concept of mining the web service that wants to change the concept of the SOAP header by adding information about the event log attribute [5]. This concept explains that the lack of information from the header of the SOAP protocol has led to difficulty in forming the event log of the web service. For this reason, this concept would like to add information to the tag <WSHeaders> which can contain information such as protocol choreography, choreography name, and chorography case.

Journal of Theoretical and Applied Information Technology
28th February 2015. Vol.72 No.3

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

In addition to the web service mining concept above, there is another concept by utilizing various logging facilities [15]. This concept distinguishes various types of mining web service that can be done by logging facilities available. There are five levels of mining web service provided by this concept as shown in Table 1.

Table 1: Tabel berbagai tingkat web service mining

Level	Logged information	Logging facilities	Mining opportunities
1	- consumer IP - invoked WS - timestamp - HTTP status code	- Web server	- service utilization - consumer tracking - failure rates
2	- level 1 + - SOAP request & response - timestamps	- HTTP listener & logger	- level 1 + - WS execution time - analysis of SOAP messages
3	- invoked WS & operation - SOAP request & response - timestamps	- WS container - SOAP handlers	- level 2
4	- level 3 + - consumer-side activity	- WS container - SOAP handlers	- level 3 + - client-side activity
5	- level 4 + - workflow information	- level 4 + - Web services	- level 4 + - Web services workflows

At the first level, logging facilities are only available on a web server where the service is located. Information that can be obtained only from the client IP address, the web service is executed, a timestamp, and the HTTP status code. With this bit of information, the event log cannot be formed. Various levels that exist in this concept don't explain how the event log information can be formed. This concept describes how the information is only contained information on the web service and SOAP protocol can be useful for mining web service.

Therefore, this study uses the system event log on the server APACHE ODE generator. This system is a logging facilities provided by the APACHE ODE. This concept is very different from the two concepts mentioned above. This system does not change the structure of the SOAP header information. The system also does not utilize HTTP logging. The system utilizes the implementation of an event listener classes available from APACHE ODE. APACHE ODE gives an abstract class that can be implemented by the user. The method of this class will be called every one of activity is called. The implementation used in this research work flow depicted in Figure 7.

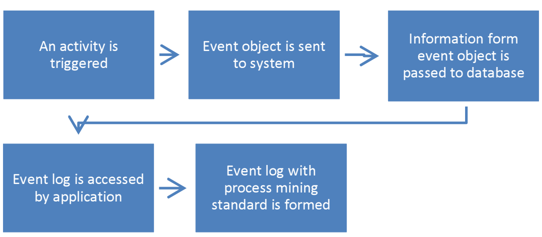

Figure 7: Stages of generating event log on WFMS

In this flow, there are several important parts. Event object delivery part occurs in the system event listener. Once the object is received, information from the event can be accessed. All information will be sent to the database, but only some important information such as caseID, activity Name, status, and timestamp. After all the event logs are in the database, the event log can be viewed via the web. Event logs can also be downloaded into a csv format. Csv format chosen because this data type that can be processed into the mining process. In addition, this format is easy and there are applications that can change the format of the data in csv format to process mining, such as mxml format.

4. PROPOSED METHOD

4.1 System Architecture

The case study used in this paper is the case of business processes for Enterprise Resource Planning. The parts used are part of the Oder Sales of ERP business processes in [16]. The chronology of sale orders in the ERP can be seen in Figure 8. Sale order is a business process flow of the ERP that focus on procurement. The flow starts from procurement request and ending on the payment of goods.

There are eight activities in this flow. The first activity, presale activities is an activity that checks for the required items and document procurement. Sales order entry is an activity that includes procurement data and examine the attributes of the procurement. Check Availability are activities that would check the availability of goods held in inventory. Activity of pack material and material pick an activity that takes a list of items needed. Good post issue activity is an activity that takes care of the transportation and receipt of goods in the warehouse. Activity of customer invoices and receipt of customer payment is an activity that takes care of an issue with the payment for goods arriving at the warehouse.

Each activity is composed of a web service below. This composition is similar to the semantic web service composition [17]. Description of the various activities will also be incorporated into the case study. This description obtained from a collection of explanations obtained from SAP ERP.

Journal of Theoretical and Applied Information Technology
28th February 2015. Vol.72 No.3

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

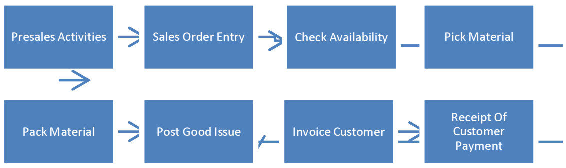

Figure 8: Flow of sales order

4.2 Evaluation

Testing of Workflow Searching Methods

This section describes the tests performed to test the searching and workflow event log generator features. This test scenario follows the scenario of semantic web service test case composition. Every activity of sales order flow in Figure 10 will be used as a web service 30 pieces that have a high similarity. It is a composite Web service, or has a part in it. In this test, we use 240 pieces of the web service to test the similarity scores method proposed in this paper. The whole web service has been converted into OWL-S has a service name and description in English.

Scenario testing is performed by splitting the web service into eight types according to their respective activities. This is done because there are eight different queries contained in the test. Each query represents a single event. Once the web service is divided into eight sections, each query will be executed. So similarity value score from each web service will appear and recorded.

Once all the data is tested and scored, the accuracy of this method can be calculated according to the formula (3). True Positive (TP) is the amount of data that has a value above the threshold and is derived from the list of web services on such activity. True Negative (TN) is the amount of data that has a value below the threshold and does not come from the list of web services on such activity. False Positive (FP) is the amount of data that has a value above the threshold but does not come from the list of web services on such activity. False Negative (FN) is the amount of data that has a value below the threshold but are from the web service in the event.

Threshold value is an average taken the best threshold value at each test activity. Results of testing an activity can be seen in Table 3. This test is a test that uses a pick activity material. Query to seek service from the pick activity materials will be incorporated into the system as a keyword. Keywords will be compared with the rest of the files existing OWL-S so that all will have a similarity value of the keyword. Then, each threshold will try and produce accuracy as Table 3. The best of threshold at every activity will be collected and averaged to produce the overall threshold as shown in Table 2.

From the test results, there are some things that should be observed. Firstly, the case of FP occurs because some of the descriptions contained in the web service has a word that is synonymous with the keyword but at a different location. The difference lies in the calculation of the word is not accommodated. Cases like this are rare, so it does not cause any significant change. Secondly, the high threshold does not guarantee high accuracy. This is due to the calculation of semantic value depends on the difference in number of words. In other words, the difference in the number of words if keywords and OWL-S is too far, the semantic value will be smaller. Thus, for cases with little semantic value that can be derived from the positive list, due to difference in the number of words in the OWL-S and keywords. Therefore, it is suggested limiting the number of words on OWL-S and keywords to maintain high accuracy.

$$accuracy = \frac{TP+TN}{TP+FP+TN+FN} \quad (3)$$

Table 2 Accuration of counting results

No	Activity	Best Threshold
1	Presale Activity Service	0.7
2	saleOrderEntryService	0.7
3	checkAvailabilityService	0.7
4	pickMaterialService	0.6
5	packMaterialService	0.6
6	postGoodIssueService	0.7
7	invoiceCustomerService	0.6
8	receiptOfCustomerPaymentService	0.7
	Mean	0.6625

Journal of Theoretical and Applied Information Technology
28th February 2015. Vol.72 No.3

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Table 3 Testing of semantic query pick material activity

Threshold	TP	FP	TN	FN	Accuracy
0.1	30	210	0	0	0.125
0.2	30	210	0	0	0.125
0.3	30	204	6	0	0.15
0.4	30	48	162	0	0.8
0.5	30	18	192	0	0.925
0.6	30	0	210	0	1
0.7	24	0	210	6	0.975
0.8	23	0	210	7	0.970833333
0.9	12	0	210	18	0.925

Testing of event log generator methods

Testing this method also uses a case study of the sales order. Sales order is composed in the form of BPEL to be uploaded first. After uploaded, the application will be implemented in the form of BPEL web service. After that, the activity of sale order will be activated and executed. After taking the event log, the event log will be opened with a spreadsheet application MICROSOFT EXCEL.

ID	processID	caseID	activity	status	timestamp
25	{http://naga-naga.com}salesOrderService	3701	receiveInput	start	5/31/2014 18:47
26	{http://naga-naga.com}salesOrderService	3701	receiveInput	complete	5/31/2014 18:47
27	{http://naga-naga.com}salesOrderService	3701	Assign	start	5/31/2014 18:47
28	{http://naga-naga.com}salesOrderService	3701	Assign	complete	5/31/2014 18:47
29	{http://naga-naga.com}salesOrderService	3701	presalesActivitiesInvoke	start	5/31/2014 18:47
30	{http://naga-naga.com}salesOrderService	3701	presalesActivitiesInvoke	complete	5/31/2014 18:47
31	{http://naga-naga.com}salesOrderService	3701	Assign1	start	5/31/2014 18:47
32	{http://naga-naga.com}salesOrderService	3701	Assign1	complete	5/31/2014 18:47

Figure 9: Piece of sales order actifity event log

Event logs that have been taken in CSV format can be seen in Figure 9. In the event log, there are five attributes that are compliant with the standard event log mining process, namely are processID, caseID, activity, status and timestamp. After that, the event log can be analyzed by using process mining applications. The results can be seen in Figure 10. This shows the event log generated by this architecture is the standard event log mining process.

Figure 10: Mining Process Activity

5. CONCLUSION

The needs for WFMS arise from the need for flexible application. The difficulty faced by developers in WFMS is the workflow searching and formation workflow activity data that can be analyzed. WFMS architecture which was built to focus on proposes a new search method and a method of generating a workflow activity data that can be analyzed further.

Searching method proposed in the architecture of WFMS is a semantic-based search method. This method is divided into two stages, namely annotation workflow and calculation of similarity values. Annotation process of workflow using ontology approaches with OWL-S standard. To query the annotation of standardized OWL-S workflow, WFMS architecture using a query ontology language, namely SPARQL. Phase compute similarity values calculated similarity between the keywords and description OWL-S. These calculations compare the service name and description, and keywords that exist on OWL-S. To assist in the counting process semantics, this architecture using the WordNet database from Princeton.

Method of forming a workflow activity data used to establish the corresponding log data with standard analysis techniques mining process. This method uses the concept of event listeners from APACHE ODE workflow engine. This method works by capturing events sent by APACHE ODE and compile them into a database.

Both methods contained on this architecture has provided a new perspective on solving search problems workflows and formation activity data. Workflows Searching method has high accuracy and has been tested on a variety of different business processes. This proves the method is successful in achieving the first objective of this WFMS architecture, namely the problem of finding

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

the concept of workflow. Method of forming a workflow activity data is also successfully generates workflow activity data that meet mining process standards. This is proven through testing of the method of formation data. From the test, the data that has been generated by the WFMS can be further analyzed using process mining applications.

ACKNOWLEDGEMENTS

We would like to thank the Higher Education Directorate of the Education Ministry of Indonesia for supporting the research.

REFRENCES:

[1] Workflow Management Coalition, The Workflow Management Coalition Specification, 2, (1996)

[2] Kacsuk P., Terstyánszky G., Balaskó Á., Karóczkai K., Farkas Z., Executing Multi-workflow simulations on a mixed grid/cloud infrastructure using the SHIWA and SCI-BUS Technology, in *Advances in Parallel Computing*, IOS Press, Amsterdam (2013), Vol. Cloud Computing and Big Data, pp.141

[3] M. L. Rosaa, H. A. Reijersb, W. M. P. V. D. Aalst, R. M. Dijkmanb, J. Mendlingc, M. Dumasd, L. García-Bañuelosd, APROMORE: an advanced process model repository, *Journal Expert Systems With Applications* Vol 38, issue 6 (2011)

[4] D. Martin, M. Burstein, J. Hobbs,O. Lassila,D. McDermott,S. McIlraith, S. Narayanan,M. Paolucci,B. Parsia,T. Payne,E. Sirin,N. Srinivasan and K. Sycara, OWL-S: Semantic Markup for Web Services (2004), http://www.w3.org/Submission/2004/07/

[5] Blockeel H. ,Kosala R., Web mining research: A survey, *ACM SIGKDD Explorations Newsletter*

[6] W.M.P. van der Aalst. Service Mining: Using Process Mining to Discover, Check, and Improve Service Behavior, *IEEE Transactions on Services Computing*, 2013

[7] Khoshkbarforoushha A., Jamshidi P., Gholami M.F.,Wang L., and Ranjan R., Metrics for BPEL Process Reusability Analysis in a Workflow System, *IEEE Journal System*, 99 (2014)

[8] Jordan D., Evdemon J., Web Services Business Process Execution Language Version 2.0, Edited Alves A.,Arkin A., Askary S., Barreto C.,Bloch B.,Curbera F.,Ford M., Goland Y.,Guízar A.,Kartha N.,Liu C.K., Khalaf R., König D., Marin M, Mehta V.,Thatte S.,Rijn D.v.D,Yendluri P.,Yiu A., (2007)

[9] Christensen E., Curbera F.,Meredith G., Weerawarana S. , Web Services Description Language (WSDL) 1.1, (2011)

[10] W. M. P. v. d. Aalst and K. M. v. Hee, Workflow Management: Models, Methods, and Systems. Cambridge, (2002).

[11] Le D., Nguyen V, Goh A., Matching WSDL and OWL-S Web Services ICSC '09 Proceedings of the 2009 IEEE International Conference on Semantic Computing, (2009);

[12] Celik D., Elci A., Computer Software and Applications, 2008. COMPSAC '08. 32nd Annual IEEE International, (2008) July 28-August 1; Turku

[13] Farrag T.A.; Saleh A.I., Ali H.A., Toward SWSs discovery: mapping from WSDL to owl-s based on ontology search and standardization engine, *IEEE Transactions on Knowledge and Data Engineering 25*, 5, (2013)

[14] Gaaloul W., Bhiri S., and Godart C., Atelier Systèmes d'Information et Services Web, (2006)

[15] S. Dustdar and R. Gombotz, Discovering Web Service Work ows Using Web Services Interaction Mining, *International Journal of Business Process Integration and Management*, 1, 256, (2007).

[16] Sarno R., Sanjoyo B.A., Mukhlash I., Astuti H.M, Petri Net Model Of ERP Business Process Variation For Small And Medium Enterprises, *Journal of Theoretical and Applied Information Technology*, Vol. 54, No. 1 (2013)

[17] Kunaefi A., Sarno R., Sunaryono D., and Mukhlash I, Semantic Web Service Composition for ERP Business Process, *Journal Kursor* , Vol. 7, No. 1 (2013)