Association Rule Mining for Failure Recovery Strategy Selection in Composite Web Services

O.Bushehrian and N.Mohammadi
Department of Computer Engineering and IT, Shiraz University of Technology, Shiraz, Iran.
bushehrian@sutech.ac.ir

Abstract— Web services composition has a special position in the industry. Using the concept of combining web services, large and complex applications can be solved easily. The most important issue in web services composition is fault tolerance. Different errors may occur in web services composition, leading to a failure in the execution of web services. There are two strategies to recover failures (forward and backward recoveries) and both strategies have overhead costs. Usually in the forward recovery, alternative web services are invoked. Although these services have low failure rate, they have high costs. Further, the backward recovery done by the compensation web services imposes a rollback cost. Hence, an accurate selection for recovery strategy is very important because it has significant impact on reducing overhead costs. This research uses association rules mining technique to predict the possible combination of the executive states. The results illustrate that the extracted rules can predict success or failure in the implementation of web services with high accuracy.

***Index Terms*—Association rules, Backward recovery, Fault-tolerance, Forward recovery, Web service composition.**

I. INTRODUCTION

The combination of web services has a great impact in solving complex requests from users. Service Oriented Architecture (SOA) has provided a framework through which, data and services can be accessed in distributed environments. Each web service has an independent functionality [1]. By combining functionality of all web services, more complex requests can be solved. There are different fault-tolerant patterns for forward recovery strategy such as the Retry, Recovery-Block, Voting, etc [2]. Each pattern may impose costs due to combined web services. For example, in Recovery-Block pattern, alternative web services are invoked. They have a higher price compared to the main web services. Hence, an appropriate recovery strategy selection is important. In this paper, the aim is to identify different types of failures (permanent- transient) [3-4]. If the type of failure is permanent, the backward recovery should be selected. Otherwise, to improve the transient failures, the forward recovery is the most appropriate selection. Having accurate rules are essential, because the most appropriate recovery strategy can be selected when a failure occurs. For example, Si → rollback expresses that if in the future web service Si fails, the best decision to repair the failure is the rollback strategy. To have these rules, the association rules mining technique have been used. In the following, Section 2 explains some of the related studies. Section 3 illustrates how a secure workflow is created by the association rules mining technique. In Section 4, the extracted rules have been described in detail and the results are shown in Section 5.

II. LITERATURE REVIEW

During the combination of web services, various errors may occur which lead to failure in the implementation of web services. The use of each backward or forward strategy will impose costs on composition. So, the deliberate choice of recovery strategy is very important. Recently, in [1] a FSP (Finite State Process) language has been used to create a fault-tolerant workflow. The FSP provides a situation wherein a sequence of the implementation of web services can be determined through it. In this case, when a web service fails, a recovery strategy runs with the lowest cost. In [2], a concept of the event calculations has been used to increase the reliability of web services and fault tolerance of combinations. This research reviews a variety of possible dependency between the web services based on event calculations. Then, according to the dependencies, some patterns in the workflow such as AND-Split, AND-Join and XOR-Split are applied.

In [3], to increase the reliability of web services and to prevent failure in their execution, transaction patterns have been used as a convergent concept between ATM and workflow patterns. This study investigates the state of using transaction patterns to increase the reliability of web services. In [4], to have a fault-tolerant combination, the transactional web services have been used. To reach this goal, a concept of risk has been introduced. The risk 0 means that after the completion of the execution of combination, the user can neutralize the effects of the executed web services. Risk 1 means that there is no such possibility. In [5], a FACETA framework has been implemented to increase the performance and fault tolerance in transactional web services. To support the failures, the implementation of a forward strategy is done by replacing the failed web service with an alternative web service, and backward recovery is based on the unrolling algorithm in the colored petri net. In [6], a FACTS framework has been introduced in order to composite the fault-tolerant web services. In this framework, some web services that cannot be neutralized, use transfer technology, which is more economical.

In [7], the forward or backward recovery has been used to recover failures. In this paper, a replica technique has been used for forward recovery, especially when the time factor is very important. However, there is no solution about selecting the most appropriate recovery strategy that consciously predicts the failure or success of web services. In [8], fault-tolerant patterns have been used in order to decrease the failure rate during the execution of composition. So far, none of the previous works have offered an accurate solution to select the most appropriate strategy to improve failures. In [9], the purpose is to separate the failure management from the combination flow. To reach this goal, an orchestration has been created. In the orchestration, the logic of failure retrieval has been considered as a separate module, using the tagged system. In the composition, each web service has two properties (rollback cost and failure rate), in which the recovery strategy is chosen based on the calculated cost in the event of failure.

The aim of this investigation is to extract the valid rules based on the history of the implementation of combined web services as the success or failure for each of them can be predicted with high accuracy. Therefore, by having such valid rules during failure, the most appropriate recovery strategy can be chosen in order to maintain the quality factors.

III. METHODOLOGY

As previously mentioned, selection of each recovery strategy (forward or backward) will have overhead costs. Aim of this research is to present a methodology to select the most appropriate recovery strategy, so that the overhead costs are minimized. Figure 1 illustrates the proposed methodology.

The proposed methodology is divided into 3 main sections that are explained in the following.

A. *Runtime Monitoring*

Runtime monitoring determines the status of executive states of web services in combination. Each web service can be in one of the states Not-Executed, Running, Succeed and Failed. The executive states monitoring is done by orchestration. Figure 2 illustrates the communication between orchestration and web services in BPEL (Business Process Execution Language) designer environment.

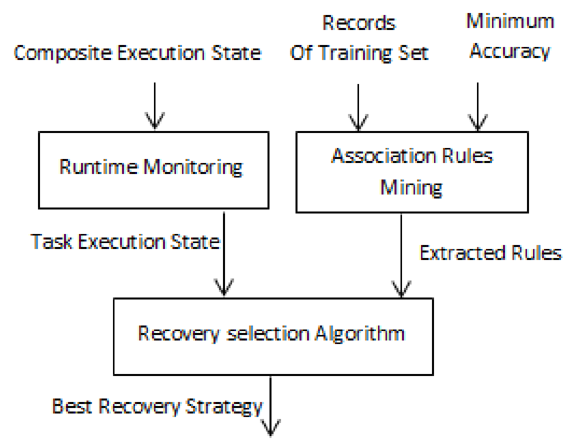


Figure 1: Proposed methodology

In this method, the orchestration designer defines the transactional dependencies between web services and the sequence of execution of web services in the workflow inside the controller. After executing each web service, the executive state (success or failure) should be notified to the controller by orchestration. It will be done via sending an index of web service and their executive state to the controller Perform-Decision (Index, State) function. If the type of failure is permanent, timeout error will happen on the BPEL environment and then the orchestration will be notified to the controller. The output of Perform-Decision (Index, State) function is a vector of integer type. If the type of failure is transient, the controller will select forward recovery to create training and test sets by giving a number to output vector elements which are related to the alternative web services. For example, if web service Si is selected to run, its element will be initialized with 1 within output vector. However, if permanent failures occur, the controller should select backward recovery. Figure 3 shows the schema of BPEL designer environment for workflow 4.

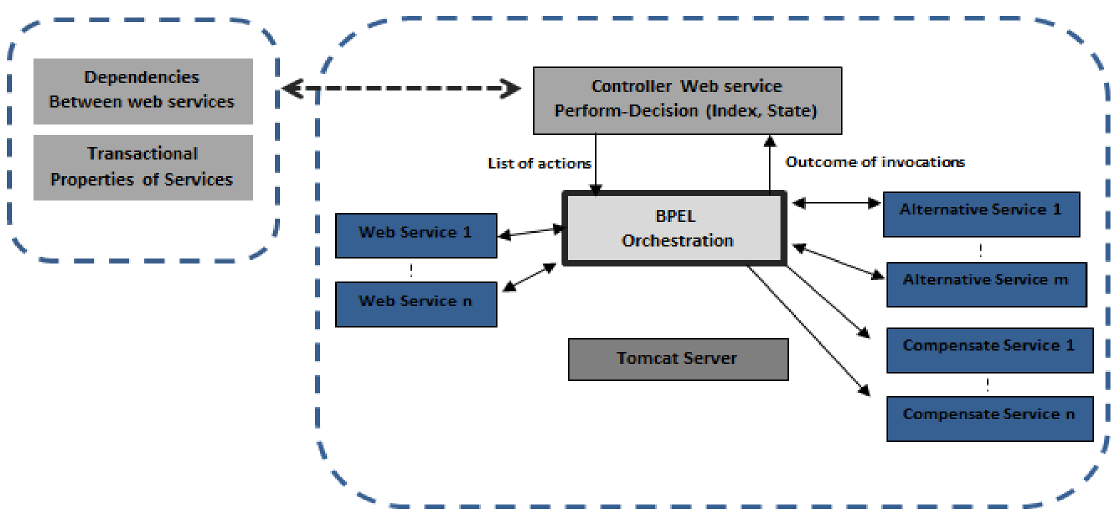


Figure 2: Communication between orchestration and web services

Figure 3: Schema of BPEL designer environment for Workflow 4

The command type (backward or forward recoveries) has been determined via an element in the end of the output vector that its amount is -1 by default. If the controller selects the backward recovery, the amount of this element will be 1. Otherwise, if forward recovery is selected, the amount of this element will be 2.

B. *Association rules mining*

Data mining is used in a wide range of different fields. It is an important tool for analyzing data in the research fields. For example, there are essential applications of data streams such as network traffic monitoring that requires extracting the relevant rules. In spite of the saved data within the static traditional database in distributed environments, data streams usually arrive continuously quickly with large volume. This causes issues that require applying techniques to explore the related rules [10]. [11] illustrates the usage of the association rule mining technique to estimate the missing data in sensor networks. Another functionality is the prediction of frequency on internet packet streams in distributed environments [12]. The association of rule mining has also been used in MAIDS projects to recognize alarming incidents in data streams [13]. In [14], it explains the similarities and differences in the preferences of customers in collaborative recommender systems or it is used for large-scale gene-expression data analysis in order to express a set of correlated genes [15].

The important issue is to get the best strategy to improve the failures so that the two properties, the compensation cost of web services and the invocation cost for alternative web services are taken into account. Thus, the rules mining technique is used to extract the valid rules in the form of event→action. For example, a rule such as Si→rollback means that if during combination, the web service Si failed, the best strategy to repair the failure is backward recovery. These rules are extracted based on the training data set for each of the workflows. Then, the rules are verified by the test set.

To assess the association rules, accuracy metric has been defined. Equation 1 shows how the accuracy metric is calculated for each rule, if a rule such as x→y existed.

$$accuracy = counts\ (a \rightarrow b) \ / \ counts\ (a) \tag{1}$$

where counts (a→b) expresses the number of records inside the test set that web services a and b have failed, together in a specified run. Counts (a) express the number of failures in web service a in all records of test set. Minimum accuracy has been determined experimentally by the designer. The accuracy of the selected rules should be greater than or equal to the minimum accuracy. Equation 2 shows this concept.

$$accuracy \geq minimum\ accuracy \tag{2}$$

To perform the rule mining steps, different workflows with different parallelism degrees have been defined. All the alternative web services have low failure rate and high cost. For the web services with no alternatives service, the 'retry' pattern is used. To extract the valid rules among the extracted rules, the minimum accuracy 30 has been defined by the designer. To create training and test sets when failure occurs, forward recovery has been chosen. Each of 4 workflows has been executed 150 times in BPEL environment for each scenario (high failure rate and low failure rate). In each workflow, 2/3 of the records are used as training set and the rest are considered as test set. Test set has been used to compute the accuracy of the final extracted rules. To extract and choose the valid rules, the following steps are followed:

Step 1: Implementing a workflow within BPEL designer environment and selecting the forward recovery strategy in case of failure for data set creation.

Step 2: Extracting all possible rules from the workflow and computing the accuracy for any rule.

Step 3: Selecting rules in which their accuracy is greater than or equal to minimum accuracy.

Step 4: Re-computing the accuracy for the selected rules based on the records inside test set.

Step 5: Choosing the rules in which their accuracy in predicting the execution of web services is more than or equal to 40 percent.

In the first step, every workflow should be implemented within BPEL environment by orchestration to extract the valid rules. As previously mentioned, to create training and test sets, if a transient failure occurred, a forward recovery is selected by the controller in order to repair the failure. Only, if the type of failure is permanent, the backward recovery must inevitably be followed. The permanent failure is detected by the orchestration within BPEL designer environment. In the second step, after creating the training and test sets, all the possible rules are made from the workflows. Making these rules depends on the sequence of the execution of web services in the composition, and the accuracy value is calculated for each rule using Equation 1. After selecting the rules extraction and calculating the accuracy based on training set records, in Step 3, the accuracy of selected rules should be greater than or equal to 30 (minimum accuracy is 30). In Step 4, the ability of the extracted rules in predicting the executive states of web services in combination will be checked.

This step reveals how much the selected rules are able to predict the executive state of web service within possible combinations. Finally in Step 5, the valid rules will be selected in which their accuracy are more than or equal to 40 percent. The amount of 40 percent accuracy is chosen by the designer.

C. Recovery Selection Algorithm

The recovery selection algorithm based on final extracted valid rules and defined fault-tolerant patterns for each web service determines that if a web service, such as Si fails and there is a rule, such as Si→Sj, and if forward recovery is selected in continuance of the combination execution, web service Sj will fail. Therefore, in such a situation, backward recovery will be proposed. Therefore, before the valid rules extraction, the controller always selects forward recovery to repair failures. This action will be costly because the cost of alternative web services is high. However, by having the valid rules with appropriate accuracy, and if in the continuance of execution, other failures are predicted, backward recovery will be proposed.

IV. IMPLEMENTATION AND EVALUATION

To evaluate the proposed methodology performance to implement web services and the final rules production, BPEL designer and eclipse software have been used. All implementations of workflows have been performed on a system with the listed properties in Table 1.

Table 1
System properties

Item	Specification
RAM	3GB
OS	Windows 7
Graphics card	GeForce 9300 MG
CPU	Intel 2.40 GHZ

Four workflows have been defined by designer in two different scenarios. High rate of failure is 40 percent and low rate of failure is 8 percent.

Table 2 illustrates the characteristics of the implemented workflows within an orchestration. In all implementations, there are five tasks inside each workflow. Equivalent to these tasks, five main web services, five compensation web services and four alternative web services in workflows (total 14 web services in orchestration) have been defined.

Table 2
Characteristics of the implemented workflows within orchestration

Parameters	Workflows			
	Wf1	Wf2	Wf3	Wf4
Number of services	14	14	14	14
The number of parallel currents in combination	1	1	2	1
Total number of parallel services	2	2	2	5
The maximum number in parallel execution of services	4	3	3	1
The minimum service with parallel execution	4	3	2	1

In fact, some web services do not have any alternative web service. Here, web service 3 in all workflows does not have any alternative web service. So, if web service 3 fails, the forward recovery will be selected by using 'retry' fault-tolerant pattern and other web services have 'recovery-block' pattern to repair failures. Table 3 illustrates the applied fault-tolerant patterns for each web service.

Table 3
Applied fault-tolerant patterns for each task

Web services	Used Pattern
WS1	Recovery-Block
WS2	Recovery-Block
WS3	Retry
WS4	Recovery-Block
WS5	Recovery-Block

For faster execution of orchestration, a simple function is used in all web services. To simulate the error in web services, each of them has a random function within itself. The random function produces a random number between [0, 100]. For example, for a web service with a failure rate of 40, if generated random number was in range of [0, 40] i.e. web service has been failed, otherwise its execution is successful.

After extracting the rules based on the described steps in section III, the selected rules are listed in Table 4. When the failure rate is high, the minimum accuracy will be 30 and for the low failure rate, 20 have been intended.

Table 4
The extracted rules on 4 workflows when failure rate is high and minimum accuracy is 30

Workflows	Rules	Accuracy
Wf1	1→2,5	0.48
	2→4,5	0.55
Wf2	1→2,4	0.59
	2→4,5	0.59
	3→3,5	0.50
Wf3	4→1	0.40
Wf4	1→2,4,5	0.90
	2→4,5	0.58
	3→3,4,5	0.80

Table 4 shows that when the failure rate is low, valid rules cannot be extracted. However, when the failure rate of web services is high (for example in work flow 4) and if the web service 1 failed, it can be predicted by 90 percent accuracy that after the failure recovery, web services 2, 4 or 5 will fail in continuance. In the event of failure in the web services, if backward recovery is selected to repair the failure, the user must pay some penalty for the execution of the compensation web services. The aim of rules extraction is to find the relations between the failures in combined web services based on the described steps in section III. These relations are expressed in the form of event→action.

Table 5 shows the extracted rules for web service 4 in case of high failure rate. The accuracy parameter in Table 5 illustrates how much the desired rule has a correct prediction for executive states of web services based on test set records.

Table 5
Extracted rules for web service 4

Record number	Rule(a→b)	Count(a)	Count(a→b)	Accuracy (%)
1	1→2	41	20	49
2	1→3	41	2	5
3	1→4	41	17	41
4	1→5	41	21	51
5	2→3	42	4	9
6	2→4	42	19	45
7	2→5	42	23	55
8	3→2	10	2	20
9	3→3	10	3	30
10	3→4	10	4	40
11	3→5	10	4	40
12	4→2	0	0	0
13	4→5	41	14	34
14	5→2	0	0	0
15	5→4	46	8	17

Figure 4 shows workflow 4.

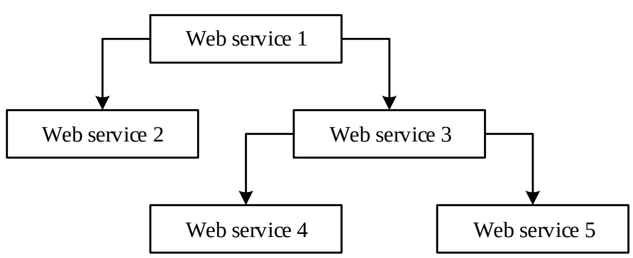


Figure 4: Sequence of web services in workflow 4

The number of test set records for each workflow is 50. In Table 1, rule 1→2 shows 41 records inside the test set, in which each of them has a failure in web service 1. Among the 41 records, there are 20 records which show that after a failure in web service 1, there will be a failure in the following web service 2. Therefore:

Accuracy (1→2) = 20/41= 0.49 * 100= 49 %

Accuracy of this rule is more than 40 percent. Thus, rule 1→2 has been considered as a valid rule.

After combining the valid rules for web service 4, three rules will be obtained as follows:

1→2,4,5
2→4,5
3→3,4,5

Since the alternative web services usually have low failure rate with high price, the forward recovery will enhance the cost overhead. If after the first failure, the produced rules predict another failure in the future, the backward recovery will be selected; otherwise, the forward recovery will be followed. The final rules based on Table 4 have been rewritten in Table 6.

Table 6
The final extracted rules on 4 workflows when failure rate is high and minimum accuracy is 30

Workflows	Rules	Accuracy
Wf1	1→ RollBack	0.48
	2→ RollBack	0.55
Wf2	1→ RollBack	0.59
	2→ RollBack	0.59
	3→ RollBack	0.50
Wf3	4→ RollBack	0.40
Wf4	1→ RollBack	0.90
	2→ RollBack	0.58
	3→ RollBack	0.80

As shown in Table 6, when the failure rate for web services is high, there will be a possibility of multiple failures in web services composition.

Precision and recall are evaluation metrics in binary classification. In information retrieval systems, precision and recall or similar metrics are the main criteria for evaluation.

Precision can also be evaluated at a given cut-off rank. This measure is called precision at (n) [16]. The amount of n expresses the number of the retrieval records inside a test set at any moment.

The precision metric specifies the ability to retrieve mostly related top-ranked executions in test set, whilst recall metric is the ability of searching to find all relevant items in the test set.

Precision and Recall are computed by Equation 3 and 4 respectively.

$$\text{Precision}=\text{n}\left(\text{ret}\cap\text{rel}\right)/\text{n}\left(\text{ret}\right) \quad (3)$$

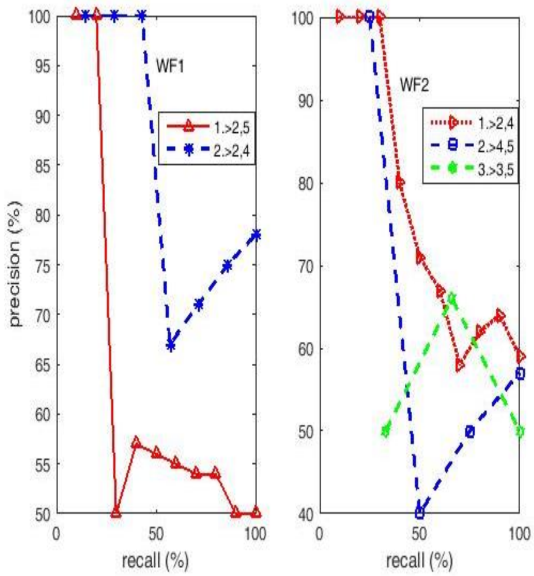
$$\text{Recall} = n\left(ret\cap rel\right)/\,n\left(rel\right) \quad (4)$$

where n(ret∩rel) is the number of retrieved relevant records. n(ret) expresses the total number of retrieved records and n(rel) shows total number of relevant records. The corresponding precision-recall curves are shown in Figure 5.

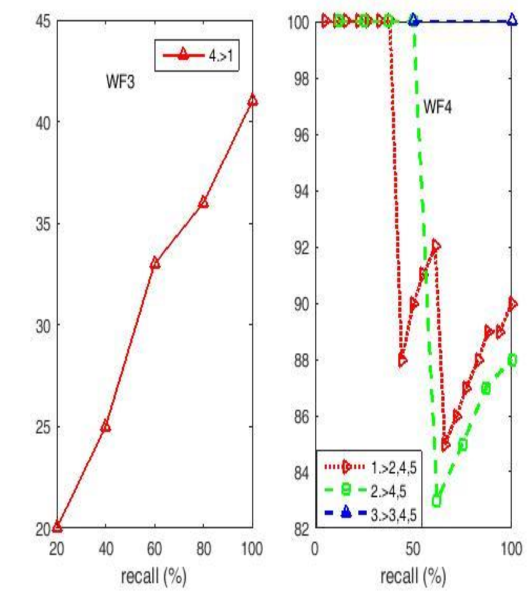
In computing the recall metric, n(rel) parameter illustrates all related records (retrieved and not retrieved). Its value is specified by the total number of related records within a test set and it is a fixed number. As previously mentioned, n(ret∩rel) parameter illustrates all retrieved related records and its value increases during reading the records of the test set.

Recall $\propto$ related records which are retrieved during reading test set records.

In computing precision metric, the amounts of parameters n(ret∩rel) and n(ret) at any moment depend on the retrieved records within a test set. In each reading of a record of test set, the amount of n(ret) increases a unit. If retrieved record is associated with the mining rule, n(ret∩rel) increases a unit, otherwise it will not change. This concept is illustrated by high and low points as shown in Figure 5.



(a) Precision-recall curves for wf1 and wf2



(b) Precision-recall curves for wf3 and wf4

Figure 5: Precision-recall curves for extracted rules on the 4 workflows

Table 7 shows the calculations for recall and precision metrics from rule 2→4, 5 in workflow 4.

Table 7
Calculations for recall and precision metrics for rule 2→4,5 in workflow 4

Retrieved record number in test set	Related record	Recall	Precision
6	Yes	1/8=0.12	1/1=1
7	Yes	2/8=0.25	2/2=1
11	Yes	3/8=0.37	3/3=1
13	Yes	4/8=0.5	4/4=1
15	No	---	---
21	Yes	5/8=0.62	5/6=0.83
26	Yes	6/8=0.75	6/7=0.85
37	Yes	7/8=0.87	7/8=0.87
42	Yes	8/8=1	8/9=0.88

V. Conclusion

The fault tolerance is one of the most important issues in combining web services. In the failure moment, forward or backward recoveries can be selected based on user constraints. Having a technique that can select the best recovery strategy in the event of failure, considering user constraints, overhead cost and other important factors is very necessary. In this paper, valid rules are created by using association rules technique and history of the implemented web services in data set. The most appropriate recovery strategy in the failure moment can be predicted by these rules. The results showed that using the association rules causes a reduction in overhead costs, and this method has a proper accuracy and good performance.

References

[1] O.Bushehrian, S.Zare and Rad, N. K. “A workflow-based failure recovery in web services composition”. Journal of Software Engineering and Applications, 2012.

[2] W.Gaaloul, S.Bhiri, and M.Rouached. “Event-based design and runtime verification of composite service transactional behavior”. Services Computing, IEEE Transactions on, 3(1), 2010, pp.32-45.

[3] S.Bhiri, C.Godart and O.Perrin. “Transactional patterns for reliable web services compositions”. In Proceedings of the 6th international conference on Web engineering, 2006, pp. 137-144. ACM.

[4] J.El Hadad, M.Manouvrier and M.Rukoz. “TQoS: Transactional and QoS-aware selection algorithm for automatic Web service composition”. Services Computing, 2010, IEEE Transactions on, 3(1), 73-85.

[5] R.Angarita, Y.Cardinale and M.Rukoz. “Faceta: Backward and forward recovery for execution of transactional composite ws”, 2012.

[6] A.Liu, Q.Li, L.Huang and M.Xiao. “ Facts: A framework for fault-tolerant composition of transactional web services”. Services Computing, IEEE Transactions on, 3(1), pp. 46-59.

[7] R.Angarita, Y.Cardinale, and M.Rukoz. “Reliable composite web services execution: towards a dynamic recovery decision”. Electronic Notes in Theoretical Computer Science, 302, 2014, pp.5-28.

[8] O.Bushehrian and MM.Mavaddat. “A Fault-Tolerance Pattern Recommender for Transactional Service Compositions”. The 2013 CSI International symposium on Computer Science and Software Engineering (CSSE), 2013.

[9] O.Bushehrian, and S.Zare. "Cost-aware execution of transactional web services using labelled transition systems". Software, IET, 8(5), 2014, pp. 232-243

[10] N.Jiang, and L.Gruenwald . ”Research issues in data stream association rule mining”. *ACM Sigmod Record*, *35*(1), pp.14-19. 2006.

[11] M. Halatchev and L. Gruenwald. “ Estimating Missing Values in Related Sensor Data Streams”. In *COMAD* (pp. 83-94) , January , 2005.

[12] E.D Demaine, A.López-Ortiz and J.I.Munro.” Frequency estimation of internet packet streams with limited space”. In *Algorithms—ESA 2002* (pp. 348-360). 2002, Springer Berlin Heidelberg.

[13] Y.D.Cai, D.Clutter, G. Pape, J. Han, M.Welge and L.Auvil.” MAIDS: Mining alarming incidents from data streams”. In *Proceedings of the 2004 ACM SIGMOD international conference on Management of data* (pp. 919-920). ACM, June, 2004.

[14] W.Lin, S.A. Alvarez and C.Ruiz. “Efficient adaptive-support association rule mining for recommender systems”. *Data mining and knowledge discovery*, *6*(1), pp.83-105, 2002.

[15] C.Becquet, S.Blachon, B. Jeudy, J.F. Boulicaut, and O.Gandrillon. “Strong-association-rule mining for large-scale gene-expression data analysis: a case study on human SAGE data”. *Genome Biology*, *3*(12), pp.1-16, 2002.

[16] https://en.wikipedia.org/wiki/Information_retrieval