I.J.Modern Education and Computer Science, **2013, 8, 18-26**
Published Online October 2013 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijmecs.2013.08.03

A Proposal for Improving Behavioral Adaptation of Web Services Integration

M. Rizwan Jameel Qureshi, Alaa Ahmed Almarzuki
Faculty of Computing and Information Technology, King Abdulaziz University, Saudi Arabia
E-mail: anriz@hotmail.com; a.almarzoki@yahoo.com

Abstract— Reuse of existing software components is the main goal of Component-Based Software Engineering. Different organizations designed different web-services as the most famous type of component. Adaptation is a critical issue when building new applications by reusing existing services. How to adapt these services to work correctly is the main goal of most of the proposed models and techniques in software component filed. Behavioral mismatches are one of the adaptation problems. Different solutions have been written to address this problem like adapter and some other techniques. Most of the papers focused on how to create the adapter to overcome the incompatibility problem. In this paper, the authors provide a simple integrated tool that can solves ordering mismatching. The results are concluded using a survey from thirty one respondents. The proposed solution needs further validation by conducting a case study.

Index Terms— Component-Based; Web Services; Adaptation; Behavioral Mismatch

I. Introduction

Composition of service as a type of component is a core issue in Service-Computing. By reusing the existing services, developers can save the time and the costs of developing new software. Some existing services developed by different vendors. However, direct integration of services is nearly impossible because of incompatibilities/mismatches of interfaces. These mismatches services may arise at different interoperability levels: technical, signature, behavioural, quality of service and semantic [1]. The behavioural level mismatches and how to ensure web services compatible at the behavioural level is an important issue for web services integration and it is the core of this research. Deadlock is a typical 'behavioural mismatch with reordering' in which ordering of messages cause the mismatch [7].

The rest of paper organized as follows, the next section presents a brief literature review. Section 3 presents the problem statement. The solution towards that problem is present in section 4. Section 5 is talking about the validation and the paper ends with the conclusion.

II. Related Work

Several approaches and techniques developed to overcome behavioural mismatch problems. All of them aiming to build one common solution called adapter, but they are differ in the used algorithms and techniques. A model-based adaptation approach proposed to focus on mismatch between behavioural interfaces of two components [1]. It is automatically generate adaptor based on Labelled Transition Systems (LTSs) and by cutting off the behaviours causing mismatches. Two algorithms that generate adaptor protocols automatically used to support the proposed approach: the first based on synchronous products and the second based on Petri net encodings. The adapter implements both to use them with a user-friendly graphical interface.

Another proposed technique that generates behavioural adapters in [2]. It is differs in the models of the services to be composed and behavioural property, so called specification of the elementary activities (SEA). This approach aims to reduces the size of the generated adapter and reduce the complexity of the result. The key insight is to simplify the adapter through removing from the adapter the places that are redundant in the context of the given services.

[3] Illustrates a novel formal model for web service interfaces, which used to reach to and use the services. They give the condition for detecting behavioural mismatches among multiple web services, which relies on an abstract notation based on (LTS). In order to support this method, an algorithm automatically builds the synchronous product for a set of web service behavioural interfaces.

Qi Huacheng, Rong Mei and Zhang Guang quan construct component behaviour ontology (CBO) which includes component entity, component behaviour and component behaviour properties. Then, they proposed a Component Interaction Adaptation Model (ClAM) based on CBO. The idea is to divide different components groups. The next stage is Detecting Behavioural Mismatch in order to obtain the autocephalous adapter for each component group [4].

Another safety and effective approach of service behaviour adaptation presented [5]. A coordinator used for interaction of web services and solution for behaviour mismatch. Authors present their approach with discussion the calculation model and the framework of service behaviour adaptation. The service behaviour specified using the Symbolic Transition Systems (STS). The behaviour protocol of adaptor – adapter definition- calculated according to the specification of adaptor in which each element called as synchronous vector. Thus, the adaptor will interact with services according to its behaviour protocol.

H.-H. Lin, T. Aoki, and T. Katayama present an approach for service adaptation. The proposed model used a pushdown automaton to support non-regular adaptation, Also, it is integrate adaptation and model checking together to overcome some time-consuming issues. The detection process of behavioural mismatches performed by novel model called checking model. In addition, the adapter generation performed by model checking of pushdown systems with the guidance of a special over-behavioural adapter called “coordinator” that guide adapter generation [6].

The paper number [7] focused on resolving behavioural mismatches for loosely coupled, asynchronously communicating services. It is provide an efficient, automated method to construct protocol adaptors. The generated adapter makes service adaptation processes only to those messages that cause a deadlock, and that is the meaning behind using (minimal) word in this approach.

A novel model called a protocol structure presented in [8]. A protocol structure used to model service protocols and represent messages and message dependencies graphically. It easily used to identify multiple message mismatches and the position of every mismatch. Based on the proposed structure, authors develop an algorithm for automatically detect multiple message mismatches and to automatically generate Business Process Execution Language (BPEL) adapters.

A proposed interface automata model with the function descriptions of semantic operations presented in [9]. "Authors present a formal approach based on interface automata to adapt components in order to eliminate possible behavioral mismatches" [9].

An attempt [10] was made to solve the adaptation by generating an adapter after cutting off the behaviour that may lead to mismatch in order to make it deadlock-free. Most of the proposed solutions above aimed to how to generate an adapter to overcome adaptation problems and behavioural mismatches. Table 1 summarizes the main limitations of related work [1-10].

Table 1: Limitations of the related work

Title	Limitation
Model-Based Adaptation of Behavioral Mismatching Components[1]	• Limited to interaction between two components • Behavior interfaces, such as LTS, only handle regular language. • Complexity • The design of an adaptation contract is error-prone task, many interactions being removed in the adaptation process -reduce the functionality of components.
Trading off concurrency to generate behavioral adapters[2]	• The behavior of the adapter will change in other contexts • Removing from the adapter some places that are redundant in the composed system
A Method for Detecting Behavioral Mismatching Web Services[3]	• Not working correctly with time constraints.
A Behavior-Driven Model of Component Interaction Adaptation [4]	Adapting group of components is: • Difficult to steer and control rather than dealing with two components • Processing is more complex • Need more time and memory
Specification and Verification of Safe Web Service Behavior Adaptation[5]	The solution is used only for • Two services • The kind of one to one behavior relation
Non-regular adaptation of services using model checking[6]	• Problems in using Büchi’s automata model to define services with unbounded messages. -Same state representing both start and end states this lead to confusion.
Generating Minimal Protocol Adaptors for Loosely Coupled Services[7]	• Complexity. • Dose not used to analyze protocols containing loops.
Message Dependency-Based Adaptation of Services [8]	• Deadlock mismatches still possibly occur.
Adapting Component Behaviors using Interface Automata [9]	• Does not include the processing for service messages. • Has not an algorithm to validate the business function of services.
Failure-free coordinators synthesis for component-based architectures[10]	• The functionality of the components involved in the adaptation process will be Restricted.

III. Problem Statement

Developers always face compatibility problems while integrating the components. The integration between services raises behavioral mismatches that prevent them to work correctly to reach the desired goal.

IV. The Proposed Solution

To find a solution for mismatches in the behavioural level, the authors are looking for new solution rather than thinking how to create an adapter. The proposed solution is explained by taking the example of two components 'A' and 'B' in figure 1. The component 'A' sends a message to component 'B'. The 'B' receives it in its first in first out (FIFO) queue, and then it picks first entry up to extract the data from the message and continues its job. There are two situations for B here. 1) If the massage was the same as what it is looking for, it will perform its job and continue to the next message in the queue. 2) If the message was not the same as what it is looking for, thus it will hang and deadlock will occur. Some of deadlock situations can be resolve and other are not as presented in [7]. Since this paper focuses on the behavioural mismatches that relates to reordering problem, thus a proposed solution aims to add a tool to a black-box component by the original developer.

The box with bold frame at left in the figure is represent component 'A' which retrieves the record for specific user in order to analyse it and create a graphic representation.

At the right Component 'B' verifies the user and password for the given user, then it sends his records back to the component 'A'. In figure 1, the processes of component 'A' and 'B' did not addressed in details while it presents detailed processes about the proposed tool to show how it should be work in case if it receives messages not ordered as 'B' requirements. Assume that each (Sn, Mn, Rn) represents an interaction between component 'A' and 'B'.

Where:

- S = Send, M= Messages, R= Receive
- n represent the interaction number.

Table 2 shows the meaning of each symbol in the figure 1 above. The simplest example provided in figure 1 that B receives password then the user name while it is waiting for the reverse. When component B receive m1 (password) it will not work because it is expect the username m2 will come first. Thus the tool should change the order to fit with component B. While in the normal case without using this tool, deadlock will occurs and developers will have to focus on how to create an adapter to solve such problem. The component that integrated with the proposed tool must be able to solve ordering mismatches by itself.

Table 1: Shows the meaning of each symbol in the figure 1

n	1	2	3	4
S	Send password	Send username	Send request	Send record
M	Password	Username	Request	Record
R	Receive password	Receive username	Receive request	Receive record

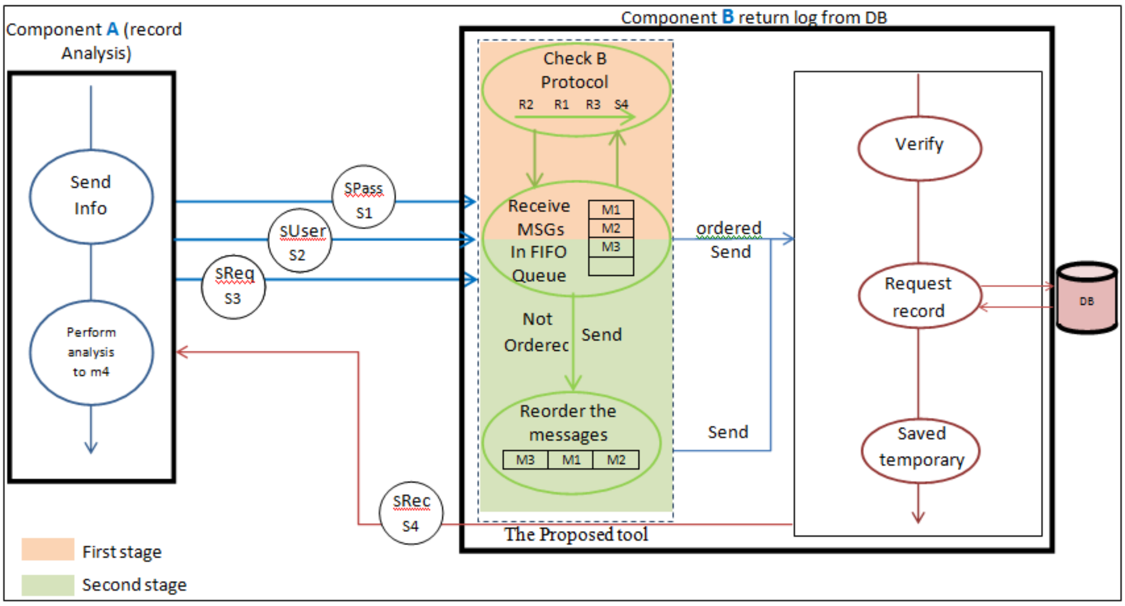


Fig. 1: Simple Simulation for the Proposed Tool

V. Validation

In order to validate the proposed solution and measure the need for such tool by highlighting the disadvantages of current solutions and find out how much they are incompetence, a questionnaire created by the authors to be distributed over number of programmers and developers to collect their opinions about the proposed solution. The questionnaire contains 18 phrases and questions, which fall under three goals, to enhance the solution proposed above. Likert scale will used for evaluating the collected answers. Questionnaire appended in Appendix A.

5.1 Findings

In order to validate the proposed solution the questionnaire divided into 3 goals as in the following:

A. The Cumulative Analysis of Goal 1. To measures the participant knowledge about components adaptation.

This goal placed to ensure the questionnaire solved by the right participants who are really having knowledge about component and reusing component. This goal also wills increments the value of the questionnaire by giving the result of validation goal greater degree of accuracy.

Table 3: Cumulative analysis of goal 1

Q No.	Very high	High	Nominal	Low	Very low
1	9	11	11	0	0
2	5	9	13	3	1
3	9	15	4	3	0
4	12	11	6	2	0
5	5	4	15	6	1
Total	40	50	49	14	2
Avg.	25.8	32.25	31.61	9.03	1.29

In the first goal, the questions focus on reusing components, the process of choosing the components those are fit to work together, the integration of component and the adaptation of components. The cumulative analysis of goal 1 is shown in Table 3.

The cumulative descriptive is shown for each question in the table. 25.8% of the participants have very high degree of knowledge area and 32.25 of them have high degree of knowledge. The percentage of 9.03% represents the participants who have little information about the components adaptation while 1.29% of them have a very low knowledge about the components adaptation. The percentage of 31.61% represents the participants who have a good knowledge area. Figure 2 represents the corresponding chart for the results represented in the table.

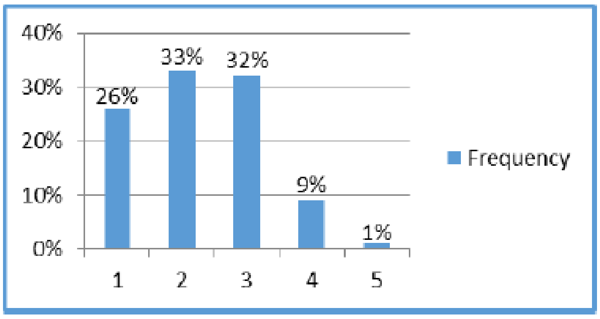


Fig. 2: Graph showing the cumulative results of questionnaire for goal 1

B. The Cumulative Analysis of Goal 2. To evaluates the existing solutions of the behavioral mismatches problem.

This goal reflects the need of new solution. The cumulative analysis of goal 2 is shown in Table 4.

Table 4: Cumulative analysis of goal 2

Q No.	Very high	High	Nominal	Low	Very low
6	4	13	10	4	0
7	3	15	8	3	2
8	7	13	9	2	0
9	14	8	7	2	0
10	5	11	4	8	3
Total	33	60	38	19	5
Avg.	21.29	38.7	24.5	12.25	3.2

By 12.25% of the participant agreed that existing solutions are satisfied the needing for components adaptation while only 3.2% highly agree on the existing solutions. While 38.7% see, the existing solutions are not efficient and take a lot of time and effort and 21.29% of the participant strongly agreed that. 24.5% are impartial. Figure 3 represents the corresponding chart for the results represented in the table.

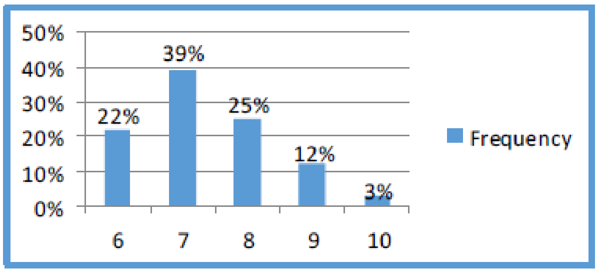


Fig. 3: Graph showing the cumulative results of questionnaire for goal 2

C. The Cumulative Analysis of Goal 3. To validates the proposed solution.

This goal reflects how much the proposed solution is desirable and recommended. The cumulative analysis of goal 3 is shown in Table 5.

Table 5: Cumulative analysis of goal 3

Q No.	Very high	High	Nominal	Low	Very low
11	12	12	5	2	0
12	12	12	4	1	2
13	9	11	9	2	0
14	6	18	7	0	0
15	9	16	6	0	0
16	5	12	6	5	3
17	13	14	4	0	0
18	14	10	7	0	0
Total	80	105	48	10	5
Avg.	32.25	42.33	19.35	4	2

The high percentage of very high and high columns in table 5 indicates the efficiency of the proposed solution were they represent 32.25% and 42.33% respectively. They agreed the proposed solution would save their time and effort. In addition, they recommend the vendors to implement the tool in the component and they will pay more for these components in the future. By 4% of the participants have the opposite opinions also 2% strongly disagreed everything about the proposed tool. While 19.35% are impartial. Figure 4 represents the corresponding chart for the results represented in the table. Appendix B show the result of all questions as a charts

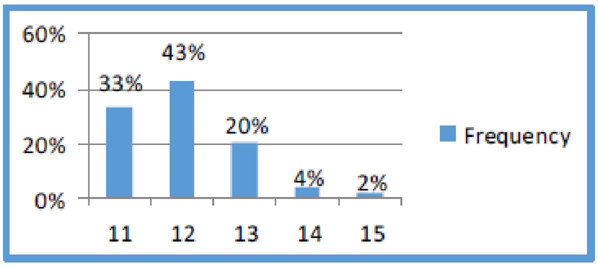


Fig. 4: Graph showing the cumulative results of questionnaire for goal 3

VI. Conclusion

Adaptation is a critical issue when building new applications by reusing existing services. The integration between components usually presents different mismatches according to several levels. Behavioral mismatches are one of the adaptation problems. Researchers have been present different solutions for that problem like adapter and some other techniques. In this paper, authors provide these solutions and their limitation. Then they proposed a solution for behavioral mismatches to solve unordered messages problem. The validation of this proposal presented using a survey. The highest results indicate the need for the proposed tool and the efficiency of appending that to into the components.

In the future, the authors want to implement this tool actually to a component at which that tool can deal with simple and complex reordering problems. In addition, they looking for improve the proposed tool to be able to solve other mismatches either in the behavioral level or in the other levels.

References

[1] Canal, C.; Poizat, P.; Salaun, G.,"Model-Based Adaptation of Behavioral Mismatching Components," *Software Engineering, IEEE Transactions on*, vol.34, no.4, pp.546-563, July-Aug. 2008.

[2] Mooij, A.J., Voorhoeve, M., "Trading Off Concurrency to Generate Behavioral Adapters", *Application of Concurrency to System Design, 2009. ACSD '09. Ninth International Conference on*, On page(s): 109 – 118

[3] Sheng Wang; Guangquan Zhang; Xiaofang Zhang; Yang Yang; , "A Method for Detecting Behavioral Mismatching Web Services," *Web Information Systems and Applications Conference, 2009. WISA 2009. Sixth*, vol., no., pp.116-121, 18-20 Sept. 2009

[4] Qi Huacheng; Rong Mei; Zhang Guangquan; , "A behavior-driven model of component interaction adaptation," *Computer Science & Education, 2009. ICCSE '09. 4th International Conference on*, vol., no., pp.871-875, 25-28 July 2009

[5] Xiong Xie, Weishi Zhang, Huawei Zhai, Licheng Cui, Jinyu Shi, "Specification and Verification of Safe Web Service Behavior Adaptation", *Parallel Architectures, Algorithms and Programming (PAAP), 2010 Third International Symposium on*, On page(s): 417 - 421

[6] H.-H. Lin, T. Aoki, and T. Katayama, “Non-regular adaptation of services using model checking,” in*ISORC10*. 2010, pp. 170–174.

[7] Seguel, R.; Eshuis, R.; Grefen, P.; , "Generating Minimal Protocol Adapt ors for Loosely Coupled Services," *Web Services (ICWS), 2010 IEEE International Conference on* , vol., no., pp.417-424, 5-10 July 2010

[8] Jian-min Jiang, Shi Zhang, Ping Gong, Zhong Hong, "Message Dependency-Based Adaptation of Services", *Services Computing Conference (APSCC), 2011 IEEE Asia-Pacific,* On page(s): 442 – 449

[9] Chouali, S.; Mouelhi, S.; Mountassir, H.; , "Adapting Component Behaviours Using Interface Automata," *Software Engineering and Advanced Applications (SEAA), 2010 36th EUROMICRO Conference on* , vol., no., pp.119-122, 1-3 Sept. 2010

[10] M. Tivoli and P. Inverardi, "Failure-free coordinators synthesis for component-based architectures," *Sci. Comput. Program.*, vol. 71, no. 3, pp. 181–212, 2008.

Appendix A

Questionnaire for Appending a Tool into Component to Solve Reordering Behavioral Mismatches

Filling Guidelines

Likert scale is ranging from 1 to 5.

- **Very low** effect indicating 1
- **Low** effect indicating 2
- **Nominal/Average** effect indicating 3
- **High** effect indicating 4
- **Very high** effect indicating 5

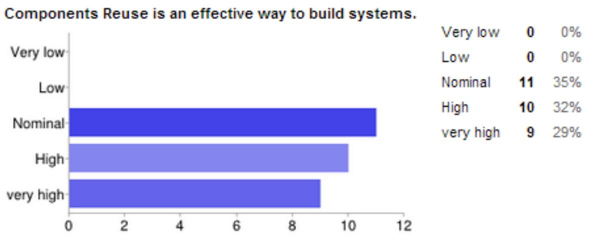
Adaptation word in the questionnaire refers to adaptation of behavioral mismatches only. As you now, behavioral mismatches (unordered messages and deadlock) arise in the behavioral level. The goal of this questionnaire is to measures how much it is useful when a component can adapt unordered messages mismatch by itself.

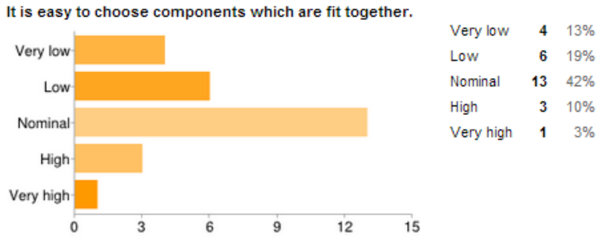
Question 1-5: Measure the participant knowledge about components adaptation		
#	**Question**	**Response**
1.	Components Reuse is an effective way to build systems.	□ Very low □ Low □ Nominal □ High □ Very high
2.	It is hard to choose components which are fit together.	□ Very low □ Low □ Nominal □ High □ Very high
3.	Usually, integration of two or more components causes a lot of error.	□ Very low □ Low □ Nominal □ High □ Very high
4.	How much the adaption process is important?	□ Very low □ Low □ Nominal □ High □ Very high
5.	How much the existing techniques that used for adaptation are satisfied their needing?	□ Very low □ Low □ Nominal □ High □ Very high
Question 6-10: Measure the drawbacks of existing adaptation solution		
6.	The existing solutions are reliable solutions that can be used to adapt to different components.	□ Very low □ Low □ Nominal □ High □ Very high
7.	It is time consuming process to choose the best model or technique that can be used to generate the appropriate adapter.	□ Very low □ Low □ Nominal □ High □ Very high

8.	How much choosing the best model or technique that used to generate appropriate adapter is hard?	□ Very low □ Low □ Nominal □ High □ Very high
9.	How much the adapter generating process is time consuming?	□ Very low □ Low □ Nominal □ High □ Very high
10.	How much the adapter generating process hard?	□ Very low □ Low □ Nominal □ High □ Very high
Question 11-18: **Measure the need for the novel tool**		
11.	Do you think it is possible to develop an intelligent component that can adapt itself with others?	□ Very low □ Low □ Nominal □ High □ Very high
12.	Do you think it is useful economically to use components that has integrated adaptation tool?	□ Very low □ Low □ Nominal □ High □ Very high
13.	How much time do you think will be saved if you use components that has integrated adaptation tool?	□ Very low □ Low □ Nominal □ High □ Very high
14.	Do you believe it will be easier to use a component that has integrated adaptation tool feature rather than regular one?	□ Very low □ Low □ Nominal □ High □ Very high
15.	As a developer you see that software development using components is more effective by using component with integrated adaptation tool.	□ Very low □ Low □ Nominal □ High □ Very high
16.	As a developer you prefer to pay for component with integrated adaptation tool even if it is a little more expensive	□ Very low □ Low □ Nominal □ High □ Very high
17.	Is it better if the vendors develop this type of components?	□ Very low □ Low □ Nominal □ High □ Very high
18.	Developing components with integrated adaptation tool will increase the strength and the quality of the vender products	□ Very low □ Low □ Nominal □ High □ Very high

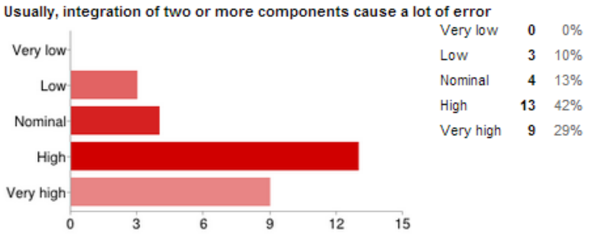
Appendix B

The total results represented with a chart for each question as in the following.

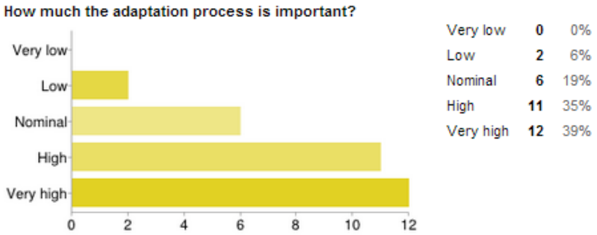



It is easy to choose components which are fit together.

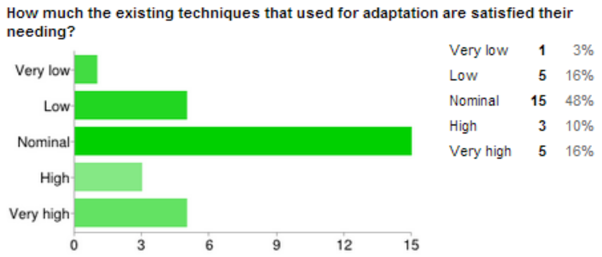
Very low	4	13%
Low	6	19%
Nominal	13	42%
High	3	10%
Very high	1	3%


Usually, integration of two or more components cause a lot of error

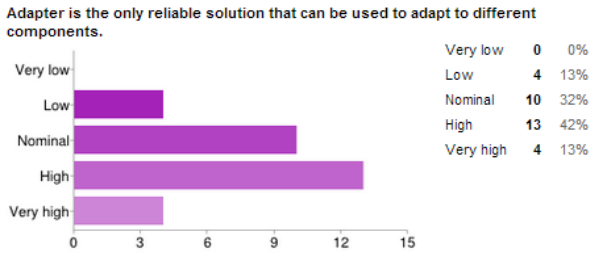
Very low	0	0%
Low	3	10%
Nominal	4	13%
High	13	42%
Very high	9	29%


How much the adaptation process is important?

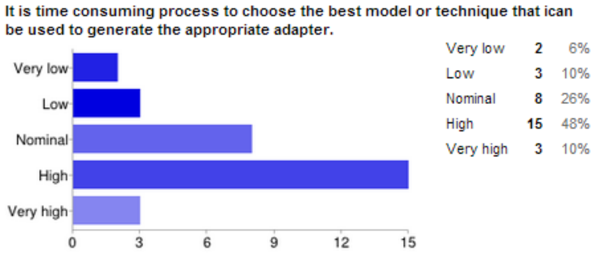
Very low	0	0%
Low	2	6%
Nominal	6	19%
High	11	35%
Very high	12	39%


How much the existing techniques that used for adaptation are satisfied their needing?

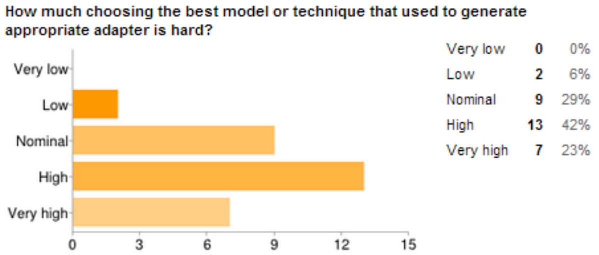
Very low	1	3%
Low	5	16%
Nominal	15	48%
High	3	10%
Very high	5	16%


Adapter is the only reliable solution that can be used to adapt to different components.

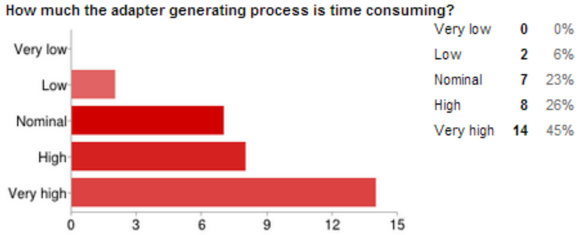
Very low	0	0%
Low	4	13%
Nominal	10	32%
High	13	42%
Very high	4	13%


It is time consuming process to choose the best model or technique that ican be used to generate the appropriate adapter.

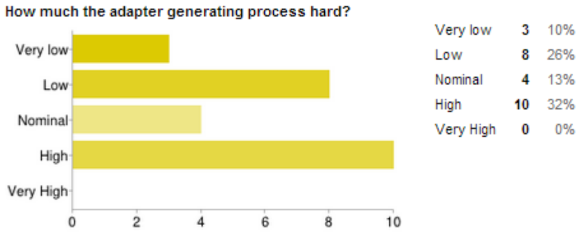
Very low	2	6%
Low	3	10%
Nominal	8	26%
High	15	48%
Very high	3	10%


How much choosing the best model or technique that used to generate appropriate adapter is hard?

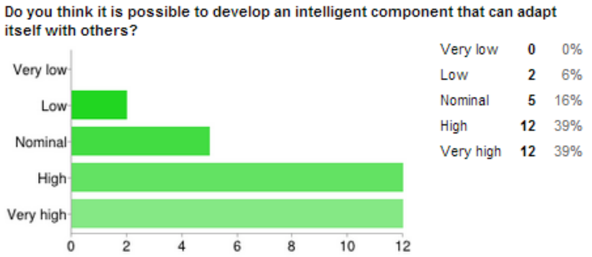
Very low	0	0%
Low	2	6%
Nominal	9	29%
High	13	42%
Very high	7	23%


How much the adapter generating process is time consuming?

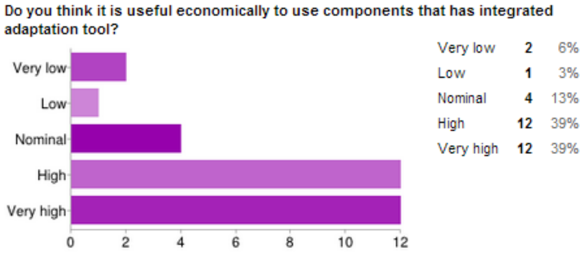
Very low	0	0%
Low	2	6%
Nominal	7	23%
High	8	26%
Very high	14	45%


How much the adapter generating process hard?

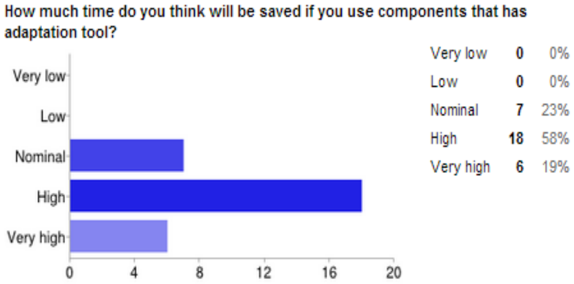
Very low	3	10%
Low	8	26%
Nominal	4	13%
High	10	32%
Very High	0	0%


Do you think it is possible to develop an intelligent component that can adapt itself with others?

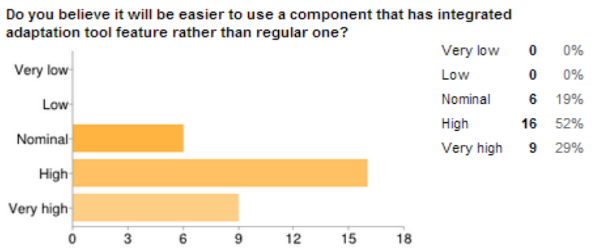
Very low	0	0%
Low	2	6%
Nominal	5	16%
High	12	39%
Very high	12	39%


Do you think it is useful economically to use components that has integrated adaptation tool?

Very low	2	6%
Low	1	3%
Nominal	4	13%
High	12	39%
Very high	12	39%

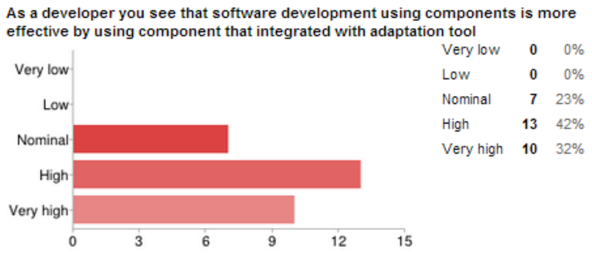

How much time do you think will be saved if you use components that has adaptation tool?

Very low	0	0%
Low	0	0%
Nominal	7	23%
High	18	58%
Very high	6	19%



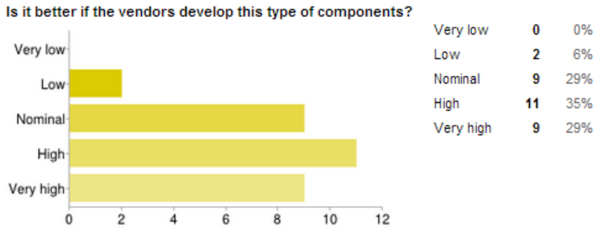
Do you believe it will be easier to use a component that has integrated adaptation tool feature rather than regular one?

Very low	0	0%
Low	0	0%
Nominal	6	19%
High	16	52%
Very high	9	29%



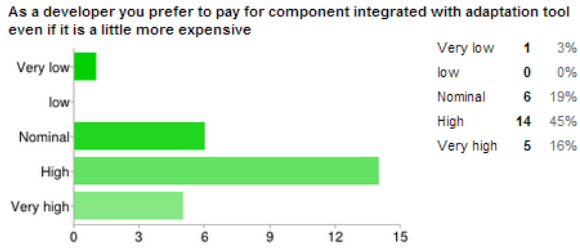
As a developer you see that software development using components is more effective by using component that integrated with adaptation tool

Very low	0	0%
Low	0	0%
Nominal	7	23%
High	13	42%
Very high	10	32%



Is it better if the vendors develop this type of components?

Very low	0	0%
Low	2	6%
Nominal	9	29%
High	11	35%
Very high	9	29%



As a developer you prefer to pay for component integrated with adaptation tool even if it is a little more expensive

Very low	1	3%
low	0	0%
Nominal	6	19%
High	14	45%
Very high	5	16%

Authors' Profiles

M. Rizwan Jameel Qureshi: Assistant Professor of Information Technology, Faculty of Computing & Information Technology, King Abdulaziz University, major in software engineering and database management systems.

Alla Ahmed Almarzuki: Post-graduate student for master degree for Information Technology in King Abdul Aziz University, major in Internet Technology and Software engineering